DOI: 10.15514/ISPRAS-2023-35(5)-14

Here We Go Again: Modern GEC Models Need Help with Spelling

V.M. Starchenko, ORCID: 0009-0004-6638-9124 <*vstarchenko@hse.ru*>
A.M. Starchenko, ORCID: 0000-0003-1650-7597 <*aleksey-starchenko@mail.ru*>

HSE University,
20, Myasnitskaya st., Moscow, 101000 Russia

Abstract. The study focuses on how modern GEC systems handle character-level errors. We discuss the ways these errors effect the performance of models and test how models of different architectures handle them. We conclude that specialized GEC systems do struggle against correcting non-existent words, and that a simple spellchecker considerably improve overall performance of a model. To evaluate it, we assess the models over several datasets. In addition to CoNLL-2014 validation dataset, we contribute a synthetic dataset with higher density of character-level errors and conclude that, provided that models generally show very high scores, validation datasets with higher density of tricky errors are a useful tool to compare models. Lastly, we notice cases of incorrect treatment of non-existent words on experts' annotation and contribute a cleared version of this dataset. In contrast to specialized GEC systems, LLaMA model used for GEC task handles character-level errors well. We suggest that this better performance is explained by the fact that Alpaca is not extensively trained on annotated texts with errors, but gets as input grammatically and orthographically correct texts.

Keywords: GEC, validation; spellcheck; preprocessing; generated datasets.

For citation: Starchenko V.M., Starchenko A.M. Here We Go Again: Modern GEC Models Need Help with Spelling. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 5, 2023. pp. 215-228. DOI: 10.15514/ISPRAS-2023-35(5)-14.

Acknowledgements. This work is an output of a research project implemented as part of the Basic Research Program at the National Research University Higher School of Economics (HSE University) in 2023. We are grateful to Anastasia Vyrenkova and Olga Lyashevskaya for extensive discussion and comments on the paper, to Olga Vinogradova for help with the research in general. All possible errors made in the study are exclusively on us.

Проблема валидации современных систем исправления грамматических ошибок: случай ошибок на уровне символов

В.М. Старченко, ORCID: 0009-0004-6638-9124 <*vstarchenko@hse.ru*>
А.М. Старченко, ORCID: 0000-0003-1650-7597 <*aleksey-starchenko@mail.ru*>

Национальный исследовательский университет «Высшая школа экономики»,
101000, Россия, г. Москва, ул. Мясницкая, д. 20

Аннотация. Исследование сосредотачивается на проблеме того, как современные системы исправления грамматических ошибок обрабатывают ошибки на уровне слова. Работа обсуждает, как подобные ошибки могут взаимодействовать с эффективностью модели, и оценивает, как модели с разными архитектурами справляется с ними. Делается вывод о том, что специализированные системы исправления грамматических ошибок сталкиваются с проблемами при исправлении ошибок, приводящих к созданию несуществующих слов, и что предобработка с помощью простой системой обработки подобных ошибок значительно улучшает общую эффективность модели. Для оценки этого работа модели тестируется для нескольких валидационных датасетах. Вдобавок к валидационному датасету соревнования CoNLL-2014 в работе предлагается синтетический датасет с повышенной плотностью ошибок на уровне слова. На основании сравнения эффективности модели на двух датасетах, работа делает вывод о том, что валидационные датасеты с высокой плотностью ошибок, представляющих проблему для моделей, — это полезный инструмент для сравнения моделей. Кроме того, работа указывает на случаи некорректной аннотации несуществующих слов в разметке экспертов и предлагает очищенную версию датасета. В отличие от специализированных систем исправления грамматических ошибок, модель LLaMA, используемся для задачи исправления грамматических ошибок хорошо справляется с ошибками на уровне слова. Мы предполагаем гипотезу, в соответствии с которой этот результат объясняется тем фактом, что эта модель не обучается на специальной аннотированной выборке, содержащей ошибки, а получает в качестве входа грамматически и орфографически корректные тексты.

Ключевые слова: автоматическое исправление грамматических ошибок; валидация; спеллчек; предобработка; синтетические датасеты.

Для цитирования: Старченко В.М., Старченко А.М. Проблема валидации современных систем исправления грамматических ошибок: случай ошибок на уровне символов. Труды ИСП РАН, 2023, том 35 вып. 5, с. 215-228 (на английском языке). DOI: 10.15514/ISPRAS–2023–35(5)–14.

Благодарности. Исследование осуществлено в рамках Программы фундаментальных исследований НИУ ВШЭ в 2023 году. Мы благодарны А. С. Выренковой и О. Н. Ляшевской за подробные комментарии к работе, О. Н. Виноградовой за помощь в исследованиях. Все возможные ошибки, которые могли остаться в настоящей работе, исключительно на нашей совести.

1. Introduction

Tools for GEC (Grammatical error correction) tasks have greatly improved over recent decades. In terms of metrics, modern big language models outperform a human annotator in the GEC task [1]; overviews [2-4] present the performance growth at different stages. GEC models are however still noticed to fail in correcting several types of errors that would be easily and necessarily corrected by a human [1].

