FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

DOCTORAL THESIS

Jakub Náplava

Natural Language Correction With Focus on Czech

Institute of Formal and Applied Linguistics

Supervisor of the doctoral thesis:	RNDr. Milan Straka, Ph.D.
Study programme:	Computer Science
Study branch:	Computational Linguistics

Prague 2022

I declare that I carried out this doctoral thesis independently, and only with the cited sources, literature and other professional sources. It has not been used to obtain another or the same degree.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In .... date ....

Author's signature

First, I would like to thank my supervisor RNDr. Milan Straka, Ph.D. for his patient guidance, undying enthusiasm and openness to discuss any problem. There is no doubt that without Milan, I would not have extended my master's studies into a Ph.D., which in retrospect would have been a great loss for me. I really enjoyed working with Milan.

Thank you to all my family members who have always been my support. The biggest thanks go to my wonderful wife Ája, who had to listen to all my complaints and yet remained positive and encouraging. My big thanks also belong to my parents Petr and Eva, my brother Lukáš and my sister Erika.

Studying at the Institute of Formal and Applied Linguistics was an amazing journey along which I met great colleagues and also made good friends. Special thanks belong to RNDr. Jana Straková, Ph.D. for her invaluable help with texts of several papers and for proofreading this thesis, and to prof. RNDr. Jan Hajič, Dr. for allocating the money for my data acquisition.

My final words belong to my son Ondřej – always try to stay curious and open-minded. You are a wonderful person.

Title: Natural Language Correction With Focus on Czech

Author: Jakub Náplava

Department: Institute of Formal and Applied Linguistics

Supervisor: RNDr. Milan Straka, Ph.D., Institute of Formal and Applied Linguistics

Abstract: Natural language correction, a subfield of natural language processing (NLP), is the task of automatically correcting user errors in written texts. It includes, but is not limited to, grammatical error correction, spelling error correction and diacritics restoration. During the course of the work on this thesis, we witnessed a great advance in this field, with the emergence of new approaches to correct user errors, new datasets and also new evaluation metrics. This thesis presents, in the form of a dissertation by publication, our contributions to this field. As Czech is the primary language of the thesis author, special focus was devoted to improving natural language correction in Czech. The main contributions are (1) the creation of the Grammar Error Correction Corpus for Czech that comprises multiple sources of noisy texts such as essays or online discussion posts, evaluation of strong neural models on this dataset, and meta-evaluation of existing metrics, (2) the development of grammar error correction systems suited to scenarios in which only low amount of annotated data is available, and (3) the development of two state-of-the-art models and the creation of the new multilingual dataset comprising 12 languages for diacritics restoration.

Keywords: natural language correction, grammatical error correction, diacritics restoration, datasets, Czech

Contents

Part I

Preamble

1. Introduction

These days, enormous amounts of text data are created every day. There are e-mails to one's colleagues, business documents written for the customers, social media posts for one's friends, newspaper articles written for the general public and many others. Although they differ in many aspects such as the form, target audience or device they have been written on, they all have been written by humans. As such, they often contain errors such as typos or missing punctuation that are caused by the lack of our time, the patience and sometimes even knowledge to review and correct the texts.

As errors make texts more difficult to read and comprehend, to be processed by automatic systems, as well as they may embarrass their authors, great efforts have been made to correct them. These include paid services employing human editors, volunteer proofreading and corrections by automatic systems. In this work, we aim at improving and developing automatic error correction systems for natural languages.

The motivation and the need for correcting texts is universally acknowledged. People also agree that the corrected text should preserve the meaning, be comprehensible and accurate. However, people differ in their needs on what errors to correct. While for example authors writing shorter texts in their native language do not typically seek for larger edits affecting the text fluency, the same people want the corrector to propose also fluency edits when writing essays in their non-native language. When compared to other natural language processing tasks like named entity recognition or morphological tagging, the wide difference in users' requirements on automatic correction makes the formalization of the task more difficult.

Despite the above-mentioned issues, there is a consensus that a large group of necessary corrections such as errors in morphology (*Every days*), typos (*He wrtes*) or errors in word order (*This great is*) should be always corrected. We certainly include this set of generally accepted errors to be attended in natural language correction. It is however the grammatical error correction that is considered the holy grail and widely acknowledged interpretation of the natural language correction task. The grammatical error correction is usually defined as correcting an original erroneous sentence with a (minimal) set of edits so that the semantics remains unchanged, and the corrected sentence is grammatically correct and fluent. Natural language correction comprises also other tasks that require correcting only a subset of all error types. An example of such tasks are spelling error correction, which aims at correcting typos, and diacritics restoration, which aims at generating diacritical marks to text without them.

The major difficulty in natural language correction tasks stems from ambiguity and that context is needed to choose the best correction variant. This can be illustrated on a simple example comprising original text *Do he go there?* that without any additional context has two very probable GEC corrections: *Does he go there?* and *Do we go there?*. For this reason, simple rule-based methods as well as specific error classifiers turned out insufficient, and large neural networks are now employed as they are capable of processing large contexts effectively. Despite that using these networks improved performance significantly, there is still a large gap to optimal performance.

Until recently, the main research focus has been devoted to English. This includes development of a large variety of datasets, metrics and models for grammatical error correction. In Czech, on the other hand, only a limited number of research has been conducted so far. This involves mainly models aiming primary at two specific subtasks of natural language correction: spelling error correction and diacritics restoration tasks. Being a native speaker of Czech, using Czech on an everyday basis and knowing that there is a non-trivial amount of people using the Czech correcting tools lagging behind English state of the art by a large margin, the focus of my work was set to natural language correction with a specific focus in Czech.

1.1 Thesis Contributions

The main contribution of the work done during my Ph.D. studies are the improvements in the Czech natural language correction. Specifically, in its two tasks: grammatical error correction and diacritics restoration. For each task I created and published a new dataset as both tasks lacked such a dataset. Furthermore, I developed two state-of-the-art neural-based models for each task, and I conducted a meta-evaluation of metrics in Czech grammatical error correction to select the metric best correlating with humans.

Apart from the Czech natural language correction, I have also worked on grammatical error correction in languages with low amount of annotated data. I have shown that in such scenarios, strategies that synthetically generate training examples using relatively simple word-replacement rules work remarkably well, and outperform contemporary state-of-the-art results.

Lastly, I conducted analysis on model robustness when presented with data comprising all types of human errors. By doing analysis over multiple languages and tasks, I have shown that even the current neural-based models are still very sensitive to input noise, and their performance significantly deteriorates when they are presented with noisy texts. Moreover, I have shown that we can improve model robustness by either training the model on a mixture of original and noisy data and also by pre-processing the texts with an external correction system.

In this thesis, I describe 7 papers that I have published in natural language correction [Náplava et al., 2018, Náplava and Straka, 2019b,a, Náplava et al., 2021, Náplava et al., 2021, 2022, Straka et al., 2021]. In all but the last paper, I was the principal investigator. Regarding the last paper, where I am the second author, I have developed the model and also co-created the alignment algorithm.

1.2 Thesis Structure

The thesis is published as a dissertation by publication.

I open the first part by introducing the natural language correction together with its main background: tasks, metrics, datasets and history. I follow with my contribution, which comprises a story of my doctorate studies describing the main approaches, decisions, models and datasets I have created. I conclude the first part of the thesis with discussion of my contributions, and discuss possible future directions.

The second part comprises 7 papers that I have published in peer-reviewed journals, conferences and their workshops in the field of natural language correction.

2. Background

Natural language correction comprises a set of tasks that aim at correcting errors in users' text. Formally, an original sequence $x = \{x_0, x_1, \ldots, x_{N_x}\}$ should be mapped to a corresponding target sequence $y = \{y_0, y_1, ..., y_{N_y}\}$ (often called gold or reference annotation) that satisfies the task requirements such as being error-free. The output of a correction system $\hat{y} = \{\hat{y}_0, \hat{y}_1, ..., \hat{y}_{N_g}\}$ is called a system hypothesis (or simply model output).

Undoubtedly, the task with the most recent focus, the most conducted research and the holy grail of natural language correction is the **grammatical error correction (GEC)**. This task requires detecting and correcting all errors present in a source text with error types ranging from the simplest ones in spelling up to more complex errors in grammar and even fluency. Although the deployed systems should work upon detokenized texts, for historical and evaluation reasons, the GEC datasets are most commonly distributed with tokenized original texts. We illustrate two examples of erroneous sentences comprising several grammatical errors with their corrections in Figure 2.1. The first example presents rather simpler corrections consisting of single word replacements and deletions, and the second example illustrates a scenario in which a larger rewrite is needed to improve the text fluency.

In fact , in ~~the~~ political , economic and defence terms,I feel this ~~realocation~~ **reallocation** of resources can and will be ~~so~~ **very** positive .

So I think we ~~can not live~~ **would not be alive** if ~~old people could not find siences and tecnologies and they did not developed~~ **our ancestors did not develop sciences and technologies** .

Figure 2.1: Examples of erroneous sentences with their corrections taken from existing English GEC corpora: W&I+LOCNESS [Bryant et al., 2019, Granger, 2014] and JFLEG [Napoles et al., 2017a].

Grammatical error correction is the task for which most datasets, metrics and models have been developed, and we will discuss them later in more detail. Until recently, **grammatical error detection** task, whose objective is only to detect erroneous spans in a text without attempting at correction, was the focused research topic. However, with recent methods and advances, it seems that the majority of research has shifted to the full-scale grammatical error correction comprising both erroneous span detection and correction, and in our work, we solve both span detection and error correction jointly.

As correcting complex grammatical and fluency errors has been too difficult problem for a long time, and also because such corrections are often not needed, several other simpler tasks are solved. One of the most popular is the **spelling error correction**. So called spell-checkers or spelling error correction tools are well known software features appearing in a variety of commercial and non-commercial writing tools, in which they help both to detect errors and for each detected error to also propose a list of correction suggestions. As the name of the task suggest, the goal of spelling error correction is to detect and correct misspellings, local errors commonly caused by the lack of exact word spelling knowledge or miss-clicking keyboard keys resulting in a typographical error (typo). We illustrate two examples of sentences with spelling errors and their corrections in Figure 2.2.

They **before**
~~**Thay**~~ arrived ~~**befor**~~ us .

Where **today**
~~**Wherre**~~ are you going ~~**tdoay**~~ ?

Figure 2.2: Examples of erroneous sentences comprising spelling errors including their corrections. While the first sentence illustrates errors originating from word pronunciation similarity, the second sentence illustrates errors originating from typography with missclicked computer keyboard keys.

In languages with letters with diacritics (e.g. Czech or Vietnamese), **diacritics restoration**, also known as **diacritics generation** or **accent restoration**, is another commonly discussed task. In these languages, people frequently write without diacritics, because it is often significantly faster and also for historical reasons as there used to be problems with encoding diacritics. Although that removing diacritics creates new groups of homonymy (*dal*/*dál*, *krize*/*kříže*), the text remains readable and makes sense. Having the text with no diacritics, the objective of the diacritics restoration task is to correctly restore diacritical marks for all letters in a text. It is important to emphasize that in the diacritics restoration setting, the source texts for the task lack any diacritics whatsoever, as opposed to (partially) omitted/misspelled diacritics in the spelling correction or the grammar correction task. An example with both undiacritized and correctly diacritized sentences in Czech is presented in Figure 2.3.

ý **ě** **ší** **ří** **ň** **é**
K nejv**y**znamn**e**j**si**m pat**ri** zmi**n**ovan**e** vily.

ů **ý** **á** **í** **í** **í** **é** **í** **é**
V r**u**zn**y**ch pod**a**n**i**ch existuj**i** v**i**ly hodn**e** a v**i**ly zl**e**.

Figure 2.3: Examples of Czech sentences without diacritics and correct diacritized variants for their characters.

Note that similarly to diacritics restoration, there are also correction tasks aiming at single error type correction such as **punctuation** and **casing restoration**, but these are typically used as post-processing techniques for other systems such as automatic speech recognition or demonstration techniques rather than being a stand-alone natural language correction task operating over users' text.

Finally, the last common task belonging to the natural language correction is **text normalization**. Generally, the task's objective is to transform a non-standard text to a standard register, commonly also formulated as converting non-canonical texts to their canonical equivalent. There are three main areas in which the task is used: historical data [Tang et al., 2018, Bollmann, 2019], medical data [Dirkson et al., 2019] and social media texts [Han et al., 2013, van der Goot et al., 2021]. There are differences in the task setting as some researchers [Han et al., 2013] restrict the task to exact one-to-one mapping (one correction word replaces one source word) between words, and consider only out-of-vocabulary words for normalization, while newer works [van der Goot et al., 2021] soften the restrictions by allowing normalization to happen also on in-vocabulary words, and also allow for one-to-many and many-to-one replacements. To illustrate the task, we provide examples in Figure 2.4.

Recall that for all the tasks, the main difficulty stems from ambiguity. This means that often, multiple correction variants for a piece of text are possible, and

The queen right greatly displeased
þe quene was ryght gretly displisyd with us both

fibromyalgia
Muscle pain from fibromalgia is reduced.

Social people are
social ppl r troublesome

Figure 2.4: Examples of original unnormalized sentences and their normalization variants in three domains: (1) historical texts, (2) medical texts and (3) social media texts.

a context is needed to disambiguate between them, and choose the best correction. Despite that ambiguity is present across all described tasks, the highest level of it is indisputably in grammar error correction, which makes it the hardest task.

During my Ph.D. studies, I mainly dealt with two specific tasks: **grammar error correction (GEC)** and **diacritics restoration**. I also dealt a bit with **spelling error correction**. Therefore, these tasks will be introduced more thoroughly in the rest of this chapter. Specifically, first, the metrics commonly used for evaluating model performance are outlined. Further, common datasets are introduced, and a short history of developing models for natural language correction is described. I end the model history section in 2017 when my doctorate studies started, and describe the models introduced later that were important to my work in the following chapter. Then, I discuss commercial systems, and compare their performance to state-of-the-art models. Finally, I discuss models and datasets specific for Czech.

2.1 Evaluating Systems

First, the GEC metrics are described as they are the most universal, i.e. generally, these metrics could be used for assessing model performance in other natural correction tasks. However, as GEC metrics are relatively new and other tasks simpler, we often refer to their metrics specifically, although they are often only specializations of the GEC metrics.

2.1.1 GEC Metrics

There are two type of metrics commonly used for evaluating system performance in GEC: *edit*-based and *machine-translation*-alike. The first group, *edit-based* methods evaluate system performance by means of correctly performed edits, where an edit consists of a part of the noisy text and its correction. The corrections are typically string to string replacements in a token span performed on the word level. Requiring to correct an entire edit is important as correcting only its subpart may lead to a text of even worse quality than the original text has. A simple example is correcting *He have eating.* to *He has eaten.*, where a full edit *have eating* → *has eaten* has to be performed to make a meaningful correction.

Supposing that a corpus contains annotated gold edits, these gold edits are compared with edits extracted from a system hypothesis and the *F-score* computed over the gold and system edits is reported.

Formally, having a set of N sentences and the respective set of gold edits $\{g_1, g_2, ..., g_N\}$ and system edits $\{e_1, e_2, ..., e_N\}$, where $g_i = \{g_i^1, g_i^2, ..., g_i^O\}$ and $e_i = \{e_i^1, e_i^2, ..., e_i^P\}$ contain gold and system edits for i-th sentence, the precision, recall and final F_β-score are computed as follows:

$$P = \frac{\sum_{i=1}^{N} |e_i \cap g_i|}{\sum_{i=1}^{N} |e_i|}$$

$$R = \frac{\sum_{i=1}^{N} |e_i \cap g_i|}{\sum_{i=1}^{N} |g_i|}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{P \cdot R}{(\beta^2 \cdot P) + R}$$

where the intersection between i-th sentence system edits e_i and gold edits g_i is defined as:

$$e_i \cap g_i = \{e \in e_i | \exists g \in g_i, match(g, e)\}$$

where the matching function $match(g, e)$ checks equality of the system edit e and gold edit f, i.e. whether they correct the same source span with the same string. Note that as multiple corrections are sometimes possible for a sentence, GEC corpora often contain multiple references for each sentence produced by multiple annotators. In this setting, the gold reference producing maximum F-score for a given system hypothesis is chosen independently for each sentence. Finally, in grammar error correction, proposing of a bad edit is considered worse than omitting an error, hence precision is emphasized over recall, and, therefore, F_β-score with $\beta = 0.5$ ($F_{0.5}$-score) is typically used:

$$F_{0.5} = \frac{(1 + 0.5^2) \cdot R \cdot P}{R + 0.5^2 \cdot P}$$

The two most popular *edit*-based scorers differ in how they extract system edits. The **MaxMatch** (M^2)-scorer [Dahlmeier and Ng, 2012a] extracts system edits that have the highest overlap with the gold standard annotation, i.e. the set of edits that yields the maximum F-score. Specifically, Dahlmeier and Ng [2012a] first construct the two-dimensional Levenshtein matrix for the tokenized source sentence s_i and system hypothesis h_i. They further use the breadth-first search algorithm to extract the shortest path lattice, in which the vertices correspond to Levenshtein matrix cells, each edge represents one of four atomic operations (token *insertion*, token *deletion*, token *substitution* and *keeping* token unchanged) and each edge has a unit cost. It is a well known fact that each path through this lattice corresponds to the shortest sequence of edits that transform s_i to h_i. To allow for phrase-level edits, transitive edges that satisfy a predefined length limit and change at least one word are added to the lattice with a cost being the sum of its parts. Finally, to support usage of gold edits, the cost of each edge corresponding to any gold edit is reduced. The start-end path with the lowest cost is then extracted and Dahlmeier and Ng [2012a] prove that with properly set constants for reducing gold edges costs, the set of edits that belong to the extracted path has the maximum overlap with the gold standard annotation.

The **ERRANT**-scorer [Bryant et al., 2017] works differently. One of the major differences to the M^2-scorer is that it does not require edit annotations for the gold data, but extracts them automatically using the same process as it extract edits for the system hypothesis. To extract edits, it uses a linguistically-enhanced alignment algorithm proposed by Felice et al. [2016]. More specifically, linguistic features such as part-of-speech tags and lemmas are used as additional cost functions in a Damerau-Levenshtein algorithm to make it more likely that tokens with similar linguistic properties are aligned, and a set of manually designed merging rules is used to create meaningful edits. The extracted edits were shown to match the human edits with circa 80% F_1. Apart from extracting edits, ERRANT can also classify individual edits into 25 predefined categories such as *PUNCT* (error in punctuation), *PREP* (wrong preposition) or *NOUN:NUM* (error in noun number) and report model performance in individual error-type categories.

The third scorer belonging to *edit*-based metrics is the **I-Measure** [Felice and Briscoe, 2015]. In comparison to M^2 and ERRANT that require edits to comprise a contiguous span of tokens and edits produced by different annotators are considered independently, the annotation scheme of Felice and Briscoe [2015] allows for non-continuous original spans and assumes that every pair of intersecting edits produced by different annotators are alternating, and that non-intersecting edits are independent, i.e. it groups errors from multiple annotators whenever they refer to the same underlying error. Consequently, it generates a reference set by taking every combination of independent edits. Moreover, it does not evaluate whole edits, but rather individual tokens. Specifically, based on the alignment between the source sentence, hypothesis and gold reference, I-score is computed using weighted accuracy where true-positives and false-positives are weighted higher than true-negatives and false-negatives. By computing weighted accuracy with true-negatives, the I-Measure can discriminate between the ”do-nothing” baseline and models proposing only wrongs edits.

GLEU [Napoles et al., 2015] is a metric that belongs to metrics inspired by *machine-translation* metrics. It is based on the famous BLEU metric [Papineni et al., 2002]. The BLEU metric is based on comparing n-grams of the system hypothesis with n-grams of the reference sentence. The BLEU score is computed as the geometric mean of n-gram precision scores (for n=1,2...,N with N being typically set to 4) with penalization for too short translations. The GLEU metric extends this machine-translation metric with two features modifying the precision calculation: (1) extra weight is assigned to n-grams present in the candidate that overlap with the reference but not the source (*correct edits*), and (2) reduced weight is assigned to n-grams that are in the hypothesis and source but not in the reference (*missed corrections*).

Apart for the four described metrics, several other metrics were proposed. These include for example scorer of [Gotou et al., 2020] that tries to take into account the difficulties of individual errors or reference-less metrics [Napoles et al., 2016, Asano et al., 2017, Choshen and Abend, 2018, Yoshimura et al., 2020]. However, to the best of our knowledge, neither of these metrics has so far been widely accepted and used.

To assess the quality of individual GEC metrics, several studies were conducted [Grundkiewicz et al., 2015, Napoles et al., 2015, Chollampatt and Ng, 2018a, Napoles et al., 2019]. A standard approach to analysing metric quality is to measure their correlation with human judgements. Grundkiewicz et al. [2015] and Napoles et al. [2015] analysed outputs of models submitted to the CoNLL

2014 Shared Task [Ng et al., 2014] and found that while M^2 has a moderate correlation with human judgements, I-Measure and BLEU have low or even negative correlation. Motivated by this, Napoles et al. [2015] proposed GLEU. Grundkiewicz et al. [2015] also found that although F-score with $\beta = 1.0$ does indeed correlate better than F-score with $\beta = 0.5$, the best correlation on the CoNLL 2014 test set is observed with $\beta = 0.2$. Later, Chollampatt and Ng [2018a] conducted sentence-level correlation experiments including significance testing and found that not only no evidence that GLEU performs better than M^2 as claimed by Napoles et al. [2015], but in certain scenarios M^2 has even higher correlation. Despite all these studies, the most used metric is still the M^2-scorer with $\beta = 0.5$, which is used to evaluate system performance not only on the CoNLL 2014 test data, but also on several other datasets in languages other than English.

2.1.2 Metrics for Spelling Error Correction and Diacritics Restoration

Unlike GEC where a single error often affects multiple words, the errors in diacritics and spelling happen separately in individual words, i.e. correcting any word that contains either a spelling or a diacritical error results in better text. Note that probably for their marginal occurrence, errors in whitespace as a result of accidentally missclicking space bar inside a word or not clicking space bar are typically not considered by spelling correction systems.

Having edits comprising exactly one word in original and corrected text, the three most popular metrics for evaluating system performance in spelling error correction are **word error rate** (WER), **word accuracy** (WAcc), and **F-score** computed over words. Following standard definition of true-positives (TP, ”count of words with correctly fixed errors”), false-positives (FP, ”count of words for which a bad correction was performed”), false-negatives (FN, ”count of words for which a correction was falsely not proposed”) and false-positives (FP, ”count of words for which a correction was falsely proposed”), the word error rate and word accuracy metrics are defined as follows:

$$WAcc = \frac{TP + TN}{TP + FP + TN + FN}$$

$$WER = 1 - WAcc$$

Recall from Section 2.1.1, that the F-score is typically defined using **precision** (P, ”how many of the changed words were corrected correctly”) and **recall** (R, ”how many of the words that should have been corrected were actually corrected”), and parametrized using a parameter β that specifies the ratio of recall to precision. The typical values of β are $\beta = 1$ to weight recall and precision equally and $\beta = 0.5$ that weights precision twice as much as recall.

Regarding diacritics restoration, all metrics described for spelling error correction are being used. Besides them, so called **alpha-word accuracy** that evaluates the word accuracy only on words comprising at least one alphabetical characters is used. The last often used metrics is the **diacritization error rate** (DER), which is the proportion of letters incorrectly labelled with diacritics to the count of all letters with diacritics in the corpus.

2.2 Datasets

To evaluate performance of developed models and compare them to other systems as well as to train new models, datasets containing original uncorrected texts with their corrected versions are needed.

Generally, the datasets can be classified based on the *language* of texts they are written in (e.g. English or Czech), the *domain* of the original texts (e.g. second learners or native speakers), the type of the text (e.g. essay or social media post), the dataset *size* and *quality* (e.g. whether texts were corrected by professional annotators or using crowdsource platforms). The individual datasets also differ in the number of references annotated by different annotators that they contain for each piece of text. The combination of domain and type of the text is often referred to as *user domain*.

While grammar error correction is well established with multiple existing and commonly used datasets, the situation in spelling error correction and diacritics restoration is more chaotic. In spelling error correction, the only available public datasets such as the dataset of Birkbeck spelling error corpus / Roger Mitton or Peter Norvig[1] consists of individual uncorrected and corrected word variants completely lacking the context. In diacritics restoration, it is important to realize that due to the fact that its inputs are completely undiacritized texts, a dataset for the task can be easily created by stripping all diacritics from a clean text, thus requiring no manual annotations such as in spelling error correction or GEC datasets. A typical scenario in spelling error correction and also diacritics restoration is thus training and evaluating systems on a custom dataset. We further discuss datasets specific to GEC.

Since the CoNLL 2013 Shared Task on GEC [Ng et al., 2013], the de facto standard format for distributing GEC datasets is the *M2 format*. In M2 format, there is a set of annotation edits for each original noisy tokenized sentence. An annotation edit comprises start and end offsets to the original sentence, edit type, correction string and also a unique ID of the annotator. Besides the edit information, the format also uses two currently useless fields for historical reasons. A simple example for original tokenized sentence *Besides that , the risk of the known genetic is very serious that it can not be described .* annotated by two annotators with ID 0 and ID 1 is provided below:

```
S Besides that , the risk of the known genetic is very serious
    that it can not be described .
A 9 9|||Wci|||disease|||REQUIRED|||-NONE-|||0
A 10 11|||Wci|||so|||REQUIRED|||-NONE-|||0
A 8 9|||Others|||disorder|||REQUIRED|||-NONE-|||1
A 10 11|||Wci|||so|||REQUIRED|||-NONE-|||1
```

We can see that while both annotators agree on changing *that* to *so*, they offer two correction variants on fixing the missing noun around *known genetic*. The annotator with ID 0 suggests adding a word *disease* resulting in *known genetic disease* and the annotator with ID 1 suggests replacing the word *genetic* by *disorder* resulting in *known disorder*. Having two gold sentences, the evaluated

[1] http://norvig.com/ngrams/spell-errors.txt

system would score positive points if proposing any of the two alternatives.

We further provide an overview of all public GEC datasets known to us in Table 2.1. We split the table into three time periods: (1) until 2017 (before the work on the doctorate thesis started), (2) between 2017 and September 2019 when the results of the BEA 2019 Shared Task on GEC [Bryant et al., 2019] were announced, and (3) between September 2019 and December 2021 when the work on the doctorate ended. For each dataset, we report its name, number of sentences it comprises, average token error rate of its texts, user domain of writers who wrote the erroneous texts and number of reference annotations that each testing sentence has.

Language	Corpus	Sentences	Err. r.	Domain	# Refs.
until 2017					
English	*Lang-8*	1 147 451	14.1%	SL	1
	NUCLE	57 151	6.6%	SL	1
	CoNLL 2014 test	1 312	8.2%	SL	2,10,8
	JFLEG	1 511	—	SL	4
	FCE	33 236	11.5%	SL	1
	AESW	over 1M	—	scientific writing	1
2017-09/2019					
English	*W&I+LOCNESS*	43 169	11.8%	SL, native students	5
German	*Falko-MERLIN*	24 077	16.8%	SL essays	1
Russian	*RULEC-GEC*	12 480	6.4%	SL, heritage speakers	1
09/2019-12/2021					
English	*GMEG*	6 000	—	web, formal articles, SL	4
	CWEB	13 574	~2%	web	2
Czech	*AKCES-GEC*	47 371	21.4%	SL & Romani heritage speakers essays	2
	GECCC	83 058	18.2%	AKCES-GEC + native + web	2
Spanish	*COWS-L2H*	12 336	—	SL, heritage speakers	2
Ukrainian	*UA-GEC*	20 715	7.1%	natives/SL, translations and personal texts	2
Romanian	*RONACC*	10 119	—	native speakers transcriptions	1

Table 2.1: Comparison of GEC corpora in size, token error rate, domain and number of annotations of the test portion. SL = second language learners. We divide the datasets into three time periods.

Table 2.1 shows that until 2017, GEC datasets were available only for English. Moreover, all but one dataset contain corrected texts of second learner students of English. We provide a short description of these datasets below:

- **NUCLE** [Dahlmeier et al., 2013] – consists of essays written by undergraduate students of the National University of Singapore. It was used as the official training data for the CoNLL 2014 Shared Task on GEC.

- **CoNLL14 Shared Task test set** [Ng et al., 2014] – this testing set consists of 50 essays written by 25 South-East Asian undergraduates and was used as the official testing data for the CoNLL 2014 Shared Task on GEC. Since then, the majority of research papers reported their performance on it using the M^2-scorer. Due to its popularity, the original 2 reference annotations were later extended by 10 additional annotations from Bryant and Ng [2015] and 8 alternative annotations from Sakaguchi et al. [2016]. We further refer to this test set in a shorter form: *CoNLL14 test set.*

- **Lang-8** [Tajiri et al., 2012] – large corpus of English language learner texts collected from the Lang-8 social networking system[2]. Because texts were

[2] https://lang-8.com/

corrected by other users of the social platform, the corrections are often of a low quality. On the other hand, the corpus is very large and allows for training data hungry [Koehn and Knowles, 2017] machine-translation models.

- **JFLEG** [Napoles et al., 2017a] – GEC corpus with focus on fluency edits in addition to usual grammatical edits. GLEU scorer is used to assess the performance on this dataset.

- **FCE** [Yannakoudakis et al., 2011] – short essays written by non-native learners for the Cambridge ESOL First Certificate in English

- AESW [Daudaravicius et al., 2016] – large corpus of scientific writing (over 1M sentences, mostly in physics and mathematics), edited by professional editors. Probably due to the domain specificity, this dataset is used marginally in GEC compared to other datasets from this time period.

In 2018, the German *FALKO-MERLIN GEC* [Boyd, 2018] was released, becoming the first non-English GEC dataset. It was compiled from two German learner corpora and as we can see in Table 2.1, it was the noisiest dataset by then with circa every 8-th token erroneous. To create the dataset from aligned pairs of original and corrected sentences, Boyd [2018] extended original English ERRANT to German. The list of non-English GEC datasets was further extended in 2019, when Rozovskaya and Roth [2019] introduced the Russian *RULEC-GEC* comprising data from Russian second learners and heritage speakers. In accordance with the CoNLL14 test set, both Boyd [2018] and Rozovskaya and Roth [2019] decided to use the M^2-scorer to assess model performance.

The BEA 2019 Shared Task on GEC introduced a *Write&Improve+Locness* (W&I+LOCNESS) dataset comprising texts from English second learners and more importantly also corrected essays originally written by English native students. Compared to the CoNLL14 test set, the W&I+LOCNESS test contains more than 3 times more sentences (4,477 vs 1 312) that were written by authors from around the world (as opposed to South-East Asian undergraduates only in the CoNLL14 test set). Moreover, the second learners' texts are split into 3 CEFR levels groups – A (beginner), B (intermediate), and C (advanced). To evaluate system performance on this dataset, Bryant et al. [2019] decided to use ERRANT, because it can provide more detailed feedback, e.g. in terms of performance on specific error type.

Shortly after the BEA 2019 Shared Task on GEC, we [Náplava and Straka, 2019a] released the first Czech GEC dataset: *AKCES-GEC*. We compiled it from Czech learner corpora with two user domains: Czech second learners and Romani speakers. As can be seen in Table 2.1, it is of a decent size and its texts are quite noisy. As opposed to the English-only time period before 2017, several other non-English datasets were futher released: Spanish *COWS-L2H* [Davidson et al., 2020], Ukrainian *UA-GEC* [Syvokon and Nahorna, 2021] and Romanian *RONACC* [Cotet et al., 2020a].

The work naturally continued also in English. Specifically, Flachs et al. [2020] released *CWEB*, a collection of website texts, aiming at contributing lower error density data and broadening the restricted variety of user domains. Napoles et al.

