Assessing Grammatical Correctness in Language Learning

Katinskaia, Anisia

2021-04

Katinskaia, A & Yangarber, R 2021, Assessing Grammatical Correctness in Language Learning. in Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications. The Association for Computational Linguistics, Stroudsburg, pp. 135-146, 16th Workshop on Innovative Use of NLP for Building Educational Applications, 20/04/2021.

http://hdl.handle.net/10138/330272

Assessing Grammatical Correctness in Language Learning

Anisia Katinskaia[†⋆] and **Roman Yangarber**[†]
University of Helsinki, Finland
[⋆]Department of Computer Science
[†]Department of Digital Humanities
first.last@helsinki.fi

Abstract

We present experiments on assessing the grammatical correctness of learner answers in the Revita language-learning platform.[1] In particular, we explore the problem of detecting *alternative-correct answers*: when more than one inflected form of a lemma fits syntactically and semantically in a given context. This problem was formulated as *Multiple Admissibility* (MA) in (Katinskaia et al., 2019). We approach the problem from the perspective of grammatical error detection (GED), since we hypothesize that models for detecting grammatical mistakes can assess the correctness of potential alternative answers in the language-learning setting. Due to the paucity of training data, we explore the ability of pre-trained BERT to detect grammatical errors and then fine-tune it using *synthetic* training data. In this work, we focus on errors in inflection. Our experiments A. show that pre-trained BERT performs worse at detecting grammatical irregularities for Russian than for English; B. show that fine-tuned BERT yields promising results on assessing correctness in grammatical exercises; and C. establish new GED benchmarks for Russian. To further investigate its performance, we compare fine-tuned BERT with one a state-of-the-art model for GED (Bell et al., 2019) on our dataset, and on RULEC-GEC (Rozovskaya and Roth, 2019). We release our manually annotated learner dataset, used for testing, for general use.

1 Introduction

Many intelligent tutoring systems (ITS) and computer-aided language learning systems (CALL) generate exercises and try to assess the learner's answers automatically. Providing feedback to the learner is difficult, due to the critical requirement of very high precision—providing incorrect feedback is much more harmful than no feedback at all. For this reason, most existing systems have prefabricated sets of exercises, with possible expected answers and prepared feedback.

Revita is an online L2 learning system for learners beyond the beginner level, which can be used in the classroom and for self-study (Katinskaia et al., 2017; Katinskaia and Yangarber, 2018; Katinskaia et al., 2018). It covers several languages, most of which are highly inflectional, with rich morphology. In contrast to the pre-fabricated approach, Revita allows the learner to upload *arbitrary* texts to be used as learning content, and automatically creates exercises based on the chosen content. At practice time, Revita presents the text one paragraph at a time with some words hidden and used as fill-in-the-blank (*cloze*) exercises. For each hidden word, Revita provides a hint—the base form (*lemma*) of the word. The learner should insert the inflected form of the lemma, given the context.

Continuous assessment of the user's answers is also performed automatically (Hou et al., 2019). Revita checks the learner's answer by comparing it with the *expected* answer—the one found in the original text. The problem arises when, for some exercise, besides the expected answer, another answer is also valid in the context. As a result, Revita may provide undesirable feedback by flagging answers that are not expected, but nonetheless correct, as "errors"—this can strongly mislead, confuse and discourage the learner. For example, both highlighted answers in the example below can be considered correct, but Revita expects the learner to use only the past tense form "сдавал" ("took"):
"*Мне приснилось, как я* ***сдавал*** *экзамены.*"
("*I saw a dream how I* ***took*** *exams.*")
"*Мне приснилось, как я* ***сдаю*** *экзамены.*"
("*I saw a dream how I* ***take*** *exams.*")

Hence, detecting alternative-correct answers is essential in the learning context. We manually checked a large set of answers, which were marked

[1] revita.cs.helsinki.fi

Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications, pages 135–146

Learner level	Advanced	Others
Gram. error	649 (62.2%)	4777 (72.8%)
Alternative corr.	395 (37.8%)	1024 (15.6%)

Table 1: Percentage of answers with real grammatical errors and alternative-correct answers for advanced and other learners among all answers which were automatically marked by Revita as incorrect.

by Revita as "erroneous" and discovered that the percentage of alternative-correct answers for advanced learners is more than double the percentage for other learners (see Table 1).

To address this problem, we build a model, which takes a paragraph with learner answers and decides whether they are *grammatically* correct. If the model is not certain about a user's answer, we can fall back on the "default" method—comparing to the expected answer. For evaluation, we created a dataset of paragraphs containing answers given by real learners, manually annotated for acceptability in their context.[2] Additionally, we do not focus on semantics, due to the current setup of the exercises—if a learner inserts an answer with a lemma that is different from the given hint, that is always considered erroneous. Learners may give such answers, but they are easily identified and are not considered and are not annotated in this set of experiments.

The paper is organized as follows. In section 2, we review prior work on GED, with a focus on limited training data. In section 3, we describe the learner corpora collected by Revita and used for evaluation and propose a novel method for generating data with *simulated* grammatical errors. In section 4, we first experiment with a pre-trained BERT as a masked language model (MLM). We fine-tune the pre-trained BERT on synthetic data and measure its ability to assess grammatical correctness on the learner data. In section 5, we discuss the results of the experiments. In section 6, we summarize our contribution and discuss future work.