Despite the part “grammatical” in GEC, the task is usually understood wider than the mere correction of illicit grammar use. As the expected result is a text judged natural by a native speaker, spelling, punctuation, word choice, stylistic and other types of errors are treated, as well.

One must note that the best-performing modern models for GEC show pure results with character-level errors, and this problem had been preserved during the last decade [5-7]. If the errors ranked according to their difficulty, this type is considered one of the easiest to correct [8].

Consider a spelling error in Table 1, which the GECToR model [9] fails to correct (*diag***o***nosed* instead of *diagnosed*). In contrast, several other errors are successfully handled, including article use, word form selection and phrasal verbs. Notice that the error in Table 1 is not challenging to detect because it results in a non-existent word, and the closest candidate in terms of Levenshtein distance is a required one. This type of errors is effectively handled with a number of tools performing with a quality acceptable for practical use for a very long time [10-11].

Table 1. Example of the failure and successes of a GEC model

source	When we are **diagonosed** out with certain genetic disease , are we suppose to disclose this result to our relatives ?
corrected = target	When we are **diagonosed** with a certain genetic disease, are we supposed to disclose this result to our relatives ?

Although some researchers apply spellcheckers or character-based models as a part of preprocessing [12-13], [7], [14], it is still not a common practice for modern GEC. For example, the possibility of preprocessing of spelling is not discussed in the recent detailed overview of approaches to GEC [4].

In this study we focus on how character-level, primarily spelling errors affect the output of best-performing GEC models with different architecture. We test 3 SOTA GEC systems with different architecture: GECToR (large) [9], BART (large) [15], and T5 (base) [16]. We also add LLaMA 7B model [17-18] fine-tuned as Alpaca 7B [19]. Large language models like LLaMA or GPT [20] have been recently tested for multiple tasks, including GEC, though at the moment they exhibit lower performance than other models [21][1] show for English[2]. Of other modern SOTA GEC system we do not separately discuss the SynGEC [23], as the cited study shows that the innovations introduced by this complex model to regular transformer-based baseline / BART worsen the performance on spelling errors [23: 2525]. The evaluation of GEC-DI [24], which has been very recently released and suggested to us by one of anonymous reviewers, we leave to the further studies.

Table 2 presents performance of the models, evaluated on the validation dataset for CoNLL-2014 [25] with annotation by 10 experts [26]. F0.5 metric is used, which is argued to represent human judgments well [27-29].

Table 2. Performance of SOTA GEC systems and human experts

model name	$F_{0.5}$
BART	78.04
GECToR	76.82
T5	74.38
LLaMA	68.58
human experts	72.58

Three best-evaluated models: BART, GECToR and T5, show higher scores than human experts do with respect to each other. Yet, we are going to confirm that they perform imperfectly with character-level errors.

Elaborating on the nature of the spelling pitfall of big language models, we notice that both training and especially validation datasets for GEC tasks are noisy when dealing with character-level errors. We contribute a cleared version of CoNLL-2014 validation dataset [25] (its 10-annotators version [26]) and a synthetic dataset with a higher density of spelling errors (but also including all other types of errors), which can be used for testing the impact of this kind of errors. We further suggest that such datasets with high error density are a useful tool to test models that generally show very high performance over tricky types of errors.

Based on both datasets we show that all the tested models designed specifically for GEC show higher performance with spelling errors corrected at the preprocessing stage. In contrast, results of LLaMA model, despite its purer performance in general, is almost not affected with the preprocessing.

The rest of the paper is organized as follows: Section 2 describes the ways in which character-level errors interact with the performance of models. Section 3 discusses the representation and the source of this kind of errors in training and validation data. Section 4 presents three datasets we work with. Section 5 presents an experiment that evaluates the influence of preprocessing of spelling errors on the performance, based on these datasets. Sections 6 interprets the experiments and presents the discussion. Section 7 is the conclusion.

[1] We use a prompt different from the one suggested for GPT 3.5/4 in [21], as our prompt gives a higher score: $F_{0.5}$*=68.58* compared to $F_{0.5}$*=64.34*. The used prompt is: *You will receive a text in English and you must check whether it contain any errors, according to English language rules. Return a corrected version of the text. Don't correct stylistic errors. Do not correct sentences that may be correct in some context. The final text should not contain errors.*

[2] Though s.f. [22] for Swedish, for which GPT 3 outperforms all other models.

2. Relationship between model performance and misspellings

In this section we discuss how character-level error may affect the performance of a model based on the CoNLL-2014 validation dataset.

We only restrict ourselves to the errors resulting in non-existent words. In most cases such errors result from misprints (*wth* instead of *with*, *otherm* instead of *other*), problems with spelling orthographically difficult words (*hypertesion* / *hypertention* instead of *hypertension*, *percieved* instead of *perceived*) and the influence of the native language of an author (e. g. insertion of *o* in consonants clusters by Singapore students: *tech**o**nology*, *diag**o**nosed*). Rarer an error emerges as a result of a morphological process including creating a non-existent word form (plural *medias* instead of *media*) or derivation (*disclosement* instead of *disclosure*). Supposed misprints that lead to the use of an existing word which does not fit the context (*brunch* instead of *bunch*) are considered as word choice errors and therefore are not discussed.