[2019] had the similar intention, and they released *GMEG*, a GEC test set spanning three user domains: native formal writing (articles from Wikipedia), native informal writing (web posts from Yahoo Answers) and second learner writing (FCE essays).

By the end of 2021, we released *Grammar Error Correction Corpus for Czech* (GECCC), which is an improved version of the AKCES-GEC dataset. It contains two new user domains: essays written by native speakers and web discussion posts, completely new annotations for the testing, development and part of the training set, and we have opted to use the M^2 scorer with β=0.5, which was shown to have the highest correlation with human judgements.

To sum up, nowadays, there is a plethora GEC datasets for training and evaluating system performance on English. The most commonly used testing sets are *CoNLL 14 test set* and *W&I+LOCNESS test set.* To train systems, any described dataset can be used and often, models are trained on their combination. Regarding other languages, there is now exactly one GEC corpus for German, Russian, Czech, Spanish, Ukrainian and Romanian.

2.3 Brief History of Developing Models for Correcting Texts up to 2017

Early natural language correction attempts focused on correcting **isolated words**. The work on computer techniques for automatic spelling correction began as early as the 1960s [Kukich, 1992]. The research mainly focused on two issues: how to detect non-words in a text and how to correct them. One of the first commercial tools to detect spelling errors is the UNIX *SPELL* [McIlroy, 1982], which contains a list of 30 000 correct English words. As the module for correcting the detected errors is not part of the program, users either had to correct the words by themselves or could use tools for isolated word spelling correction such as *grope* [Taylor, 1981].

The first systems that aimed to correct a larger variety of errors employed **hand-coded rules**. One of such tools is Writer's Workbench [Macdonald et al., 1982] included with Unix systems as far back as the 1970s. It was based on simple pattern matching and string replacement and its *style* and *diction* tools could highlight common grammatical and stylistic errors and propose corrections. Other systems such as *GramCheck* [Bustamante and León, 1996] or *EPISTLE* [Heidorn et al., 1982] employed syntactic analysis with manually designed grammar rules. The great advantage of the hand-coded rules is their interpretability and also the fact that for certain error types, they can be implemented easily. On the other hand, it is nearly impossible to define rules to cover all errors in grammar (or fluency), therefore, not much of current research is conducted in this direction.

Despite that mainstream research slowly changed its focus from rule-based systems, several systems have been further developed using this approach. A great example is a Czech system *Kontrola české gramatiky* [Petkevič, 2014] that was used as the official grammar checker in Microsoft Word software product from version 2003 up to version 2013. It employs circa 2 000 disambiguation and 820 grammar hand-coded rules operating over word morphological tags and lemmas. These are used alongside a complex pipeline to detect mainly grammatical and also some orthographic and stylistic errors and propose their corrections in Czech texts. Petkevič [2014] reported that their tool detects 30-40% of errors in Czech texts and has a low rate of false positives.

In the 1990s, researchers in natural language processing started utilizing **data-driven** approaches and applied machine learning to NLP tasks. Because article and preposition errors are both difficult for manual rules but have a small span, multiple machine learning models were proposed to tackle them [Knight and Chander, 1994, Minnen et al., 2000, Han et al., 2004, Nagata et al., 2005]. Features encoding context such as neighbouring words or their part-of-speech tags are typically used as inputs into a machine learning classification model. For example, Han et al. [2004] trained a max-entropy classifier to detect article errors and achieved an accuracy of 88%.

As each trained classifier can only correct a single error type, several such classifiers, one per each error type, must be **combined** to allow more realistic usage for GEC. Dahlmeier et al. [2012] used a pipeline system comprising several sequential steps. [Dahlmeier and Ng, 2012b] employ specific classifiers together with a language model to score a beam of hypotheses. These are iteratively generated by so-called proposers, each allowed to propose only a small incremental change. Although their system worked quite well, it has many flaws such as the beam size growing with the number of proposers or that designing a classifier for certain more complex error types might be complicated.

The Czech system for context-sensitive spelling correction and diacritics restoration *Korektor* of Richter et al. [2012] does not use any specific classifiers, but its approach can be seen as a generalization of the approach of Dahlmeier and Ng [2012b]. It uses the **noisy channel approach** with a candidate model, that for each word proposes its variants up to a predefined edit distance. As it would be intractable to make a beam of all the hypothesis, the authors employ a Hidden Markov Model with vertices being the variants of words proposed by the candidate model. Instead of using separate error classifiers, the transition costs are determined from three N-gram language models built over word forms, lemmas and part-of-speech-tags. To find an optimal correction, the Viterbi algorithm [Forney, 1973] is used.

Brockett et al. [2006] proposed to consider GEC to be a **machine translation problem** of translating grammatically incorrect sentences into correct ones. Although they used a **statistical machine translation (SMT)** system for correcting only mass noun errors, such model was already powerful enough to correct a variety of error types as well as make stylistic changes if trained on large enough data [Leacock et al., 2010]. Since then, several other papers utilizing SMT were proposed Mizumoto et al. [2011], but the real advent of GEC started with the Helping Our Own and CoNLL shared tasks between 2011 and 2014 [Dale and Kilgarriff, 2011, Dale et al., 2012, Ng et al., 2013, 2014]. Two out of three top performing teams in CoNLL 2014 shared task [Felice et al., 2014, Junczys-Dowmunt and Grundkiewicz, 2014] used machine translation approaches.

While the SMT approach has also been successfully used in diacritics restoration [Diab et al., 2007, Pham et al., 2013] and spelling error correction [Hasan et al., 2015, Chiu et al., 2013], by 2017 the approaches utilizing large language models to choose the best variant proposed by relatively simple methods such as the described *Korektor* or diacritization system of Ljubešić et al. [2016] still provided comparable results.

On the other hand in GEC, the prevailing number of papers utilized machine translation approach in the following years. Grundkiewicz and Junczys-Dowmunt [2014] trained an SMT system with filtered sentences from Wikipedia revisions that matched a set of rules derived from NUCLE training data (GEC corpus). The SMT system was further tuned and extended by a rich set of task-specific features and incorporation of large language models in Junczys-Dowmunt and Grundkiewicz [2016]. An SVM ranking model to re-rank correction candidates proposed by SMT output was then implemented by Yuan [2017].

With the continuing success of neural networks in machine translation [Cho et al., 2014, Sutskever et al., 2014, Heidorn et al., 1982, Bahdanau et al., 2015] and the inability of SMT to capture long range dependencies and generalize beyond patterns seen during training, it was only a matter of time, when the **neural models** strike into the area of GEC.

Yuan and Briscoe [2016] proposed a first GEC system based on **neural machine translation**. Its backbone was a classical **encoder-decoder** word-level **recurrent** model, and they use a two-step approach to address out-of-vocabulary words, which may occur quite frequently due to errors in spelling. The two-step approach started by aligning the unknown words in the target sequence to their origins in the source sentence with an unsupervised aligner and translating these words with a word-level translation model. Xie et al. [2016] proposed to operate on character level and implemented a neural sequence-to-sequence recurrent model comprising a character level pyramidal encoder and a character decoder with an attention mechanism. Although the use of characters as basic units eliminated the problem with out-of-vocabulary words and the pyramidal encoder reduced the size of potentially large attention matrices, we speculate that its inability to effectively leverage word-level information and longer training time caused that this model was surpassed by Ji et al. [2017]. Similarly to Yuan and Briscoe [2016], they utilized a word-level encoder-decoder model, but the decoder used two nested levels of attention to overcome out-of-vocabulary problem: word level and character level. The word level was used in the classical manner, but whenever there was an out-of-vocabulary word in the target sequence, they used hard attention mechanism and character level decoder to output the target word character by character. The important aspect of the model was its combined loss term, which allowed the character level decoder to be trained jointly. The SMT approach once reappeared when Grundkiewicz and Junczys-Dowmunt [2018] achieved new state-of-the-art results with neural machine translation model being a re-scoring component in its SMT system. However, since then the backbone of most models in GEC became either the **Transformer** architecture [Vaswani et al., 2017] or the **convolutional encoder-decoder** model proposed by Chollampatt and Ng [2018b]. Both models use subword units to mitigate out-of-vocabulary issue and replace the slow-to-train recurrent units with either self-attention mechanism or convolution operations.

2.4 Applications

Nowadays (and even in 2017), there are various commercial and open source tools for natural language correction. Among the most known are spelling and grammar features of *Office 365 Word*[3] and *Google Docs*[4] that work across multiple languages, and also English-specific *Grammarly*[5] and open-source *LanguageTool*[6]. Recently, also *Writefull*[7] that aims at scientific writing correction appeared. As these tools rarely provide any information on their internal workings, we decided to conduct a simple extrinsic analysis by evaluating their performance on the popular CoNLL 2014 test set.

We evaluated the following five systems:

- Office 365 Word (build *16.0.14709.41008*)
- Google Docs
- Writefull (version *v2021.18.1*, *premium* licence)
- LanguageTool (release *5.5.1*, *basic* account)
- Grammarly (*edu* account, we used only the *Correctness* suggestion type. We also experimented with correcting errors proposed by all suggesters, but it provided worse results in terms of final $F_{0.5}$)

When there were multiple correction proposals, we always took the first of them, and repeated this process until there were no proposals left. All corrections were performed on 21st November 2021. Note that only random 400 sentences of the entire test set were corrected as this procedure requires manual user interaction. Also note that as the CoNLL 2014 test set comprises tokenized sentences, we first detokenized them using Moses detokenizer,[8] corrected detokenized data using the individual correcting tools, and tokenized them back using the UDPipe tokenizer [Straka et al., 2016].

Before presenting the results themselves, note that we compared only the system performance by means of correctly performed edits, while sometimes users may also benefit from further error explanations that some tools provide. Furthermore, please be aware that the CoNLL 2014 test set that we used for system comparison contains texts written by South-East Asian undergraduates. Our analysis thus evaluates systems only on one specific user domain, and more user domains such as noisy text written by English native speakers must be tested to draw any general conclusions. Nevertheless, our analysis still provides useful information on what performance to expect from the tested systems, and how these deployed systems compare to models from research papers.

The results of the analysis are presented in Table 2.2. To put the system results into context, Table 2.2 also contains results of several systems presented in Section 2.3. Furthermore, as the commercial and open source tools were evaluated in 2021, we also present results of several chosen research papers conducted after 2017. Note that for better clarity, we split the results of research papers into three time periods similarly to what we did with English datasets in Table 2.1.

[3] https://www.office.com/

[4] https://docs.google.com/

[5] https://www.grammarly.com/

[6] https://languagetool.org/

[7] https://www.writefull.com/

[8] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/detokenizer.perl

System	P	R	$F_{0.5}$	System Approach
until 2017				
Felice et al. [2014]	39.71	30.10	37.33	rules, SMT, LM
Yuan and Briscoe [2016]	–	–	39.90	NMT (word RNN seq2seq)
Xie et al. [2016]	49.24	23.77	40.56	NMT (char RNN seq2seq, LM)
Junczys-Dowmunt and Grundkiewicz [2016]	61.27	27.98	49.49	SMT + task features
Ji et al. [2017]	–	–	45.15	NMT (word RNN seq2seq)
2018 – 06/2019				
Grundkiewicz and Junczys-Dowmunt [2018]	66.77	34.49	56.25	hybrid SMT + NMT, LM
Lichtarge et al. [2018]	62.2	37.8	54.9	NMT (iter. Transformer + Wiki edits)
Junczys-Dowmunt et al. [2018]	63.0	38.9	56.1	NMT (Transformer with low-res. tricks)
Chollampatt and Ng [2018b]*	65.49	33.14	54.79	NMT (CNN seq2seq, LM)
Ge et al. [2018]	74.12	36.30	61.34	NMT (iter. CNN seq2seq + non-public CLC)
06/2019 – 2021				
Grundkiewicz et al. [2019]	–	–	61.30	NMT (Transformer, synth. data)
Kiyono et al. [2019]	73.3	44.2	64.7	NMT (Transformer, backtrans. synth. data)
Kaneko et al. [2020]	72.6	46.4	65.2	NMT (Transformer + BERT, synth. data)
Omelianchuk et al. [2020]	77.5	40.1	65.3	BERT, XLNet, word-rules
Rothe et al. [2021]	–	–	68.87	mT5-xxl (13B params), filtered Lang8
Commercial&other systems				
Google Docs	66.12	40.25	58.59	
Grammarly	56.51	39.20	51.92	
LanguageTool	37.99	15.08	29.13	
Office 365 Word	33.98	12.03	24.90	
Writefull	48.27	30.13	43.08	

Table 2.2: Comparison of $F_{0.5}$ of selected models on the popular CoNLL 2014 test set. The models are split into four sections: (1) *until 2017* that shows systems before my doctorate started, (2) *2018 – 06/2019* before the BEA 2019 Shared Task on GEC happened (3) *06/2019 – 2021* after it, and (4) *Commercial and other systems* that presents performance of commercial and other open source systems. Asterisk next to a system denotes an ensemble.

The results of the commercial and open source systems presented in Table 2.2 reveal two interesting facts: (1) their quality varies a lot – while the best performing text correction module of Google Docs reaches $F_{0.5}$ score of 58.59, the corrections proposed by text correction module of Office 365 Word have only 24.90 $F_{0.5}$ score, and (2) the results of the best performing Google Docs (and Grammarly) are on the examined domain comparable to the best models proposed in the research area up to 2019. The research models developed later seem to outperform the examined commercial systems significantly. While we expected commercial tools to differ from those from the research area by having higher precision and possibly lower recall, the observed results do not confirm this hypothesis.

Note that we also perform a similar analysis comparing commercial and research systems in Czech in Section 3.5.3. We do not present it here, because no publicly available Czech dataset with a variety of error types existed by 2017 when my doctorate studies started.

2.5 Czech Systems

Probably the first complex system for automatic text correction in Czech is a grammar checker *Kontrola české gramatiky* [Petkevič, 2014]. As we already discussed in Section 2.3, the system employs several thousands of hand-coded rules to correct users' texts. It was used alongside spelling correction module developed at Lingea[9] in Microsoft Word software product from version 2003 up to version 2013, and Petkevič [2014] reported that it corrects 30-40% errors in Czech texts. Historically, Lingea also offered its own grammar error correction tool *Grammaticon*, but it is not sold anymore, and we did not find any information on its inner workings.

In Section 2.3, we also discussed *Korektor* [Richter et al., 2012], a Czech statistical and occasional grammar checker offered as a command line utility and a web service. Unlike the tool of Petkevič [2014], it mainly aims at correcting spelling errors occurring in individual words, and instead of using hand-coded rules, it is based on a statistical approach. Its main power stems from the three language models that can judge individual candidate hypothesis proposed by a simple suggester based on the Levenshtein distance. Richter et al. [2012] tested *Korektor* on three test sets, but, sadly, none of them is publicly available. The first test set is an error corpus *Chyby* [Pala et al., 2003] that consists of corrected texts written by Czech university students. Out of all annotated error types, Richter et al. [2012] evaluated *Korektor* on spelling and morphological errors. Richter et al. [2012] created the second test set by transcribing an audiobook and the third test set by marking spelling errors in web texts by *Korektor* and annotating the reported errors using human annotators. On the three test sets, *Korektor* accuracy is 73-92%, with worst results on *Chyby* test set.

Apart from the two described systems and approaches, a few other tools were also developed. These are mostly undocumented spelling correction and diacritization online tools. An example of one of the documented tools is an online diacritization tool *CzAccent* [Rychlý, 2012] that for each undiacritized word suggests its most frequent diacritized variant as observed in a large corpus.

Being on the edge between deployed systems and research is the prototype demo system for grammar checking of Holan et al. [1997] developed as early as 1997. Their system heavily relies on syntactic parsing to reveal syntactic inconsistencies. These are then checked for errors.

In the research area, three years before Holan et al. [1997], Kubon and Platek [1994] sketched a specification for grammar-based grammar checker that iteratively runs two automata: the first one removing error-free subsequences of input and the second one correcting the remaining blocks with errors. Not much research has been further conducted on the full-scale GEC, but rather attempts on developing models for specific areas of GEC were performed. These include models for fixing errors in punctuation [Jakubíček and Horák, 2010, Kovář, 2014, Kovář et al., 2016] or for errors in subject-predicate agreement [Kovář, 2014, Novotná, 2018]. The proposed models usually comprise rules operating over syntactic trees.

The first and to the best of our knowledge only attempts to experiment with neural networks for correcting Czech texts were performed by the thesis author in his diploma thesis [Náplava, 2017]. The experiments utilized two models: (1) a simple character-level neural model comprising a bidirectional LSTM [Hochreiter and Schmidhuber, 1997] network for diacritics restoration, and (2) a similar model operating over words for spelling error correction. To test the spelling error correction model, *CzeSL Grammatical Error Correction Dataset (CzeSL-*

[9] `https://www.lingea.cz/`

GEC) [Šebesta et al., 2017], a parallel dataset with original and corrected texts, was compiled from two Czech learner corpora: Czesl-man corpus [Rosen, 2016] with manually annotated transcripts of essays of non-native speakers of Czech and unreleased parts of RoMI corpus with essays of Romani pupils with Romani ethnolect. The accuracy of the model for diacritics restoration reached 99.25%, and the accuracy of the spelling error correction model reached 98.70%. However, both models remained prototypes, and were never deployed.

Regarding datasets for training and/or evaluating models in Czech, we already mentioned non-public error corpus *Chyby* containing corrected essays and reviews of software products written by Czech university students. This dataset contains circa 410 000 original word forms and 8 639 errors annotated with their correction and with one of seven error types. We also discussed publicly available *CzeSL-GEC* that contains over 12k aligned sentences curated for spelling error correction. Besides these two datasets, we are not aware of any other Czech dataset that would be ready for training/testing model performance.

When comparing the GEC research conducted in Czech with research in English (see Section 2.3), it is evident that while Czech systems mostly utilize hand-coded rules and simple statistical approaches, the research in English moved its focus to utilizing the SMT and NMT approaches. With large enough training corpora, the machine translation approaches were shown to outperform all previous methods. Moreover, with good error type coverage, there is a great chance that the model generalizes well, and corrects such errors also in unseen texts. This also holds for difficult grammatical errors such as errors in agreement or punctuation as well as stylistic errors, for which it is nearly impossible to define manual rules nor to correct them based on simple statistical approaches.

3. The Story of My Doctorate

The following text aims at describing the work done by the thesis author in the field of natural language correction. The chapter is written in the chronological order of publication with most technical details such as exact model hyperparameters or learning rates omitted for better readability. For the technical details, we refer the reader to the Part II of the thesis, which contains the published work. We also note that the majority of the published work's source code is publicly available.

We illustrate our work on natural language correction in Figure 3.1. For each topic we have worked on, we display two points indicating the start and end of our work on the particular topic.

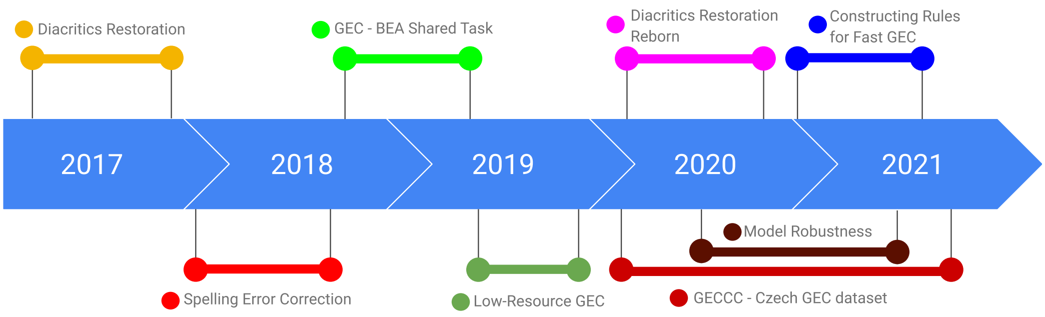


Figure 3.1: Illustration of the topics we worked on in the field of natural language correction. For each work, we display its start date and end date as two separate points.

We started by developing models for **diacritics restoration**. We proposed and implemented a novel combination of a recurrent neural model and an external language model and reached new state of the art on multiple languages. We further created and published a multilingual dataset comprising 12 languages for diacritics restoration as we found out that the field of diacritics restoration lacks such a dataset. **The paper comprising our work was published on LREC 2018 as Náplava [2017].** We discuss this work in Section 3.1.

We followed by discussing models that would be able to correct a larger variety of errors. A natural choice seemed to be to focus on **spelling error correction**. However, we found it extremely difficult to distinguish between typos and local grammatical errors, and rather moved our focus to GEC.

As the new shared task on **English GEC** (*BEA 2019 Shared Task on Grammatical Error Correction*) was announced, we fully concentrated on developing the best possible models for it. We developed a strong neural machine translation based model and participated in all three Shared Task's Tracks. **Our work was published on the BEA 2019 Workshop as Náplava and Straka [2019b]**, and we discuss it in Section 3.3.

One of the most important outcomes of the Shared Task was the creation and use of synthetic data. It was shown [Grundkiewicz et al., 2019] that we can automatically create synthetic data that boost model performance by a significant margin in English. Following this observation, we experimented with developing

GEC models for languages that have low amount of annotated data, and substantially outperformed previous state of the art. As no large enough Czech GEC dataset with annotated errors in the M2 format existed, we further compiled a first GEC dataset for Czech (*AKCES-GEC*) from existing learner corpora. **We presented this work on the 5th Workshop on Noisy User-generated Text as Náplava and Straka [2019a]**, and discuss it in Section 3.4.

As we compiled *AKCES-GEC* from learner corpora, it naturally lacked texts written by Czech natives. Furthermore, we opted to use the popular $M^2_{\beta=0.5}$ as the metric, however, there were some studies indicating that in certain scenarios, other values of β or other metrics might correlate better with human judgements. Therefore, we decided to create a new dataset that we named **Grammar Error Correction Corpus for Czech (GECCC)**, which includes texts from four user domains, for which we created completely new annotations. We also conducted a meta-evaluation of common GEC metrics against human judgements to select the best metric. The final *GECCC* dataset ranks amongst the largest and most diverse GEC datasets available in all languages. **The paper describing our work is accepted to be published in the TACL journal**. We discuss this work in Section 3.5.

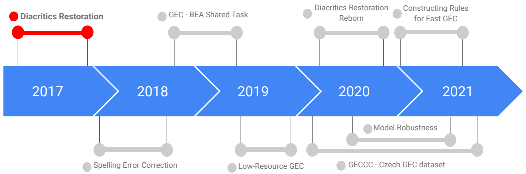
During 2019, **BERT [Devlin et al., 2019]** was shown to outperform many models in a variety of NLP tasks. We were naturally curious on how would it perform in natural language correction, and started by proposing a BERT-based model for **diacritics restoration**. We trained and evaluated the model on our multilingual dataset, and showed that on 9 of 12 of them, the new model significantly outperforms previous state-of-the-art combination of RNN with an external language model. We further inspected model performance on Czech in detail by evaluating it on a large set of domains, and also analysed model errors. **The paper was published in the Prague Bulletin of Mathematical Linguistics journal as Náplava et al. [2021].** We discuss it in Section 3.6.

We continued by building a GEC model based on BERT. We put an effort into creating efficient rules and using these, trained the correction model. Although the results are not state of the art, they are faster than our previous neural machine translation based models. **We presented our work on the 7th Workshop on Noisy User-generated Text as Straka et al. [2021]**. We discuss our work in Section 3.7.

Finally, we moved our focus to testing model robustness in noisy scenarios. We developed a tool for inserting natural noise into texts. The tool tries to mimic humans by estimating human errors' distribution on GEC datasets. We used the tool to evaluate various models in multiple languages, and confirmed that even current state-of-the-art models suffer from natural noise. We further proposed two ways to mitigate noise and evaluated them. **We presented our work on the 7th Workshop on Noisy User-generated Text as Náplava et al. [2021]**. We discuss our work in Section 3.8.

3.1 Diacritics Restoration

The full paper is available in Chapter 5, and it was published as Náplava et al. [2018].


Diacritics Restoration
GEC - BEA Shared Task
Diacritics Restoration Reborn
Constructing Rules for Fast GEC
2017
2018
2019
2020
2021
Model Robustness
Spelling Error Correction
Low-Resource GEC
GECCC - Czech GEC dataset

We started our work in natural language correction with diacritics restoration task. Recall that we introduced the diacritics restoration task and provided several examples in Chapter 2. Although this task covers only one of the plethora of possible error types, it is a well-defined and practical task as in certain languages, people for various reasons often write without diacritics. Apart from solving the task, we also hoped that experiences gained from developing models for diacritics restoration could also help us in the future with developing models capable of correcting a larger variety of error types.

Back in 2017, the state-of-the-art approaches for diacritics restoration mostly utilized large statistical n-gram language models [Richter et al., 2012, Ljubešić et al., 2016]. These were used to disambiguate between several variants obtained either by variants of the Viterbi algorithm applied on the prebuilt Hidden Markov model or lexicon approaches proposing dictionary variants. Meanwhile, in other areas of natural language processing, recurrent-neural networks seemed to excel. They were shown to outperform previous state of the art statistical models in machine translation [Sutskever et al., 2014, Cho et al., 2014, Bahdanau et al., 2015], named-entity-recognition [Huang et al., 2015, Lample et al., 2016], part-of-speech-tagging [Huang et al., 2015, Ling et al., 2015] and many other areas. One of the key factors behind their success was their ability to consider a larger context. While statistical n-gram language models need to, for computational reasons, use a limited context of for example 5 neighbouring tokens, the recurrent neural networks operate over the whole input. Therefore, we were interested in whether we can combine the two approaches (the recurrent neural network, which has a large perceptive field, and language model that on the other hand gains general language knowledge from large monolingual corpora), in the diacritics restoration task to achieve superior results. Moreover, we hypothesized that a recurrent model may exhibit good results even alone, i.e. without a language model.

As we discussed in Section 2.2, in diacritics restoration it was, unfortunately, a common practice that each new paper presented their model performance on a new dataset, and unlike GEC, there were no standardized datasets. On the other hand, diacritics restoration was an active area of research in many languages such as Czech [Richter et al., 2012], Vietnamese [Nguyen and Ock, 2010, Pham et al., 2013] or Croatian [Ljubešić et al., 2016]. To make the future research and model comparison easier, we decided to create a multilingual corpus for diacritics restoration covering a broad range of languages.

Objectives of the work

1) Explore the possibilities of using models based on recurrent neural networks and their combination with statistical language models in diacritics restoration task.
2) Create a dataset covering a broad range of languages for which diacritics restoration is a relevant task.

3.1.1 Model

One of the most popular recurrent neural architectures is the sequence-to-sequence (*seq2seq*) architecture [Sutskever et al., 2014, Cho et al., 2014]. The seq2seq approach comprise a bidirectional encoder [Schuster and Paliwal, 1997] that from given tokenized text produces a vector representation of the input sequence. This vector is then processed by a decoder that outputs the desired target sequence one token at a time autoregressively (autoregressive = predictions from a previous state are used to generate the next step). To overcome the bottleneck of the single vector representation of the input sequence, an attention mechanism [Bahdanau et al., 2015] was proposed, and it allows the decoder to attend to specific tokens when decoding. This model is particularly useful for tasks with an unknown mapping between the input and target tokens such as machine translation, and it could also be useful for other tasks such as diacritics restoration. Given enough training data, it outperforms previous state-of-the-art results. By 2017, such models came also to grammatical error correction, when Xie et al. [2016] proposed a character-level seq2seq model for grammatical error correction in English with promising results.

In our first attempt at diacritization model with RNN, we considered that diacritics restoration has a clear mapping between source and target tokens. Therefore, a simpler sequence tagging approach was considered and implemented. It is based on a bidirectional encoder with LSTM unit [Hochreiter and Schmidhuber, 1997] that for each input character predicts its correctly diacritized variant. Additionally, our model uses two stacked LSTM layers with residual connections [He et al., 2016], because our experiments proved them efficient. An illustration of the model architecture without residual connections is displayed in Figure 3.2

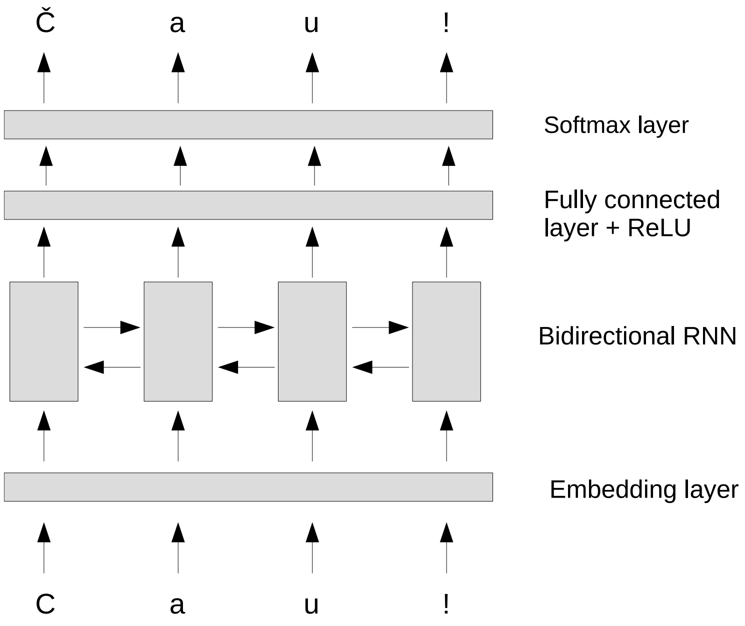


Figure 3.2: Recurrent neural model for diacritics restoration. Embedded characters are inputted into a bidirectional recurrent-neural network, whose outputs are projected using a linear layer and softmax classifier into a distribution over a set of possible diacritized variants of input characters.

Operating over characters instead of whole words was decided for two reasons: 1) to mitigate the issue with out-of-vocabulary words, 2) to reduce the size of an output layer which now has to accommodate the (much smaller) character vocabulary instead of the entire word vocabulary. Although we considered it the best option at that time, retrospectively, using intermediate subword units such as that outputted by the WordPiece algorithm [Schuster and Nakajima, 2012] would most probably be a better choice. While keeping the size of input and output vocabularies decent, using subwords would result in shorter distances between relevant pieces of text that would probably make the distant reasoning easier.

We further extended the recurrent neural model with an external language model in the decoding phase. Technically, in each decoding step, a beam of hypothesis is extended by the most probable variants as proposed by the recurrent model, and each time a whole word is decoded, the probabilities of items in the beam are weighted using external language model probabilities. Given that x denotes the input sequence, y stands for the sequence of decoded symbols, P_{NN} and P_{LM} are neural network and language model probabilities and α specifies the weight of the language model, the hypothesis probability in step k was computed as:

$$P(y_{1:k}|x) = (1 - \alpha) \log P_{NN}(y_{1:k}|x) + \alpha \log P_{LM}(y_{1:k}),$$

3.1.2 Model Evaluation

To evaluate system performance, we use the standard diacritics measure called alpha-word accuracy (see Section 2.1.2 for overview of common metrics in diacritics restoration) in all our experiments.

One of our main intentions when planning the work was to improve performance on Czech. However, the datasets used by prior works [Rychlý, 2012, Richter et al., 2012] were not available to us. Therefore, we first evaluated our model on a dataset of Ljubešić et al. [2016] comprising 3 Slavic languages (Croatian, Serbian and Slovene) from two sources (Wikipedia articles and Twitter posts). Moreover, we created a new dataset for Czech from the SYN2010 corpus [Křen et al., 2010] and PDT3.0 [Hajič et al., 2018] and compared our model to all relevant baselines and prior works. Note that in English, diacritics restoration is not a discussed task as the standard English alphabet does not contain letters with diacritics. Finally, for the apparent lack of supervised diacritization data, we collected a multilingual dataset for diacritization as a follow-up work to the development of our model, and we describe the resulting multilingual corpus in Section 3.1.3.