2 Related Work

Early experiments with GED utilized rules (Foster and Vogel, 2004) and supervised learning from error-annotated corpora (Chodorow et al., 2007). Much work focused on detection of particular types of errors, e.g., verb forms (Lee and Seneff, 2008). Later work is mostly independent of the type of errors, and explores various neural architectures.

Rei and Yannakoudakis (2016) first approach error detection by bi-LSTM models, which achieved strong results on in-domain data. Rei et al. (2016) added character-level embeddings to capture morphological similarities between words. Rei (2017) experiment with using a secondary language modeling (LM) objective. Rei and Yannakoudakis (2017) perform experiments with adding multiple auxiliary tasks for error detection. The best result was achieved by combining the main error detection task with predicting error types, POS tags, and types of grammatical relations. In subsequent experiments, the architecture was modified for jointly learning to label tokens and sentences (Rei and Søgaard, 2019). Bell et al. (2019) extended the above model by incorporating contextualized word embeddings produced by BERT, ELMO, and Flair (Peters et al., 2017; Devlin et al., 2018; Akbik et al., 2018). BERT embeddings produced the best performance across all test sets.

Previous work on GED mostly uses bi-LSTM as a classification model, combined with various approaches for augmenting the training data (Liu and Liu, 2017; Kasewa et al., 2018), or creating new, grammatically-specific word embeddings (Kaneko et al., 2017). More recent work utilizes transformer models (Kaneko and Komachi, 2019; Kaneko et al., 2020; Li et al., 2020; Chen et al., 2020).

Several works on GEC focus on lower-resource languages, including Russian, using the RULEC-GEC dataset for training or fine-tuning (Rozovskaya and Roth, 2019; Náplava and Straka, 2019; Katsumata and Komachi, 2020). Náplava and Straka (2019) outperformed results of Rozovskaya and Roth (2019) by over 100% on $F_{0.5}$, but still showed poor performance compared with other languages in the experiment. GEC for Russian is demonstrated to be the most challenging task, which is explained in part by the small size of RULEC-GEC.

The problem of scarce training data for GED can be approached by using pre-trained language models. Linzen et al. (2016) explored the ability of a LSTM model trained without grammatical supervision to detect grammatical errors by performing an *unsupervised cloze test*. The authors use a dataset of sentence pairs: an error-free original and an erroneous one. The erroneous sentence can be built manually or automatically, and differs from the original by only one word—the target position. They

[2] The annotated data is released with this paper. The dataset contains only *replacement* errors due to the current design of the practice mode in Revita.

feed complete sentences into the model, collect all predictions for the target position, and compare the scores assigned to the original correct word and the incorrect one, e.g., *write* vs. *writes*. Errors should have a lower probability than correct forms. The LM performs much worse than supervised models, especially in case of long syntactic dependencies (Jozefowicz et al., 2016; Marvin and Linzen, 2018; Gulordava et al., 2018). This work was done on Italian, Hebrew, and Russian.

Goldberg (2019) adapted the described evaluation methods and applied them to pre-trained BERT models by masking out the target words. BERT showed high scores on all test cases with subject-verb agreement and reflexive anaphora, except for sentences with relative clauses. The experiments were extended by Wolf (2019) by evaluating the OpenAI Generative Pre-trained Transformer (GPT) of Radford et al. (2018). BERT outperformed the OpenAI GPT on the datasets from Linzen et al. (2016) and Goudalova et al. (2018), but not on the dataset from Marvin and Linzen (2018).

The problem of data scarcity can be addressed by generating artificial training data. Among the existing approaches are oversampling a small learner corpus (Junczys-Dowmunt et al., 2018; Aprosio et al., 2019), utilizing additional resources, such as Wikipedia edits (Grundkiewicz and Junczys-Dowmunt, 2014; Boyd, 2018), or introducing *natural* and *synthetic* noise into error-free data (Belinkov and Bisk, 2017; Felice and Yuan, 2014). *Natural noise* means harvesting naturally occurring errors from the available corpora and creating a look-up table of possible replacements. Using natural noise also tries to imitate the distribution of errors in the available learner corpora. *Synthetic noise* can be generated by probabilistically injecting character-level or word-level noise into the source sentence, as shown in (Lichtarge et al., 2019; Kiyono et al., 2019; Zhao et al., 2019).

Synthetic error generation based on confusion sets extracted from a spellchecker was used by one of the top-scoring systems at the Restricted and the Low Resource tracks at the BEA-2019 Shared task (Grundkiewicz et al., 2019). Both tracks suppose limited use of available learner corpora. This method was compared in (White and Rozovskaya, 2020) with another top scoring approach (Choe et al., 2019) which relies on token-based and POS-based confusion sets extracted from a small annotated sample of the W&I +LOCNESS dataset (Yannakoudakis et al., 2018). Extensive evaluation showed that the methods are better suited for correcting different types of errors. In general, the token- and POS-based pattern method demonstrated stronger results.

If enough training data is available, errors can be generated by back-translation from correct data to data with errors (reverse error correction), which can be modified by additional random noise (Rei et al., 2017; Kasewa et al., 2018; Xie et al., 2018; Kiyono et al., 2019).

3 Data

First, we describe our real learner data. This data was used as the test set for all experiments presented below. Then, we present the method for generating ungrammatical data for training.