Noticeably, in most cases relevant errors of the considered dataset are not ordinary misprints. More than in half cases, misspelling result from inability of a speaker to deal with phonology—orthography incongruity or forming a wrong morphological pattern, rather than from their inaccuracy. While some of these misspellings still do not fall under the narrow definition of grammatical errors (that is, related to ungrammaticality), they can definitely be viewed as a part of the language system (undertrained mental phonological—orthographic interface) and not an accidental typesetting problem.

Proceeding to the interaction of the character-level errors and model performance, trivially, a model may fail to correct a spelling error (Table 3, BART [15]) and leave a incorrectly spelled word as it is.

Table 3. Example of a simple spelling error

source & corrected	However , it is a good practice not to **intesively** use social media all the time .

In some cases, the model tries to deal with a misspelling, but fails to fix it correctly. In the example in Table 4 by BART [15], instead of inserting the missing letter *r* into the word *concurrently*, the model replaces the whole word, which leads to a semantic distortion of the source sentence. Preprocessing of the text with a spellchecker prevents the model from this alteration.

Table 4. Example of a spelling error leading to semantic distortion

source	… he or she **concurently** has a knowledge about others .
corrected	… he or she **definitely** has knowledge about others .
spellcheck + corrected	… he or she **concurrently** has knowledge about others .

Lastly, a misspelling may affect processing of other types of errors, either close or long-distance. In the example in Table 5 by GECToR [9], in addition to the leaving the misspelling in the word *dilenma*, the output of the model contains collocation *feel into*, which is syntactically related to the word *dilemma* and infelicitous in the context. Furthermore, the model does not handle the word *reflects*, which is semantically incorrect and stands further away from the misspelling in the sentence. If the misspelling is corrected prior to model application, both *reflect* and *feel into* are handled better, although the latter case is still an imperfect correction.

Table 5. Example of spelling error interacting with other error types

source	During that period , if one of the family member reflects genetic disorder symptoms , he will fell in an ethical **dilenma** for sure .
corrected	During that period , if one of the family members reflects genetic disorder symptoms , he will feel into an ethical **dilenma** for sure .
spellcheck + corrected	During that period , if one of the family members has genetic disorder symptoms , he will feel in an ethical **dilemma** for sure .

Summing up, the output of the discussed GEC model is influenced by spelling errors at different levels, starting from the mere inability to handle spelling and up to preventing correction of other errors at the scale of the whole sentence. Similar distortions can be noticed for other considered models, as well.

In the next section we suggest an explanation for this notorious inability to correct character-level errors, which at least partially lies in the annotation for training and validation datasets.

3. Noise in training and validation data

3.1 Validation data

CoNLL-2014 dataset contains 137 character-level errors that produce non-existent words. Despite elaborate and thorough annotation of different types of errors in other domains, the coverage of character-level errors producing non-existent words in the annotation is not high. Of all them, 94 were missed by at least one annotator, 266 cases of unspotted misspellings were found in total. It means that more than a half of character-level errors cannot be accounted properly during the evaluation process.

Table 6. Number of uncorrected non-existent words grouped by number of annotators

annotators	1	2	3	4	5	6	7	8	9	total
errors	32	23	10	9	8	5	5	1	1	94

The detailed statistics on how many errors were missed by the annotators is presented in Table 6. One can notice that some errors are remarkably stealthy, unnoticed by most of annotators. The highest scores are: 9 annotators, *newpaper* — *newspaper*; 8 annotators *techonology* — *technology*; 7 annotators *subconsiously* — *subconsciously*, *covenient* — *convenient*, *againt* — *against*, *simliar* — *similar*, *acccount* — *account*.

The high quantity of non-annotated errors poses a problem for validation. This problem is especially significant due to the validation procedure, used in the setup with multiple annotations [26].

In order to account for multiple annotations by multiple experts, the output of the model is evaluated over all annotations and the highest $F_{0.5}$-score is assigned to the model.

This approach is suggested to capture the cases in which annotators correct an error in different yet equally grammatical ways. Otherwise, the availability of paraphrases with similar meanings could not be properly accounted for.

However, in case of inaccuracy in the annotation, all other equal, it is the erroneous target sentence that receives the highest score for the model that does not make a correction. Thus, presence of both correct and incorrect options among annotations yields to indistinguishability of correctly and incorrectly working models.

To avert this problem, we contribute a new version of the dataset, in which all listed cases of missed character-level errors are manually added to the annotation[3].

Notice that such corrections may not be considered as interfering with a personal choice of a rarer yet well-formed construction by the expert. All the corrected words are not found in any dictionary of Standard English (and are not among commonly used spellings that are not yet represented in dictionaries) and therefore must be viewed as overlooked unintentionally. The exact inserted annotations, including the error type, are based on the most frequent option among annotators who corrected a particular error.

[3] Additionally, rare occasions of misspellings in the corrections suggested by annotators were fixed.