Regarding the Slavic dataset of Ljubešić et al. [2016], we compared our model with two methods of Ljubešić et al. [2016]: *Lexicon* and *Corpus*. The simple *Lexicon* method replaces each undiacritized word with its most frequent variant as seen in the training data (in Czech, this is the same approach as Rychlý [2012]). The state-of-the-art *Corpus* method incorporates a pretrained language model via a log-linear model (in Czech, this is similar to Richter et al. [2012]). As can be seen in Table 3.1, the combination of the recurrent model and the language model outperforms the *Corpus* method by a large margin, and reduces its error by more than 20% on all languages. Moreover, even the recurrent model alone performs satisfactorily. It outperforms *Corpus* method on clean Wikipedia data, but has slightly worse results on noisier data from Twitter.

System	wiki			tweet		
	hr	sr	sl	hr	sr	sl
Lexicon	99.36%	99.24%	99.33%	99.17%	98.93%	98.20%
Corpus	99.57%	99.47%	99.62%	99.38%	99.17%	99.12%
RNN model	99.67%	99.61%	99.70%	99.32%	99.39%	98.82%
RNN model + LM	**99.73%**	**99.68%**	**99.74%**	**99.51%**	**99.44%**	**99.30%**
Error reduction	36.81%	39.74%	30.45%	21.62%	32.14%	20.77%

Table 3.1: Diacritics restoration results obtained Croatian (HR), Serbian (SR) and Slovenian (SL) Wikipedia and Twitter testing sets of Ljubešić et al. [2016].

Regarding the experiments on Czech, we used the large SYN2010 corpus [Křen et al., 2010] with more than 8M sentences collected from Czech literature and newspapers for training. To evaluate the model, PDT3.0 [Hajič et al., 2018] testing set with 13 136 sentences originating from Czech newspapers was used. We compared our system to several systems available at that time: Microsoft Office Word 2010, ASpell [Atkinson, 2006], *Lexicon* and *Corpus* methods as proposed by Ljubešić et al. [2016], CZAccent [Rychlý, 2012] and a former state-of-the-art model for Czech diacritics restoration Korektor [Richter et al., 2012]. Because Microsoft Office Word and ASpell require user interaction, only a decent subset of the testing set was corrected and compared. As can be seen in Table 3.2, our recurrent model reaches the best results. Moreover, when combined with a language model, it reduces error of the former state-of-the-art Korektor by more than 64%.

System	Word accuracy
Microsoft Office Word 2010 (*)	89.10%
ASpell (*)	88.39%
Lexicon	95.27%
CZACCENT	96.07%
Corpus	97.13%
Korektor	98.61%
RNN model	98.87%
RNN model + LM	**99.51%**

Table 3.2: Diacritics restoration results obtained on Czech. The (*) denotes reduced test data.

3.1.3 Multilingual Dataset

During development of the diacritization model, we suffered from the lack of publicly available datasets for diacritics restoration. Although there is a large amount of languages with diacritics, the vast majority of works on diacritization focused on either a single language or small set of languages. Moreover, in a non-trivial amount of cases, the training and evaluation datasets were newly created for each new model using custom pipelines such as in works of Ljubešić et al. [2016], Richter et al. [2012] or Rychlý [2012]. To help future research, we decided to assemble and publish a new dataset covering a broad range of languages.

An obvious way of creating diacritization corpus is to strip the diacritics. However, the standard method for stripping diacritics has some drawbacks, which prelude the straightforward usage for creation of diacritization datasets as means for supervised machine learning: Specifically, the standard method for stripping diacritics was to convert input word to NFD [The Unicode Consortium, 2017, Normalization Form D] which decomposes composite characters into a base character and a sequence of combining marks, then remove the combining marks, and convert the result back to NFC. However, we noted that the method does not strip diacritics from some letters (e.g. ø, ł or đ) and proposed a new method. For each character of a word, it inspects its name in the Unicode Character Database. If it contains a word *WITH*, we remove the longest suffix starting with it, try looking up a character with the remaining name and yield the character if it exists. The method is illustrated below:

ø (hex code: 00F8)	→	LATIN SMALL LETTER O **WITH STROKE**
	→	LATIN SMALL LETTER O
	→	o (hex code: 006F)

We followed by identifying languages for which the diacritics restoration is a relevant and non-trivial task. Naturally, to include a language, its alphabet must contain characters with diacritics. Moreover, so that the task itself is not trivial, multiple diacritized variants should exist for an undiacritized word. To identify a potential set of languages, we considered every language contained within UD 2.0 [Nivre et al., 2017], took its plain texts, and stripped diacritics using the described method from them. This provided us with word diacritics ratio in each language. To satisfy the task complexity needs, we estimated a ratio of words for which the diacritics restoration could be difficult. Specifically, we estimated a word error rate of the *Lexicon* baseline method: for each undiacritized word, its most frequent diacritized variant was obtained from CoNLL 2017 Shared Task raw data [Ginter et al., 2017], and the word error rate was then estimated on a dedicated different part of the same dataset. Given the percentages of words and word error rate numbers, we selected 12 languages with higher error rate as the most relevant and non-trivial for the diacritization task, and present them in Table 3.3.

Comparing the 12 selected languages with published literature, we found that the selected set contains relevant languages, but also misses Arabic. Although the diacritization task is also performed in Arabic, the diacritics in Arabic behaves differently to already discussed languages. Instead of the diacritics being represented as variants of the underlying letters, the diacritical marks (short vowels and consonant length) are represented as stand-alone Unicode combining characters in Arabic script, plus several such mark characters can be combined. Given that we lacked the expertise in the Arabic language and were not confident that we would create a high-quality dataset, we therefore decided not to include Arabic.

Language	Words with diacritics	Word error rate of *Lexicon*
Vietnamese	88.4%	40.53%
Romanian	31.0%	29.71%
Latvian	47.7%	8.45%
Czech	52.5%	4.09%
Slovak	41.4%	3.35%
Irish	29.5%	3.15%
French	16.7%	2.86%
Hungarian	50.7%	2.80%
Polish	36.9%	2.52%

Table 3.3: 12 languages selected for the new multilingual diacritics restoration dataset. For each language, we report the percentage of words with diacritics and the word error rate of Lexicon.

Having identified the 12 interesting languages from the diacritization point of view, we compiled the multilingual dataset for 12 languages from two sources: W2C corpus [Majliš, 2011] with texts from Wikipedia and the general web, and the CommonCrawl corpus with language annotations generated by Buck et al. [2014] with a substantially larger amount of general web texts. While the Wikipedia data were distributed into both training, development and testing sets, the general web texts comprise only a training part of the dataset due to their potential noisiness. The basic dataset statistics together with performance of 4 systems (*Lexicon*, *Corpus*, *Recurrent Model*, *Recurrent Model with LM*) are presented in Table 3.4. The dataset is publicly available at `https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-2607`.

Language	Wiki sentences	Web sentences	Lexicon	Corpus	Our model	Our model +LM
Vietnamese	819 918	25 932 077	71.64%	86.39%	96.22%	97.73%
Romanian	837 647	16 560 534	85.33%	90.46%	90.18%	98.37%
Latvian	315 807	3 827 443	91.01%	94.57%	96.08%	97.49%
Czech	952 909	52 639 067	95.90%	98.14%	98.52%	99.06%
Polish	1 069 841	36 449 109	97.08%	98.41%	98.91%	99.55%
Slovak	613 727	12 687 699	97.34%	98.37%	98.68%	99.09%
Irish	50 825	279 266	97.35%	98.00%	98.42%	98.71%
Hungarian	1 294 605	46 399 979	97.49%	98.32%	98.88%	99.29%
French	1 818 618	78 600 777	97.93%	99.31%	99.48%	99.71%
Turkish	875 781	72 179 352	98.78%	99.05%	99.12%	99.28%
Spanish	1 735 516	80 031 113	99.11%	99.53%	99.56%	99.65%
Croatian	802 610	7 254 410	99.31%	99.47%	99.51%	99.67%

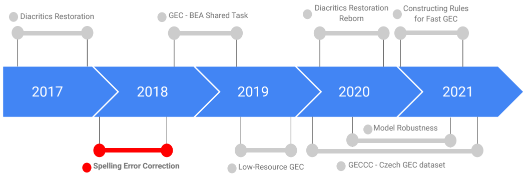
Table 3.4: Statistics of the new multilingual dataset for diacritics restoration. We also report results of 4 automatic systems.

Given the results obtained from experiments in Section 3.1.2 as well as from experiments conducted on the new multilingual dataset, we could state that the proposed at-that-time-novel combination of the neural recurrent model and the external language model greatly outperforms the previous models and is a state-of-the-art method for four languages (Czech, Slovenian, Croatian and Serbian) in two datasets. Moreover, we have shown that even the recurrent model alone performs satisfactorily, which may be beneficial for cases in which the external language model would be too slow.

Main outcomes and conclusions

1) We proposed a new architecture for diacritics restoration based on a recurrent neural network and an external language model.
2) We compiled a new multilingual dataset for diacritics restoration covering 12 languages, and evaluated two baseline methods on it.
3) We evaluated the proposed model on one existing dataset comprising 3 Slavic languages, one new Czech corpus and the newly created multilingual dataset. In all cases, the combination of recurrent model with external statical language model outperformed previous results. Moreover, even without language model, the recurrent neural network achieved surprisingly solid results.

3.2 Towards Spelling Error Correction


Diacritics Restoration
GEC - BEA Shared Task
Diacritics Restoration Reborn
Constructing Rules for Fast GEC
2017
2018
2019
2020
2021
Model Robustness
Spelling Error Correction
Low-Resource GEC
GECCC - Czech GEC dataset

After achieving state-of-the-art results on the diacritics restoration task, we moved our focus to developing systems capable of correcting larger variety of error types. A natural choice was extending our system with spelling correction as it is a common part of many current tools, and given that spelling errors affect only single words, similar model to diacritics restoration could be used.

According to Jurafsky and Martin [2000], spelling errors can be divided into two categories: non-word errors or real-world errors, motivated by the fact that accidentally misclicking a computer key may result either in an invalid or a valid word. Given that even a valid word may actually be wrong makes the task more difficult as a simple error detection method based on existence in a dictionary would provide suboptimal results.

Though we were not happy with the fact that there was no standard dataset for diacritics restoration, the situation in the spelling error correction appeared to be even worse. The available public datasets such as the dataset of Birkbeck spelling error corpus / Roger Mitton or Peter Norvig[1] consisted of individual uncorrected and corrected word variants completely lacking context. If the datasets contained context such as the datasets developed by Brill and Moore [2000] or Li et al. [2018], they were not publicly available [Hagiwara and Mita, 2020].

Having no dataset for spelling error correction, we decided to create a new dataset on our own. We started writing annotation rules for what a spelling error in a text is. However, it turned out extremely difficult, as many local errors

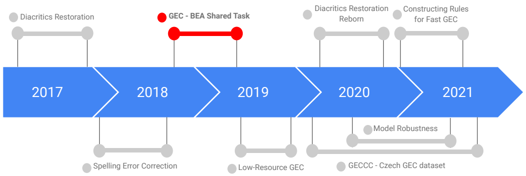
[1] http://norvig.com/ngrams/spell-errors.txt

differing in a single character are actually not spelling errors but other error types such as subject-verb-agreement errors (*he appear* vs *he appears*) or verb-tense errors (*he likes* vs *he liked*). Moreover, standard approaches did omit cases where a word could be split by accident into two words and vice versa, which, however, should belong there according to us.

By the time we were at a loss about how to approach the task, a new shared task on grammatical error correction on English appeared. It was a great incentive to change our focus, and, instead of spending time on an intermediate task of vaguely defined spelling error correction, to start developing a general system capable of correcting any grammatical error.

3.3 Grammatical Error Correction

The full paper is available in Chapter 5, and it was published as Náplava and Straka [2019b].


Diacritics Restoration
GEC - BEA Shared Task
Diacritics Restoration Reborn
Constructing Rules for Fast GEC
2017
2018
2019
2020
2021
Model Robustness
Spelling Error Correction
Low-Resource GEC
GECCC - Czech GEC dataset

Latest CoNLL 2014 Shared Task on Grammatical Error Correction [Ng et al., 2014] introduced a dataset of uncorrected and corrected English essays written by South-East Asian undergraduates. Compared to three previous shared tasks on error correction [Dale and Kilgarriff, 2011, Dale et al., 2012, Ng et al., 2013], the task's main objective changed to correcting all error types instead of only a subset of error types. Moreover, the metric was decided to be $F_{0.5}$ instead of F_1 to emphasize precision over recall (see Section 2.1.1 for more details on metrics). The introduction of the dataset together with an improved metric and strong baselines proposed by the Shared Task participants attracted research interest into the area of grammatical error correction. When compared to the situation previously described in the diacritics restoration and spelling error correction, the shared task's contributions to future research were extremely valuable.

As we discussed in Section 2.3, between 2014 and the end of 2018, when the next shared task was announced (*Building Educational Applications 2019 Shared Task: Grammatical Error Correction* [Bryant et al., 2019]), much research was done on grammatical error correction. The system architecture has slowly changed from using a combination of separate classifiers [Dahlmeier et al., 2012] and statistical machine translation approaches [Brockett et al., 2006, Felice et al., 2014, Junczys-Dowmunt and Grundkiewicz, 2014] to using neural based models, and especially, neural machine translation [Sutskever et al., 2014, Cho et al., 2014, Vaswani et al., 2017]. Using neural models and public datasets, the scores on the de-facto-standard CoNLL 2014 test set improved from 37.33% [Felice et al., 2014] to more than 56% $F_{0.5}$ score [Junczys-Dowmunt et al., 2018, Lichtarge et al., 2018] (see Table 2.2). Training on a non-public Cambridge Learner Corpus [Nicholls, 2003], Ge et al. [2018] even report reaching human-level performance scores.

Apart from the progress in the development of models, several tools and data appeared. One of the most important is ERRANT [Felice et al., 2016, Bryant et al., 2017], which is a tool for automatic annotation and evaluation for grammatical error correction. Furthermore, as neural models require large training data, Wikipedia revision histories were found to be a valuable source of large amounts of rather noisy training data [Grundkiewicz and Junczys-Dowmunt, 2014]. Finally, to test not only grammaticality of the text but also its fluency, JFLEG dataset [Napoles et al., 2017a] was proposed.

3.3.1 BEA-2019 Shared Task on Grammatical Error Correction

At the beginning of year 2019, Building Educational Applications Workshop announced a new shared task on grammatical error correction: The BEA-2019 Shared Task on Grammatical Error Correction [Bryant et al., 2019]. One of its main objectives was to provide a platform on which systems could be evaluated under controlled conditions. To ensure this, three tracks were defined:

- Restricted Track – limits the annotated training data to specified datasets (FCE, Lang-8 Corpus of Learner English, NUCLE, W&I+L – see Table 2.1 for more information on their sizes and user domains)
- Unrestricted Track – allows using any data for training systems
- Low-Resource Track – forbids using any annotated data different from W&I+L development set for training systems

Along with the shared task, a new dataset was introduced: the Cambridge English Write & Improve and LOCNESS corpus. The corpus originates from two sources: Write & Improve [Yannakoudakis et al., 2018] with annotated texts of second learners of English and LOCNESS [Granger, 2014] that contains decent amount of annotated essays of native English students. Note that the corpus is often referred to as *W&I+L* or *BEA 2019*. Compared to the CoNLL 2014 test set, the W&I+L test set is almost 4 times bigger (1 312 vs 4 477 sentences), it contains larger variety of user domains (note that user domain is a combination of the domain of the user and the type of the text and was mentioned in Section 2.2) with annotated English levels, and was written by people all around the globe (as opposed to South-East Asian undergraduates only in the CoNLL 2014 test set). The scorer used to compute the $F_{0.5}$ score was changed from MaxMatch to ERRANT scorer as ERRANT can also report performance with respect to individual error types (see Section 2.1.1).

Objectives of the work

1) Propose and implement the best possible system for each of the three Tracks of the Shared Task.

3.3.2 BEA 2019 Submission

Our work on grammatical error correction started with discussions of the model architecture. By the beginning of 2019, the recurrent models seemed to already be outperformed by the Transformer architecture [Vaswani et al., 2017], which was on the model side attributed to the self-attention mechanism that removed recurrences and allowed to model long range dependencies better, and also to the availability of large enough training data [Junczys-Dowmunt et al., 2018]. In GEC, two papers Junczys-Dowmunt et al. [2018], Lichtarge et al. [2018] using the Transformer architecture and having state-of-the-art results were already published. An alternative GEC research was represented by a convolutional encoder-decoder model [Chollampatt and Ng, 2018b] that used convolutions to replace recurrences. We decided to use the Transformer model, which showed promising results across the NLP field.

We re-implemented the model proposed by Junczys-Dowmunt et al. [2018]. Specifically, we extended the standard Transformer model by source word dropout [Srivastava et al., 2014], and its loss by term that assigns higher weight to words that should change. To make regularization even more effective, we decided to dropout also whole target word embeddings randomly in training.

As the so called iterative decoding [Lichtarge et al., 2018, Ge et al., 2018] turned out effective, we implemented it as well. In iterative decoding, the trained system is used incrementally until a stopping criterion is met. While Ge et al. [2018] use additional language model probabilities to estimate sentence fluency and use these as a stop criterion, we follow an approach of Lichtarge et al. [2018], who rely solely on the trained model probabilities. Specifically, we use the model log-likelihood as the cost function, and run iterative decoding as long as the cost of the correction is less than the cost of the identity translation times a learned constant.

Finally, to reduce variance during training and hopefully also achieve better results, we used checkpoint averaging as described by Popel and Bojar [2018].

3.3.3 Data

We decided to participate in all three tracks of the shared task, therefore, we needed to think about what data sources to use.

In the Restricted track, the possible data sources were strictly defined. We did not find it meaningful to omit any data source, and used all allowed data for training the model. As the majority of data came from the noisiest Lang-8 corpus, we however oversampled other datasets to make the proportions of noisy and cleaner data more equal.

Regarding the Low-Resource track which forbids to use any annotated data, we decided to incorporate Wikipedia revisions. These contain complete edit history for each page on Wikipedia, and were already previously successfully used for training systems for GEC. To create a training corpus from Wikipedia revisions, Grundkiewicz and Junczys-Dowmunt [2014] defined a set of rules employing edit distances that were used to decide, whether a sentence before the revision and its edited variant are similar enough to be considered a training pair. Although the resulting corpora are typically quite noisy as many sentences do not contain grammatical errors but for example factual improvements, their size is enormous. Lichtarge et al. [2018] have shown that in a two-stage training, in which a system is first trained on the Wikipedia corpus and then finetuned on authentic data (e.g. NUCLE corpus), the system performance is significantly better than when only one-stage training with authentic data is used. Moreover, even when the system is trained only with the Wikipedia data, the results seemed quite strong, especially with the iterative decoding. We followed an approach of Lichtarge et al. [2018], and created 190M aligned pairs of noisy and corrected segments from Wikipedia revision history dump, and used them for training.

Finally, in the Unrestricted Track, we used the data from the Restricted and Unrestricted Tracks.

3.3.4 Training and Results

We used the Restricted Track as an experimental playground, on which we tried numerous hyperparameters and settings. Specifically, we experimented with the value of source and target word dropout, weight for non-identity words in the modified objective, constant used in iterative decoding, oversampling cleaner data and the differences between lighter *Transformer-BASE* and heavier *Transformer-BIG* architecture. As can be seen in Table 3.5, the chosen hyperparameters improved the performance of the baseline Transformer model by almost 12 points in the final score. It is also evident that each proposed improvement boosted the model performance greatly. Note that while the column *Combined* in Table 3.5 shows scores on the entire W&I+L development set, the columns *A*, *B* and *C* present scores on specific subsets of the development set that differ in the CEFR level of the writer of the original noisy text (A – beginner, B – intermediate, C – advanced), and the *N* column presents scores on the subset of the development set that comprises original noisy texts written by native speakers.

System	A	B	C	N	Combined
Transformer-BASE architecture	39.98	32.68	23.97	14.49	32.47
Transformer-BIG architecture	39.70	35.13	26.22	20.20	34.20
+ 0.2 src drop, 0.1 tgt drop, 3 MLE	42.06	38.25	28.72	23.80	38.15
+ Extended dataset	45.99	41.79	32.52	27.89	40.86
+ Averaging 8 checkpoints	47.90	44.13	36.19	29.05	43.29
+ Iterative decoding	48.75	45.46	37.09	30.19	44.27

Table 3.5: $F_{0.5}$ scores of incremental improvements of our system on the W&I+L development set.

In the Low Resource Track, we used the best hyperparameters from the Restricted Track to train the model on the Wikipedia data only.

Finally, in the Unrestricted Track, we used the best system from the Low-Resource track that was trained on parallel data from Wikipedia revisions, and finetuned it on the training data from the Restricted Track.

The results of our systems together with performance of best performing systems and number of participants are summarized in Table 3.6. Our system ranked 10th out of 21 systems in the Restricted Track, 3rd out of 7 submitted systems in the Unrestricted Track and 5th out od 9 systems in the Low-Resource Track.

There were two reasons behind the performance gap between ours and the best performing models. First, the best models typically used multiple models combined into ensembles, and, more importantly, they utilized large synthetically generated noisy data for pretraining. It is important to describe the concept of

Track	P	R	$F_{0.5}$	Best	Rank
Restricted	67.33	40.37	59.39	69.47	10 / 21
Unrestricted	68.17	53.25	64.55	66.78	3 / 7
Low Resource	50.47	29.38	44.13	64.24	5 / 9

Table 3.6: Official results of Building Education Applications 2019 Shared Task on Grammatical Error Correction. The reported scores are precision, recall and $F_{0.5}$ measured on the test set.

pretraining systems and synthetic data here as we will further use this approach. In the traditional approach to machine learning, models are typically trained in a single stage to model the authentic training data, and by doing so hopefully also generalize to unseen test data. As deep learning models, such as the recurrent neural networks or models based on Transformer, require large amounts of training data to work well and these are often not available, a different two-stage strategy appeared. The so-called synthetic (sometimes also known as artificial) data that try to mimic the real data are first automatically created, and the model is trained with them. This process is typically called *pretraining*. A simple synthetic data for GEC can be for example created by taking a clean text and inserting random words to it and deleting some of its words. Although the quality of generated synthetic data is lower than of the authentic annotated data, we can generate large amount of them as they typically require only clean texts. Given large enough synthetic data with decent quality, we hope that in the pretraining phase the model learns patterns that will be useful for the second stage, in which the authentic annotated data are used. The second stage is typically called *finetuning* phase, and the pretrained model's weight are further readjusted with authentic data. As the model was already pretrained, to reach a similar model performance on separate testing data, the amount of authentic data needed is typically lower than when no pretraining is used.

Both Grundkiewicz et al. [2019], who won the shared task, and Choe et al. [2019], who placed the second, generated synthetic data to boost their model performance. Choe et al. [2019] first extracted edits from W&I+L training set and also defined custom manual noising scenarios for preposition, nouns and verbs, and used these to noise a large clean text. Grundkiewicz et al. [2019], on the other hand, used an unsupervised approach, and when noising a text, they allowed deleting a word, inserting a random word, shuffling of nearby words and also replacing a word with one of its spell-checker suggestions. Especially the simple replacing rule turned out particularly effective as it can generate examples that are hard to detect and correct.

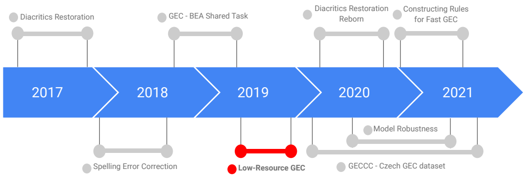
Recall, that our model in the Unrestricted Track was first pretrained on Wikipedia revisions, and then finetuned with the authentic data. As can be seen from that our system is in the Unrestricted Track better by circa 5 points than our model in the Restricted Track, the large Wikipedia data also helped the model to better generalize, and thus provide better corrections. However, probably due to the noisy nature of the Wikipedia data, and the fact that not all filtered corrections between two consecutive snapshots need to be grammatical errors, the approach of two best teams of Grundkiewicz et al. [2019], Choe et al. [2019] turned out better.

Main outcomes and conclusions

1) We developed strong Transformer-based models for GEC and submitted them to all three Tracks of the shared task. Our systems ranked 10th out 21 systems in the Restricted Track, 3rd out of 7 systems in the Unrestricted Track and 5th out of 9 systems in the Low Resource Track.

3.4 GEC in Low Resource Scenarios

The full paper is available in Chapter 5, and it was published as Náplava and Straka [2019a].


Diacritics Restoration
GEC - BEA Shared Task
Diacritics Restoration Reborn
Constructing Rules for Fast GEC
2017
2018
2019
2020
2021
Model Robustness
Spelling Error Correction
Low-Resource GEC
GECCC - Czech GEC dataset

GEC in English is a long studied problem, with many existing systems and datasets. However, there has been only a limited research on error correction of other languages. As we discussed in Section 2.2, by the end of 2019, datasets existed for German [Boyd, 2018] and Russian [Rozovskaya and Roth, 2019], and efforts to create annotated learner corpora were also done for Chinese [Zhao et al., 2018], Japanese [Koyama et al., 2020] and Arabic [Alfaifi and Atwell, 2013]. One of the main reasons why the majority of research has been conducted on English is the availability of data. While there are at least 6 datasets for GEC in English with millions of sentences altogether (see Table 2.1), in languages other than English, we know of only a single dataset consisting of at most tens of thousands of sentence pairs. This is naturally an issue as the current approaches based on neural machine translation require large corpora to train properly.

One of the outcomes of the BEA Shared Task 2019 on Grammatical Error Correction (see Section 3.3.4) was the fact that generating synthetic errors for training is surprisingly effective on English. It helps to overcome the lack of annotated data, and also improves model performance even if large annotated data are available. As we can see from Table 3.6, Grundkiewicz et al. [2019] managed to train models without any authentic annotated data that are worse only by as little as 5 points than the best models utilizing also annotated data.

Motivated by the fact that synthetic data help models in English even in the Restricted Track setting with large annotated data, the fact that $F_{0.5}$ scores of state-of-the-art models in languages such as German (*45.22* [Boyd, 2018]) or Russian (*21.0* [Rozovskaya and Roth, 2019]) lag behind those of English (*69.47*, *61.30* [Grundkiewicz et al., 2019]) by a large margin, and the fact that datasets of these languages are relatively small, we decided to concentrate on these low-resource languages.

Key research questions of the work

1) Explore, whether incorporating synthetic data into training helps in case of low-resource languages such as German or Russian?
2) Do synthetic data help also in case of Czech?

3.4.1 Synthetic Data Generation

We identified two languages other than English with datasets prepared for grammatical error correction: German *FALKO-MERLIN* [Boyd, 2018] and Russian *RULEC-GEC* [Rozovskaya and Roth, 2019]. As can be seen in Table 2.1, the German dataset consists of 24 077 sentences, and the Russian dataset has 12 480. Having experience in training models for English on various datasets, we knew that training a model solely on these data would result in heavily overfitted models with poor results. We therefore decided to generate synthetic data to enlarge the training set. There are three main approaches for generating synthetic data:

1. **Rule-based approach** – the synthetic data are created from clean texts by rule-based substitutions or by using a subset of the following operations: *token replacement*, *token deletion*, *token insertion*, *multitoken swap* and *spelling noise introduction*. In Section 3.3.4, we have already described two approaches of Grundkiewicz et al. [2019] and Choe et al. [2019] that were the newest representatives of this approach. To name a few others, Brockett et al. [2006] used a set of hand-constructed regular expressions to create noisy data exhibiting characteristics of countability errors associated with mass nouns as produced by Chinese second learners such as *much advice* → *many advice* or *much good advice* → *many good advices*, Yuan and Felice [2013] extracted edits from NUCLE dataset and applied them on a clean text, and Zhao et al. [2019] use a straightforward approach and allow deleting a word, inserting a random word, replacing a word with random word and shuffling of neighbouring words.

2. **Model-based approach** – is based on the so-called backtranslation model. The core of this approach is a machine translation model trained in the opposite direction, i.e. it learns to translate from correct into incorrect sentences [Rei et al., 2017].

3. Generating from **other sources** – by 2019, such a source was mainly Wikipedia revision histories as described for example by Grundkiewicz and Junczys-Dowmunt [2014]. Shortly after publicizing this work [Náplava and Straka, 2019a], GitHub Typo Corpus Hagiwara and Mita [2020] with errors extracted from GitHub appeared.

After considerations, we decided to use the first approach and method similar to Grundkiewicz et al. [2019] for the following reasons: (1) it was shown to provide high-quality difficult examples on English and can be easily extended to other languages, (2) we already tried using Wikipedia revisions and we already confirmed that using Wikipedia revisions is ineffective, (3) the backtranslation approach requires large volumes of training data, which is in the case of low resource languages intractable.

The algorithm of Grundkiewicz et al. [2019] works as follows. Given a clean sentence, it first samples a fixed number of words to modify. For each chosen word, one of the following operations is performed with a predefined probability: *substituting* the word with one of its ASpell proposals, *deleting* it, *swapping* it with its right-adjacent neighbour, or *inserting* a random word from a dictionary after the current word. To make the system more robust to spelling errors, the same operations are also used on individual characters. Grundkiewicz et al. [2019] set the probability of performing a word-level operation to 0.15 and 0.02 for character-level operations, and additionally set the probability for word substitution arbitrarily to 0.7, and the three remaining operations are chosen with a probability of 0.1. Recall that Grundkiewicz et al. [2019] performed the experiments on English, therefore, when extending the approach to other languages, we ran a small grid-search for each language that modified the probabilities slightly.

Because models after training often failed to correct errors in casing, we further extended the word-level operations of the algorithm by adding an operation that allows to change word casing. When later applied to generate synthetic samples also for Czech, we additionally included also an operation which changes diacritics.

We illustrate several sentences generated by the algorithm together with the original clean sentences in Table 3.7. Note that while some errors such as simple spelling errors or the subject-verb agreement error in the third sentence resemble real human errors, there are errors such as the newly introduced word *promulgating* in the fourth example that are on the other hand unlikely to be produced by a human.

O:	Best **known** **a**s a novelist , **M**atar was born in New York , grew up in Tripoli and Cairo , and now **lives in** England .
N:	Best **is** a novelist , **m**atar was**p** **in** born New York , grew up in Tripoli and Cairo , and now England .
O:	**The** machines became more **lighthearted** in the enlig**h**tened 18th century .
N:	machines **The** became more **IN** the enligtened 18**Y** centu**e**ry ,
O:	Do I think it ’s essential **that** a member wear**s** a tie ?
N:	Do I think it ’s essential a member wear a tie ?
O:	Jimmy Kimmel is a guest on the series premiere .
N:	Jimmy Kimmel is a guest**s** on t**eh** series premiere **promulgating** .
O:	She made lots of stir - frie’s and curries . . .
N:	She made lot**us** o**e** stir - f**ire**’s and curries . . .

Table 3.7: Examples of synthetic sentences generated for English. The sentences preceded by O: are the original tokenized clean sentences, and the sentences preceded by N: are the generated noisy synthetic sentences.

We would like to stress the importance of using ASpell dictionary to propose word candidates as part of our method for synthetic data generation. ASpell generates confusion sets of lexically and phonetically similar words, and thus introduces errors that are both hard to detect and correct. It is for example capable of generating errors in subject-verb agreement (*he agrees* vs *he agree*), errors in verb-tense (*I did see it* vs *I did seen it*) or errors in morphology (*He is smart than me* vs *He is smarter than me*). We would also like to note that two years later, in 2021, Flachs et al. [2021a] have experimented with using ASpell and Unimorph,[2] and have shown that although using ASpell works better than using Unimorph alone, mixing errors produced by ASpell with errors produced by Unimorph leads to better results on Czech and Russian, while it does not improve results on German and Spanish.

Having the algorithm for generating synthetic samples, we generated artificial data for German and Russian, and trained the GEC models.