3.1 Learner Data

While students perform exercises using the Revita language-learning platform, it continuously collects[3] and automatically annotates *ReLCo*—the longitudinal Revita Learner Corpus (Katinskaia et al., 2020), where each record includes:

- an authentic learner error in the context;
- unique anonymized internal identifiers (ID) of the learner;
- the type of exercise which was practiced.
- the timestamp;

Error category	%
Noun: Case	32.1
Noun: Number	16.4
Adj.: Case	13.0
Verb: Number	10.7
Verb: Tense	10.4
Adj.: Gender	8.8
Verb form	8.2
Verb: Person	6.0
Pron.: Case	7.9
Adj.: Number	7.0

AC category	%
Verb: Tense	31.3
Noun: Number	27.2
Noun: Case	17.5
Verb: Gender	14.7
Verb: Number	13.0
Adj.: Case	10.0
Verb form	9.0
Verb: Mood	6.8
Adj.: Gender	5.9
Pron.: Case	2.3

Table 2: Most frequent grammatical errors and alternative-correct (AC) answers in the annotated dataset.

Revita generates exercises—"cloze", multiple-choice, listening, etc.—with hidden words in each paragraph. Learner answers that differ from the expected answers but have the same lemma are automatically flagged as grammatical errors, e.g., ел

[3] Students are aware that data is collected when they register on the platform.

Data	Sentences	Tokens	Errors per sentence	Grammatical errors	Correct
Real	7 869	120 420	1.9	4 704	693
Simulated	6 891 517	106 767 033	1.7	11 510 977	—

Table 3: The real dataset collected from learners, and the simulated dataset. The column "Correct" shows learner answers, which were manually labeled as correct by the annotators.

"(he) ate" in place of ест "(she/he/it) eats". Our goal is to improve this step—we aim to provide to the learners better feedback on the grammatical correctness of their answers, and in addition to improve the quality of automatic annotation.

Exercises include words of various parts of speech (POS); in this work, we focus only on the inflected POSs. A total of 10K of such flagged answers were manually checked, and annotated as correct or incorrect. Annotation was performed by two native speakers, with 91% agreement. Cases where annotators did not agree were resolved by consensus. Answers with spelling errors or with incorrect lemmas are ignored, since we focus only on grammatical errors (see the most frequent types in Table 2). We label as "unsure" cases when we could not decide whether the answer is correct. There were 194 such answers (2% of the annotated data), and they were not used for evaluation.

We assume that the context for annotation is *one paragraph*—earlier text is not used; all following sentences are also ignored (because they are not seen by the learner at the time of practice).[4] It is important to note that we annotate *jointly* all answers—which may affect each other—given by the learner in the paragraph at the same time during practice. In total, we have collected 3004 paragraphs, with an average of 2.6 sentences per paragraph. We include the same paragraph in the data multiple times if it had different exercises when it was shown to the learners, or if the same exercises were given, but they were answered differently. Statistics about this dataset of real errors are given in Table 3.

The manually annotated data is released to the community.[5] It includes the answers to exercises practiced in 2017–2020 by 150 learners, user IDs (anonymized), timestamps, and the corresponding correct sentences.

[4]For example, if the gender of a pronoun in the current paragraph is answered as feminine, but from the previous paragraph we know that it should be masculine, we do not mark such an answer as an error, if it suits the context of the *current* paragraph.

[5]github.com/Askinkaty/Russian_learner_corpora

3.2 Generating Training Data

As the source of error-free data, we used the open-source "Taiga" Russian corpus (Shavrina and Shapovalova, 2017), which is arranged into several segments based on genre. We used all news segments and a part of a literary text segment. Details about the data are presented in Table 3.

Keeping in line with the current design of Revita's practice mode—where the learner may not change the word order, nor the number of words in the sentence, nor the lemma of the hidden word—we generate errors by replacing some of the words by other random forms from their paradigms. During the pre-processing all sentences are parsed by a rule-based shallow parser, which is implemented as a component of Revita. It identifies which words belong to *chunks*—constructions based on syntactic agreement and government. We use about 30 types of chunks, e.g., `Prep+Adj+Noun` or `Noun+Conj+Noun`.[6]

A synthetic sentence $\widetilde{X}$ is produced from a source sentence $X = (x_1, x_i, ..., x_n)$ with n words by replacing the i-th word x_i by a form from the paradigm of x_i. The word is replaced, if: it has a valid morphological analysis; it is present in a frequency dictionary, which was computed from the entire "Taiga" corpus; and it has an inflected POS. Paradigms are generated by pymorphy2 (Korobov, 2015). Using the paradigm as a confusion set is similar to the approach in (Yin et al., 2020).

For every x_i, we pick a random sample from the uniform distribution. The word x_i is replaced, if it does not belong to a chunk and the picked value is above the threshold $\theta_p = p(error)$ = 0.1. The word x_i is also replaced, if it belongs to a chunk and the picked value is above the threshold $\theta_{p,c} = p(error, chunk)$ = 0.04. The thresholds denote a probability of inserting an error, and their values were chosen to reflect the distributions of errors in chunks and single tokens in the learner data.

[6]For example, in Russian, as in many languages, prepositions govern nouns in a specific case; adjective and noun must agree in gender, number and case; etc.

4 Models

We explore two ways to tackle the problem of scarce data: 1. use a LM in an unsupervised fashion to detect grammatical irregularities; 2. train GED models with supervision on synthetic data.

4.1 BERT as a Masked Language Model

We evaluate BERT as a masked language model (MLM)—to check how well it can distinguish correct answers from grammatical errors in the annotated learner data by performing an unsupervised cloze test, similar to that described by Goldberg (2019). The pre-trained BERT Base[7](Kuratov and Arkhipov, 2019) is used for all experiments.