3.2 Training data

Having found from the validation dataset that experts often disregard spelling errors, one could suppose that training datasets must show even more noise of this type. Training datasets are much larger and are often annotated much less thoroughly (for example, one of frequently used training datasets cLang-8 [30-31] was annotated by language learners).

In contrast, training datasets appear to be much clearer with respect to character-level errors.

To estimate it, we check the training part of the FCE dataset [32], which contains about 18k sentences from texts by ESL learners, annotated by a single expert. For a random subset of the dataset, we automatically located all words that are marked as non-dictionary ones and then manually annotated about whether they contain an error.

Table 7 compares the number of uncorrected character level errors in training and validation datasets and the density of sentences with character-level errors (both corrected and uncorrected).

Table 7. Number of uncorrected character-level errors and the overall density of character-level errors in training and validation datasets

	uncorrected errors, % of all character-level errors	**% of sentences with character-level errors of all sentences**
FCE, training	3%	13.5%
CoNLL-2014[4], validation	19% per annotator 69% for 10 annotators	8.6%

Table 7 shows that in the validation dataset, character-level errors are not included in the annotation much more often than in the training dataset. Consequently, even if this kind of errors is corrected well by the model, it will be difficult to evaluate when compared with models that demonstrate lower performances. This difficulty is enlarged with a low density of character-level errors in the dataset (only 8.6% of sentences in the dataset have them).

On the other hand, some proportions of uncorrected character-level errors are included in the training dataset and thus may impact performance of the model.

3.3 The source of noise and a way to prevent it

Spelling errors are in most cases easy to correct. If there are no errors of other types or poor word choice in a fragment, annotators perfectly agree on how misspellings must be corrected. The main problem is to notice them.

It is known that a person does not read familiar words letter by letter, but processes words or at least parts of words as a whole [33], [34]. For this reason, slight distortions of visual appearance of words are not necessarily perceived while regular reading. In contrast, grammatical errors which lead to infelicity at the sentence-level are expected to be conceived during regular reading easier.

Therefore, correcting spelling errors practically requires from an expert to process the text twice, performing both natural reading and the other task, which due to its untypicality takes more effort.

It is natural that without constant conscious effort even an utmost high-skilled reader is going to miss some of character-level errors. Provided that an expert is also expected to annotate other types of errors, the doing so is inevitable.

As for the testing datasets, one can notice that because of their big sizes and costly annotation process, they are usually partially automatically annotated, which naturally includes spellcheck. As a result, the percentage of non-annotated character-level errors in the training dataset is lower.

[4] Based on the dataset with 10 annotators, missed errors per one annotator are calculated as total number of errors divided by number of annotators: *266/10=26.6*. The value of 69% for 10 annotators is calculated for errors that were missed by at least one annotator.

The problem with the higher can be fixed by adding a simple spellchecker that locates non-existent words in am annotator's interface. It must not be a more complicated tool, as it may interfere with the way an expert corrects errors.

4. Datasets

4.1 Introduction of three datasets

Further on, we will focus on the problem of character-level errors in validation data. We elaborate on the problem of partially correct annotation of the validation dataset. Then we proceed to the problem of low density of a particular type of errors in a dataset, which does not allow to test a model over this type of errors properly.

As discussed in Section 3.1, we build on the validation dataset for the CoNLL-2014 [25] and use its version that was independently annotated by 10 experts [26]. Henceforth, we call it Original dataset.

By correcting inaccuracies in annotation, discussed in Section 2, we create **Corrected dataset**[5].

Lastly, we generate **Synthetic dataset**[6] with a higher density of spelling errors in order to highlight the trends that emerge from comparing Corrected and Original datasets.

4.1 Generation of Synthetic dataset

We build on the CoNLL-2014 validation dataset, rather than create a new one with only synthetically induced errors, to capture the interaction of character- and word-level errors, described in Section 2. We preserve all non-character-level errors, in order to capture their interaction with the character-level ones.

We rely on the algorithm for generating datasets suggested in [35], that is, probabilistically introduce spelling errors in the source sentences at a rate of 1–3 per sentence, randomly selecting deletion, insertion, replacement, or transposition of adjacent characters for each introduced error.

The new density of character-level errors is much higher than the one in Original dataset. Yet it is not unrealistically high and does not make texts impossible to understand.

Corrections to all errors induced into the dataset were added to each annotation.

5. Performance of the models on character-level errors with different validation datasets

5.1 Experiment 1: Original and Corrected datasets

Table 8. Performance of four models on the original and corrected validation datasets, with and without preprocessing

Model name	**original dataset**		**corrected dataset**	
	only model, *$F_{0.5}$*	**spellchecker + model, *$F_{0.5}$***	**only model, *$F_{0.5}$***	**spellchecker + model, *$F_{0.5}$***
BART	78.04	78.47	78.07	78.57
GECToR	76.82	77.03	76.84	77.13
T5	74.38	74.68	74.44	74.81
LLaMA	68.58	68.58	68.77	68.78

[5] drive.google.com/drive/folders/169Xvvgn4eBIhSIzYjPE8YsTL93JwjlB2

[6] drive.google.com/drive/folders/1lruoHhAyTrvMaJniAUz0I6G6H9hF57dP

5.1.1 Setup

Noise in training and validation data affects performance and evaluation of a model.