3.4.2 German and Russian Results

We employed the described pipeline to generate large synthetic data comprising 10M sentences for English, Russian and German from clean WMT News Crawl monolingual training data [Bojar et al., 2017]. We opted to prepare data for and train also the English models to compare our models to current state of the art in English.

We used the synthetic data to pretrain the modified Transformer model, which is exactly the same as we used in the BEA Shared Task and described in Section 3.3.2. We trained a separate model for each language. As can be seen from Table 3.8 and Table 3.9, although these models never saw any authentic data, they were already better than the previous state-of-the-art systems on German [Boyd, 2018] and Russian [Rozovskaya and Roth, 2019] (see row *Our work – pretrained*).

System	Precision	Recall	$F_{0.5}$
Rozovskaya and Roth [2019]	38.0	7.5	21.0
Our work – pretrained	47.76	26.08	40.96
Our work – finetuned	63.26	27.50	50.20

Table 3.8: Comparison of our systems on Russian RULEC-GEC test set.

System	Precision	Recall	$F_{0.5}$
Boyd [2018]	51.99	29.73	45.22
Our work – pretrained	67.45	26.35	51.41
Our work – finetuned	78.21	59.94	73.71

Table 3.9: Comparison of our systems on German Falko-Merlin test set.

We further finetuned the pretrained models on a mixture of authentic[3] and synthetic data, which increased the performance even further (see row *Our work – finetuned* in Table 3.8 and Table 3.9). We found it important to use a mixture of authentic and synthetic data for finetuning, especially for German and Russian. When only the authentic data were used for finetuning, the model quickly

[2] Unimoph (`https://unimorph.github.io/`) is a database of morphological variants of words.

[3] For Russian, we used training data from RULEC-GEC, for German from Falko-Merlin and for English Lang-8 data, FCE, NUCLE and W&I+L.

System	W&I+L test	CoNLL 14 test No W&I+L	 With W&I+L
	including ensembles		
Lichtarge et al. [2019]	–	60.40	–
Zhao et al. [2019]	–	61.15	–
Xu et al. [2019]	67.21	–	63.20
Choe et al. [2019]	69.06	57.50	–
Grundkiewicz et al. [2019]	**69.47**	**61.30**	**64.16**
	no ensembles		
Lichtarge et al. [2019]	–	**56.80**	–
Xu et al. [2019]	**63.94**	–	60.90
Choe et al. [2019]	63.05	–	–
	no ensembles		
Our work – pretrained	51.16	41.85	44.12
Our work – finetuned	69.00	60.76	63.40

Table 3.10: Comparison of systems on two English GEC datasets. CoNLL 2014 Test Set is divided into two system groups (columns): those which do not train on W&I+L training data and those which do.

overfitted, and the results were significantly worse than when finetuning with the data mixture.

Regarding the results on English, the comparison is slightly more difficult. Many papers publish the results using ensemble models, which are however not directly comparable with a single system. As we can see from Table 3.10, our finetuned model outperformed all single models, while performing slightly worse than ensembles of Grundkiewicz et al. [2019] and Zhao et al. [2019] on the W&I+L and CoNLL14 test sets.

3.4.3 AKCES-GEC – First Czech GEC Dataset

As we mentioned in the Introduction of this thesis (see Section 1), developing models for Czech is one of our long-term goals. Unfortunately, no Czech dataset for GEC was available by 2019. Probably closest to it was a dataset of Náplava [2017], which consists of aligned sentences extracted from Czech learner corpora. Although the dataset could be used for training models, the absence of extracted edits make it impossible for evaluation by any of the standard GEC scorers.

Despite that no GEC datasets for Czech existed, several annotated learner corpora were being developed. These, in spite of existing in learner corpora specific formats, should have contained all information required to compile a GEC dataset. Therefore, we decided to compile a Czech GEC dataset from these learner corpora on our own.

Building the Dataset

Recall from Section 2.2 that the standard format for distributing and evaluating GEC datasets is the M2 format [Dahlmeier and Ng, 2012a]. This contains the original uncorrected sentences and for each original sentence a list of edits comprising its span, a replacement string and an error type. M2 format is used for the majority of GEC datasets, and other formats are used rarely. An example of dataset with different format is JFLEG [Napoles et al., 2017b], where different format was used because the dataset addresses fluency rewrites with large edits, for which the M2 format may not be optimal. Given the fact the learner corpora contain standard grammatical errors, we decided to use the M2 format.

The Czech learner corpora have been collected under an umbrella project AKCES [Šebesta, 2010]. This project comprises several acquisition resources – CzeSL (learner corpus of Czech as a second language), ROMi (Romani ethnolect of Czech Romani children and teenagers) and SKRIPT and SCHOLA (written and spoken language collected from native Czech pupils, respectively).

By the time we were performing the work, only the annotated Czesl-man corpus [Rosen, 2016] consisting of manually annotated transcripts of essays of non-native speakers of Czech was publicly available. Apart from it, we also managed to obtain additional data from CzeSL and ROMi authors: unreleased parts of CzeSL-man and also essays of Romani pupils with Romani ethnolect of Czech as their first language from ROMi corpus. Given these resources, we built a first Czech GEC corpus: *AKCES-GEC*. We tried to keep the error annotations where possible, and present the list of them in Table 3.11.

Error type	Description	Example	Occ
unspec	unspecified error type		69 123
incorBase	incorrect word base	musíš to [**posvětlit** → *posvětit*]	20 334
incorInfl	incorrect inflection	[**pracovají** → *pracují*] v továrně	8 986
dep	error in valency	bojí se [**pes** → *psa*]; otázka [**čas** → *času*]	6 733
agr	violated agreement rules	to jsou [**hezké** → *hezcí*] hoši; Jan [**čtu** → *čte*]	5 162
lex	error in lexicon or phraseology	dopadlo to [**přírodně** → *přirozeně*]	3 967
stylColl	colloquial expression	viděli jsme [**hezký** → *hezké*] holky	3 533
use	error in the use of a grammar category	pošta je [**nejvíc blízko** → *nejblíže*]	1458
wbdOther	other word boundary error	[**mocdobře** → *moc dobře*]; [**atak** → *a tak*]	1 326
rflx	error in reflexive expression	dívá [∅ → *se*] na televizi; Pavel [**si** → *se*] raduje	915
sec	secondary error (supplementary flag)	stará se o [**našich holčičkách** → *naše holčičky*]	866
vbx	error in analytical verb form or compound predicate	musíš [**přijdeš** → *přijít*]; kluci [**jsou**] běhali	864
wbdPre	prefix separated by a space or preposition w/o space	musím to [**při pravit** → *připravit*]	817
ref	error in pronominal reference	dal jsem to jemu i [**jejího** → *jeho*] bratrovi	344
problem	supplementary label for problematic cases		175
fwNC	foreign word	váza je na [**Tisch** → *stole*]	166
stylOther	bookish, dialectal, slang, hyper-correct expression	rozbil se mi [**hadr**]	156
neg	error in negation	[**půjdu ne** → *nepůjdu*] do školy	111
wbdComp	wrongly separated compound	[**český anglický** → *česko-anglický*] slovník	92
fwFab	non-emendable, „fabricated“ word	pokud nechceš slyšet [**smášky**]	78
disr	disrupted construction	znám [**hodné spoustu** → *spoustu hodných*] lidí	64
flex	supplementary flag used with fwFab and fwNC marking the presence of inflection	jdu do [**shopa** → *obchodu*]	34
stylMark	redundant discourse marker	[**no**]; [**teda**]; [**jo**]	15

Table 3.11: Error types used in the *AKCES-GEC* corpus taken from Jelínek et al. [2012], including the number of occurrences in the dataset.

The original data the corpus was built from contain explicit metadata on whether text were written by Romani or Czech second learners. Second learners are further divided based on their mother tongue, specifically, whether they come from the Slavic group or not. We enhanced our dataset with this information, and also used it when performing the dataset split into training, development and testing sets (in a ratio 90:5:5). As original texts sometimes contain two annotations from two different users for the same document, we also considered this fact when performing the dataset split by including the documents with two annotations in the development and testing set to make the evaluation of future systems more fair. The detailed statistics of the final *AKCES-GEC* dataset are presented in Table 3.12.

To evaluate models on the new *AKCES-GEC* dataset, we opted to use the M^2-scorer as it requires only model outputs, reference M2 file, and is actively used in many other datasets (see Section 2.1.1).

		Train				Dev				Test			
		Doc	Sent	Word	Error r.	Doc	Sent	Word	Error r.	Doc	Sent	Word	Error r.
Foreign.	Slavic	1 816	27 242	289 439	22.2 %	70	1 161	14 243	21.8 %	69	1 255	14 984	18.8 %
	Other					45	804	8 331	23.8 %	45	879	9 624	20.5 %
Romani		1 937	14 968	157 342	20.4 %	80	520	5 481	21.0 %	74	542	5 831	17.8 %
Total		3 753	42 210	446 781	21.5 %	195	2 485	28 055	22.2 %	188	2 676	30 439	19.1 %

Table 3.12: Statistics of the AKCES-GEC dataset – number of documents, sentences, words and error rates.

We made the AKCES-GEC dataset publicly available at `http://hdl.handle.net/11234/1-3057`.

Developing Strong Baselines

Having created a new Czech GEC dataset, we trained models similarly to German and Russian (see Section 3.4.2). First, we generated additional synthetic data using the process described in Section 3.4.1 with additional character-level operation for simulating errors in diacritics. We generated 10M sentences from clean WMT News Crawl monolingual training data [Bojar et al., 2017].

Similarly to the previous experiments on English, German and Russian, we used the two-stage training scheme. First, we used the synthetic data to pretrain the model. Then, we finetuned it with a mixture of synthetic and authentic training data from the AKCES-GEC dataset. We present the results in Table 3.13 and compare our model to previous state-of-the-art model Korektor [Richter et al., 2012]. Despite that Korektor can make only local changes and cannot for example insert a new word, it reaches surprisingly solid results. Nevertheless, both the pretrained-only model (*Our work – pretrained*) and the finetuned model (*Our work – finetuned*) outperform it by a large margin in both precision, recall and combined $F_{0.5}$ score. For the finetuned model, we also provide results on individual user domains of the test set: second learners with Slavic background (*Foreigners – Slavic*), second learners with other than Slavic background (*Foreigners – Other*) and Romani students (*Romani*). The best results are achieved on Romani subset, followed by the Slavic group of second learners and worst results are observed on the rest second learners. We hypothesize that this effect is caused by the fact that Romani as an ethnolect group with Czech as their mother tongue do produce texts with errors that are easiest to correct, and similarly people with Slavic background do learn Czech faster than when coming from different background. As can be seen in Table 3.12, this fact is also supported by individual error rates of the specific groups: on the development set, the error rate of Romani is 21.0%, Foreigners–Slavic 21.8% and the Foreigners–Other 23.8%.

System	Test Subset	P	R	$F_{0.5}$
Richter et al. [2012]	All	68.72	36.75	58.54
Our work – pretrained	All	80.32	39.55	66.59
Our work – finetuned	Foreigners – Slavic	84.34	71.55	81.43
	Foreigners – Other	81.03	62.36	76.45
	Romani	86.61	71.13	83.00
	All	83.75	68.48	80.17

Table 3.13: Results on AKCES-GEC Test Set (Czech).

We would like to make a small proposal for future experimenting with Czech. As we have already described, Flachs et al. [2021a] have later shown that mixing ASpell and Unimorph proposals transitively results in better model results. For Czech, other source of morphologically relevant words exists – MorfFlex [Hajič et al., 2020] contains data larger by more than two order of magnitude (26 511 962 unique forms vs 134 527 for Unimoph). Moreover, there exists the DeriNet project [Vidra et al., 2021] that for circa million lemmas also contains their derivations, and can provide even more useful noisy examples such as *Praha*^{noun} → *pražský*^{adjective} or *dům*$^{a\ house}$ → *domek*$^{a\ small\ house}$. An example for English would be *dog* → *doggie*.

Main outcomes and conclusions

1) We have shown that utilizing synthetic data for low-resource languages provides a great performance boost. We used it to pretrain the model based on the Transformer architecture, and managed to achieve new state-the-art results on German, Russian and Czech.
2) We compiled the first dataset for Czech GEC called *AKCES-GEC* from the available learner corpora, and made it publicly available.

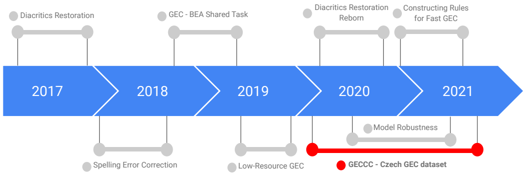
3.4.4 Remark on Future Development

Since 2019, when we proposed our state-of-the-art models, several studies have been conducted for the low-resource languages [Takahashi et al., 2020, Yamashita et al., 2020, Flachs et al., 2021b, Rothe et al., 2021]. Although these comprised interesting ideas, Rothe et al. [2021] were the only work that actually outperformed our system by using an enormously large multilingual mT5 model [Xue et al., 2021] in the *xxl* setting with 13B parameters. Apart to the standard mT5 pretraining, Rothe et al. [2021] also additionally pretrained their model on a synthetic corpus created by corrupting text by a manually designed set of rules such as dropping a span of tokens or token swap. Having the model pretrained, they finally finetuned it on the authentic data. Apart for the *xxl* configuration, Rothe et al. [2021] also reported results using more decent *base* configuration with 580M parameters, which, however, did not reach state-of-the-art results.

3.5 GECCC – New Czech Multi-Domain GEC Dataset

The full paper is available in Chapter 5, and it is currently in print and available on arXiv as Náplava et al. [2022].

Previously (see Section 3.4.3), we have compiled AKCES-GEC, the first Czech dataset for grammatical error correction. To create it, we utilized available annotated learner corpora comprising second learner and Romani user domains. For practical reasons, we decided to use the M2 format and the M^2-scorer to evaluate the best models. This resulted in a dataset of decent size – circa twice as big as German FALKO-MERLIN [Boyd, 2018] and circa four times bigger than Russian RULEC-GEC [Rozovskaya and Roth, 2019]. Later, we found four issues with the new dataset:


Diacritics Restoration
GEC - BEA Shared Task
Diacritics Restoration Reborn
Constructing Rules for Fast GEC
2017
2018
2019
2020
2021
Model Robustness
Spelling Error Correction
Low-Resource GEC
GECCC - Czech GEC dataset

1. **User Domains** – AKCES-GEC dataset contains texts written by Czech second learners and Romani. It thus completely lacks texts from native speakers. This is a big concern as we naturally believe that our correcting tools will be also used by native Czech speakers.

2. **Annotation Style** – We found that the edits proposed by annotators when annotating the learner corpora are rather decent and local. This issue is most prominent in cases when either two sentences should be merged into one or a larger edit would be needed to improve overall fluency, but annotators fixed only the basic spelling, and did not propose a larger rewrite.

3. **Tokenization** – The learner corpora annotations are provided on tokenized texts. This leads to losing information about possible bad spacing around punctuation (mostly commas).

4. **Scorer** – To evaluate model performance on AKCES-GEC dataset, we have decided to use M^2-scorer with $\beta = 0.5$. Although that we argued that it is a common metric used by majority of existing GEC datasets, it should be validated whether the selected metric correlates well with human judgements on the corpus.

Given the described issues, we decided to create a new corpus that we named **Grammar Error Correction Corpus for Czech (GECCC)** comprising also texts written by Czech natives, annotated as original detokenized texts, containing data with proper fluency rewrites and the metric selected as the one that correlates the best with human judgements.

Objectives of the work

1) Create a new dataset for GEC in Czech that would cover a variety of user domains including annotated texts written by Czech natives.
2) The metric selected for evaluating model performance on the dataset should be the metric with the highest correlation with human judgements.
3) Apart for the dataset itself, models should be trained and evaluated to set baseline performance.

3.5.1 Historical Window

The narrow range of user domains that the *AKCES-GEC* dataset may suffer from, has actually been a common thing in GEC for a long time. As can be seen from Table 2.1, up to 2017, the existing GEC datasets consisted of only a single user domain which was in all but a single case second learners. This even includes data in the popular CoNLL 2014 Shared Task in which both the training and testing texts were written solely by English second learners. The situation turned better with the BEA 2019 Shared Task on GEC. It's official dataset *W&I+L* covered both texts written by second learners and native speakers. Several month later, the Russian *RULEC-GEC* dataset with the same user domains was introduced. Later, the Spanish *COWS-L2H* [Yamada et al., 2020], the Ukrainian *UA-GEC* [Syvokon and Nahorna, 2021] and the English *GMEG* [Napoles et al., 2019] GEC datasets covering both the second learners and native speakers were introduced.

Despite being only a test set of decent size, the work of Napoles et al. [2019] on the *GMEG* dataset was probably the closest to what we planned to achieve. *GMEG* test set contains 6k English sentences written by second learners and natives that were further divided into formal texts with original texts downloaded from Wikipedia and informal texts with original texts from Yahoo web posts. Each text was corrected by 4 annotators. Except for the dataset itself, Napoles et al. [2019] also analysed the correlation between human judgements and automatic metrics and ultimately proposed a new metric being a linear combination of existing metrics which was shown to correlate with human judgements the best.

As we have described in Section 2.1.1, there are several automatic metrics for evaluating GEC systems. Probably the most popular metric has been the M^2-scorer. Although it has initially been used with $\beta = 1.0$, i.e. weighting precision equally to recall, authors of the CoNLL 14 Shared Task decided to change the β to 0.5 to emphasize precision over recall. A year after the end of the Shared Task, Grundkiewicz et al. [2015] asked annotators to rank the outputs of systems submitted to the Shared Task on the Shared Tasks's test data and used these annotations to assess the correlation between human judgements and automatic metrics. They found that although the M^2-scorer with $\beta = 0.5$ correlates with human judgements better than the M^2-scorer with $\beta = 1.0$, it is even better to use $\beta = 0.18$ on the CoNLLL 14 test set. They also found that I-measure has very weak negative correlation and BLEU negatively correlates with the human judgements. Later, Napoles et al. [2019] have shown that $\beta = 0.5$ correlates better than $\beta = 0.2$ on the *FCE* dataset (theirs second learner domain), but that

$\beta = 0.2$ correlates much better than $\beta = 0.5$ on Yahoo web posts and Wikipedia. Napoles et al. [2019] further formulated a hypothesis that larger $\beta = 0.5$ correlates better on higher error density domains and vice versa. Despite all these works, we are not aware of any dataset that would use M^2-scorer with β different from 0.5.

3.5.2 Creating the Dataset

Having the need for better annotations and new domains, it was obvious that completely new annotations are needed. As such, the high-level dataset creation had the following stages.

1. Selecting **user domains** that should be contained within a dataset
2. Acquiring **source texts** with errors for each selected user domain
3. Defining **annotation rules** and **annotation scheme**
4. Data **preprocessing**
5. **Annotation** of the texts
6. **Annotation analysis** such as computing agreements between annotators
7. **Finalising** dataset and making it publicly available

Selecting User Domains

As already described, one of the most prominent issues with the AKCES-GEC dataset is that it does not represent Czech native speakers. While thinking about it, we identified two subdomains of Czech native writing that in our opinion differed in their writing that much that each of them should be represented as a separate dataset domain. The first of them is the *native formal domain*, which represents longer pieces of texts that are of high quality such as essays or articles. The second of them is then *native informal domain*, which represents shorter pieces of texts of lower quality such as social media posts or text messages.

Regarding other user domains, similarly to AKCES-GEC, we decided to include also the second learner and the Romani user domain in the dataset. These cover texts written in a different user style to texts from natives and are definitely an important part of the dataset as we suppose that the developed correcting tools will also aim at these two user groups.

To sum up, 4 user domains were identified and will be represented in the dataset: *Natives-Formal* and *Natives-Informal*, *Second Learners* and *Romani*.

Selecting Source Texts with Errors

Although that we decided not to use the old annotations from learner corpora as we did when creating AKCES-GEC, we could use the original noisy texts as input texts for annotators. Apart from the released Czesl-man and unreleased parts of Czesl-man and ROMi corpus that we used for compiling AKCES-GEC, we also identified and used original Czech texts from multilingual learner corpus MERLIN [Boyd et al., 2014]. These corpora provided us with enough data representing the second learners and the Romani domain.

Regarding the Czech natives, we decided to use original texts from SKRIPT 2012, which was compiled in the AKCES project and released shortly before we started working on the *GECCC* dataset. These texts were written by Czech pupils of Czech elementary schools and thus represent a formal domain of texts with decent quality. To create texts for informal domain, we decided to use website discussion from Czech Facebook Dataset [Habernal et al., 2013] and also discussions from Czech news site `novinky.cz` that were provided us by Seznam.cz a.s.[4] As the informal texts come solely from web domains, we decided to rename the user domain from *Natives-Informal* to *Natives Web Informal*.

Annotation Scheme

We formulated the annotation rules as following: The corrected text must not contain any spelling or grammatical errors and should sound fluent. The text semantics must be preserved. Whenever multiple corrections exist, the correction that affects the least tokens should be preferred. Fluency edits are allowed if the original text is incoherent. Note that annotators can join or split sentences and even paragraphs if needed.

The entire document was given as a context for the annotation. Annotators were further instructed to remove too incomprehensible documents or those that contain private information, which was an issue mostly for Czech discussions.

Regarding the annotation tool, we considered two options: using simple text editor such as Notepad[5] or using more complex annotation tools such as Teitok [Janssen, 2016] that is being used for annotating learner corpora. The great advantage of the first approach is the simplicity of the annotation process as annotators are only correcting texts by simple rewriting. To propose a word correction using the second approach, annotators typically need to mark a word span containing an error and fill in the correction in a new window. On the other hand, when using simple text editor, annotators cannot mark and classify individual edits. After considerations, we chose the first simple approach for following reasons: (1) we believed that the tool simplicity will not discourage annotators from making fluency edits, (2) using the simple tool will allow us to collect more annotations quickly, (3) there are tools such as ERRANT that can infer the edits automatically. Naturally, using automatic tools such as ERRANT for inferring edits provides only suboptimal results when compared to manual annotation and also the error types that can be automatically classified are typically more high-level, nevertheless, it is quite common decision also in other languages [Bryant et al., 2019, Boyd, 2018, Cotet et al., 2020b].

Data Preprocessing and Annotation

Having acquired the input data, we selected a subset of documents for annotation. Naturally, not all documents from each domain could be annotated for annotation

[4] `https://seznam.cz`

[5] `https://notepad.org/`

budget reasons as for example there are 10 000 discussion posts in the Facebook dataset and much more also in discussion posts from Novinky.cz. For each of 4 user domains, we sampled a set of documents so that each domain has roughly the same number of sentences. Moreover, we utilized available additional metadata such as user proficiency level in MERLIN to support the split balance. Table 3.14 shows the original number of documents in each data source and also the number of documents that we took for annotation. Note that all documents from the development and testing sets of AKCES-GEC dataset are selected for annotation.

Dataset	Documents	Selected
AKCES-GEC-test	188	188
AKCES-GEC-dev	195	195
MERLIN	441	385
Novinky.cz	—	2 695
Facebook	10 000	3 850
SKRIPT 2012	394	167
ROMi	1 529	218

Table 3.14: Data resources for the new Czech GEC corpus.

Our main objective was to perform data split so that the development and testing data contain enough documents to maintain representativeness, coverage and are also backwards compatible to AKCES-GEC.[6] Similarly to selecting documents for annotations, we split the selected documents so that (i) test and development data contain roughly the same amount of annotated data from all domains, (ii) original AKCES-GEC dataset splits remain unchanged, (iii) available additional detailed annotations such as user proficiency level in MERLIN are leveraged to support the split balance. Regarding the training data, we decided to use new annotations for the Natives Web Informal domain as there are no such previously annotated data and also as a small part in the Second Learners domain. For other domains, we collected existing annotations from SKRIPT 2012; the MERLIN corpus and the AKCES-GEC and thus cover the Natives Formal, the Romani and partially also the Second Learners domain.

Because certain data sources such as AKCES-GEC provide only tokenized data which comes from the fact that their underlying texts come from tokenized second learner corpora, and we wanted to operate over detokenized data, we had to detokenize all data from such sources. To detokenize data, we used the Moses detokenizer[7] with slight changes. We publish the corpus in two variants: tokenized and detokenized.

Having selected, preprocessed and split the data, we hired 5 annotators to annotate texts. All test data were annotated by two annotators to increase chances of covering multiple alternatives for sentences with multiple appropriate corrections. The half of development data was annotated twice and the second half of the development set as well as training data then received one annotation.

[6] The development data of AKCES-GEC are fully contained within the development set of the new dataset and this holds similarly also for the testing sets.

[7] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/detokenizer.perl

Once the data were annotated, we used ERRANT to enhance the raw document-aligned data with edits. As ERRANT was originally developed for English, we had to make several changes. Similarly to Boyd [2018] who adapted ERRANT for German, we adjusted the original English error types to the Czech language and present all error types together with examples in Table 3.15 and their detailed description below:

- *POS* – the part-of-speech-tag of an original word and a corrected word are the same. It is additionally subtyped by *:INFL* in cases where the original and corrected words share a common lemma. The *POS* error types are based on the UD POS tags [Nivre et al., 2020]
- *MORPH* – original and corrected words share a common lemma, but have a different POS
- *ORTH* – we distinguish between errors in casing (*ORTH:CASING*) and errors differing in spacing (*ORTH:WSPACE*).
- *SPELL* – incorrect word spelling leading to original word being a non-word. We use the word list from MorfFlex [Hajič et al., 2020] to detect non-words.
- *WO* – words are in a different order. It is subtyped by *:SPELL* when there are also spelling errors.
- *QUOTATION* – wrongly used quotation marks.
- *DIACR* – original and corrected words differ in diacritical marks
- *OTHER* – all errors not covered by any other rule. It often comprises edits that have a different number of words in the original and corrected text.

Error Type	Subtype	Example
POS (15)		*tažené* → *řízené*
	:INFL	*manželka* → *manželkou*
MORPH		*maj* → *mají*
ORTH	:CASING	*usa* → *USA*
	:WSPACE	*přes to* → *přesto*
SPELL		*ochtnat* → *ochutnat*
WO		*plná jsou* → *jsou plná*
	:SPELL	*blískají zeleně* → *zeleně blýskají*
QUOTATION		” → „
DIACR		*tiskarna* → *tiskárna*
OTHER		*sem* → *jsem ho*

Table 3.15: Czech ERRANT Error Types.

Having classified individual edits, we measured proportions of individual error types in the dataset. Specifically, we sorted error types according to their occurrence count in each user domain and display the top 10 most occurring in Figure 3.3. It is evident that each domain has another distribution of error types, however, errors in punctuation (*PUNCT*) are the most common error in 3 domains. Errors in either missing or wrongly used diacritics (*DIACR*), spelling errors (*SPELL*) and errors in orthography (*ORTH*) are also common, with varying frequency across domains.

As errors in punctuation (*PUNCT*) are the most common error, we further inspected them in more detail and found out that 9% (*Natives Formal*) - 27% (*Natives Web Informal*) are caused by either missing or wrongly used punctuation at the end of the sentence (linguistically uninteresting as simple rules could be possibly written to fix them) and the rest (75-91%) appears in a sentence, most of which (35% [*Natives Web Informal*] - 68% [*Natives Formal*]) is either a missing or a redundant comma. Errors that require splitting a sentence into two or joining two sentences are also quite common as they occur in 5-7%.

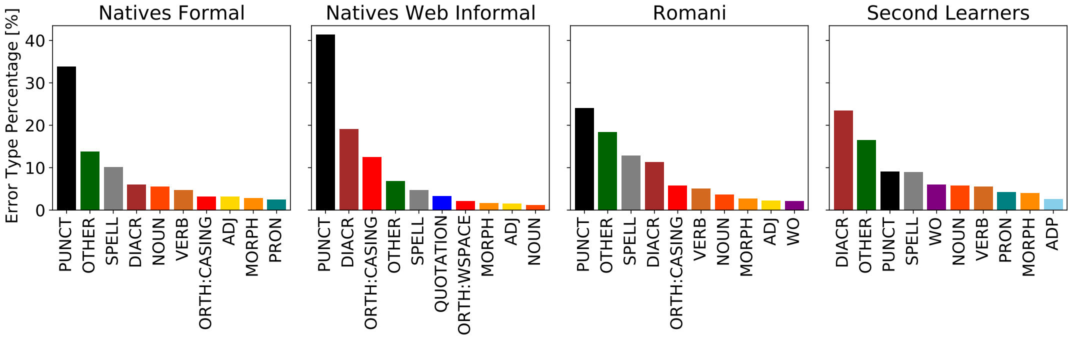


Figure 3.3: Distribution of top-10 ERRANT error types per domain in the development set of the GECCC dataset.

Annotation Analysis

As multiple corrections often exist for a text, the inter-annotator agreement cannot be computed by simply comparing equality on two references of the testing set. Instead, a different approach is used commonly in GEC [Rozovskaya and Roth, 2010, 2019, Syvokon and Nahorna, 2021]: random sentences corrected by one annotator are shown to a different annotator who should judge the need for a correction, i.e. whether he would make any modifications if he saw such data in the original first annotation round. We followed this approach and employed three annotators from the first annotation round to judge the sentence correctness in the second pass. In the second pass, each of the three annotators annotated a disjoint set of 120 sentences. Once the second pass was annotated, we computed the inter-annotator agreement scores as the percentage of sentences judged correct and provide these scores in Table 3.16. To put the scores into context, we computed the mean and standard deviation: 82.96 ± 12.12 and compared them to numbers reported on English (63 ± 18.46 [Rozovskaya and Roth, 2010]), Russian (80 ± 16.26 [Rozovskaya and Roth, 2019]) and Ukrainian (69.5 ± 7.78 [Syvokon and Nahorna, 2021]). As our numbers seem similar or even slightly better than others, we considered our first round annotations strong.

First → Second ↓	A1	A2	A3	A4	A5
A1	—	93.39	97.96	89.63	72.50
A2	84.43	—	95.91	90.18	78.15
A3	68.80	87.68	—	79.39	57.50

Table 3.16: Inter-annotator agreement based on second-pass judgements for GECCC dataset: numbers represent percentage of sentences judged correct in second-pass proofreading. Five annotators annotated the first pass, three annotators judged the sentence correctness in the second pass.

Dataset Publishing

Traditionally, models for GEC have been operating over a single sentence. However, by the beginning of 2021 when we were finishing the work on the dataset, preliminary approaches on document-level GEC were published [Chollampatt et al., 2019b, Yuan and Bryant, 2021]. The models were shown to benefit from larger context as certain errors such as errors in tense choice or errors in articles often require large context. To ease the future work with our dataset, we decided to release it on three alignment levels: (1) traditional sentence-level, (2) paragraph-level and (3) document-level.

Apart for the three alignment levels, we further decided to publish the dataset in two formats: the traditional tokenized M2 format and the detokenized format. While the tokenized M2 is the de-facto standard for distributing GEC datasets and in comparison to detokenized format also contains the error type for each correction edit, it does in certain cases lose information about errors in original spacing. Moreover, despite that GEC models are typically trained in a tokenized mode for evaluation reasons, one would typically need to have them in a detokenized mode when used in practice. For these reasons, we release also the dataset in the detokenized format which retains full information about the original spacing and allows for training detokenized models.

We present the statistics of the GECCC dataset in Table 3.17. GECCC dataset contains more than 83k sentences in total out of which about 8k newly annotated sentences are in the development and testing sets. It is evident that apart for edit types, individual user domains also differ in error rates: while in the *Natives Formal* domain the average number of erroneous tokens is about 6%, circa every fourth token contains error in the *Romani* and the *Second Learners* domain.

GECCC dataset contains more than 83k sentences total that makes it the largest among GEC datasets other than English. It is surpassed in size only by the English Lang-8 and AESW datasets (see Table 2.1).