Joint assessment of answers: We need to assess more than one target word *jointly*, because correctness depends on the joint fills in *all* exercises in a paragraph. Experiments described by Linzen et al. (2016) and Goldberg (2019) mask only *one* target word at a time in the original sentence and the sentence with the error. However, as the following example shows, two different sets of answers can suit the same context, as long as they are considered *jointly*. The words in the brackets are the hints (lemmas), which the user should replace:
"*Я [идти] по улице и [увидеть] пуделя.*"
("*I [walk] down the street and [see] a poodle.*")
The expected answers may be: "*Я **иду** по улице и **вижу** пуделя.*"
("*I **walk** down the street and **see** a poodle.*") But the learner may provide different answers, which are *alternatively correct*, if inserted jointly:
"*Я **шёл** по улице и **увидел** пуделя.*"
("*I **walked** down the street and **saw** a poodle.*")

We adapted the approach of (Linzen et al., 2016; Goldberg, 2019) to our setup and applied two masking strategies: 1. mask *one target* token in a sentence (Table 4, left side) before feeding it to BERT, and 2. mask *multiple targets* to be predicted jointly (right side). We use WordPiece (Schuster and Nakajima, 2012) to segment tokens for BERT, so some target words missing in the pre-trained model's vocabulary are split into sub-tokens. Because of this, we compared the mean log-probabilities of all of the target's sub-tokens. Acc_{err} denotes accuracy calculated on MLM predictions for only erroneous answers. We also evaluated predictions using different BERT layers.

Alternative-correct answers: The method of Linzen et al. (2016) and Goldberg (2019) is based on comparing the model's probabilities predicted for the original word and for the replacement, with the assumption that the replacement is incorrect. This gives us only the absolute difference in probabilities returned by the LM, which cannot be used to determine whether the learner's answer is also correct in the context. When comparing BERT's predictions for the masked original word and the masked alternative-correct word, we conjecture that the model recognizes an alternative answer as grammatical if its predicted probability is *at least as high* as the probability of the expected answer. We also applied two masking strategies (*one target* vs. *multiple targets*), see accuracy Acc_{corr} in Table 4.

4.2 Supervised Model Architecture

Following prior experiments—which show that fine-tuning BERT for NER (Peters et al., 2019) and error detection (Kaneko et al., 2020) gives better performance than using the contextual representation of words from pre-trained BERT—we also fine-tune the pre-trained model. We modified the Huggingface Pytorch implementation of BERT for token classification and the code for the NER task[8] (Debut et al., 2019). Hyper-parameters for fine-tuning BERT are the same as for the NER task: maximum number of epochs is 3, maximum input sequence length is 256, dropout rate is 0.1, batch size is 32, Adam optimizer, and the initial learning rate is set to 5E-5. We split the generated dataset into a training set, a development set, and a test set. Real learner data was *not* used for optimizing hyper-parameters or regularization—only for the final testing.

Tokens: To process words, we did not use the only first sub-token per token, as is usually done when fine-tuning BERT for NER, but assigned the error/correct label of the entire token to all of its sub-tokens. We also tried labeling as errors only those sub-tokens that are actually erroneous, but that did not improve performance. This may be due to the segmentation and BERT's deficiency in capturing morphological features.

Training sequence: We experimented with using one sentence as the training instance (padded or cut to the maximum input length). However, using a paragraph as input decreases training time and gives better performance (see Table 5, where `s` denotes *sentence instances* and `p` means *paragraph instances*). The results were the same with

[7] docs.deeppavlov.ai/en/master/features/models/bert.html

[8] github.com/huggingface/transformers

BERT Layers	One target word			Multiple target words		
	Acc_{err}	Acc_{corr}	$bAcc$	Acc_{err}	Acc_{corr}	$bAcc$
all layers	64.5	61.4	63.0	65.7	57.8	61.8
layer 10	50.9	59.0	55.0	51.9	55.9	53.9
layer 9	50.2	58.4	54.3	51.7	55.8	53.8
layer 8	49.4	60.1	54.8	51.5	57.1	54.3

Table 4: Accuracy of BERT as a MLM on detecting errors. The 3 left columns present results on masking only one target word in the sentence; the 3 right columns present masking multiple learner answers jointly; err and $corr$ denote accuracy for masked grammatical errors and alternative-correct answers, respectively. Balanced accuracy ($bAcc$) is calculated for both classes: errors and alternative-correct answers.

paragraph length from 128 up to 256.

Layers: As multiple studies show that syntactic information is most prominent in the *middle* layers (6-9 for BERT-base), while the final layer is the most task-specific (Rogers et al., 2020; Yin et al., 2020), we also experimented with middle layers from several models with 12, 8, and 6 layers. For the classification task, we use a softmax output layer.

Loss: The training data is very skewed—over 90% of tokens are correct words, so negative examples far outnumber the positive ones. This particularly complicates the process of training and evaluation. To handle this, we use the weighted cross-entropy loss, wCE. It is a variant of cross-entropy where all classes are given weight coefficients:

$$wCE = -\sum_{c=1}^{C} w_c p_c \log \hat{p}_c$$

where the weight of a class c is calculated as: $w_c = \frac{N}{CN_c}$, where N is the total number of samples in the dataset, C is the number of classes, and N_c is the number of samples within the class.