Noise in the training data is expected to worsen its overall result. In order to evaluate how good a model performs at the task of correcting character-level errors, we compare its metrics on validation to the metrics of the model after preprocessing of the source with a simple spellchecker (its architecture is described in more detail in Section 5.1.2).

Noise in the validation data may not allow us to evaluate the performance of a model properly. This problem is aggravated by the way in which multiple unequal annotations are accounted for (see Section 3.1). This approach neatly captures the possibility of variation between different grammatical options of error correction. However, if the annotation includes an erroneous option, it is not necessarily outweighed by corrections of other annotators. That is, if at least one of the annotators missed an error, everything else equal, it may be this annotation which will be accounted for during validation.

To evaluate the impact of the noise in the validation dataset, we compare performance of models on Original and Corrected datasets. For each model, we separately evaluate the performance of a model on its own and the performance of the system of both a spellchecker as a preprocessing tool and the model.

Before proceeding to the scores of the models, we describe how the spellchecker system is organized.

5.1.2 Spellchecker

A spellchecker used for preprocessing should eliminate non-existent words and not affect the rest of the text. Non-existent words are a type of errors which is handled well by different spelling correction systems [36]. On the other hand, big GEC models under consideration handle non-character-level errors, including word choice and discourse incongruence, well. For this reason, spelling errors that lead to creation of an existent word (bunch — brunch) are not corrected and are left to GEC models. We also do not correct spelling issues related to British / American orthography differences like the contract of -ise and -ize derivational suffixes (organised vs. organized).

In the outlined setup, the most reasonable is a dictionary-based approach, which is not expected to creatively alter the source text.

To enlarge the dictionary, we use multiple available spellcheckers, showing high quality on the task of correcting non-existent words: *hunspell*[7], *autocorrect*[8] and *spellchecker*[9].

5.1.3 Evaluation

Table 8 presents how models perform on the original and corrected validation datasets with and without preprocessing by the spellchecker.

The difference in the evaluation results is not large, which may be expected. 94 changes in 116 of 1342 sentences have been made, so the changes on the third–forth significant digit is reasonable. Despite modest differences of scores, some feasible trends can be noticed.

Firstly, the performance on the original and the corrected datasets either differs just slightly or is higher for the latter (up to $\Delta F_{0.5}=0.19$). Models, for which evaluation grew, are better at correcting spelling errors (see the next section form more detailed discussion) than whose performance didn't change significantly. Provided all that, one can conclude that in the new version of the dataset models are being punished less for correcting non-existent words and exhibit higher scores.

[7] Availabe at: https://pypi.org/project/hunspell/.

[8] Availabe at: https://github.com/filyp/autocorrect/.

[9] Availabe at: https://pypi.org/project/pyspellchecker/.

Noticeably, in combination with spellchecking, Corrected dataset produces consistently higher scores than Original dataset does. This result confirms that the corrected dataset is more sensitive to the character-level errors correction and awards better models that show higher quality with this type of errors.

Secondly, all models except for LLaMA perform better when preprocessed with a spellchecker. The growth of the score is low (0.21–0.5), but consistent and goes along with the hypothesis that these models do not perform well for character-level errors.

This means that applying a simple spellchecker to SOTA GEC models results in higher scores and this improves their overall performance.

Ultimately, we test our models against a synthetic dataset that has a higher density of character-level errors to provide a more robust confirmation of our hypothesis.

5.2 Experiment 2: Synthetic dataset

We present the performance of models with and without spellchecker preprocessing in Table 9.

Table 9. Performance of models with and without preprocessing, dataset with the high density of spelling errors

model	**only model, $F_{0.5}$**	**spellchecker + model, $F_{0.5}$**
BART	76.79	**83.6**
GECToR	75.34	82.3
T5	76.89	81.77
LLaMA	**82.25**	82.68

For Synthetic dataset, scores with and without spellchecker-assisted preprocessing are significantly higher than for Original or Corrected dataset. It is expected, provided that the former contains by far more character-level errors than the latter two.

Evaluation reveals that four models handle character-level errors to a different degree. BART and GECToR, though best performing on the Original and Corrected dataset, lowered their results on the Synthetic dataset. In contrast, adding multiple character-level errors increased the scores in T5 and LLaMA, meaning that the models are better at correcting these types of errors.

Three models: BART, GECToR and T5 show a considerable growth in metrics in combination with spellchecking. It strikingly differentiates them from LLaMA, which is less affected by the spellcheck. While on the original dataset, LLaMA model performs worst, the dataset with the high density of character level errors promotes it on top. The score of the model with and without a spellchecker are almost equal, meaning that LLaMA perfectly handles character-level errors induced into the dataset.

Yet the spellcheck preprocessing allows the BART model to regain its first place. Therefore, for Synthetic dataset, SOTA GEC model combined with a simple spellchecker shows the best performance, while all three specialized GEC models without spelling check perform quite poorly.