3.5.3 GEC Models

We further evaluated six existing systems on the *GECCC* dataset and also trained one new model:

- **Office 365 Word** – We used the *Spelling & Grammar* module of Office 365 Word. For each proposed correction, we applied the first proposal variant.

	Sentence-aligned #sentences			Paragraph-aligned #paragraphs			Doc-aligned #docs			Error Rate
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test	
Natives Formal	4 060	1 952	1 684	1 618	859	669	227	87	76	5.81%
Natives Web Informal	6 977	2 465	2 166	3 622	1 294	1 256	3 619	1 291	1 256	15.61%
Romani	24 824	1 254	1 260	9 723	574	561	3 247	173	169	26.21%
Second Learners	30 812	2 807	2 797	8 781	865	756	2 050	167	170	25.16%
Total	66 673	8 478	7 907	23 744	3 592	3 242	9 143	1 718	1 671	18.19%

Table 3.17: GECCC dataset statistics at three alignment levels: sentence-aligned, paragraph-aligned and doc-aligned. Average Error rate was computed on the concatenation of development and test data in all three alignment levels.

We ran the correction process until no other corrections were proposed. As this requires manual user interaction, we did not correct the whole dataset, but only 400 randomly sampled sentences from each domain. The texts were corrected on November 8, 2021 using build *16.0.14704.41015*.

- **Google Docs** – We used the *Spelling and grammar* module of Google Docs. For each proposed correction, we applied the first proposal variant. We ran the correction process until no other corrections were proposed. As this requires manual user interaction, we did not correct the whole dataset, but only 400 randomly sampled sentences from each domain (same sentences as for *Office 365 Word*). The texts were corrected on November 8, 2021.

- **Korektor** – a pre-neural state-of-the-art statistical spellchecker and (occasional) grammar checker (see Section 2.3 and Section 2.5 for more details on the system)

- **Synthetic trained** – Transformer-based model pretrained on synthetic data and described in Section 3.4.3. We previously referred to this model as *Our work – pretrained*.

- **AKCES-GEC finetuned** – Transformer-based model pretrained on synthetic data and finetuned on authentic train data of *AKCES-GEC*. We described the system in Section 3.4.3. We previously referred to this model as *Our work – finetuned*.

- **GECCC finetuned** – newly trained model. Similarly to *AKCES-GEC finetuned*, we pretrained the model on synthetic data, but instead of finetuning it on *AKCES-GEC*, we finetuned it on the new train set of the *GECCC* dataset.

- **Joint GEC+NMT** – Transformer-based model trained in multitask setting with two objectives: GEC in English and Czech and translation between Czech and English. The training data for GEC were created by the KaziText tool (see Section 3.8.1 for more information on KaziText), which is our statistical tool that tries to mimic human errors by estimating their characteristics from training data. The training data for translation are from the parallel CzEng corpus [Kocmi et al., 2020]. We trained this model with initial hypothesis that the combined objective might provide some benefits (e.g. model may learn paraphrases better), but its final performance did not meet our expectations, and we used it only for this analysis.

The left part of Table 3.18 summarizes the evaluation of the seven grammar error correction systems, evaluated with the highest-correlating and widely used metric, the M^2 scorer with $\beta = 0.5$. For the meta-evaluation of GEC metrics against human judgements, see the following Section 3.5.4.

	$M^2_{0.5}$-score					Mean human score				
System	NF	NWI	R	SL	Σ	NF	NWI	R	SL	Σ
Original	—	—	—	—	—	8.47	7.99	7.76	7.18	7.61
*Office 365 Word**	50.57	51.20	48.76	54.40	51.21	–	–	–	–	–
*Google Docs**	38.17	28.10	48.48	47.74	43.37	–	–	–	–	–
Korektor	28.99	31.51	46.77	55.93	45.09	8.26	7.60	7.90	7.55	7.63
Synthetic trained	46.83	38.63	46.36	62.20	53.07	8.55	7.99	8.10	7.88	7.98
AG finetuned	65.77	55.20	69.71	71.41	68.08	8.97	8.22	**8.91**	8.35	8.38
GECCC finetuned	**72.50**	**71.09**	**72.23**	**73.21**	**72.96**	**9.19**	**8.72**	**8.91**	**8.67**	**8.74**
Joint GEC+NMT	68.14	66.64	65.21	70.43	67.40	9.06	8.37	8.69	8.19	8.35
Reference	—	—	—	—	—	9.58	9.48	9.60	9.63	9.57

Table 3.18: Mean score of human judgements and $M^2_{0.5}$ score for each GEC system in domains (NF = *Natives Formal*, NWI = *Natives Web Informal*, R = *Romani*, SL = *Second Learners*, Σ = whole dataset). Note that models denoted by asterisk (*Office 365 Word* and *Google Docs*) were evaluated on a reduced test set.

The first three systems in Table 3.18 (*Office 365 Word*, *Google Docs* and *Korektor*) are the commercial baselines being actively used in practice. Out of them, *Office 365 Word* works surprisingly solid while outperforming all other commercial baselines on all domains as well as on the whole dataset. While the performance of *Google Docs* and *Korektor* seems to deteriorate badly on the *Natives Formal* and *Natives Web Informal* domains, *Office 365 Word* provides stable results across all domains.

Despite the solid performance of *Office 365 Word*, all our neural-based models outperformed it on the whole dataset. When the authentic training data are used for training, the models (*AG finetuned*, *GECCC finetuned* and *Joint GEC+NMT*) surpass *Office 365 Word* in all domains by a large margin. The incorporation of authentic data clearly improves results as can be observed on the performance gap between *Synthetic trained* and the other neural models. Moreover, utilizing larger and domain-richer training data from new *GECCCC* dataset as opposed to *AKCES-GEC* training set leads to significant improvements (see the difference between *AG finetuned* and *GECCC finetuned* that differ only in the finetuning data). Finally, using the authentic data directly seems superior to using them indirectly for more accurate synthetic data creation (see the slightly worse results of *Joint GEC+NMT*).

The best model *GECCC finetuned* significantly[8] outperformed all other systems in all domains as well as on the whole dataset. Being the best model, we further analysed its performance with respect to individual error types. For simpler analysis, we grouped all POS-related errors into two error types: *POS* and

[8] With p-value < 0.001, using the Monte Carlo permutation test with 10M samples and probability of error at most 10^{-6} [Fay and Follmann, 2002, Gandy, 2009]

POS:INFL for words which are erroneous only in inflection and share the same lemma with their correction.

As we can see in Table 3.19, the model is very good at correcting local errors in diacritics (*DIACR*), quotation (*QUOTATION*), spelling (*SPELL*) and casing (*ORTH:CASING*). Unsurprisingly, small changes are easier than longer edits: similarly, the system is better in inflection corrections (*POS:INFL*, words with the same lemma) than on *POS* (correction involves finding a word with a different lemma).

Should the word be split or joined with an adjacent word, the model does so with a relatively high success rate (*ORTH:WSPACE*). The model is also able to correctly reorder words (*WO*), but here its recall is rather low. The model performs the worst on errors categorized as *OTHER*, which includes edits that often require rewriting larger pieces of text. Generally, the model has higher precision than recall, which suits the needs of standard GEC, where proposing a bad correction for a good text is worse than being inert to an existing error.

Error Type	#	P	R	$F_{0.5}$
DIACR	3 617	86.84	88.77	87.22
MORPH	610	73.58	55.91	69.20
ORTH:CASING	1 058	81.60	55.15	74.46
ORTH:WSPACE	385	64.44	74.36	66.21
OTHER	3 719	23.59	20.04	22.78
POS	2 735	56.50	22.12	43.10
POS:INFL	1 276	74.47	48.22	67.16
PUNCT	4 709	71.42	61.17	69.10
QUOTATION	223	89.44	61.06	81.83
SPELL	1 816	77.27	75.76	76.96
WO	662	60.00	29.89	49.94

Table 3.19: Analysis of *GECCC finetuned* model performance on individual error types. For this analysis, all POS-error types were merged into a single error type POS.

3.5.4 Meta-evaluation of Metrics

Having created the dataset, we were further interested in what metric has the highest correlation with human judgements and thus should be used for evaluating system performance. As we have described in Section 2.1.1, there are several metrics that can be used and some of them such as M^2 and ERRANT must be further specified by the exact value of β parameter, which defines the ratio of precision and recall.

Human Judgements Annotation

In order to evaluate correlations of several GEC metrics with human judgements, we first needed to collect human judgements of texts corrected by a range of GEC systems. For ranking the corrections, we used the suitable hybrid *partial ranking with scalars* [Sakaguchi and Van Durme, 2018] approach in which the annotators judge the original erroneous sentence, the manually corrected gold references and automatic corrections made by GEC systems on a scale 0–10 (from ungrammatical to correct). We employed three annotators who judged the sentences with respect to the context of the document. As the commercial systems (*Office 365 Word* and *Google Docs*) were not included in the original analysis, but were added later for the completeness of this dissertation thesis itself, they were not annotated in this annotation round. Note that we decided to use the partial ranking with scalars approach, as it was previously shown [Sakaguchi and Van Durme, 2018, Novikova et al., 2018, Napoles et al., 2019] to be more reliable than other approaches such as direct assessment framework used by the Workshop (Conference) on Machine Translation [Bojar et al., 2016] and earlier GEC approaches [Grundkiewicz et al., 2015, Napoles et al., 2015].

We selected a representative subset of the *GECCC* dataset for annotation. It contained circa 1 100 documents with about 4 300 original sentences and about 15 500 unique corrected variants and gold references of the sentences. To make the inter-annotator agreement analysis possible, the annotators annotated 127 documents jointly and the rest were annotated by a single annotator.

Once the annotation process was over, we computed the agreement in human judgements on the set of jointly annotated documents. We performed the analysis on individual domains and also on the whole dataset on the *system*-level: the annotator's judgements for each system were averaged over individual sentences and the correlation was computed for each pair of the three annotators. We report the common *Pearson correlation* and *Spearman's rank correlation coefficient* in Table 3.20.

Domain	r	ρ
Natives Formal	92.01	92.52
Natives Web Inf.	95.33	91.80
Romani	88.73	85.90
Second Learners	96.50	97.23
Whole Dataset	96.11	95.54

Table 3.20: Inter-annotator agreement between the human judgements on the system-level: Pearson (r) and Spearman (ρ) mean correlation between 3 human judgements of 5 sentence versions.

Metrics Correlations with Judgements

For the metric correlation analysis, we selected the following most common GEC metrics: M^2, ERRANT, GLEU and I-Measure. Moreover, for the M^2 and ERRANT, we varied the value of β parameter, i.e. the proportion of precision and recall.

For a given domain and metric, we computed the correlation between the automatic metric evaluations of the five GEC systems on one side and the human judgements on the other side. Similarly to the analysis of inter-annotator agreement in human judgements, we performed the metric correlation analysis on the system level, i.e. the human score for a GEC system is an average of all human scores of its sentences in a domain (or whole dataset respectively). In order to obtain smoother estimates and also to estimate standard deviations, we employed the bootstrap resampling method, with 100 samples. We report the final *Pearson correlation* in Table 3.21 and further visualize correlations for different values of β for M^2 and ERRANT in more detail in Figure 3.4.

Metric	Pearsons' r
GLEU	97.37 ± 1.52
I-measure	95.37 ± 2.16
$M^2_{0.2}$	96.25 ± 1.71
$M^2_{0.5}$	98.28 ± 1.03
$M^2_{1.0}$	95.62 ± 1.81
$\text{ERRANT}_{0.2}$	94.66 ± 2.44
$\text{ERRANT}_{0.5}$	98.28 ± 1.04
$\text{ERRANT}_{1.0}$	95.70 ± 1.80

Table 3.21: System-level Pearson correlation r between the automatic metric scores and human annotations.

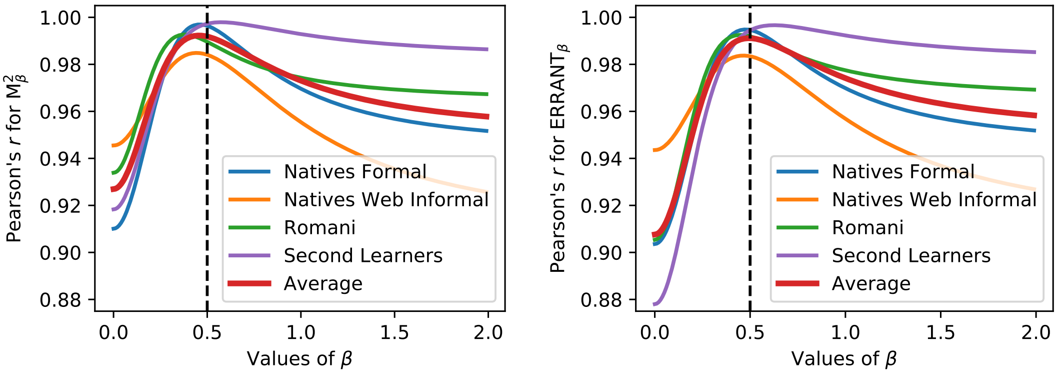


Figure 3.4: **Left:** System-level Pearson correlation coefficient r between human annotation and M^2_β-scorer for various values of β. **Right:** The same correlation for ERRANT_β.

Table 3.21 reveals that although the GLEU and I-measure correlate with human judgements quite well, the metrics that correlates best are $M^2_{0.5}$ and $\text{ERRANT}_{0.5}$. Figure 3.4 further reveals that for the M^2-scorer, slightly smaller $\beta = 0.48$ correlates even better. Interestingly, as can be seen in Figure 3.4, the value of best correlating β remains roughly the same across all domains for both the M^2-scorer and ERRANT.

Out of the two best correlating metrics, we decided to opt for the M^2-scorer mainly because it is language agnostic and does not need an additional POS tagger, lemmatizer, morphological dictionary and language specific rules as ERRANT does. We also decided to use the value of $\beta = 0.5$ as the difference in the correlation to the optimal $\beta = 0.48$ is only marginal and $\beta = 0.5$ is a standard common choice in other datasets.

3.5.5 Human Evaluations of GEC Systems

We previously reported results of the selected GEC systems obtained using the automatic $\mathrm{M}^2_{\beta=0.5}$-scorer in Section 3.5.3 and presented them in the left part of Table 3.18. Having the human scores for the subset of sentences annotated for the metric correlation in Section 3.5.4, we further averaged the sentence scores to obtain human scores for the GEC systems in individual domains and present them in the right part of Table 3.18. Recall that since we did not include *Office 365 Word* and *Google Docs* in the human score annotation, they do not have the human scores and whenever we further refer to *all systems*, they are excluded.

As we can see in Table 3.18, the GEC systems are ordered in the same way using both the automatic metric and the human scores when measured over the entire dataset. When inspecting the ordering in specific domains, minor differences can be observed such as between *Akces-GEC finetuned* and *GECCC finetuned* in the *Romani* domain.

Compared to the M^2-scorer, we can use the human scores to reveal whether using a GEC system for a particular domain provides better results than the simple ”do nothing” baseline. As we can see, measured over the entire dataset, all system have higher human score than the ”do nothing” baseline (row *Original*). This implies that having documents mixed from all domains, using any of the GEC systems would result in a better text. However, when we focus on individual domains, we can see that *Korektor* scores are worse than the ”do nothing” baseline on two domains: *Natives Formal* and *Natives Web Informal.* When inspecting the issue more thoroughly, we identified that this is mainly due to named entities that occur frequently in two domains and which upon an eager change disturb the meaning of a sentence, leading to severe penalization by human annotators.

The human judgements also confirmed that there was still a large gap between the optimal *Reference* score and the best performing models. Regarding the domains, the neural models in the finetuned mode that had access to data from all domains seemed to improve the results consistently across each domain. However, given the fact that the source sentences in the *Second Learners* domain received the worst scores by human annotators, this domain seems to hold the greatest potential for future improvements.

Finally, we used the human scores to illustrate capabilities and also differences between three systems: *Korektor*, *Synthetic trained* and *GECCC finetuned.* While *Korektor* represents the statistical approach with relatively straightforward suggester, two other systems represent the newer neural based approach capable of performing any operations. Although there are cases in which *Korektor* corrects sentences better than even *GECCC finetuned*, we picked the examples following the order of human evaluations over whole systems and show them in Table 3.22.

As can be seen in Table 3.22, *Korektor* replaces original non-words by existing words in text. Although the proposed variants are often correct in the sense that they replace out-of-vocabulary words for their correct counterparts, there are cases in which the replacements are wrong given the context such as replacing *kamarádmi* by *kamarádi* in the second example. Also, *Korektor* sometimes fails to correct an existing but, given the context, wrongly used word – keeping *hraji* in the third example or *sem* in the fourth example. The *Synthetic trained* model seems to disambiguate between these cases better and does also insert new words (*a* in the fifth example), delete redundant words (*že* in the first example) and also reorders words (*můžu si* → *si můžu* in the second example). The main benefit of using the best *GECCC finetuned* seems to be in a better correction of complicated cases such as by proposing more appropriate preposition (*na* instead of *v* in the first example) or inserting words for better fluency (*si* in the fourth example).

System	Correction	Human Score
Original	To je, myslím že, nejkrásnější město ve célem světě.	7
Korektor	To je, myslím že, nejkrásnější město v celém světě.	8
Synthetic trained	To je, myslím, nejkrásnější město v celém světě.	9
GECCC finetuned	To je, myslím, nejkrásnější město na celém světě.	10
Original	Chtelá bych také vedet jestli můžu si u Vas objednat jídlo.	6
Korektor	Chtěla bych také vedet jestli můžu si u Vas objednat jídlo.	7
Synthetic trained	Chtěla bych také vědět jestli si můžu u vás objednat jídlo.	9
GECCC finetuned	Chtěla bych také vědět, jestli si u vás můžu objednat jídlo.	10
Original	Nejdůležitým je, co má clověk na duši, jaké vstahy má s lidmi (kamarádmi, kolegami atd), co dělá pro ně.	5
Korektor	Nedůležitým je, co má člověk na duši, jaké vztahy má s lidmi (kamarádi, kolega atd), co dělá pro ně.	6
Synthetic trained	Nejdůležitější je, co má člověk na duši, jaké vztahy má s lidmi (kamarády, kolegy atd), co dělá pro ně.	7
GECCC finetuned	Nejdůležitější je, co má člověk na duši, jaké vztahy má s lidmi (kamarády, kolegy atd.), co pro ně dělá.	8
Original	Tatinek a syn hraji v moří.	6
Korektor	Tatínek a syn hraji v moři.	7
Synthetic trained	Tatínek a syn hrají v moři.	8
GECCC finetuned	Tatínek a syn si hrají v moři.	10
Original	Máma vaří uklízí, chodí do práce pere pro nás , stará se o nás.	5
Korektor	Máma vaří uklízí, chodí do práce pere pro nás, stará se o nás.	7
Synthetic trained	Máma vaří a uklízí, chodí do práce a pere pro nás, stará se o nás.	8
GECCC finetuned	Máma vaří, uklízí, chodí do práce, pere pro nás, stará se o nás.	9

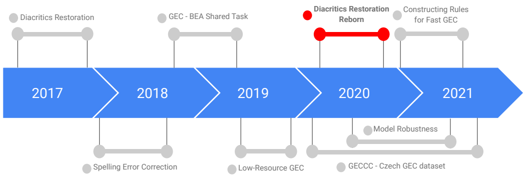
Table 3.22: Examples of corrections and the scores assigned to them by human annotators illustrating the improvements between the original sentence and three automatic systems: *Korektor*, *Synthetic trained* and *GECCC finetuned*.

Main outcomes and conclusions

1) We have created a new dataset for GEC in Czech: *Grammar Error Correction Corpus for Czech (GECCC)* with 83 058 sentences that cover four diverse domains including essays written by native students, informal website texts, essays written by Romani ethnic minority children and teenagers and essays written by non-native speakers. All domains were professionally annotated for GEC errors in a unified manner. The dataset is the largest non-English GEC dataset and is publicly available.
2) We compared several strong Czech GEC systems including commercial baselines and the newly trained system. The neural models set a strong baseline for further research.
3) We conducted a meta-evaluation of common GEC metrics across domains in our data. We concluded that M^2 and ERRANT scorers with $\beta = 0.5$ are the measures most correlating with human judgements on our dataset, and we chose the $M^2_{0.5}$ as the preferred metric for the *GECCC* dataset.

3.6 Diacritics Restoration Reborn

The full paper is available in Chapter 5, and it was published as Náplava et al. [2021].


Diacritics Restoration
GEC - BEA Shared Task
Diacritics Restoration Reborn
Constructing Rules for Fast GEC
2017
2018
2019
2020
2021
Model Robustness
Spelling Error Correction
Low-Resource GEC
GECCC - Czech GEC dataset

In the beginning of 2020, while the annotation process for the new *GECCC* dataset was running, we focused on new developments in NLP and their possible utilization for (Czech) GEC. During 2019, the BERT model [Devlin et al., 2019] revolutionized NLP, and became the backbone of plentiful state-of-the-art models [Devlin et al., 2019, Kondratyuk and Straka, 2019, Wang et al., 2019, Pires et al., 2019, Wu and Dredze, 2019]. From the architecture perspective, BERT consists of the Transformer encoder. It thus processes input subwords and using several encoder layers comprising multi-head self-attention followed by dense layers, computes for each input subword its contextualized embedding. The great advantage of BERT is that it is (unsupervisedly) pretrained on enormously large corpora, and it has been empirically shown that the model can be quickly and using a relatively small amount of supervised data finetuned for various NLP tasks with great success.

Given the success of applying BERT to various tasks, the natural question was whether and to what extent would it be possible to build a good GEC system based on BERT. The most prominent issue with the BERT model is that it is just an encoder that outputs only as many contextualized embeddings as there are subwords on the input. Building a model that for each input subword predicts the possible replacement string would lead to the model that can correct local errors, but would not be able to cover complex rewrites that change multiple nearby tokens. Nevertheless, the model would most likely be much faster than the Transformer model that we previously used for GEC (see Section 3.3.2) as it does not contain an autoregressive decoder.

While discussing whether giving up several complex error types in favour of higher speed allowed by non-autoregressive decoding and possibly also higher accuracy on perhaps more frequent local errors, we decided to first try the model on the diacritics restoration task. In the diacritics restoration task, there is a one-to-one mapping between input and target characters (subwords) and thus this task is an ideal candidate for experimenting with the BERT model. As we have already developed the multilingual dataset covering 12 languages [Náplava et al., 2018], it would be meaningful to train and evaluate the model on this dataset, and compare the new results to the previous ones.

Apart for the model results themselves, we were further interested in a detailed model analysis on Czech. Specifically, we considered three research questions:

- While the testing set of our previous multilingual dataset consists of only clean Wikipedia sentences, it is a well known fact that the (deep neural) models may deteriorate substantially when the input domain is changed [Belinkov and Bisk, 2017, Rychalska et al., 2019]. We were therefore interested, how well would the model perform on user domains different from Wikipedia.

- Our multilingual dataset was created by acquiring Wikipedia articles and by stripping the diacritics from them. This naturally implies that for each undiacritized text, there is a single gold reference text. It is, however, possible that the diacritization stripping process introduces ambiguity, i.e. that for an undiacritized text, multiple texts with diacritics are correct and plausible. Moreover, even the original texts on Wikipedia may had contained wrong diacritics. Given these two issues, we formulated the second question, which was to explore the model errors in detail and classify them to four groups: (1) real errors, (2) plausible variants, (3) mistakes in the annotation, (4) mistakes in both the annotation and prediction.

- Finally, we were interested, whether there is an observable characteristic in the real model errors that will be identified by the previously described analysis.

Objectives of the work

1) Implement and train a model for diacritics restoration based on BERT, and evaluate it on our multilingual dataset comprising 12 languages.
2) Test model robustness on domains other than Wikipedia.
3) Analyse model errors on the Czech language in detail. Specifically, categorize them based on whether they are real errors or plausible variants. Further, inspect the real errors thoroughly.

3.6.1 Model

Vaswani et al. [2017] proposed to finetune BERT for token classification tasks by adding a single dense layer that is applied on all outputs. When compared to the recurrent model that we used previously for the diacritics restoration (see Section 3.1.1 and Figure 3.2), the models differ in two main aspects: (1) bidirectional RNN got replaced by BERT model, and (2) input tokenization changed from individual characters to subwords.

Using BERT model instead of RNN layers provides multiple benefits. First, several pretrained checkpoints exist for BERT, and these make the finetuning for various tasks easier. We remark that before BERT, RNNs were pretrained on large unlabelled data to produce contextualized word embeddings [Peters et al., 2018], however, the pre-trained BERT model reaches considerably better results. Second, the self-attention mechanism used in BERT layers allows capturing long range dependencies easier. And third, the BERT model does not employ recurrent layers and thus can be parallelized better.

Changing input tokenization from characters to subwords provides a good trade-off between the size of the input vocabulary and the actual size of the input passed to the model. However, when used naively for diacritics restoration, it brought a potential issue with the size of the output vocabulary. If we used the naive approach that would for each input subword predict its replacement as its diacritized variant, the size of the output vocabulary may happen to be quite large as it contains all the subwords used in the language. Having the large output vocabulary could make the training difficult. Therefore, we decided to

input	instruction	result	note
dite	1:CARON;3:ACUTE	dítě	optimal instruction
dite	1:CARON	díte	
dite	3:ACUTE	ditě	
dite	<KEEP>	dite	no change
dite	2:RING ABOVE	dite	impossible instruction ignored

Figure 3.5: Diacritization instructions examples for input "dite (dítě)" with 4 characters, indexed from 0 to 3. Index-instruction tuples generate diacritics for the given input.

instead predict instructions on how to restore diacritics. Specifically, one such instruction consists of index-diacritical mark tuples that define on what index of the input subword a particular diacritical mark should be added. It is evident that the size of the final output vocabulary is in this case at most the size of the direct diacritized variant approach, but usually much smaller as for certain different subwords and their different diacritized variants, the same instruction exist. A simple example are two undiacritized subwords *dam* and *sam* with correct diacritized variants *dám* and *sám* covered by a single instruction *1:CARON*.

An example of a diacritization instructions set can be seen in Figure 3.5. Given an input subword *dite (dítě)*, with four characters indexed from 0 to 3, the appropriate diacritization instruction is *1:ACUTE;3:CARON*, in which acute is to be added to *i* and caron is to be added to *e*, resulting in a properly diacritized word *dítě*. Obviously, the network can choose to leave the (sub)word unchanged, for which a special instruction *<KEEP>* is reserved. Should the network accidentally select an impossible instruction, no operation is carried out and the input (sub)word is left unchanged.

To construct the set of possible diacritization instructions, we tokenize the undiacritized text of the particular training set, and align each input token to the corresponding token in the diacritized text variant. The diacritical mark in each instruction is obtained from the Unicode name of the diacritized character (see Section 3.1. We keep only those instructions that occurred at least twice in a training set to filter out extremely rare instructions that originate for example from foreign words or wrong spelling.

To assess the extent to which the usage of instructions reduced the size of output vocabulary when compared to using direct diacritized variants, we constructed vocabularies for both cases. For each language, we used the training data from the multilingual dataset for diacritics restoration. To filter out extremely rare cases, we again removed all diacritized variants and instructions that occurred only once. As can be Seen in Table 3.23, the instruction approach shrinks the size of the output vocabulary by an order of magnitude.

As we already stated, the great advantage of BERT is that it comes with a set of pretrained models. We use the well-known *bert-base-multilingual-uncased* model[9]. It was trained on 102 languages, and all 12 languages that we use are among them. The input texts are tokenized using WordPiece algorithm, and the vocabulary consists of 110 000 subwords.

[9] https://github.com/google-research/bert

Language	Direct Variants Set Size	Instructions Set Size
Czech	38 635	1 005
Vietnamese	23 106	2 018
Latvian	30 026	720
Polish	20 034	1 005
Slovak	37 851	785
French	23 577	681
Irish	6 846	189
Spanish	26 233	492
Croatian	16 180	541
Hungarian	35 124	767
Turkish	17 565	1 005
Romanian	33 349	1 677

Table 3.23: Comparison of final output set size for using the direct variants and intructions options.

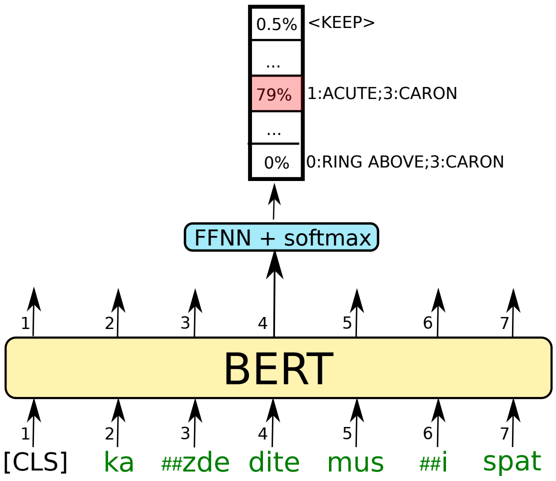


Figure 3.6: Model architecture. Text without diacritics, tokenized into subwords, is fed to BERT, and for each of its outputs, a fully-connected layer followed by softmax is applied to obtain the most probable instruction for diacritization. The ##-prefixes of some subwords are added by the BERT tokenizer.

We present the final model architecture in Figure 3.6. The input texts are tokenized into subwords, these are then processed by BERT, and for each input subword, a distribution over a set of instructions is obtained by applying a dense layer followed by a softmax. We select the instruction with maximum probability.

3.6.2 Evaluation on the Multilingual Dataset

We trained a separate BERT model on each of the 12 languages present in our previously created multilingual dataset (see Section 3.1.3). We present the models' alpha-word accuracy including 95% confidential intervals computed using the bootstrap resampling method in Table 3.24. For each language, we further show the size of the instruction set, and we also include results of our previous RNN model and the former state-of-the-art combination of RNN model and an external language model (*RNN+LM*) for comparison. Finally, we report the error reduction of our BERT model with respect to *RNN+LM*.

On 9 out of 12 languages, our approach significantly outperforms previous

Language	Instruction Set Size	RNN [Náplava et al., 2018]	RNN+LM [Náplava et al., 2018]	Ours	Error Reduction
Czech	1005	98.71	99.06	**99.22 ±0.046**	17 %
Vietnamese	2018	97.55	97.73	**98.53 ±0.037**	35 %
Latvian	720	96.57	97.49	**98.63 ±0.045**	45 %
Polish	1005	99.03	99.55	**99.66 ±0.041**	24 %
Slovak	785	98.84	99.09	**99.32 ±0.030**	25 %
French	681	99.54	**99.71**	**99.71 ±0.016**	0 %
Irish	189	98.46	98.71	**98.88 ±0.040**	13 %
Spanish	492	98.46	**99.65**	99.62 ±0.018	− 9 %
Croatian	541	99.51	99.67	**99.73 ±0.018**	18 %
Hungarian	767	99.02	99.29	**99.41 ±0.038**	17 %
Turkish	1005	99.18	**99.28**	98.95 ±0.046	− 46 %
Romanian	1677	97.99	98.37	**98.64 ±0.056**	17 %

Table 3.24: Comparison of alpha-word accuracy of our model including 95% confidential intervals to previous state-of-the-art on 12 languages.

state-of-the-art *RNN+LM*. The most significant improvements are achieved on Vietnamese and Latvian, for which the error gets reduced by more than 30%. On 11 out of 12 languages, our approach significantly outperforms *RNN* approach that does not utilize an external language model. Although we tried to inspect, why BERT performs substantially worse on Turkish, we did not find any obvious reason.

3.6.3 Model Domain Robustness

Having trained the new state-of-the-art model for Czech, we were further interested, how well such model performs on various text domains. As we previously created the new Czech dataset for grammatical error correction *GECCC* that comprises four different user domains, we decided to test the model robustness on these data.