5 Results and Discussion

BERT as MLM: We calculated balanced accuracy scores $bAcc$ for both classes—grammatical errors and alternative-correct answers (see Table 4). The results for the one-target approach are not strictly comparable with the results in Goldberg (2019) because our data includes grammatical errors in many syntactic relations, not only in subject-verb agreement or reflexive anaphora. However, we can conclude that the pre-trained BERT models capture syntactic-sensitive dependencies markedly worse for Russian than for English, especially if multiple target words are masked. Exploring different layers showed that the lower layers of the pre-trained model are weaker at detecting the errors. Figure 1 presents histograms and kernel density estimation over log-probabilities that BERT as MLM assigns to a. errors, b. correct words sampled randomly from the learner corpus (which were not exercised) and c. alternative-correct answers. These three groups of words clearly have different distributions, but they are not easily separable to assess the learner answers in a reliable fashion.

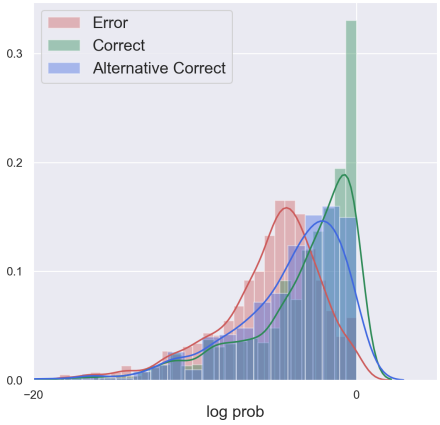

Figure 1: Histograms and kernel density estimations of log-probabilities assigned by BERT as MLM to: grammatical errors, randomly chosen correct words (not used in exercises), and alternative-correct answers.

Fine-tuned BERT: Table 5 presents the metrics calculated on *all words* in the test set and only on the *target words* (with superscript T)—i.e., words which were inserted by the learner: errors or alternative-correct answers. MCC ranges between $[-1, 1]$, and is high if the model correctly predicts a high percentage of negative *and* positive instances (Powers, 2020). We use $F_{0.5}$ because it favors precision over recall; this is important for our task, since providing incorrect feedback on learner answers is far more harmful than no feedback at all. MCC and $F_{0.5}$ do not completely agree, be-

Model	MCC	P	R	$F_{0.5}$	F_1	P^T	R^T	$F_{0.5}^T$	F_1^T	$bAcc^T$
CE+12+p	**63.7**	79.3	84.6	**80.3**	81.9	68.0	77.6	**69.7**	72.5	77.6
CE+12+s	58.5	77.6	81.0	78.2	79.3	66.8	76.9	68.6	71.5	76.9
CE+8+p	**63.4**	79.2	84.4	**80.2**	81.7	68.2	78.1	**70.0**	72.8	78.1
CE+8+s	58.9	77.6	81.4	78.3	79.5	67.1	77.4	68.9	71.9	77.4
wCE+12+p	**55.1**	70.0	87.9	**73.0**	77.9	68.8	74.5	**69.9**	71.5	74.5
wCE+12+s	50.0	68.2	84.4	70.9	75.4	65.9	73.2	67.2	69.4	73.2
wCE+8+p	**56.1**	70.6	88.1	**73.5**	78.4	69.5	75.5	**70.6**	72.4	75.5
wCE+8+s	48.2	67.1	83.9	69.9	74.6	66.3	73.9	67.9	70.0	73.9
CE+12+p+t	66.4	86.3	80.3	85.0	83.2	68.0	77.6	69.7	72.5	77.6
CE+8+p+t	**66.8**	86.5	80.6	**85.3**	83.4	68.4	78.7	**70.2**	73.2	78.7
wCE+12+p+t	65.5	87.2	78.9	85.4	82.8	68.7	76.6	70.1	72.4	76.6
wCE+8+p+t	**66.7**	87.7	79.5	**85.9**	83.4	68.6	77.3	**70.2**	72.7	77.3

Table 5: Results of evaluation of fine-tuned BERT models on assessing grammatical correctness: `CE`—*cross-entropy loss*, `wCE`—*weighted cross-entropy*; the numbers denote the number of layers; `s`—*sentence training instance*, `p`—*paragraph training instance*, `t`—scores after moving decision thresholds. MCC—Matthews correlation coefficient, P and R—macro-averaged precision and recall, $F_{0.5}$ and F_1—macro F-measures, $bAcc$—balanced accuracy. Metrics are calculated for *all* tokens, except where superscript^T—calculated only for the *target* words.

cause $F_{0.5}$ does not take into account the true negatives. We report macro-averaged scores, as they reflect how well the model performs for all classes, which is important for our task—assessing the erroneous vs. the alternative-correct answers. Macro-averaging treats all classes as equal, including the minority class—grammatical errors. We calculate the balanced accuracy $bAcc^T$ on the target tokens (errors and alternative-correct) for the fine-tuned models for comparison with BERT as MLM. We consider a word to be tagged as an error if at least one of its sub-segments was tagged as an error.

The fine-tuned models show better results on evaluating the correctness of learner answers. The best performing models are highlighted. We also report the metrics for the best 4 models after moving the decision thresholds (denoted by `t` in the model name), chosen based on the highest values of $F_{0.5}$ and MCC. The thresholds are shown in Table 6. All evaluation methods show that training with paragraphs outperforms training with sentence instances. This may be due to the wider context available during training and evaluation.

On the target positions, the fine-tuned models perform better with 8 layers, regardless of the loss function, which is consistent with experiments for English (Yin et al., 2020).[9] Performance on the target positions is worse than for all tokens because all models, even trained considering unbalanced data, tend to predict more often that a word is correct, which is true for most of the tokens in a paragraph.