6. Discussion

The experiment performed in the previous section confirms the hypotheses made in Section 5.1. Three big models, trained specifically for the GEC task: BART, GECToR and T5 perform worse than a simple spellchecker, when dealing with non-existent words. The experiment also allows to differentiate between these three models: T5 performs on this task better than two other models.

Quite the opposite, LLaMA perfectly deals with character-level errors.

This result contradicts the idea that big word or sentence-level models perform bad with character-level errors, suggested in different studies [5-7].

What distinguishes LLaMA from all other models, making it good in correcting one (or, possibly, some) types of errors, but leaving it with a generally worse score?

One possible explanation is exposure or lack of exposure of a model to special training GEC datasets. Specialized GEC models are trained on grammatically and orthographically incorrect input, partially inaccurately annotated. Errors in annotation may lead to incorrect patterns being learned by a model. In contrast, LLaMA is (mostly) trained on grammatically correct texts produced by native speakers and is not generally expected to produce ungrammatical text if not asked otherwise (and does not do so in our data).

Therefore, LLaMA usually does not preserve or produce wrongly spelled or ungrammatical output. What it is often punished for by the metrics is being over-creative, producing sentences that are too different from the initial ones. If a dataset is annotated by an expert who aims to keep the initial sentence as close to the original as possible, provided its grammaticality, a creative correction is going to receive a lower score. On the other hand, in the task of returning of not just a grammatically correct, but also natively sounding English sentence (for JFLEG dataset [37]), large language models like LLaMA or GPT models may perform better than specialized models [21]. On the other hand, a creative correction produced by such model may become semantically unequal to the source, making this correction erroneous.

Further tuning of the large language models is therefore not aimed at correcting grammatical or orthographic errors in the input, but rather restraining it from changing the source too much.

7. Conclusions

Our study evaluated the performance of modern SOTA GEC systems on character-level errors. We described how this type of errors interferes with the performance of GEC systems and confirmed that they still struggle through handling character-level errors, like they deed over recent decade [5-7]. In contrast to the cited studies, we however notice that not all large language models perform badly with character-level errors: LLaMA, though being worse in the GEC task in general, performs on this particular error type better. We relate this difference to the exposure to annotated ungrammatical texts, which contain noise in training data.

An immediate practical output of the study is the suggestion of performing spellcheck preprocessing as a common practice with GEC models. Some studies do so for English [12-13], [7], [14] and it is a regular practice for Chinese because of the peculiar traits of its graphical system [38-39]. Still, handling character-level errors is not discussed in many recent studies.

In a longer-term perspective, this suggestion cannot be considered most suitable: one could desire for a big language model to correct all kinds of errors, rather than just a spellcheck including system. Suggested steps to achieve this result are to clean training (at least from character-level errors) and validation datasets for specialized models.

The last important result is that the sensitivity of a particular validation dataset may not be sensitive enough to evaluate performance of a model for a particular type of errors. Study [1] lists several types of errors that modern models are unable to handle adequately, including the correction of unnatural phrases, correction of patterns requiring information about sentence structure, and correction of errors that involve inter-sentence relationships. In such cases, to capture the difference in model performance, validation datasets with a higher error density can be done.

In this study, we start with character-level errors, for which a high-density dataset is relatively easy to synthesize and show that it allows to highlight differences in performance of models. This dataset can be used in further studies to report on the results for specifically character-level errors, though without missing the information about their interaction with other types of errors. To obtain such an opportunity for other kinds of errors, more work on collecting natural examples with them or more elaborate synthesizing is to be done.

References

[1]. Qorib M. R., Ng H. T. Grammatical error correction: Are we there yet? In Proceedings of the 29th International Conference on Computational Linguistics. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, 2022, pp. 2794–2800.

[2]. Leacock C., Chodorow M., Gamon M., Tetreault J. Automated Grammatical Error Detection for Language Learners. Morgan & Claypool Publishers, 2014. 154 p.

[3]. Wang Y., Wang Y., Dang K., Liu J., and Liu Z. A comprehensive survey of grammatical error correction. ACM Trans. Intell. Syst. Technol., 12(5), 2021, pp. 1–51. doi: 10.1145/3474840.

[4]. Bryant C., Yuan Z., Qorib M. R., Cao H., Ng H. T., Briscoe T. Grammatical Error Correction: A Survey of the State of the Art. Computational Linguistics, 49 (3), 2023, pp. 643–701. doi: 10.1162/coli_a_00478.

[5]. Susanto R. H., Phandi P., Ng H. T. System combination for grammatical error correction. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, 2014, pp. 951–962. [Online]. doi: 10.3115/v1/D14-1102.

[6]. Rozovskaya A., Roth D. Grammatical error correction: Machine translation and classifiers. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Berlin, Germany: Association for Computational Linguistics, 2016, pp. 2205–2215. doi: 10.18653/v1/P16-1208.

[7]. Chollampatt S., Wang W., Ng H. T. Cross-sentence grammatical error correction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019, pp. 435–445. doi: 10.18653/v1/P19-1042.