The input texts in the *GECCC* dataset comprise all types of different errors. To create target data for our experiment, we applied all correcting edits that fix errors in diacritics and casing.[10] We left other errors intact, but did not evaluate on words that contain these errors, because they are not directly relevant to diacritics, and in many cases, the errors are so severe that evaluation would be controversial. Although the severely perturbed words are omitted from evaluation, they still remain in the sentence context, and may still confuse the diacritization system, making the task potentially more difficult.

We evaluated our model on all domains of the *GECCC* dataset, and present the results in Table 3.25 in the column *Original*. For comparison, we also report the results previously obtained on the multilingual dataset (row *Wiki*). Surprisingly, the model performance remains roughly stable on all domains despite their noisiness. On each domain, the results stay above the 99% alpha-word accuracy indicating strong performance. We hypothesise that one of the facts behind the strong results is the model itself, and especially that it operates over subwords. These might help the model to recover in cases with smaller typos. Further, we also hypothesise that although the writers produced quite noisy texts, they at the same time avoided foreign words that are generally harder to correctly diacritize.

[10] We corrected casing due to the manual error analysis performed further.

Domain	Original	Annotated	Annotated w/o annotated typos
Wiki	99.22	99.49	99.66
Natives Formal	99.50	99.75	99.75
Natives Web Informal	99.12	99.53	99.62
Romani	99.11	99.46	99.54
Second Learners	99.18	99.73	99.79

Table 3.25: Alpha-word accuracy of Czech model on 5 datasets from various domains.

3.6.4 Detailed Error Analysis on Czech

Having trained the new state-of-the-art model for Czech, we were further interested in the remaining potential on Czech diacritics restoration, i.e. in a detailed error classification. Recall that we identified two main subtasks:

1. Classify the reported errors into 4 categories: (1) system correct, gold correct (*plausible variant*), (2) system correct, gold wrong (*system corrects a data error*), (3) system wrong, gold wrong (*uncorrected error in data*) and (4) system wrong, gold correct (*real errors*).

2. Obtain better insight into real errors.

We employed annotators to classify each reported model error. For each model error, annotators classified two annotation items: whether the model prediction is correct, and whether the gold variant is correct. For each variant, they responded to three questions: (1) Is the word correct given a context of the current sentence? (2) Is the word still correct if the context is extended by two previous and two following sentences? (3) Does the word contain a spelling error? The motivation behind the first two questions was that certain ambiguities may be resolved on a document context, but not on a sentence context. A simple example of this phenomenon is the following sentence: *K nejvýznamnějším patří zmiňované vily/víly.* The third question was added to clean the data as they sometimes contained uncorrected spelling errors.

Annotators classified 4 702 mispredicted words from the Wikipedia domain from the multilingual dataset (see Section 3.6.2) and also from the four domains of the *GECCC* dataset (see Section 3.6.3).

The basic analysis of the annotated system errors revealed that out of 4 702 wrongly diacritized words in the all our data, 960 of the mispredicted words contain a spelling error, and we do not consider them further. The remaining 3 742 mispredicted words were categorized as follows:

1. System correct, gold correct: 19% (694 of 3 742) (*plausible variants*)

2. System correct, gold wrong: 25% (964 of 3 742) (*system corrects a data error*)

3. System wrong, gold wrong: 1% (31 of 3 742) (*uncorrected error in data*)

4. System wrong, gold correct: 55% (2 084 of 3 742) (*real errors*)

Interestingly, the annotations revealed that about 44% of originally reported errors are not errors at all. In 694 cases (19%) both the system word and the gold word are correct, which is justified by the plausible variants. In 964 cases (25%) the original gold annotation was wrong whereas the system annotation was correct, which means that the system effectively corrected some errors in the original data. We used these observations to refine results previously reported in Table 3.25 and extended them by two types of results: *Annotated*, which uses the new annotations to change the false negative data to correct in the evaluation, and *Annotated w/o annotated typos*, which also removes words with reported spelling errors from the evaluation. As can be seen, using the new annotations we report 35% to 67% error reduction. Moreover, when the spelling errors are further excluded from evaluation, the error gets additionally reduced by up to 33%.

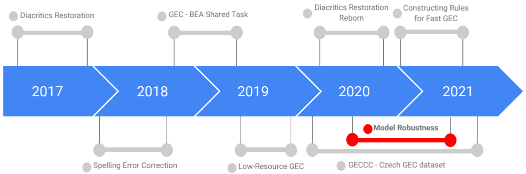
The annotations also revealed 2 084 real system errors. In these errors, annotations confirmed the existence of an interesting discourse phenomenon, in which a word is correctly diacritized given the current sentence, however it is incorrectly diacritized given a larger context. As these cases constituted only 50 out of 2 084 real system errors, improving diacritization model to process larger contexts effectively to correct them promises only marginal improvements. To further inspect the real errors in more detail, we employed morphological annotations. Specifically, we analysed real errors by means of the Universal POS tags and Universal features [Nivre et al., 2020]. For better insight into obtained characteristics, we also analysed morphological features of plausible variants, and found out that the morphological characteristics of the real errors and plausible variants bear similarities. Unlike plausible variants, the real errors differ more often also in their lemma.

Main outcomes and conclusions

1) We implemented a model for diacritics restoration based on BERT. When evaluated on the multilingual dataset with 12 languages, it outperformed previous state-of-the-art combination of recurrent neural model combined with an external language model on 9 languages, and the sole recurrent neural model without an external language model on 11 languages significantly.
2) The model exhibited stable performance even if we tested it on other more noisy domains.
3) We annotated all reported mispredictions on Czech, and found out that 44% of the model errors were not errors, but either plausible variants or errors in the original data. Finally, we analysed the real errors using morphological annotations.

3.7 Rules and Model For Fast Grammatical Error Correction

The full paper is available in Chapter 5, and it was published as Straka et al. [2021].


Diacritics Restoration
GEC - BEA Shared Task
Diacritics Restoration Reborn
Constructing Rules for Fast GEC
2017
2018
2019
2020
2021
Model Robustness
Spelling Error Correction
Low-Resource GEC
GECCC - Czech GEC dataset

As we were training and evaluating models for diacritics restoration based on BERT, Omelianchuk et al. [2020] proposed a model named *GECTOR* that employed BERT for English GEC. They reported new state-of-the-art results as well as speed-ups up to 10 times compared to the previous state-of-the-art seq2seq models based on Transformer. Omelianchuk et al. [2020] in their approach first construct a dictionary comprising word-level rules that are of two types: *basic* (*KEEP*, *DELETE*, 1 167 token-dependent *APPEND*s and 3 802 *REPLACE*s) and *g-transformations* (custom-made rules for e.g. casing change, singular to plural noun change or verb tense change). The constructed rules are then used to create training data for the model so that exactly one tag is predicted for every word. The model itself is based on BERT, upon which a linear layer followed by the softmax activation is stacked to predict the most probable instruction. As the rules are constructed on the word level, outputs are considered only for the first subword of each word. Finally, authors employ several other tricks to achieve state-of-the-art results as for example a detection layer to be able to tweak precision in favour of recall, three-stage training and iterative decoding.

The set of rules constructed by Omelianchuk et al. [2020] was efficient for English corpora, which rarely contain spelling errors, and for English language, which does not have diacritization marks, and its morphology is very modest compared to morphologically rich languages such as Czech or Russian. Using a set of word-level transformations designed for English, all character-level corrections would have to be handled by the generic word-for-word *REPLACE* rule, leading to an explosion of rules. As our primary focus was on developing models for Czech, we were naturally interested in a set of rules of modest size and good coverage for a broad variety of Czech errors, including even complicated categories such as spelling errors and errors in diacritics. As opposed to Omelianchuk et al. [2020], who manually designed their *g-transformations* to boost the coverage of errors, we were rather interested in an approach for constructing the rules without manual intervention, which could be easily applied also to other languages. Naturally, having good error coverage does not necessary imply good model performance, therefore, we were also interested in the results of a GEC model trained using the best rule set.

Objectives of the work

1) Propose and evaluate different approaches to creating rules for GEC model based on BERT.
2) Evaluate the best rule set by training the BERT GEC model.

3.7.1 Constructing Rules

Operating on the word-level rules allowed Omelianchuk et al. [2020] to define specific *g-transformations* for changing the word grammatical properties such as the tense in case of a verb or number in case of nouns. On the other hand, as we already described, the word-level approach poses issues with representing errors for example in spelling, as naively representing every possible misspelling with one *REPLACE* rule would result in a dictionary of enormous size. Given that we encode an input sentence using subwords for BERT, it also seemed natural to create rules already on the subword level. As it is definitely more probable to share rules on the subword level than on the word level, it is likely that the final dictionary will be smaller for the subword level approach.

Recall that Omelianchuk et al. [2020] proposed two types of word-level rules: *basic* and *g-transformations*. As *g-transformations* are manually created specifically for English, we decided to keep only the *basic* rules comprising *KEEP*, *DELETE*, token dependent *APPEND*s and *REPLACE*s operations for further experiments and refer them as *string* later. Although there are now two modifying operations (*REPLACE* and *APPEND*), the situation still resembles the previously discussed issue in diacritics restoration (see Section 3.6.1), where we proposed to use character-level instructions instead of direct diacritized variants. These turned out being often shared between individual subwords, and utilizing them reduced the dictionary size significantly. Similarly, in the GEC scenario, we considered using character-level instructions instead of the word-level *basic* transformations. This time, we decided to base them on minimal *edit scripts*, i.e. the sequence of character-level *inserts*, *replaces* and *deletes* that convert the original subword to a correct one. To further reduce the dictionary size, we decided to index each edit operation either from the beginning of the input subword (if it involves the first half of it) or from the end of it (otherwise).

Overall, we identified two dimensions of the rule construction: the unit on which the rules are applied (*word* vs *subword*) and the granularity of the transformations (*string* or *char*). These give rise to a set of 4 possible combinations:

- character transformations applied on each subword separately (*char-at-subword*),
- character transformations applied on each complete word (*char-at-word*),
- string transformations applied on each subword (*string-at-subword*),
- string transformations applied on each complete word (*string-at-word*).

In such terminology, the transformations proposed by Omelianchuk et al. [2020] can be referred to as *string-at-word*, and our previous diacritical transformations as *char-at-word*. An example of the described transformation types is illustrated in Figure 3.7.

To assess the effect of the number and the type of transformations, we computed the potential maximum $F_{0.5}$ score with the M^2-scorer. We evaluated it for 4 languages: Czech, German, Russian and English by first generating transformation dictionary from a mixture of authentic training data and a small portion of synthetic data (see Section 3.4.1), and then measuring the coverage on the particular language test set (*AKCES-GEC* for Czech[11], *FALKO-MERLIN GEC* for German, *RULEC-GEC* for Russian and *CoNLL 2014 test set* for English). As we were computing the potential maximum $F_{0.5}$ score, we allowed to use any transformation from the generated set of rules for each gold correction which results

[11] Note that by the time of performing the experiments the *GECCC* dataset was not yet compiled.

Input	gatherin		leafes	
Correct	**G**atherin**g**		lea**v**es	
Subwords	␣gathe	rin	␣lea	fes
string-at-word	␣Gathering		␣leaves	
string-at-subword	␣Gathe	APPEND g	KEEP	ves
char-at-word	APPEND g, UPPERCASE 2nd		REPLACE 3rd *from end* *with* v	
char-at-subword	UPPERCASE 2nd	APPEND g	KEEP	REPLACE 1st *with* v

Figure 3.7: Example of the four types of transformations that we proposed and evaluated for BERT-based GEC.

in the desired target. Whenever a particular error was unrepresentable using the generated dictionary, the original input was copied. For each transformation type and language, we generated three dictionaries differing in the minimal count the transformation had to appear in training data to be kept: once, twice and three times. Moreover, we experimented with both the *cased* and *uncased* versions of the multilingual BERT. We visualize the results of the experiment in Figure 3.8.

From Figure 3.8 it was evident that across all four measured languages, the character-level transformations applied at subwords (*char-at-subword*, green) had the highest potential in terms of upper-bound $F_{0.5}$ score. At the same time, word-level replacements (*string-at-word*, red) did not scale well. This effect was particularly apparent for languages other than English, for which the upper-bound string replacement $\mathrm{F}_{0.5}$ (*string-at-word*, red) falls below the current GEC systems state-of-the-art $\mathrm{F}_{0.5}$. This means that even if the model was trained to predict everything perfectly, it would still not reach state of the art. On the other hand, the baseline scores of all transformations sets provided solid results for English, which confirmed that the case is not that difficult for English.

3.7.2 Model

We trained the GEC model using the *char-at-subword* transformations, which achieved the best upper-bound score. We used the same model as for the diacritics restoration, i.e. input subwords are passed into the pretrained BERT, BERT creates for each input subword a contextualized embedding vector, and the embedding vectors are fed into a simple softmax classifier that projects embedding vectors into a distribution over a set of transformations from which the instruction with the highest probability is selected. We trained the model for Czech, German and Russian, in two stages: first, models were pretrained on a large synthetic corpus, and then finetuned on a mixture of synthetic and authentic data.

Similarly to our experiments on the BEA 2019 Shared Task on Grammatical Error Correction, we applied two additional tricks: *iterative decoding*, which reruns the model iteratively on the corrected output, and *class weighting*, which assigns different weight to the *KEEP* class and other classes in the training objec-

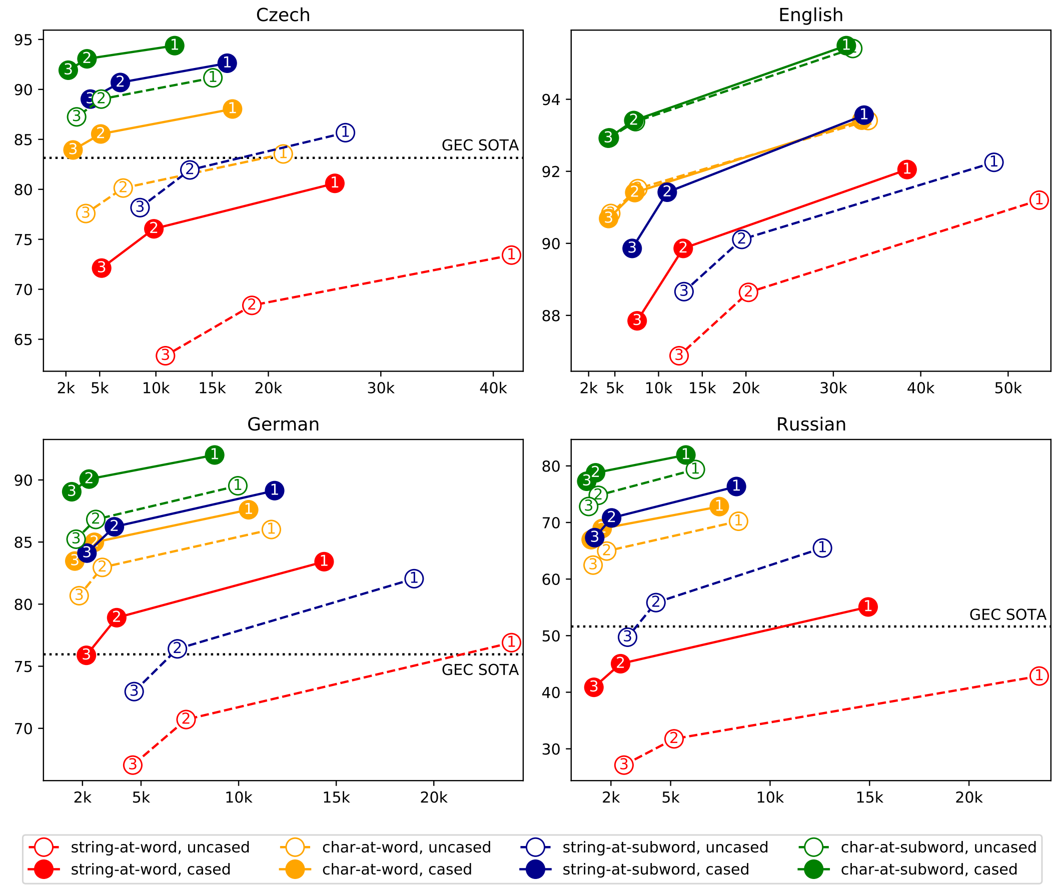


Figure 3.8: $F_{0.5}$ depending on number and type of transformations, if all transformations were correctly predicted (upper bound). Top and left is better (higher $F_{0.5}$, fewer rules), bottom and right is worse (lower $F_{0.5}$, more rules). Circled numbers ①, ② and ③ denote that we kept transformations present at least once, twice or three times in the training data, respectively (larger means less transformations).

tive. We present the results on Czech, German and Russian in Table 3.26 for the three versions of the model: (1) model that was only pretrained on synthetic data (*Ours pretrained*), (2) model that was also finetuned on the mixture of authentic and synthetic data (*Ours finetuned*) and (3) the same model when applied iteratively (*Ours finetuned iterative*). We compare our results to the results that we previously achieved with the seq2seq Transformer model (see Section 3.4.2 and Section 3.4.3), to the models of Rothe et al. [2021] and to Korektor [Richter et al., 2012] on Czech, Boyd [2018] on German and Rozovskaya and Roth [2019] on Russian.

Table 3.26 shows that our prototype models exhibit solid results, which are however inferior to their former seq2seq counterparts. We hypothesized that similarly to Omelianchuk et al. [2020], more training tricks such as the two-stage finetuning or an additional error detection branch would be needed to make our prototype model more competitive. In the end, we had to perform several tweaks previously also in the case of the seq2seq Transformer model.

Following Omelianchuk et al. [2020], who reported speed-ups of the model up to 10 times when compared to the autoregressive baselines, we also compared

Model	Params	$F_{0.5}$		
		Czech	German	Russian
Richter et al. [2012]		58.54	–	–
Boyd [2018]		–	45.22	–
Rozovskaya and Roth [2019]		–	–	21.00
Náplava and Straka [2019a]^{synt}	210M	66.59	51.41	40.96
Náplava and Straka [2019a]^{fine}	210M	80.17	73.71	50.20
Rothe et al. [2021] *base*	580M	71.88	69.21	26.24
Rothe et al. [2021] *xxl*	13B	83.15	75.96	51.62
Ours pretrained	172M	64.29	44.29	25.36
Ours finetuned	172M	72.86	62.92	36.62
Ours finetuned iterative	172M	75.06	65.95	38.68

Table 3.26: Comparison of our GEC models based on BERT evaluated on Czech (*AKCES-GEC*), German (*FALKO-MERLIN GEC*) and Russian (*RULEC-GEC*). For each system, we report the $F_{0.5}$ score on the particular test set.

the runtime performance of our BERT base model to the previous seq2seq model based on Transformer architecture, and found out that our model based on BERT is circa four times faster as can be seen in Table 3.27.

Model	Time Per Sentence	
	CPU	GPU
Seq2Seq Transformer	162.34	22.36
BERT-GEC	41.26	5.09

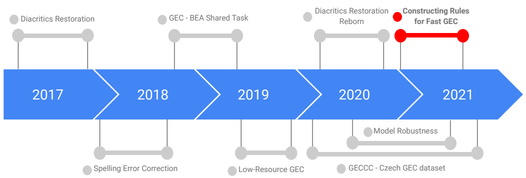
Table 3.27: Average time in milliseconds required to process a single sentence in the Czech test set, measured using both (a) CPU decoding (32-core Intel Xeon) and (b) GPU decoding (Nvidia Quadro P5000).

Main outcomes and conclusions

1) We compared the character transformations to previously used word-level transformation instructions in a BERT GEC model, and have shown that character-based rules have better coverage and scale better in Czech, German and Russian.
2) We also trained character-based GEC tagging models for these languages with promising results.

3.8 Understanding Model Robustness to User Generated Noisy Data

The full paper is available in Chapter 5, and it was published as Náplava et al. [2021].


Diacritics Restoration
GEC - BEA Shared Task
Diacritics Restoration Reborn
Constructing Rules for Fast GEC
2017
2018
2019
2020
2021
Model Robustness
Spelling Error Correction
Low-Resource GEC
GECCC - Czech GEC dataset

The last topic we have worked on slightly differs from the previously discussed topics as it does not directly aim at developing systems or datasets for natural language correcting systems, but rather tests state-of-the-art systems under noisy data. It is a well known fact that when models are run with data containing natural noise such as spelling errors, their performance deteriorates [Belinkov and Bisk, 2017, Heigold et al., 2018, Glockner et al., 2018, Ribeiro et al., 2018, Rychalska et al., 2019]. This is naturally an important concern for practical applications as system results are often reported on clean test sets, but they are often significantly worse when deployed and tested with real user texts, which often contain natural errors.

Although several studies were conducted on assessing performance drop of models under noisy data, the studies were mostly limited either to a single system (e.g. machine translation), a single user domain (e.g. errors from second learners), a single language (e.g. English) or the errors used to test models did not resemble the real human errors (e.g. the probability of introducing spelling error was chosen arbitrarily).

Motivated by these issues, we decided to create a framework that would model real human errors realistically, and use the system to test the robustness of multiple state-of-the-art systems for various NLP tasks in several languages. Furthermore, if our evaluation confirms the performance drop in model performance in noisy settings, we wanted to propose and evaluate strategies to mitigate the performance drop.

Objectives of the work

1) We want to implement a system capable of introducing errors that would resemble real human errors. The framework should work in multiple languages, and be capable of modelling errors of a particular user group.
2) The proposed system should be used to test performance drop of the current state-of-the-art systems.
3) If the performance drop is substantial, strategies to mitigate it should be proposed and evaluated.

3.8.1 KaziText

Robustness of NLP models to natural noise would ideally be evaluated on texts with authentic noise, with error corrections annotated by humans. This perfect-world setting, however, requires an immense annotation effort, as multiple target domains have to be covered by well-educated human annotators for multiple NLP tasks in a range of languages. To ease the annotation burden, we proposed a framework, which we named *KaziText*, for introducing errors likely produced by a human in a text.

To make the introduced errors and their distribution similar to human errors, we decided to estimate them from existing GEC corpora. We considered basic edits as present in the typical M2 format, and represented them as rewrites *corrected fragment* → *original fragment*. For each such pair, the probability of replacing the corrected fragment with the original fragment in a text would be proportional to how often the *corrected fragment* got actually replaced from *original fragment* in the corpus. The main issue of this simple approach lies in the fact that it can only introduce previously seen patterns, and does not generalize to other unseen errors. Ideally, when seeing for example that a lower-cased word was all upper-cased, one would want to obtain a generic rule applicable to all such words and not only to one specific. To accomplish that, apart for this simple error rewrite category (*Common Other* aspect), we defined a set of additional 7 error categories, in our work called *aspects*, that model the most typical error types, and their internal probabilities are estimated from GEC corpora:

1. **Diacritics** Strip diacritics either from a whole sentence or randomly from individual characters.
2. **Casing** Change casing of a word, distinguishing between changing the first letter and other ones.
3. **Spelling** Insert, remove, replace or swap individual characters (for example *wrong* → *worng*) or use ASpell[12] to transform a word to other existing word (*break* → *brake*).
4. **Suffix/Prefix** Replace common suffix (*do* → *doing*) and prefix (*bid* → *forbid*).
5. **Punctuation** Insert, remove or replace punctuation.
6. **Whitespace** Remove or insert spaces in text.
7. **Word Order** Reorder several adjacent words.
8. **Common Other** Insert, replace or substitute common phrases as seen in data (*the* → *a*, *a lot of* → *many*). This is the aspect which should learn language specific rules.

The probability of each aspect is estimated separately, and in general, any subset of them can be used to noise a text. In our experiments, we either took only the first four of them in case that we were testing systems for which the number of input words should have remained unchanged, or all of them in the rest of the cases. The framework does allow to noise the text at a particular error rate of the final text by scaling up or down its internal probabilities. In this context, we refer to a *corpus error level* being the error rate corresponding to the error rate of a particular GEC corpus.

A complete set of estimated aspects' probabilities is called a *user profile*. We estimated such user profiles for 4 languages, and for each language also for multiple user domains:

- English: natives and second learners
- Czech: natives, natives web informal, second learners and Romani
- German: second learners
- Russian: second learners

[12] http://aspell.net/

3.8.2 Assessing Performance Drop in Noisy Settings

We identified 5 common NLP tasks morpho-syntactic analysis, named entity recognition (NER), neural machine translation (NMT), GLUE benchmark and reading comprehension), obtained the existing state-of-the-art systems, and tested them against natural noise in two settings: (1) with respect to amount of noise in the text, and (2) with respect to error types.

,We present a visualization of robustness of the state-of-the-art BERT model trained on 4 subtasks of the GLUE benchmark to the varying text noisiness noised using two English profiles (English natives and English second learners) in Figure 3.9. We can see that the performance decreases roughly linearly with respect to the amount of erroneous tokens for all subtasks. The performance drop is the largest for the Quora Question Pairs subtask (*QQP*), for which we can see that when the text of the questions is noised using a natives' *corpus level error*, the expected performance drop is circa 3 relative points, and when noised on the second learners' *corpus level error*, the expected performance drop is more than 8 relative points. We can also see that the performance drop is circa the same for the lines with the same colour. This is an interesting observation, as the dashed line and the dotted line show two different user domains with significantly different errors.

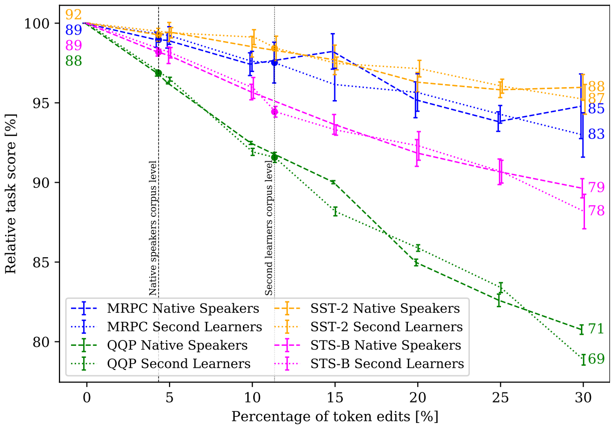


Figure 3.9: GLUE model performance deterioration with respect to the amount of token edits in its texts. We present a relative task score and each tasks' text is noised using two user profiles: English native speakers (dashed line) and English second learners (dotted line).

After running multiple experiments with multiple systems on multiple tasks and languages, we found out that the following observations applies across all the measurements: (1) **the system performance decreases approximately linearly with the amount of token edits**, and (2) **it is the sheer amount of noise rather than the distribution of aspects that contributes to the model performance deterioration**. The second observation comes from the fact that models behave similarly on texts noised with different user profiles, while their performance decreases with increasing text noisiness.

We also analysed model robustness with respect to error types. One such experiment for NER with additively stacked error types is presented in Figure 3.10. We did not find any general conclusions from these experiments that would hold for all measurements such as we found out when doing measurements with respect to the text noisiness level. Nevertheless, the experiments revealed that some tasks are more sensitive to certain error types: (1) spelling and affixes make for the major performance drop in morpho-syntactic analysis, NER and NMT, (2) casing is a crucial aspect for NER, and (3) punctuation is important for NMT and reading comprehension. Finally, some tasks, most notably lemmatization, were shown to be more sensitive to noise than others.

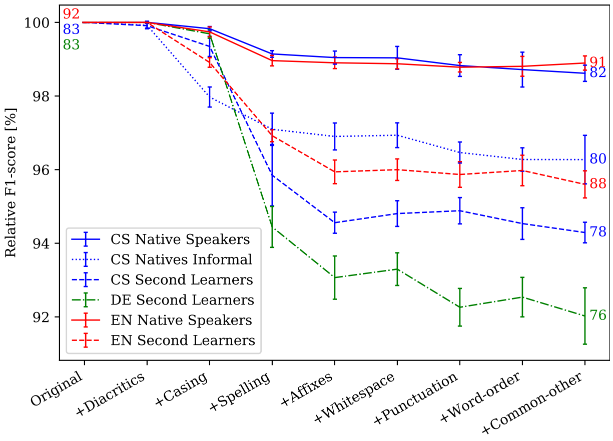


Figure 3.10: NER model performance deterioration with respect to additively stacked error types. We present a relative task score for models in 3 languages and texts noised multiple user profiles.

3.8.3 Noise Mitigating Strategies

Having observed a significant performance drop for all measured tasks, we were further interested in strategies to alleviate the model deterioration. The majority of research on improving model robustness is dedicated to training on a mixture of original and noisy data. The same procedure is usually used for generating both the test corpus and the training data [Belinkov and Bisk, 2017, Heigold et al., 2018, Ribeiro et al., 2018, Rychalska et al., 2019]. This approach is often called *adversarial training*, although the training data are typically not found using adversarial attacks [Kurakin et al., 2018], but rather using synthetic approaches or backtranslation [Sennrich et al., 2016]. We decided to examine this approach, and to generate the noisy part of the training corpus using *KaziText*.

The second approach we came up with was text preprocessing with a GEC model. In this approach, we first correct the noisy texts using a GEC model, and then input the corrected data into the tested systems. For correcting texts, we use our best models for Czech, English, Russian and German as described in Section 3.4. We further refer to this approach as the *external approach*, given that the corrections are performed externally. The first approach, in which the model is trained on a mixture of original and noisy data, is further referred to as the *internal approach* as the model must perform corrections internally.

Similarly to experiments on testing model robustness in the previous section, we performed extensive measurements by testing the models on all 5 selected tasks in multiple languages. Note that for the *internal approach*, this involved retraining noise-aware models, while the *external approach* reuses the original ones. We present one such experiment in Figure 3.11 (the complete results can be found in the full paper in Chapter 5), which visualizes experiments performed for lemmatization models in Czech and English with the original models (solid lines), *internal approach* (dotted lines) and *external approach* (dashed lines). We can see that despite that the original model performs slightly better on the original clean texts (0% token edits), from circa 3% error rate, both noise mitigating strategies start outperforming the original model. It is evident that the slope of lines depicting the original approach is significantly steeper than the slope of lines for both noise mitigating strategies. It is also evident that for lemmatization, the *external approach* works better. When the noise is introduced on the *corpus error level* (see vertical lines *EN corpus level* and *CS corpus level*), the error measured for the *external approach* is smaller by more than half when compared to the error of the original model.

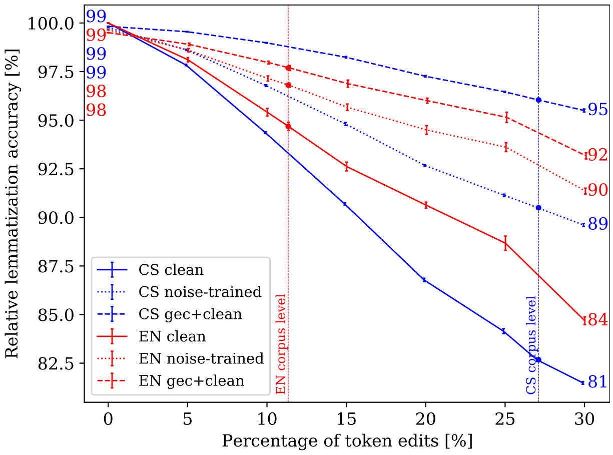


Figure 3.11: Comparison of different noise coping strategies on the lemmatization task. The solid lines depict the original model, the dotted lines represent the *internal* approach to alleviate the performance drop and the dashed lines represent the *external approach*.

After running multiple experiments with multiple systems on multiple tasks and languages, we found out two interesting observations that hold for all settings:

- Despite the original model working better on error-free texts, **from relatively low noise levels of circa 5% token edits, both the *internal* and the *external* approaches alleviating the performance drop perform better.** With increasing amount of noise in a text, the difference between the original and our two noise mitigating approaches grows. Table 2.1 and Table 3.17 illustrate that error rates of circa 5% are typical for texts written by native speakers, and that other user domains such as second learners produce texts with significantly higher amount of noisiness. This observation indicates that for any system that processes real human data, either one of the noise mitigating strategies should be considered.