Model	All words	Targets
CE+12+p+t	0.83	0.50
CE+8+p+t	0.83	0.56
wCE+12+p+t	0.98	0.75
wCE+8+p+t	0.98	0.85

Table 6: Decision thresholds for best fine-tuned BERT models.

Comparison with other models: We use the model proposed for GED by (Bell et al., 2019) as the *baseline*—a bi-LSTM trained with a second LM objective and combined with a character-level bi-LSTM model. We took the best performing configuration, which utilizes BERT contextual embeddings. The baseline was trained only on the real learner datasets with cross-validation (CV).

We used RULEC-GEC (see Table 9) as a second dataset to evaluate how well our fine-tuned model can generalize on other learner corpora, despite the fact that the synthetic training dataset was generated to imitate our learner data. RULEC-GEC is a corrected and error-tagged corpus of learner writing. It is almost double in size, has different error types and higher error rate than in our learner dataset. We performed evaluation on all types of

[9] Results with 6 layers are the worst for all models and are not reported in Table 5.

Model	Our dataset				RULEC-GEC			
	P	R	$F_{0.5}$	F_1	P	R	$F_{0.5}$	F_1
Baseline	82.9	70.1	80.0	76.0	84.5	63.4	79.2	72.4
BERT + synthetic data	79.3	84.6	80.3	81.9	72.3	62.4	70.5	68.0
BERT + synthetic data + spellchecker	-	-	-	-	82.1	91.5	83.8	86.5
BERT + real learner data	85.2	78.1	83.7	81.5	96.5	90.9	95.3	93.6

Table 7: Macro precision, recall, $F_{0.5}$, and F_1 evaluated on our learner dataset and RULEC-GEC. "Baseline" refers to a retrained model by (Bell et al., 2019), with using BERT contextualized embeddings. BERT refers to the fine-tuned models, with CE-loss and 12 layers.

Model	Noun	Adj.	Verb	Pron.	Num.
CE+12+p	82.0	79.7	67.9	77.9	73.2
CE+8+p	82.6	79.9	67.9	77.2	70.7
wCE+12+p	86.7	84.3	68.9	87.8	73.2
wCE+8+p	87.9	85.1	70.0	85.1	80.5

Table 8: Accuracy of predicting correctness on the target positions for different parts of speech by the models fine-tuned on synthetic data.

Tokens	Sentences	Errors	Total error rate
206 258	12 480	13 047	6.3%

Table 9: Statistics for the data in RULEC-GEC.

replacement and deletion errors in RULEC-GEC, not only inflection errors.

BERT, fine-tuned on synthetic data, performs comparably with the baseline (see Table 7). It has worse results on RULEC-GEC; however, it is mostly unable to detect spelling errors, as well as other error types, which were not present in the synthetic dataset (preposition, conjunction, and insertion/deletion errors). In combination with the Deep Pavlov spelling correction pipeline,[10] the fine-tuned model can achieve much higher performance without any additional training.

We also experimented with fine-tuning BERT on original learner data with CV. For our dataset, results of the model fine-tuned solely on synthetic data are comparable with the model fine-tuned and tested on the original data with CV. Moreover, recall is better for the model trained on synthetic data. The reason for this might be that our learner data is too scarce. The BERT model fine-tuned and tested solely on RULEC-GEC with CV achieves much better results than any of the other tested systems. We performed this evaluation primarily to compare the performance of fine-tuned BERT and the baseline on the same dataset.

Model confidence: To evaluate confidence of the fine-tuned models, we apply the method of Monte Carlo dropout (Gal and Ghahramani, 2016). By keeping dropout activated at test time, we can repeatedly sample T predictions for every input and estimate the predictive uncertainty by measuring the variance and entropy of the scores. We sampled $T = 20$ scores of the BERT model (CE-loss, 12 layers) fine-tuned on synthetic data for each test input and calculated their variance and entropy.

A deeper analysis is beyond the scope of the paper, but we observe that the scores have higher uncertainty when the models make mistakes in the predictions, see Figure 2. To use the model in Revita, we can compute the entropy of predicted scores and disregard the predictions when the entropy is high. In that case, we can fall back on our standard procedure of evaluation of learner answers. One disadvantage of this method is that it increases the inference time by a factor of T.

Error Analysis: Analysis of errors shows multiple problems experienced by all fine-tuned BERT models. The most prominent are due to inverted word order and long-range dependencies; many verb forms are classified incorrectly (see Table 8), rare names and non-Cyrillic words are mostly classified as errors as well. This result is consistent with the previous research, which showed that fine-tuned BERT struggles with word order errors and verb forms for English (Yin et al., 2020).