[8]. Gotou T., Nagata R., Mita M., Hanawa K. Taking the correction difficulty into account in grammatical error correction evaluation. In Proceedings of the 28th International Conference on Computational Linguistics. Barcelona, Spain (Online): International Committee on Computational Linguistics, 2020, pp. 2085–2095. doi: 10.18653/v1/2020.coling-main.188.

[9]. Omelianchuk K., Atrasevych V., Chernodub A., Skurzhanskyi O. GECToR – grammatical error correction: Tag, not rewrite. In Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications. Seattle, WA, USA → Online: Association for Computational Linguistics, 2020, pp. 163–170. doi: 10.18653/v1/2020.bea-1.16.

[10]. Cargill T. The design of a spelling checker's user interface. ACM SIGOA Newsletter, 1(3), 1980, pp. 3-4.

[11]. Bentley J. Programming pearls: A spelling checker. Communications of the ACM, 28(5), 1985, pp. 456–462.

[12]. Chollampatt S., Ng H. T. Connecting the dots: Towards human-level grammatical error correction. In Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications. Copenhagen, Denmark: Association for Computational Linguistics, 2017, pp. 327–333. doi: 10.18653/v1/W17-5037.

[13]. Ge T., Wei F., Zhou M. Fluency boost learning and inference for neural grammatical error correction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 1055–1065. doi: 10.18653/v1/P18-1097.

[14]. Sakaguchi K., Post M., Van Durme B. Grammatical error correction with neural reinforcement learning. In Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers). Taipei, Taiwan: Asian Federation of Natural Language Processing, 2017, pp. 366–372.

[15]. Katsumata S., Komachi M. Stronger Baselines for Grammatical Error Correction Using a Pretrained Encoder-Decoder Model. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, Suzhou, China: Association for Computational Linguistics, 2020, pp. 827–832.

[16]. Rothe S., Mallinson J., Malmi E., Krause S., Severyn A. A Simple Recipe for Multilingual Grammatical Error Correction. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Online, Association for Computational Linguistics, 2021, pp. 702–707. doi: 10.18653/v1/2021.acl-short.89.

[17]. Touvron H., Lavril T., Izacard G., Martinet X., Lachaux M. A., Lacroix T., Rozière B., Goyal N., Hambro E., Azhar F., Rodriguez A., Joulin A., Grave E., Lample, G. (2023) Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (online). Available at: https://arxiv.org/abs/2302.13971v1, accessed 18.12.2023.

[18]. Wang, Y., Kordi, Y., Mishra, S., Liu, A., Smith, N. A., Khashabi, D., Hajishirzi, H. (2022) Self-instruct: Aligning language model with self-generated instructions. *arXiv preprint arXiv:2212.10560* (online). Available at: https://arxiv.org/abs/2212.10560, accessed 18.12.2023.

[19]. Taori R., Gulrajani I., Zhang T., Dubois Y., Li X., Guestrin C., Liang P., Hashimoto T. B. Alpaca: A Strong, Replicable Instruction-Following Model. The Center for Research on Foundation Models of Stanford Institute for Human-Centered Artificial Intelligence. Available at: https://crfm.stanford.edu/2023/03/13/alpaca.html, accessed 18.12.2023.

[20]. Floridi L., Chiriatti M. Gpt-3: Its nature, scope, limits, and consequences. Minds and Machines, 30, 2020, pp. 1–14.

[21]. Coyne S., Sakaguchi K., Galvan-Sosa D., Zock M., Inui K. Analyzing the Performance of GPT-3.5 and GPT-4 in Grammatical Error Correction. arXiv e-prints, p. arXiv:2303.14342 (online). Available at: https://arxiv.org/abs/2303.14342, accessed 18.12.2023.

[22]. Östling R., Gillholm K., Kurfalı M., Mattson M., and Wirén M. (2023) Evaluation of really good grammatical error correction. arXiv e-prints, p. arXiv:2308.08982 (online). Available at: https://arxiv.org/abs/2308.08982v1, accessed 18.12.2023. doi: 10.18653/v1/2022.emnlp-main.162.

[23]. Zhang Yu., Zhang B., Li Zh., Bao Z., Li Ch., Zhang M. SynGEC: Syntax-Enhanced Grammatical Error Correction with a Tailored GEC-Oriented Parser. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, 2022, pp. 2518–2531.

[24]. Zhou, H., Liu, Y., Li, Z., Zhang, M., Zhang, B., Li, C., Zhang J., Huang, F. (2023) Improving Seq2Seq Grammatical Error Correction via Decoding Interventions. *arXiv preprint arXiv:2310.14534* (online). Available at: https://arxiv.org/abs/2310.14534, accessed 18.12.2023.

[25]. Ng H. T., Wu S. M., Briscoe T., Hadiwinoto C., Susanto R. H., Bryant C. The CoNLL-2014 shared task on grammatical error correction. In Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task. Baltimore, Maryland: Association for Computational Linguistics, 2014, pp. 1–14. doi: 10.3115/v1/W14-1701.

[26]. Bryant C., Ng H. T. How far are we from fully automatic high quality grammatical error correction? In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Beijing, China: Association for Computational Linguistics, 2015, pp. 697–707. doi: 10.3115/v1/P15-1068.