- **The *external* approach works better than *internal* approach on low resource tasks** (e.g. morphosyntactic analysis and NER), for which the training data are rather scarce. On the other hand, **the *internal***

approach works better on the machine translation task, for which there is a large amount of training data and a model with greater capacity. We illustrate this observation on three tasks: morpho-syntactic analysis, NER and NMT or Czech models in Figure 3.12. Its first row presents performance of models on error-free texts, and the second row illustrates performance on texts noised using the second learners user profile at the *corpus error level*, i.e. tested on texts that have the same error rate as the typical texts produced by second learners.

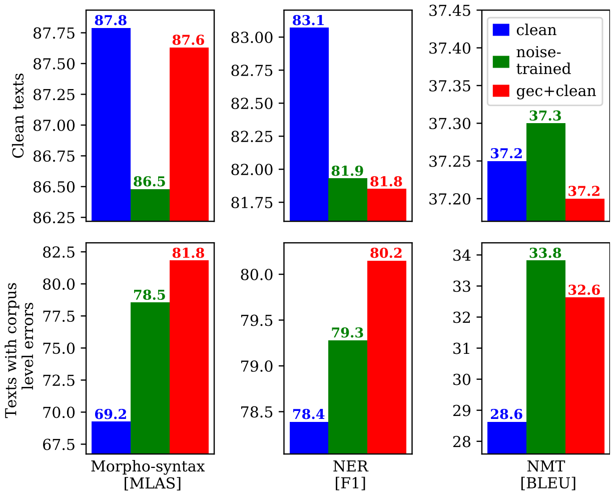


Figure 3.12: Comparison of three models – clean, noise-trained (*internal*) and GEC-preprocessed (*external*) – on three tasks in Czech second learners profile. The upper row presents the results on the original clean test data, the lower row on the test data with corpus error level noise.

Main outcomes and conclusions

1) We proposed a tool named *KaziText* for statistical modelling of natural noise that estimates the error probabilities from GEC corpora.
2) We extensively evaluated several state-of-the-art NLP downstream systems with respect to their robustness to input noise, both in increasing level of text noisiness and in variations of error types. We confirmed that the noise hurts the model performance substantially.
3) We compared two coping strategies: training with noise and preprocessing with GEC, concluding that each strategy is beneficial in different scenarios.

4. Conclusion

Over the course of my Ph.D. studies, the state of natural language correction has advanced significantly. The deep neural models have pushed the original statistical models, such as the phrase-based machine translation systems used in grammatical error correction, to the sideline, datasets that in 2017 existed only for English and only for specific user domains, such as English second learners, now exist for multiple languages and for multiple user domains, and consequently, also the correction models now work decently in multiple languages. Finally, evaluation metrics have been better analysed to find the metrics that correlate the best with human judgements. Our work has contributed to natural language correction in three major areas: (1) diacritics restoration, (2) grammatical error correction and (3) testing model robustness under noisy scenarios. In the majority of our work, we concentrated on Czech, which in our opinion lacked behind the work done in English in many aspects such as non-existence of standardized datasets or using only pre-neural models. In nearly five years of our work, the situation turned significantly better – we created multiple datasets and models, analysed evaluation metrics, and altogether brought the Czech field closer or even matched it to the research being conducted on English.

Starting with diacritics restoration, we have compiled and made freely available a dataset comprising 12 languages, for which the diacritics restoration is relevant and non-trivial task. We used the dataset to build two state-of-the-art models. The first one combines a bidirectional recurrent neural network with an external statistical n-gram language model, while the second model developed three years later is based purely on a neural network called BERT. After reaching new state-of-the-art results, we further analysed the latter model more thoroughly on Czech, and found that manual evaluation of model outputs is needed to assess the real model performance as multiple diacritization variants often exist for an undiacritized text. Our analysis also revealed that the performance of our model is surprisingly stable across texts that contain also other type of errors such as grammatical or spelling ones that come from different user domains such as native speakers or second learners.

One of the major outcomes of our work in grammatical error correction is that we have shown that using synthetic data in languages with low amount of annotated data works remarkably well. Using a relatively simple process to generate artificial data, our model based on a seq2seq Transformer outperformed the baselines in low-resource German and Russian significantly. Moreover, we outperformed previous state of the art in German and Russian even when using only synthetic data and no annotated data at all. We have also compiled a new dataset for grammatical error correction in Czech (AKCES-GEC) from available learner corpora, and have shown that the same approach with synthetic data works for Czech as well. As our proposed state-of-the-art model based on Transformer is rather slow in both training and inference, we further proposed another model that is based on BERT and a subword tagging approach. Although its performance turned out slightly worse than the Transformer-based model baseline, it is on the other hand significantly faster, and offers an interesting trade-off between speed and performance. Finally, we created a large and diverse grammatical error correction dataset for Czech (GECCC), analysed several strong Czech models, and conducted a meta-evaluation of metrics to select the metric that correlates the best with human judgements on the new dataset. The newly developed GECCC dataset comprises four types of user domains (essays from native speakers and second learners, online discussion posts written by native speakers and texts written by students from Romani ethnolect minority), and is the largest available dataset for grammatical error correction in languages other than English.

Our last contribution to the natural language correction is in the area of testing model robustness to user generated noisy texts. We created a tool named KaziText that models user errors and can introduce them in a text. We used the tool to test robustness of multiple models for multiple tasks in multiple languages, and we showed that even current state-of-the-art systems are very sensitive to noise, and their performance deteriorates roughly linearly with the amount of noise in the text. Furthermore, we evaluated two strategies to cope with the observed performance degradation, and we showed that using a grammatical error correction system before inputting the data into the model is beneficial in scenarios in which there is a low amount of annotated training data, while in scenarios with large amount of training data, it is better to re-train the model on a mixture of authentic annotated data and noisy examples. These observations are important as many even commercial tools report their performance on testing sets lacking no natural errors, and their user might be surprised that they often perform significantly worse when presented with real world human data. With our tool KaziText, the expected performance deterioration can be estimated easily for a model, and one of the two noise coping strategies can be incorporated.

As we already stated, we have witnessed great progress in developing models for natural language correction over the last five years. This can be illustrated on GEC models evaluated on the popular English ConLL14 test set. As presented in Table 2.2, the nowadays best $F_{0.5}$-score of Rothe et al. [2021] is almost twice as big as the score of Felice et al. [2014], who won the CoNLL 2014 Shared Task on GEC. Despite the great improvements, correction models are still not perfect. In our analysis described in Section 3.6.4, we have shown that even the state-of-the-art model for the simplest task of diacritics restoration still has a non-trivial error rate. In grammatical error correction, which as a task contains substantially larger range of errors than diacritics restoration and is thus a more challenging task, the gap between current models and perfect model is naturally even bigger.

One possible direction to improve current GEC models is to use byte-level models utilizing large pretrained models instead of currently used subword-level models. The byte-level models already proved efficient in the lexical normalization task [Samuel and Straka, 2021]. Furthermore, we believe that one of the future directions in GEC is also incorporation of a larger context into models. Traditionally, correction models have been operating over a single sentence. In English GEC, Chollampatt et al. [2019a] and later Yuan and Bryant [2021] have shown that taking into account larger context shows promising results. A context larger than a sentence is necessary for correcting certain error types in certain situations (such as the use of definite and indefinite articles), and it may also help the model to better understand the situation in which it operates. In Czech diacritics restoration, we have shown that only a marginal amount of errors theoretically requires larger context, however, it does not mean that it may not help models to generalize better.

Another direction of possible future research is to model multilingualism, i.e. into developing models that can correct texts in multiple languages. The multilingual models were shown to perform well in many other NLP tasks such as machine translation [Johnson et al., 2017] or question answering [Lewis et al., 2019], and also pretrained multilingual BERT models exist. One of the benefits of having multilingual models may be that patterns learnt in languages with a large amount of high-quality annotated data might help the model to work better at languages in which only a limited amount of annotated data exists.

Part II

Published Works

5. Published Works

The second part of the thesis comprises the original prints of the 7 papers that I have published in the area of natural language correction and described in detail in Section 3. I provide their list below:

- **Diacritics Restoration Using Neural Networks** [Jakub Náplava, Milan Straka, Pavel Straňák, Jan Hajič], 11th Edition of Language Resources and Evaluation Conference (LREC 2018)

- **CUNI System for the Building Educational Applications 2019 Shared Task: Grammatical Error Correction** [Jakub Náplava, Milan Straka], Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2019)

- **Grammatical Error Correction in Low-Resource Scenarios** [Jakub Náplava, Milan Straka], 5th Workshop on Noisy User-generated Text (W-NUT 2019)

- **Czech Grammar Error Correction with a Large and Diverse Corpus** [Jakub Náplava, Milan Straka, Jana Straková, Alexandr Rosen], Transactions of the Association for Computational Linguistics (TACL)

- **Diacritics Restoration using BERT with Analysis on Czech language** [Jakub Náplava, Milan Straka, Jana Straková], The Prague Bulletin of Mathematical Linguistics 116 (PBML 116)

- **Character Transformations for Non-Autoregressive GEC Tagging** [Milan Straka, Jakub Náplava, Jana Straková], Seventh Workshop on Noisy User-generated Text (W-NUT 2021)

- **Understanding Model Robustness to User-generated Noisy Texts** [Jakub Náplava, Martin Popel, Milan Straka, Jana Straková], Seventh Workshop on Noisy User-generated Text (W-NUT 2021)

Diacritics Restoration Using Neural Networks

Jakub Náplava, Milan Straka, Pavel Straňák, Jan Hajič
Institute of Formal and Applied Linguistics
Charles University, Faculty of Mathematics and Physics
Malostranské náměstí 25, Prague, Czech Republic
{naplava,straka,stranak,hajic}@ufal.mff.cuni.cz

Abstract

In this paper, we describe a novel combination of a character-level recurrent neural-network based model and a language model applied to diacritics restoration. In many cases in the past and still at present, people often replace characters with diacritics with their ASCII counterparts. Despite the fact that the resulting text is usually easy to understand for humans, it is much harder for further computational processing. This paper opens with a discussion of applicability of restoration of diacritics in selected languages. Next, we present a neural network-based approach to diacritics generation. The core component of our model is a bidirectional recurrent neural network operating at a character level. We evaluate the model on two existing datasets consisting of four European languages. When combined with a language model, our model reduces the error of current best systems by 20% to 64%. Finally, we propose a pipeline for obtaining consistent diacritics restoration datasets for twelve languages and evaluate our model on it. All the code is available under open source license on `https://github.com/arahusky/diacritics_restoration`.

Keywords: neural networks, diacritics, diacritics generation, error correction

1. Introduction

When writing emails, tweets or texts in certain languages, people for various reasons sometimes write without diacritics. When using Latin script, they replace characters with diacritics (e.g. c with acute or caron) by the underlying basic character without diacritics. Practically speaking, they write in ASCII. We offer several possible reasons for this phenomenon:

- Historically, many devices offered only an English keyboard and/or ASCII encoding (for example older mobile phones and SMS).
- Before Unicode became widespread, there were encoding problems among platforms and even among programs on the same platform, and many people still have this in mind.
- Even though text encoding is rarely a problem any more and all modern devices offer native keyboards, some problems persist. In situations of frequent code switching between English and a language with a substantially different keyboard layout, it is very hard to touch type in both layouts. It is much easier to type both languages using the same layout, although one of them without proper diacritics.
- In some circumstances typing with diacritical marks is significantly slower than using just basic latin characters. The most common example is on-screen keyboards on mobile devices. These keyboards do not include the top row (numerical on US English), so languages that use that row for accented characters are much slower to type. Naturally, users type without explicit accents and rely on the auto-completion systems. However, these systems are usually simple, unigram-based, and based on the word form ambiguity for a given language (cf. Table 1), which introduces many errors. Postponing the step of diacritics generation would be beneficial both for typing speed and accuracy.
- For example in Vietnamese, the language with most diacritics in our data (cf. Table 1), both the above problems are very pronounced: Because Vietnamese uses diacritical marks to distinguish both tones (6) and quality of vowels (up to 3), a vowel can have (and often has) 2 marks. This need to provide efficient typing of very many accented characters led to the invention of systems like unikey.org that allow a user to type all the accented characters using sequences of basic letters. For instance to typeset "đường" a user types "dduwowngf". While this system elegantly solves the problems above with switching keyboard layouts and missing top row of keys, it requires a special software package and it still results in 9 keystrokes to type 5 characters. That is why typing without accents in informal situations like emails or text messages is still common and system for efficient generation of diacritics would be very useful.

Typical languages where approximately half of the words contain diacritics are Czech, Hungarian or Latvian. Nevertheless, as we discuss in Sections 2 and 3, diacritics restoration (also known as diacritics generation or diacritization) is an active problem also in many languages with substantially lower diacritics appearance.

Current approaches to restoration of diacritics (see Section 3) are mostly based on traditional statistical methods. However, in recent years, deep neural networks have shown remarkable results in many areas. To explore their capabilities, we propose a neural based model in Section 4 and evaluate its performance on two datasets in Section 5.

In Section 6, we describe a way to obtain a consistent multilingual dataset for diacritics restoration and evaluate our model on them. The dataset can be downloaded from the published link. Finally, Section 7 concludes this paper with a summary of outcomes.

Language	Words with diacritics	Word error rate of dictionary baseline
Vietnamese	88.4%	40.53%
Romanian	31.0%	29.71%
Latvian	47.7%	8.45%
Czech	52.5%	4.09%
Slovak	41.4%	3.35%
Irish	29.5%	3.15%
French	16.7%	2.86%
Hungarian	50.7%	2.80%
Polish	36.9%	2.52%
Swedish	26.4%	1.88%
Portuguese	13.3%	1.83%
Galician	13.3%	1.62%
Estonian	19.7%	1.41%
Spanish	11.3%	1.28%
Norwegian-Nynorsk	12.8%	1.20%
Turkish	30.0%	1.16%
Catalan	11.1%	1.10%
Slovenian	14.0%	0.97%
Finnish	23.5%	0.89%
Norwegian-Bokmaal	11.7%	0.79%
Danish	10.2%	0.69%
German	8.3%	0.59%
Croatian	16.7%	0.34%

Table 1: Analysis of percentage of words with diacritics and the word error rate of a dictionary baseline. Measured on UD 2.0 data, using the *uninames* method, and CoNLL 17 UD shared task raw data for dictionary. Only words containing at least one alphabetical character are considered.

2. Diacritics Restoration in Languages using Latin Script

Table 1 presents languages using (usually some extended version of) a Latin script. Employing UD 2.0 (Nivre et al., 2017) plain text data, we measure the ratio of words with diacritics, omitting languages with less than 5% of words with diacritics. In eleven of the languages, at least every fifth word contains diacritics; in another eleven languages, at least every tenth word does.

Naturally, high occurrence of words with diacritics does not imply that generating diacritics is an ambiguous task. Consequently, we also evaluate word error rate of a simple dictionary baseline to diacritics restoration: according to a large raw text corpora we construct a dictionary of the most frequent variant with diacritics for a given word without diacritics, and use the dictionary to perform the diacritics restoration.

Table 1 presents the results. We utilized the raw corpora by Ginter et al. (2017) released as supplementary material of CoNLL 2017 UD Shared task (Zeman et al., 2017), which contain circa a gigaword for each language, therefore providing a strong baseline. For nine languages, the word error rate is larger than 2%, and eight more languages have word error rate still above 1%. We conclude that even with a very large dictionary, the diacritics restoration is a challenging task for many languages, and better method is needed.

Letter	Hex code	Unicode name
ø	00F8	LATIN SMALL LETTER O *WITH STROKE*
ł	0142	LATIN SMALL LETTER L *WITH STROKE*
đ	0111	LATIN SMALL LETTER D *WITH STROKE*
ʂ	0282	LATIN SMALL LETTER S *WITH HOOK*
ç	00E7	LATIN SMALL LETTER C *WITH CEDILLA*
š	0161	LATIN SMALL LETTER S *WITH CARON*

Table 2: Unicode characters that cannot be decomposed using NFD (first 4 lines), and those that can. The suffix of the name removed by the *uninames* method is show in italics. As we can see, the strucutre of names is identical, so the method works for all of these characters.

2.1. Methods of Diacritics Stripping

Although there is no standard way of stripping diacritics, a commonly used method is to convert input word to NFD (The Unicode Consortium, 2017, Normalization Form D) which decomposes composite characters into a base character and a sequence of combining marks, then remove the combining marks, and convert the result back to NFC (Unicode Normalization Form C). We dub this method *uninorms*.

We however noted that this method does not strip diacritics for some characters (e.g. for đ and ł).[1] We therefore propose a new method *uninames*, which operates as follows: In order to remove diacritics from a given character, we inspect its name in the Unicode Character Database (The Unicode Consortium, 2017). If it contains a word *WITH*, we remove the longest suffix starting with it, try looking up a character with the remaining name and yield the character if it exists.

The method is illustrated in Table 2, which presents four characters that do not decompose under NFD, but whose diacritics can be stripped by the proposed method.

As shown in Table 3, the proposed *uninames* method recognizes all characters the *uninorms* method does, and some additional ones. Therefore, we employ the *uninames* method to strip diacritics in the paper.

3. Related Work

One of the first papers to describe systems for automatic diacritics restoration is a seminal work by Yarowsky (1999), who compares several algorithms for restoration of diacritics in French and Spanish. Later, models for diacrization in Vietnamese (Nguyen and Ock, 2010), Czech (Richter et al., 2012), Turkish (Adali and Eryiğit, 2014), Ara-

[1] What constitutes a "diacritic mark" is a bit of a problem. On one hand not all characters with a graphical element added to a letter letter contain diacritics, e.g. ¥ (symbol of Japanese Yen) or Đ/ð (Icelandic "eth"). On the other end of the spectrum we have clear diacritics with Unicode canonical decomposition into a letter and a combining mark. Between these clear borders there are the characters that do not have a unicode decomposition, but their names still indicate they are latin letters with some modifier and often they are used the same as characters that do have decomposition. E.g. Norwegian/Danish ø is used exactly like ö in Swedish, it is just an orthographic variation. However while the latter has canonical decomposition in Unicode, the first does not. This is why we opted to treat these characters also as "letters with diacritics".

Lowercased letters with diacritics

í 15.7%	ů 2.5%	ș	ī	ņ	ō
á 11.7%	ú 1.6%	ā	ē	ū	ŕ
é 9.8%	ö 1.5%	ñ	ą	ï	ĺ
ě 6.6%	ă 1.3%	**ł**	ò	ļ	ţ
ä 6.0%	**ø** 1.1%	ć	ż	ń	ì
č 5.5%	à 0.9%	ň	ğ	ù	ó
ř 5.2%	ç 0.9%	â	ś	û	ḥ
ž 4.9%	ü 0.8%	õ	ď	ű	ǎ
ý 4.5%	ã 0.8%	ť	ô	ķ	ί
š 4.4%	è 0.6%	ê	**đ**	ģ	έ
ó 3.2%	î 0.5%	ę	ő	ź	ά
å 2.9%	ț 0.5%	ş	ľ	ë	**ş**

Table 3: Most frequent characters with diacritics from data listed in Table 1, together with their relative frequency. The bold characters are recognized only using the *uninames* method.

bic (Azmi and Almajed, 2015) Croatian, Slovenian, Serbian (Ljubešic et al., 2016), and many other languages were published. The system complexity ranges from simplest models, that for each word apply its most frequent translation as observed in the training data, to models that incorporate language models, part-of-speech tags, morphological and many other features. One of the most similar model to ours is a system by Belinkov and Glass (2015) who used recurrent neural networks for Arabic diacritization.

4. Model Architecture

The core of our model (see Figure 1) is a bidirectional recurrent neural network, which for each input character outputs its correct label (e.g. its variant with diacritics). The input and output vocabularies contain a special out-of-alphabet symbol.

The input characters are embedded, i.e. each character in the input sentence is represented by a vector of d real numbers. The character embeddings are initialized randomly and updated during training.

The embeddings are fed to a bidirectional RNN (Graves and Schmidhuber, 2005). The bidirectional RNN consists of two unidirectional RNNs, one reading the inputs in standard order (forward RNN) and the other in reverse order (backward RNN). The output is then a sum of forward and backward RNN outputs. This way, bidirectional RNN is processing information from both preceding and following context. The model allows an arbitrary number of stacked bidirectional RNN layers.

The output of the (possibly multilayer) bidirectional RNN is at each time step reduced by an identical fully connected layer to an o-dimensional vector, where o is the size of the output alphabet. A nonlinearity is then applied to these reduced vectors.

Finally, we use a softmax layer to produce a probability distribution over output alphabet at each time step.

The loss function is the cross-entropy loss summed over all outputs.

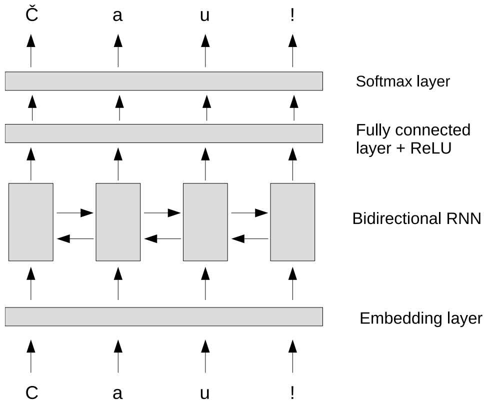


Figure 1: Visualisation of our model.

4.1. Residual connections

The proposed model allows an arbitrary number of stacked RNN layers. The model with multiple layers allows each stacked layer to process more complex representation of current input. This naturally brings potential to improve accuracy of the model.

As stated by (Wu et al., 2016), simple stacking of more RNN layers works only up to a certain number of layers. Beyond this limit, the model becomes too difficult to train, which is most likely caused by vanishing and exploding gradient problems (Pascanu et al., 2013). To improve the gradient flow, (Wu et al., 2016) incorporate residual connections to the model. To formalize this idea, let RNN_i be the i-th RNN layer in a stack and $x_0 = (\mathit{inp}_1, \mathit{inp}_2, \dots, \mathit{inp}_N)$ input to the first stacked RNN layer RNN_0. The model we have proposed so far works as follows:

$$
\begin{aligned}
o_i, c_i &= \mathit{RNN}_i(x_i)\\
x_{i+1} &= o_i\\
o_{i+1}, c_{i+1} &= \mathit{RNN}_{i+1}(x_{i+1}),
\end{aligned}
$$

where o_i is the output of i-th stacked RNN layer and c_i is a sequence of its hidden states. The model with residual connections between stacked RNN layers then works as follows:

$$
\begin{aligned}
o_i, c_i &= \mathit{RNN}_i(x_i)\\
x_{i+1} &= o_i + x_i\\
o_{i+1}, c_{i+1} &= \mathit{RNN}_{i+1}(x_{i+1})
\end{aligned}
$$

4.2. Decoding

For inference we use a left-to-right beam search decoder combining the neural network and the language model likelihoods. The process is a modified version of standard beam search used by Xie et al. (2016) for decoding sequence-to-sequence models.

Let b denote the beam size. The hypotheses in the beam are initialized with the b most probable first characters. In each step, all beam hypotheses are extended with b most probable variants of the respective character, creating b^2 hypotheses. These are then sorted and the top b of them are kept.

Whenever a space is observed in the output, all affected hypotheses are reranked using both the RNN model output probabilities and language model probabilities. The hypothesis probability in step k can be computed as:

$$P(y_{1:k}|x) = (1-\alpha)\log P_{NN}(y_{1:k}|x) + \alpha \log P_{LM}(y_{1:k}),$$

where x denotes the input sequence, y stands for the decoded symbols contained within current hypothesis, P_{NN} and P_{LM} are neural network and language model probabilities and the hyper-parameter α determines the weight of the language model. To keep both $\log P_{NN}$ and $\log P_{LM}$ terms within a similar range in the decoding, we compute the $\log P_{NN}$ as the mean of output token log probabilities and additionally normalize P_{LM} by the number of words in the sequence.

To train the language model as well as to run it, we use the open-source KenLM toolkit (Heafield, 2011).

5. Experiments

To compare performance of our model with current approaches, we perform experiments using two existing datasets. The first one, created by Ljubešic et al. (2016), consists of Croatian, Serbian and Slovenian sentences from three sources: Wikipedia texts, general Web texts and texts from Twitter. Since Web data are assumed to be the noisiest, they are used only for training. Wikipedia and Twitter testing sets should then cover both standard and non-standard language. The second evaluation dataset we utilize consists of Czech sentences collected mainly from newspapers, thus it covers mostly standard Czech.

5.1. Training and Decoding Details

We used the same model configuration for all experiments. The bidirectional RNN has 2 stacked layers with residual connections and utilizes LSTM units (Hochreiter and Schmidhuber, 1997) of dimension 300. Dropout (Srivastava et al., 2014) at a rate of 0.2 is used both on the embedded inputs and after each bidirectional layer. All weights are initialized using Xavier uniform initializer (Glorot and Bengio, 2010).

The vocabulary of each experiment consists of top 200 most occurring characters in a training set and a special symbol (<UNK>) for unknown characters.

To train the model, we use the Adam optimizer (Kingma and Ba, 2014) with learning rate 0.0003 and a minibatch size of 200. Each model was trained on a single GeForce GTX 1080 Ti for approximately 4 days. After training, the model with the highest accuracy on the corresponding development set was selected.

To estimate the decoding parameter α, we performed an exhaustive search over [0,1] with a step size of 0.1. The parameter was selected to maximize model performance on a particular development set. All results were obtained using a beam width of 8.

5.2. Croatian, Serbian and Slovenian

The original dataset contains training files divided into Web, Twitter and Wikipedia subsets. However, Ljubešic et al. (2016) showed that concatenating all these language-specific sets for training yields best results. Therefore, we used only concatenated files for training the models for each of three languages in our experiments. The training files contain 17 968 828 sentences for Croatian, 11 223 924 sentences for Slovenian and 8 376 810 sentences for Serbian.[2] All letters in the dataset are lowercased.

To remove diacritics from the collected texts, Ljubešic et al. (2016) used a simple script that replaced four letters (ž, ć, č, š) with their ASCII counterparts (z, c, c, s), and one letter (đ) with its phonetic transcription (dj). This results in the input and target sentences having different length. Since our model requires both input and target sentences to have the same length, additional data preprocessing was required before feeding the data into the model: we replace all occurrences of the *dj* sequence in both input and target sentences by a special token, and replace it back to *dj* after decoding.

The results of the experiment with comparison to previous best system (*Lexicon*, *Corpus*) are presented in Table 4. The *Lexicon* method replaces each word by its most frequent translation as observed in the training data. The *Corpus* method extends it via log-linear model with context probability. These methods were evaluated by Ljubešic et al. (2016) and the *Corpus* method is to the best of our knowledge state-of-the-art system for all three languages. System accuracy is measured, similarly to the original paper, on all words which have at least one alphanumerical character.

We incorporated the same language models as used by the authors of the original paper. There are two points in the results we would like to stress:

- Our system with language model reduces error by more than 30% on wiki data and by more than 20% on tweet data. Moreover, our model outperforms the current best system on wiki data even if it does not incorporate the additional language model, which makes the model much smaller (˜30MB instead of several gigabytes of the language model).
- Diacritics restoration problem is easier on standard language (wiki) than on non-standard data (tweets). This has, in our opinion, two reasons. First, the amount of wiki data in the training sets is substantially higher than the amount of non-standard data (tweets). This makes the model fit more standard data. Second, due to lower language quality in Twitter data, we suppose that the amount of errors in the gold data is higher.

5.3. Czech

The second experiment we conducted is devoted to diacritics restoration in Czech texts. To train both the neural network and language models, we used the SYN2010 corpus (Křen et al., 2010), which contains 8 182 870 sentences collected from Czech literature and newspapers. To evaluate the model, PDT3.0 (Hajič et al., 2018) testing set with 13 136 sentences originating from Czech newspapers is used. Both the training and testing set, thus, contain mainly standard Czech. For language model training, we consider only those {2,3,4,5}-grams that occurred at least twice, and use default KenLM options.

Table 5 presents a comparison of our model performance with Microsoft Office Word 2010, ASpell, CZACCENT (Rychlý, 2012) and Korektor (Richter et al., 2012), the latter being the state-of-the-art system of diacritics restoration

[2] The Serbian dataset is based on a Latin script.

System	wiki			tweet		
	hr	sr	sl	hr	sr	sl
Lexicon	0.9936	0.9924	0.9933	0.9917	0.9893	0.9820
Corpus	0.9957	0.9947	0.9962	0.9938	0.9917	0.9912
Our model	0.9967	0.9961	0.9970	0.9932	0.9939	0.9882
Our model + LM	**0.9973**	**0.9968**	**0.9974**	**0.9951**	**0.9944**	**0.9930**
Error reduction	36.81%	39.74%	30.45%	21.62%	32.14%	20.77%

Table 4: Results obtained on Crotian (HR), Serbian (SR) and Slovenian (SL) Wikipedia and Twitter testing sets. Note that the word accuracy presented in the table is not measured on all words, but only on words having at least one alphanumerical character.

System	Word accuracy
Microsoft Office Word 2010 (*)	0.8910
ASpell (*)	0.8839
Lexicon	0.9527
CZACCENT	0.9607
Corpus	0.9713
Korektor	0.9861
Our model	0.9887
Our model + LM	**0.9951**
Error reduction	64.75%

Table 5: Comparison of several models of restoration of diacritics for Czech. The (*) denotes reduced test data (see text).

for Czech. Note that evaluation using Microsoft Office Word 2010 and ASpell was performed only on the first 746 (636) sentences, because it requires user interaction (confirming the suggested alternatives).

As the results show, models that are not tuned to the task of diacritics restoration perform poorly. Our model combined with a language model reduces the error of the previous state-of-the-art system by more than 60%; our model achieves slightly higher accuracy than Korektor even if no language model is utilized.

5.3.1. Ablation Experiments

One of the reasons why deep learning works so well is the availability of large training corpora. This motivates us to explore the amount of data our model needs to perform well. As Figure 2 shows, the RNN model trained on 50 000 random sentences from SYN2010 corpus performs better on the PDT3.0 testing set than the *Lexicon* baseline trained on full SYN2010 corpus. Further, up to 5M sentences the performance of the RNN model increases with the growing training set size. We do not observe any performance difference between the RNN model trained on 5M and 8M sentences.

The second ablation experiment examines the effect of residual connections. We trained models with 2, 3, 4 and 5 stacked layers each with and without residual connections. We also trained a simple model with 1 bidirectional layer without residual connections. The results of this experiment are presented in Figure 3. Apart from the big difference in word accuracy between the model with 1 layer and other models, we can see that models with residual connections perform generally better than when no residual connections are incorporated. It is also evident that when more layers are added in stack, performance of models without residual connections deteriorates while performance of models with additional residual connections does not.

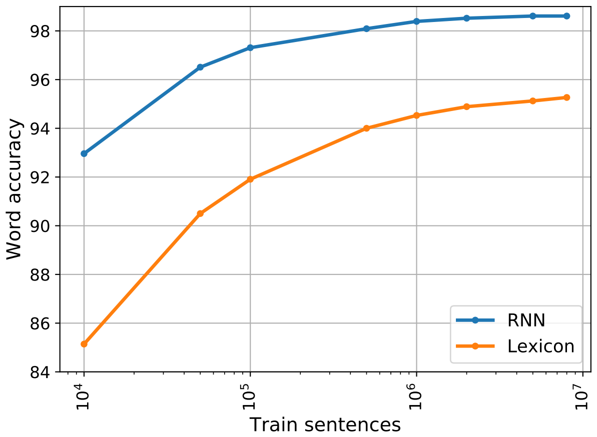


Figure 2: Comparison of RNN and Lexicon performance with varrying training data size.