Errors are frequently related to conjoined elements in the sentence. For instance, in case of two subjects with one common predicate (e.g., "Peter and John talk on the phone every day."), BERT cannot detect errors in the number of the predicate ("talk" vs. "talks"). Moreover, BERT often marks

[10] docs.deeppavlov.ai

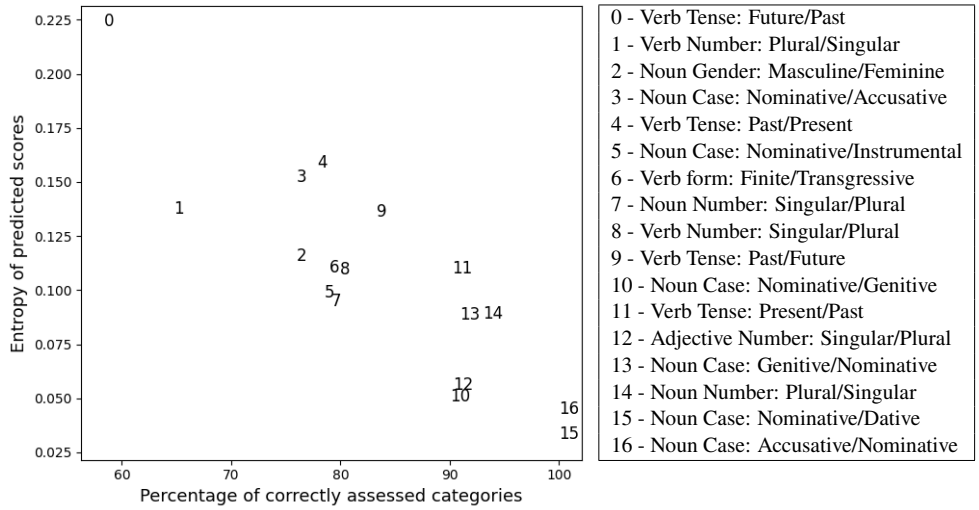

Figure 2: Percentage of 17 the most frequent correctly assessed categories in alternative-correct answers (X-axis) vs. entropy of predicted scores (Y-axis), sampled from 20 BERT models fine-tuned on synthetic data.

an erroneous word as an error along with other words syntactically related to it. This applies to both shallow and long-range relations. So, the presence of an error affects other words which are syntactically related to the erroneous one.

Another interesting problem related to multiple valid possibilities to correct an erroneous sentence. The models for GED, which we have been experimenting with, have no information about where the learners' answers are located in the sentence. In some erroneous sentences, it is possible to correct a hypothetical error (not a wrong learner's answer) and to obtain a corrected sentence with a meaning which is different from the original one but also grammatically valid. When labeling such sentences, fine-tuned BERT can consider an erroneous answer as correct and predict other words in the sentence as errors, which do not agree with the inserted form. For example: "*Я был в Африке и меня там* ***крокодила*** *съел.*" ("*I was in Africa and I was eaten by a* ***crocodile****.*"). The highlighted word "crocodile" should be in the nominative rather than accusative case. However, BERT predicts the word "*меня*" (*"me"*) as error, likely expecting the nominative "*я*" ("*I*"), i.e., changing the meaning of the sentence to *"I was in Africa and I ate a crocodile."*

6 Conclusions and Future Work

We present a study on assessing grammatical correctness in the context of language learning. Our focus is on assessing *alternative-correct* answers to cloze exercises—such answers are given more frequently by the more advanced learners. This work was done with the Russian version of Revita, using a learner corpus collected automatically and annotated manually. We release the corpus to the research community with this paper.

The motivation behind approaching the problem of alternative-correct answers as GED is based on the hypothesis that models for error detection can assess the correctness of potentially valid answers. Because learner data is limited, we experimented with pre-trained BERT as a MLM, and with several BERT models fine-tuned on *synthetic* data, which we generated for the task. The evaluation shows that the pre-trained BERT is not able to assess grammatical correctness of learner answers; the performance for Russian is considerably lower than for similar experiments with English. Comparison with a baseline model and evaluation on another leaner corpus demonstrates that fine-tuning on synthetic data is a promising approach and generalizes well.

We plan to improve the generation of synthetic data based on error analysis, to cover a wider range of error types, and continue work on estimation of the confidence of the model predictions, since it is critical to provide reliable feedback to the learners. We also plan to specify the *positions* of answers as part of the model's input, which is natural for the exercise-oriented set-up in Revita.

Acknowledgements

This work was supported in part by the Academy of Finland, HIIT—Helsinki Institute for Information Technology, and Tulevaisuus Rahasto (Future Development Fund), University of Helsinki.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Alessio Palmero Aprosio, Sara Tonelli, Marco Turchi, Matteo Negri, and Mattia A Di Gangi. 2019. Neural text simplification in low-resource conditions using weak supervision. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 37–44.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*.

Samuel Bell, Helen Yannakoudakis, and Marek Rei. 2019. Context is key: Grammatical error detection with contextual word representations. *arXiv preprint arXiv:1906.06593*.

Adriane Boyd. 2018. Using wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84.

Mengyun Chen, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020. Improving the efficiency of grammatical error correction with erroneous span detection and correction. *arXiv preprint arXiv:2010.03260*.

Martin Chodorow, Joel R Tetreault, and Na-Rae Han. 2007. Detection of grammatical errors involving prepositions. In *Proceedings of the fourth ACL-SIGSEM workshop on prepositions*, pages 25–30. Association for Computational Linguistics.

Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. 2019. A neural grammatical error correction system built on better pre-training and sequential transfer learning. *arXiv preprint arXiv:1907.01256*.

Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, , Jamie Brew, and Thomas Wolf. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Mariano Felice and Zheng Yuan. 2014. Generating artificial errors for grammatical error correction. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 116–126.

Jennifer Foster and Carl Vogel. 2004. Parsing ill-formed text using an error grammar. *Artificial Intelligence Review*, 21(3-4):269–291.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.

Yoav Goldberg. 2019. Assessing BERT's syntactic abilities. *arXiv preprint arXiv:1901.05287*.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2014. The WikEd error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction. In *International Conference on Natural Language Processing*, pages 478–490. Springer.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. *arXiv preprint arXiv:1803.11138*.