[27]. Grundkiewicz R., Junczys-Dowmunt M., Gillian E. Human evaluation of grammatical error correction systems. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon, Portugal: Association for Computational Linguistics, 2015, pp. 461–470. doi: 10.18653/v1/D15-1052.

[28]. Napoles C., Sakaguchi K., Post M., Tetreault J. Ground truth for grammatical error correction metrics. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 588–593. doi: 10.3115/v1/P15-2097.

[29]. Chollampatt S., Ng H. T. A reassessment of reference-based grammatical error correction metrics. In Proceedings of the 27th International Conference on Computational Linguistics. Santa Fe, New Mexico, USA: Association for Computational Linguistics, 2018, pp. 2730–2741.

[30]. Mizumoto T., Hayashibe Y., Komachi M., Nagata M., Matsumoto Y. The effect of learner corpus size in grammatical error correction of ESL writings. In Proceedings of COLING 2012: Posters, Kay M. and Boitet C., Eds. Mumbai, India: The COLING 2012 Organizing Committee, 2012, pp. 863–872.

[31]. Tajiri T., Komachi M., Matsumoto Y. Tense and aspect error correction for ESL learners using global context. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Li H., Lin C.-Y., Osborne M., Lee G. G., and Park J. C., Eds. Jeju Island, Korea: Association for Computational Linguistics, 2012, pp. 198–202.

[32]. Yannakoudakis H., Briscoe T., Medlock B. 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, Lin D., Matsumoto Y., Mihalcea R., Eds. Portland, Oregon, USA: Association for Computational Linguistics, 2011, pp. 180–189.

[33]. Coltheart M., Rastle K., Perry C., Langdon R., Ziegler J., Drc: a dual route cascaded model of visual word recognition and reading aloud. Psychological review, 108(1), 2001, pp. 204–256. doi: 10.1037/0033-295X.108.1.204.

[34]. Castles A., Rastle K., Nation K., Ending the reading wars: Reading acquisition from novice to expert. Psychological Science in the Public Interest, 19(1), pp. 5–51, 2018, pMID: 29890888. doi: 10.1177/1529100618772271

[35]. Lichtarge J., Alberti C., Kumar S., Shazeer N., Parmar N., Tong S., "Corpora generation for grammatical error correction," in Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Burstein J., Doran C., Solorio T., Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 3291–3301. [Online]. Available: https://aclanthology.org/N19-1333

[36]. Näther M. An in-depth comparison of 14 spelling correction tools on a common benchmark. In Proceedings of the Twelfth Language Resources and Evaluation Conference, Calzolari N., Béchet F., Blache P., Choukri K., Cieri C., Declerck T., Goggi S., Isahara H., Maegaard B., Mariani J., Mazo H., Moreno A., Odijk J., Piperidis S., Eds. Marseille, France: European Language Resources Association, 2020, pp. 1849–1857.

[37]. Napoles C., Sakaguchi K., Tetreault J., JFLEG: A fluency corpus and benchmark for grammatical error correction. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers, Lapata M., Blunsom P., Koller A., Eds. Valencia, Spain: Association for Computational Linguistics, 2017, pp. 229–234.

[38]. Qiu Z., Qu Y. A two-stage model for chinese grammatical error correction, IEEE Access, 7, pp. 146 772–146 777, 2019.

[39]. Hinson C., Huang H.-H., Chen H.-H. Heterogeneous recycle generation for Chinese grammatical error correction. In Proceedings of the 28th International Conference on Computational Linguistics, Scott D., Bel N., Zong C., Eds. Barcelona, Spain (Online): International Committee on Computational Linguistics. 2020, pp. 2191–2201. doi: 10.18653/v1/2020.coling-main.199.

Информация об авторах / Information about authors

Владимир Миронович СТАРЧЕНКО — аспирант Школы лингвистики Факультета гуманитарных наук НИУ ВШЭ, стажёр-исследователь Научно-учебной лаборатории учебных корпусов НИУ ВШЭ. Сфера научных интересов: автоматическое исправление грамматических ошибок, корпусная лингвистика, распределённые вычисления.

Vladimir Mironovich STARCHENKO is a graduate student at the School of Linguistics of the Faculty of Humanities of the HSE University, a research intern at the Laboratory of Learner's Corpora of the Higher School of Economics. Research interests: grammatical error correction, corpus linguistics, distributed systems.

Алексей Миронович СТАРЧЕНКО — аспирант и преподаватель Школы лингвистики Факультета гуманитарных наук НИУ ВШЭ, стажёр-исследователь Научно-учебной лаборатории по формальным моделям в лингвистике НИУ ВШЭ. Сфера научных интересов: аргументная структура, номинализации, фокусные частицы, полевая лингвистика, корпусная лингвистика.

Aleksey Mironovich STARCHENKO is a graduate student and lecturer at the School of Linguistics of the Faculty of Humanities of of the HSE University, a research intern at the Laboratory on Formal Models in Linguistics of the Higher School of Economics. Research interests: argument structure, nominalizations, focus particles, field linguistics, corpus linguistics.