Jue Hou, Maximilian W Koppatz, José Marıa Hoya Quecedo, Nataliya Stoyanova, Mikhail Kopotev, and Roman Yangarber. 2019. Modeling language learning using specialized Elo ratings. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, ACL: 56th annual meeting of the Association for Computational Linguistics*, pages 494–506.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. 2018. Approaching neural grammatical error correction as a low-resource machine translation task. *arXiv preprint arXiv:1804.05940*.

Masahiro Kaneko and Mamoru Komachi. 2019. Multi-head multi-layer attention to deep language representations for grammatical error detection. *arXiv preprint arXiv:1904.07334*.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. *arXiv preprint arXiv:2005.00987*.

Masahiro Kaneko, Yuya Sakaizawa, and Mamoru Komachi. 2017. Grammatical error detection using error-and grammaticality-specific word embeddings.

In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 40–48.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a right: Generating better errors to improve grammatical error detection. *arXiv preprint arXiv:1810.00668*.

Anisia Katinskaia, Sardana Ivanova, and Roman Yangarber. 2020. Toward a paradigm shift in collection of learner corpora. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 386–391.

Anisia Katinskaia, Sardana Ivanova, Roman Yangarber, et al. 2019. Multiple admissibility in language learning: Judging grammaticality using unlabeled data. In *The 7th Workshop on Balto-Slavic Natural Language Processing Proceedings of the Workshop*. The Association for Computational Linguistics.

Anisia Katinskaia, Javad Nouri, and Roman Yangarber. 2017. Revita: a system for language learning and supporting endangered languages. In *6th Workshop on NLP for CALL and 2nd Workshop on NLP for Research on Language Acquisition, at NoDaLiDa*, Gothenburg, Sweden.

Anisia Katinskaia, Javad Nouri, Roman Yangarber, et al. 2018. Revita: a language-learning platform at the intersection of ITS and CALL. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).

Anisia Katinskaia and Roman Yangarber. 2018. Digital cultural heritage and revitalization of endangered Finno-Ugric languages. In *Proceedings of the 3rd Conference on Digital Humanities in the Nordic Countries*, Helsinki, Finland.

Satoru Katsumata and Mamoru Komachi. 2020. Stronger baselines for grammatical error correction using pretrained encoder-decoder model. *arXiv preprint arXiv:2005.11849*.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. 2019. An empirical study of incorporating pseudo data into grammatical error correction. *arXiv preprint arXiv:1909.00502*.

Mikhail Korobov. 2015. Morphological analyzer and generator for Russian and Ukrainian languages. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 320–332. Springer.

Yuri Kuratov and Mikhail Arkhipov. 2019. Adaptation of deep bidirectional multilingual transformers for Russian language. *arXiv preprint arXiv:1905.07213*.

John Lee and Stephanie Seneff. 2008. Correcting misuse of verb forms. In *Proceedings of ACL-08: HLT*, pages 174–182.

Yiyuan Li, Antonios Anastasopoulos, and Alan W Black. 2020. Towards minimal supervision BERT-based grammar error correction. *arXiv preprint arXiv:2001.03521*.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. *arXiv preprint arXiv:1904.05780*.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Zhuo-Ran Liu and Yang Liu. 2017. Exploiting unlabeled data for neural grammatical error detection. *Journal of Computer Science and Technology*, 32(4):758–767.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. *arXiv preprint arXiv:1808.09031*.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. *arXiv preprint arXiv:1910.00353*.

Matthew E Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, page 1756–1765.

Matthew E Peters, Sebastian Ruder, and Noah A Smith. 2019. To tune or not to tune? Adapting pretrained representations to diverse tasks. *arXiv preprint arXiv:1903.05987*.

David MW Powers. 2020. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. *arXiv preprint arXiv:1704.07156*.

Marek Rei, Gamal KO Crichton, and Sampo Pyysalo. 2016. Attending to characters in neural sequence labeling models. *arXiv preprint arXiv:1611.04361*.

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns. *arXiv preprint arXiv:1707.05236*.

Marek Rei and Anders Søgaard. 2019. Jointly learning to label sentences and tokens. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6916–6923.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. *arXiv preprint arXiv:1607.06153*.

Marek Rei and Helen Yannakoudakis. 2017. Auxiliary objectives for neural error detection models. *arXiv preprint arXiv:1707.05227*.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how BERT works. *arXiv preprint arXiv:2002.12327*.

Alla Rozovskaya and Dan Roth. 2019. Grammar error correction in morphologically rich languages: The case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and Korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Tatiana Shavrina and Olga Shapovalova. 2017. To the methodology of corpus construction for machine learning: "Taiga" syntax tree corpus and parser. *Corpus Linguistics*, page 78.

Max White and Alla Rozovskaya. 2020. A comparative study of synthetic data generation methods for grammatical error correction. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 198–208.

Thomas Wolf. 2019. Some additional experiments extending the tech report "Assessing BERT's syntactic abilities" by Yoav Goldberg. Technical report.

Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Y Ng, and Dan Jurafsky. 2018. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 619–628.

Helen Yannakoudakis, Øistein E Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. Developing an automated writing placement system for ESL learners. *Applied Measurement in Education*, 31(3):251–267.

Fan Yin, Quanyu Long, Tao Meng, and Kai-Wei Chang. 2020. On the robustness of language encoders against grammatical errors. *arXiv preprint arXiv:2005.05683*.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. *arXiv preprint arXiv:1903.00138*.