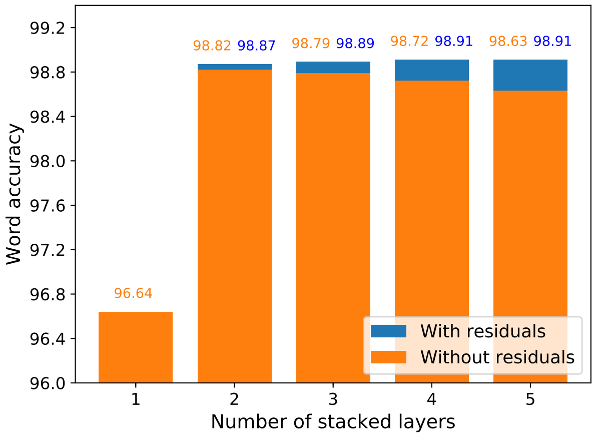


Figure 3: Effect of using residual connections with respect to the number of stacked layers.

6. New Multilingual Dataset

As discussed in the preceding sections, diacritics restoration is an active field of research. However, to the best of our knowledge, there is no consistent approach to obtaining datasets for this task. When a new diacritics restoration system is published, a new dataset is typically created both for training and testing. This makes it difficult to compare performance across systems. We thus propose a new pipeline for obtaining consistent multilingual datasets for the task of diacritics restoration.

6.1. Dataset

As the data for diacritics restoration need to be clean, we decided to utilize Wikipedia for both development and testing sets. Because there may be not enough data to train potential diacritics restoration systems on Wikipedia texts only, we further decided to create training sets from the (general) Web. We chose two corpora for this task: the W2C corpus (Majliš, 2011) with texts from Wikipedia and the general Web in 120 languages, and the CommonCrawl corpus with language annotations generated by Buck et al. (2014) with a substantially larger amount of general Web texts in more than 150 languages.

To create training, development and testing data from the Wikipedia part of the W2C corpus, its data are first segmented into sentences, these are then converted to lowercase and finally split into disjoint training, development and testing set. The split was performed in such a way that all three sets consist of sentences collected from whole articles rather then being randomly sampled across all articles. Each testing set consists of 30 000 sentences, development set of approximately 15 000 sentences and the rest of the data are preserved for training set.

The pipeline for creating additional training data from the CommonCrawl corpus starts with the removal of invalid UTF8 data and Wikipedia data. These filtered data are then segmented into sentences and converted to lowercase. Since these data come from general Web and may be noisy (e.g. contain sentences with missing diacritics), only those sentences that have at least 100 characters and contain at least a certain amount of diacritics are preserved. The constant determining the minimum amount of diacritics is language specific and is derived from Table 1. Finally, sentence intersection with existing development and testing set is removed and maximally ten similar sentences are preserved in the training data. Since both baseline methods (*Lexicon* and *Corpus*) require data to be word tokenized, all texts are also word tokenized.

The dataset was created for 12 languages (see Table 6), where the additional training sets were generated from the *2017_17* web crawl. Complete dataset can be downloaded from `http://hdl.handle.net/11234/1-2607`.

6.2. Experiments

We train and evaluate our model on the created dataset and compare its performance to two baseline methods. The same model hyperparameters as described in Section 5.1 are used, except for the RNN cell dimension, which is 500.

Training was performed in two phases. First, each language specific model was trained on particular CommonCrawl (Web) training set for approximately four days. Then, each model was fine-tuned with a smaller learning rate 0.0001 on respective Wikipedia training set for three more days. Finally, as all models seemed to be continuously improving on the development sets, we took the last model checkpoints for evaluating.

Both baseline methods and language models were trained on concatenation of Wikipedia and CommonCrawl training data. For language model training, we considered only those {2,3,4,5}-grams that occurred at least twice, and used default KenLM options.

To measure model performance, modified word error accuracy is used. The alpha-word accuracy considers only words that consist of at least one alphabetical character, because only these can be potentially diacritized. The testing set results of *Lexicon* and *Corpus* baselines, as well as of our models before and after fine-tuning, and with a language model are presented in Table 6.

As results show, our model outperforms both baselines even if no language model is used. Moreover, incorporation of the language model helps the model perform better as well as does model fine-tuning. Without fine-tuning, all models but the Romanian outperform baselines. We suspect that the reason why the Romanian model before fine-tuning performs worse than the *Corpus* method is that non-standard Web data differ too much from standard data from Wikipedia. It is also an interesting fact that the biggest error reduction is at Vietnamese and Romanian which seem to be most difficult for both baseline methods.

7. Conclusion

In this work, we propose a novel combination of recurrent neural network and a language model for performing diacritics restoration. The proposed system is language agnostic as it is trained solely from parallel corpora of texts without diacritics and diacritized texts. We test our system on two existing datasets comprising of four languages, and we show that it outperforms previous state-of-the-art systems. Moreover, we propose a pipeline for generating consistent multilingual diacritics restoration datasets, run it on twelve languages, publish the created dataset, evaluate our system on it and provide a comparison with two baseline methods. Our method outperforms even the stronger contextual baseline method on the new dataset by a big margin.

Future work includes detailed error analysis, which could reveal types of errors made by our system. Since certain words may be correctly diacritized in several ways given the context of the whole sentence, such error analysis could also set the language specific limit on the accuracy that can be achieved. Further, when designing our multilingual dataset we decided to use testing sets with sentences from Wikipedia articles. This was well motivated as we wanted it to contain sentences with proper diacritics. However, such testing sets contain mainly standard language and are thus worse for comparison of models aiming to generate diacritics for non-standard language. Therefore, we plan to create additional development and testing sets in the future work.

While experimenting with the model on Czech we found out that when it is trained to output instructions (e.g. add caron) instead of letters, it performs better. Future work thus also includes thorough inspection of this behavior when applied to all languages.

Finally, the system achieves better results when a language model is incorporated while inferring. Because the use of an external model both slows down the inferring process and

Language	Wiki sentences	Web sentences	Words with diacritics	Lexicon	Corpus	Our model w/o finetuning	Our model	Our model + LM	Error reduction
Vietnamese	819 918	25 932 077	73.63%	0.7164	0.8639	0.9622	0.9755	0.9773	83.33%
Romanian	837 647	16 560 534	24.33%	0.8533	0.9046	0.9018	0.9799	0.9837	82.96%
Latvian	315 807	3 827 443	39.39%	0.9101	0.9457	0.9608	0.9657	0.9749	53.81%
Czech	952 909	52 639 067	41.52%	0.9590	0.9814	0.9852	0.9871	0.9906	49.20%
Polish	1 069 841	36 449 109	27.09%	0.9708	0.9841	0.9891	0.9903	0.9955	71.64%
Slovak	613 727	12 687 699	35.60%	0.9734	0.9837	0.9868	0.9884	0.9909	44.21%
Irish	50 825	279 266	26.30%	0.9735	0.9800	0.9842	0.9846	0.9871	35.55%
Hungarian	1 294 605	46 399 979	40.33%	0.9749	0.9832	0.9888	0.9902	0.9929	58.04%
French	1 818 618	78 600 777	14.65%	0.9793	0.9931	0.9948	0.9954	0.9971	58.11%
Turkish	875 781	72 179 352	25.34%	0.9878	0.9905	0.9912	0.9918	0.9928	24.14%
Spanish	1 735 516	80 031 113	10.41%	0.9911	0.9953	0.9956	0.9958	0.9965	25.57%
Croatian	802 610	7 254 410	12.39%	0.9931	0.9947	0.9951	0.9951	0.9967	36.92%

Table 6: Results obtained on new multilingual dataset. Note that the alpha-word accuracy presented in the table is measured only on those words that have at least one alphabetical character. The last column presents errror reduction of our model combined with language model compared to the Corpus method.

requires significantly more memory, it would be desirable to train the model in such way that no additional language model is needed. We suspect that multitask learning (e.g. training the model also to predict next/previous letter) may compensate for the absence of a language model.

8. Acknowledgements

The research described herein has been supported by the Grant No. DG16P02R019 of the Ministry of Culture of the Czech Republic. Data has been used from and stored into the repository of LINDAT/CLARIN, a large research infrastructure supported by the Ministry of Education, Youth and Sports of the Czech Republic under projects LM2015071 and CZ.02.1.01/0.0/0.0/16_013/0001781.

Thanks to NGỤY Giang Linh for the information and examples on Vietnamese.

9. Bibliographical References

Adali, K. and Eryiğit, G. (2014). Vowel and diacritic restoration for social media texts. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 53–61.

Azmi, A. M. and Almajed, R. S. (2015). A survey of automatic arabic diacritization techniques. *Natural Language Engineering*, 21(3):477–495.

Belinkov, Y. and Glass, J. (2015). Arabic diacritization with recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2281–2285.

Buck, C., Heafield, K., and Van Ooyen, B. (2014). N-gram counts and language models from the common crawl. In *LREC*, volume 2, page 4.

Ginter, F., Hajič, J., Luotolahti, J., Straka, M., and Zeman, D. (2017). CoNLL 2017 shared task - automatically annotated raw texts and word embeddings. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256.

Graves, A. and Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, pages 5–6.

Hajič, Jan and Bejček, Eduard and Bémová, Alevtina and Buráňová, Eva and Hajičová, Eva and Havelka, Jiří and Homola, Petr and Kárník, Jiří and Kettnerová, Václava and Klyueva, Natalia and Kolářová, Veronika and Kučová, Lucie and Lopatková, Markéta and Mikulová, Marie and Mírovský, Jiří and Nedoluzhko, Anna and Pajas, Petr and Panevová, Jarmila and Poláková, Lucie and Rysová, Magdaléna and Sgall, Petr and Spoustová, Johanka and Straňák, Pavel and Synková, Pavlína and Ševčíková, Magda and Štěpánek, Jan and Urešová, Zdeňka and Vidová Hladká, Barbora and Zeman, Daniel and Zikánová, Šárka and Žabokrtský, Zdeněk. (2018). *Prague Dependency Treebank 3.5*. Institute of Formal and Applied Linguistics, LINDAT/CLARIN, Charles University.

Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kingma, D. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Křen, M., Bartoň, T., Cvrček, V., Hnátková, M., Jelínek, T., Kocek, J., Novotná, R., Petkevič, V., Procházka, P., Schmiedtová, V., et al. (2010). Syn2010: žánrově vyvážený korpus psané češtiny. *Ústav Českého národního korpusu FF UK, Praha*.

Ljubešic, N., Erjavec, T., and Fišer, D. (2016). Corpus-based diacritic restoration for south slavic languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016). European Language Resources Association (ELRA)(may 2016)*.

Majliš, M. (2011). W2c–web to corpus–corpora.

Nguyen, K.-H. and Ock, C.-Y. (2010). Diacritics restoration in vietnamese: letter based vs. syllable based model. *PRICAI 2010: Trends in Artificial Intelligence*, pages 631–

636.

Nivre, J., Agić, Ž., Ahrenberg, L., et al. (2017). Universal Dependencies 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University, Prague.

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. *ICML (3)*, 28:1310–1318.

Richter, M., Straňák, P., and Rosen, A. (2012). Korektor-a system for contextual spell-checking and diacritics completion. In *COLING (Posters)*, pages 1019–1028.

Rychlý, P. (2012). Czaccent–simple tool for restoring accents in czech texts. *RASLAN 2012 Recent Advances in Slavonic Natural Language Processing*, page 85.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

The Unicode Consortium. (2017). *The Unicode Standard, Version 10.0.0*. The Unicode Consortium, Mountain View, CA.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Xie, Z., Avati, A., Arivazhagan, N., Jurafsky, D., and Ng, A. Y. (2016). Neural language correction with character-based attention. *arXiv preprint arXiv:1603.09727*.

Yarowsky, D. (1999). A comparison of corpus-based techniques for restoring accents in spanish and french text. In *Natural language processing using very large corpora*, pages 99–120. Springer.

Zeman, D., Popel, M., Straka, M., Hajič, J., Nivre, J., Ginter, F., Luotolahti, J., Pyysalo, S., Petrov, S., Potthast, M., Tyers, F., Badmaeva, E., Gökırmak, M., Nedoluzhko, A., Cinková, S., Hajič jr., J., Hlaváčová, J., Kettnerová, V., Urešová, Z., Kanerva, J., Ojala, S., Missilä, A., Manning, C., Schuster, S., Reddy, S., Taji, D., Habash, N., Leung, H., de Marneffe, M.-C., Sanguinetti, M., Simi, M., Kanayama, H., de Paiva, V., Droganova, K., Martínez Alonso, H., Uszkoreit, H., Macketanz, V., Burchardt, A., Harris, K., Marheinecke, K., Rehm, G., Kayadelen, T., Attia, M., Elkahky, A., Yu, Z., Pitler, E., Lertpradit, S., Mandl, M., Kirchner, J., Fernandez Alcalde, H., Strnadova, J., Banerjee, E., Manurung, R., Stella, A., Shimada, A., Kwak, S., Mendonça, G., Lando, T., Nitisaroj, R., and Li, J. (2017). CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics.

CUNI System for the Building Educational Applications 2019 Shared Task: Grammatical Error Correction

Jakub Náplava and **Milan Straka**
Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics
{naplava,straka}@ufal.mff.cuni.cz

Abstract

In this paper, we describe our systems submitted to the Building Educational Applications (BEA) 2019 Shared Task (Bryant et al., 2019). We participated in all three tracks. Our models are NMT systems based on the Transformer model, which we improve by incorporating several enhancements: applying dropout to whole source and target words, weighting target subwords, averaging model checkpoints, and using the trained model iteratively for correcting the intermediate translations. The system in the Restricted Track is trained on the provided corpora with oversampled "cleaner" sentences and reaches 59.39 F0.5 score on the test set. The system in the Low-Resource Track is trained from Wikipedia revision histories and reaches 44.13 F0.5 score. Finally, we finetune the system from the Low-Resource Track on restricted data and achieve 64.55 F0.5 score, placing third in the Unrestricted Track.

1 Introduction

Starting with the 2013 and 2014 CoNLL Shared Tasks on grammatical error correction (GEC), much progress has been done in this area. The need to correct a variety of error types lead most researchers to focus on models based on machine translation (Brockett et al., 2006) rather than custom designed rule-based models or a combination of single error classifiers. The machine translation systems turned out to be particularity effective when Junczys-Dowmunt and Grundkiewicz (2016) presented state-of-the-art statistical machine translation system. Currently, models based on statistical and neural machine translation achieve best results: in restricted settings with training limited to certain public training sets (Zhao et al., 2019); unrestricted settings with no restrictions on training data (Ge et al., 2018); and also in low-resource track where the training data should not come from any annotated corpora (Lichtarge et al., 2018).[1]

In this paper, we present our models and their results in the restricted, unrestricted, and low-resource tracks. We start with a description of related work in Section 2. We then describe our systems together with the implementation details in Section 3. Section 4 is dedicated to our results and ablation experiments. Finally, in Section 5 we conclude the paper with some proposals on future work.

2 Related Work

Transformer (Vaswani et al., 2017) is currently one of the most popular architectures used in machine translation. Its self-attentive layers allow better gradient flow when compared to recurrent neural models and the masking in decoder provides faster training. Junczys-Dowmunt et al. (2018) propose several improvements for training Transformer on GEC: using dropout on whole input words, assigning weight to target words based on their alignment to source words, and they also propose to oversample sentences from the training set in order to have the same error rate as the test set.

Majority of work in grammatical error correction has been done in restricted area with a fixed set of annotated training datasets. Lichtarge et al. (2018), however, show that training a neural machine translation system from Wikipedia edits can lead to surprisingly good results. As the authors state, corpus of Wikipedia edits is only weakly supervised for the task of GEC, because most of the edits are not corrections of grammatical errors and also they are not human curated specifically for GEC. To overcome these issues, the authors use iterative decoding which allows for incremental corrections. In other words, the model can re-

[1] Note that in this settings Wikipedia revisions are allowed

Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications, pages 183–190
Florence, Italy, August 2, 2019. ©2019 Association for Computational Linguistics

peatedly translate its current output as long as the translation is more probable then keeping the sentence unchanged. Similar idea is also presented in (Ge et al., 2018), where the translation system is trained with respect to the incremental inference.

3 Our System

In this section, we present our three systems submitted to each track of the BEA 2019 Shared Task. We start with the Restricted Track In Section 3.1, where we present a series of improvements to the baseline Transformer model. In Section 3.2, we describe our model trained on Wikipedia revisions which was submitted to the Low-Resource Track. Finally, in Section 3.3, we describe the model submitted to the Unrestricted Track.

All our models are based on the Transformer model from Tensor2Tensor framework version 1.12.0.[2]

3.1 Restricted Track

In the Restricted Track, we use the 5 provided datasets for system development: FCE v2.1 (Yannakoudakis et al., 2011), Lang-8 Corpus of Learner English (Mizumoto et al., 2011; Tajiri et al., 2012), NUCLE (Dahlmeier et al., 2013), Write & Improve (W&I) and LOCNESS v2.1 (Bryant et al., 2019; Granger, 1998). From Lang-8 corpus, we took only the sentences annotated by annotators with ID 0 (A0) and ID 1 (A1). All but the development sets from W&I and LOCNESS datasets were used for training. The simple statistics of these datasets are presented in Table 1. The displayed error rate is computed using maximum alignment of original and annotated sentences as a ratio of non-matching alignment edges (insertion, deletion, and replacement).

Dataset		Sentences	Average error rate
Lang8	A0	1 037 561	13.33 %
	A1	67 975	25.84 %
FCE v2.1	train	28 350	11.31 %
	dev	2 191	11.67 %
	test	2 695	12.87 %
NUCLE		57 151	6.56 %
W&I	train A	10 493	18.13 %
	train B	13 032	11.68 %
	train C	10 783	5.62 %
	dev A	1 037	18.32 %
	dev B	1 290	12.46 %
	dev C	1 069	5.91 %
LOCNESS	dev N	998	4.72 %

Table 1: Statistics of available datasets. The error rate is computed as a ratio of non-matching alignment edges.

We use the *transformer_base* configuration of Tensor2Tensor as our baseline solution. The training dataset consists of 1 230 231 sentences. After training, beam search decoding is employed to generate model corrections and we choose the checkpoint with the highest accuracy on a development set concatenated from the W&I and LOCNESS development sets.

3.1.1 Transformer Big

The first minor improvement was to use the *transformer_big* configuration instead of *transformer_base*. This configuration has bigger capacity and as Popel and Bojar (2018) show, it reaches substantially better results on certain translation tasks.

3.1.2 Source and Target Word Dropout

Dropout (Srivastava et al., 2014) is a regularization technique that turned out to be particularly effective in the field of neural networks. It works by masking several randomly selected activations during training, which should prevent the neural network from overfitting the training data. In the area of NLP, it is a common approach to apply dropout to whole embeddings, randomly zeroing certain dimensions. As Junczys-Dowmunt et al. (2018) show, we can also apply dropout to whole source words to reduce trust in the source words. Specifically, full source word embedding vector is set to zero vector with probability p. We further note this probability as the *source_word_dropout*.

To make regularization even more effective, we decided to dropout also whole target word embeddings. We refer to the probability with which we dropout entire target word embeddings as the *target_word_dropout*.

3.1.3 Edited MLE

Compared to traditional machine translation task, whose goal is to translate one language to another, GEC operates on a single language. Together with the relatively low error rate, the translation system may converge to a local optimum, in which the

[2] https://github.com/tensorflow/tensor2tensor

model copies the input unchanged to the output. To overcome this issue, Junczys-Dowmunt et al. (2018) propose to change the maximum likelihood objective to assign bigger weights to target tokens different from the source tokens. More specifically, they start by computing the word alignment between each source $x = (x_0, x_1, ..x_N)$ and target sentence $y = (y_0, y_1, ...y_M)$. Then they set the weight λ_t of the target word y_t to 1 if it is matched, and otherwise, if it is an insertion or replacement of a source token, λ_t is set to some predefined constant. Modified log-likelihood training objective then takes following form:

$$L(x, y) = -\sum_{t=1}^{M} \lambda_t \log P(y_t|x, y_0, \ldots, y_{t-1}).$$

3.1.4 Data oversampling

It is crucial to have training data from the same domain as the test data, i.e., training data containing similar errors with similar distribution as the test data. As we can see in the Table 1, the vast majority of our training data comes from the Lang-8 corpus. However, as it is quite noisy and of low quality, it matches the target domain the least. Therefore, we decided to oversample other datasets. Specifically, we add the W&I training data 10 times, all FCE data 5 times and NUCLE corpus 5 times to the training data. The oversampled training set consists of 1 900 551.

In Table 1, we can also see token error rate of each corpus. The development error rate in W&I and LOCNESS varies from 5.91% up to 18.32%. This gives us a basic idea how the test data looks like, and since the test data does not contain annotations from which set (A, B, C, N) it comes, we decided not to optimize the training data against the token error rate any further.

3.1.5 Checkpoint Averaging

Popel and Bojar (2018) report that averaging several last Transformer model checkpoints during training leads both to lower variance results and also to slightly better performance than the baseline without averaging. They propose to save checkpoints every one hour and average either 8 or 16 last checkpoints. Since we found out that the model overfits the oversampled dataset quite quickly, we save checkpoints every 30 minutes.

3.1.6 Iterative decoding

A system for grammatical error correction should correct all errors in the text while keeping the rest of the text intact. In many situations with multiple errors in a sentence, the trained system, however, corrects only a subset of its errors. Lichtarge et al. (2018) and Ge et al. (2018) propose to use the trained system iteratively to allow the system to correct certain errors during further iterations. Iterative decoding is done as long as the cost of the correction is less than the cost of the identity translation times a predefined constant. While Lichtarge et al. (2018) use the same trained model log-likelihoods as the cost function, Ge et al. (2018) utilize an external language model for it. Because the restricted track does not contain enough training data to train a quality language model, we adopted the first approach and utilize the trained system log-likelihoods as a stopping criterion.

```
Data: input_sent; max_iters; threshold
for iter in [1,2,..,max_iters] do
    beam_results = decode(input_sent);
    identity_cost = $+\infty$;
    non_identity_cost = $+\infty$;
    non_identity_sent = None
    for beam_item in beam_results do
        text = beam_item[”text”];
        cost = beam_item[”cost”];
        if text == input_sent then
            identity_cost = cost;
        else if cost $<$ non_identity_cost then
            non_identity_cost = cost;
            non_identity_sent = text;
    end
    if non_identity_cost $\leq$ threshold $\cdot$ identity_cost then
        input_sent = non_identity_sent;
    else
        break;
    end
end
return input_sent;
```

Algorithm 1: Iterative decoding algorithm

The iterative decoding algorithm we use is presented in Algorithm 1. Note that when the resulting beam does not contain the identical (non-modified) sentence, the correction with the lowest cost is returned regardless of the provided threshold. We adopted this approach for two reasons – efficiently obtaining the log-likelihood of the identical sentence would require non-trivial modification of the Tensor2Tensor framework, and for *threshold* > 1 (i.e., allow generating changes which are less likely than identical sentence) the results are the same.

3.1.7 Implementation Details

Apart from the first experiment in which we use *transformer_base* configuration, all our experiments are based on *transformer_big* architecture. We use Adafactor optimizer (Shazeer and Stern, 2018), linearly increasing the learning rate from 0 to 0.011 over the first 8000 steps, then decrease it proportionally to the number of steps after that.[3] We also experimented with Adam optimizer with default learning rate schedule, however, training converged poorly. We hypothesise that this was caused by the higher learning rate.

All systems are trained on 4 Nvidia P5000 GPUs for approximately 2 days. The vocabulary consists of approximately 32k most common word-pieces, batch size is 2000 word-pieces per each GPU and all sentences with more than 150 word-pieces are discarded. Model checkpoints are saved every 30 minutes. We ran a grid search to find values of all hyperparameters described in the previous sections.

At evaluation time, we run iterative decoding using a beam size of 4. Beam-search length-balance decoding hyperparameter alpha is set to 0.6. This applies to all further experiments.

3.2 Low-Resource Track

The dataset for our experiments in the Low-Resource Track consists of nearly 190M segment pairs extracted from Wikipedia XML revision dumps. To acquire these, we downloaded all English Wikipedia revision dumps (155GB in size) and processed them with the *WikiRevision* dataset problem from Tensor2Tensor. The processing pipeline extracts individual pages with chronological snapshots, removes all non-text elements and downsamples the snapshots. With low probability, additional spelling noise is added by either inserting a random character, deleting a random character, transposing two adjacent characters or replacing a character with a random one. With the same low probability, a random text substring (up to 8 characters) may also be replaced with a marker, which should force the model to learn infilling. Finally, the texts from two consecutive snapshots are aligned and sequences between matching segments are extracted to form a training pair. Only 4% of identical samples are preserved.

Despite having an enormous size compared to 1.2M sentences in the Restricted Track, the training pairs extracted from Wikipedia are extremely noisy, containing a lot of edits that are in no sense grammatical correction. It is also worth noting that the identical data modified by the spelling and infilling operations form nearly 50% of the training pairs.

Since we want to re-use the system in other scenarios, we train the model on the original (untokenized) training data. To evaluate the model on the BEA development and test data, we detokenize the data using Moses,[4] run model inference and finally tokenize corrected sentences using spaCy.[5]

The training segments may contain newline and tab symbols; therefore, we applied additional post-processing in which we replaced both these symbols with spaces.

Because overfitting should not be an issue with the Wikipedia data, we decided to use *transformer_clean_big_tpu* configuration, following Lichtarge et al. (2018). This configuration, compared to *transformer_big*, performs no dropouts. The vocabulary consists of approximately 32k most common word-pieces, batch size is 2000 word-pieces per each GPU and all sentences with more than 150 word-pieces are discarded. We train the model for approximately 10 days on 4 Nvidia P5000 GPUs. After training, the last 8 checkpoints saved in 1 hour intervals are averaged. Finally, we run a grid search to find optimal values of *threshold* and *max_iters* in iterative decoding algorithm.

3.3 Unrestricted Track

Our system submitted to the Unrestricted Track is the best system from the Low-Resource Track finetuned on the oversampled training data as described in Section 3.1.4. Since our system in the Unrestricted Track was trained on detokenized data, the training sentences for finetuning were also detokenized. The tokenization and detokenization was done in the same way as described in Section 3.2.

[3] We use 8000 warmup steps and learning_rate_schedule=rsqrt_decay

[4] We use mosestokenizer v1.0.0 and its detokenizer.

[5] We use spaCy v1.9.0 and the en_core_web_sm-1.2.0 model.

Track	P	R	$F_{0.5}$	Best	Rank
Restricted	67.33	40.37	59.39	69.47	10 / 21
Unrestricted	68.17	53.25	64.55	66.78	3 / 7
Low Resource	50.47	29.38	44.13	64.24	5 / 9

Table 2: Official shared task $F_{0.5}$ scores on the test set.

System	A	B	C	N	Combined
Transformer-base architecture	39.98	32.68	23.97	14.49	32.47
Transformer-big architecture	39.70	35.13	26.22	20.20	34.20
+ 0.2 src drop, 0.1 tgt drop, 3 MLE	42.06	38.25	28.72	23.80	38.15
+ Extended dataset	45.99	41.79	32.52	27.89	40.86
+ Averaging 8 checkpoints	47.90	44.13	36.19	29.05	43.29
+ Iterative decoding	48.75	45.46	37.09	30.19	44.27

Table 3: Development combined $F_{0.5}$ score of incremental improvements of our system.

We finetune the system with the Adafactor optimizer. The learning rate linearly increases from 0 to 0.0003 over the first 20 000 steps and then remains constant. We employ source word dropout, target word dropout and weighted MLE. The training data for finetuning and the rest of the training scheme are identical to Section 3.1.7.

4 Results

We now present the results of our system. Additionally, we present several ablation experiments, which are evaluated on the concatenation of W&I and LOCNESS development sets (the *Dev combined*).

4.1 Shared Task Results

The official results of our three systems on the blind test set are presented in Table 2. All our systems have substantially higher precision than recall. It is an interesting observation that the system in the unrestricted track has similar precision as the model in the restricted track while having higher recall.

4.2 Restricted Track

The first experiment we conducted is devoted to the incremental enhancements that we proposed in Section 3.1. As Table 3 indicates, applying each enhancement results in higher performance on the development set. By applying all incremental improvements, total $F_{0.5}$ score on the development set increases by 11.8%.

Source word dropout	Target word dropout	MLE	Dev combined $F_{0.5}$
0	0	1	34.20
0.1			37.89
0.2			38.26
	0.1		35.43
	0.2		33.98
		2	34.56
		3	34.28
		4	34.17
0.2	0.1		37.89
0.2		3	38.68
0.2	0.1	3	38.15

Table 4: The effect of source word dropout, target word dropout, and MLE weight on development combined $F_{0.5}$ score.

We improved the $F_{0.5}$ score by adding *source_word_dropout*, *target_word_dropout* and MLE weighting by almost 4%. To find out optimal values of all three hyper-parameters, we ran a small grid search. The results of this experiment are presented in Table 4. The source-word dropout improves the results the most, MLE provides minor gains, while the influence of target-word dropout on the results is unclear.

In the next experiment, we examined the effect of checkpoint averaging. Table 5 presents results of the model without averaging and with averaging 4, 6, and 8 model checkpoints. The best results are achieved when 8 checkpoints are used and the results indicate that the more checkpoints are av-

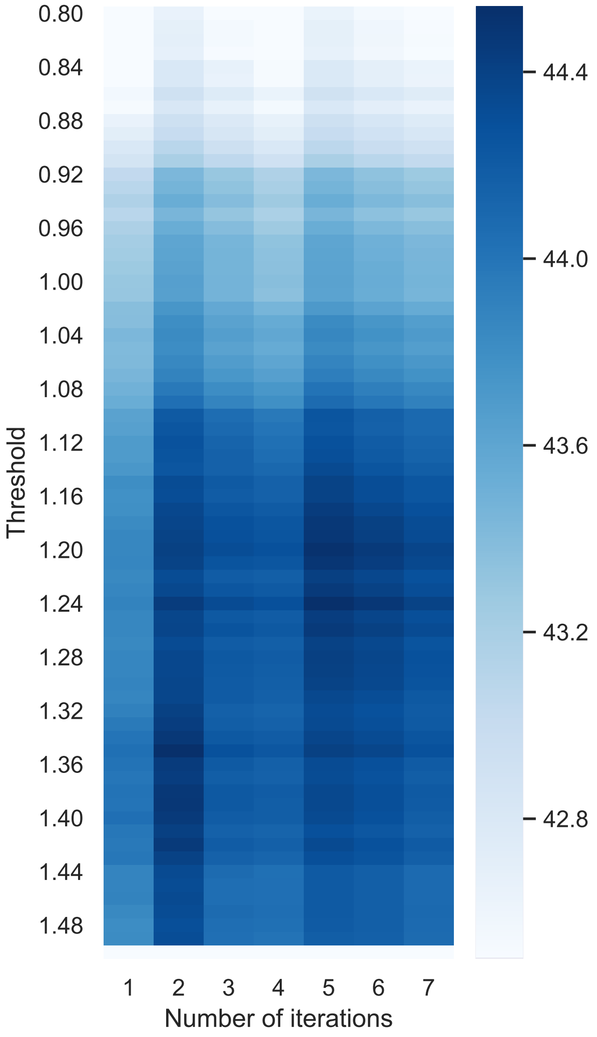


Figure 1: Performance of iterative decoding depending on number of iterations and threshold parameters.

eraged the better the results are.

Finally, we inspect the effect of iterative decoding. Specifically, we run an exhaustive grid search to find optimal values of *threshold* and *max_iters*. The results of this experimented are visualised in Figure 1. We can see that increasing *threshold* from 1 to values around 1.20 leads to substantially better results. Moreover, using more iterations also has a positive impact on the model performance. Both of these improvements are caused by the model generating more corrections which are deemed less likely to the model, i.e., we increase recall at the expense of precision.

4.3 Low-Resource Track

We train following models in the Low-Resource Track:

1. the *transformer_big* configuration with *input_word_dropout* set to 0.2 and *target_word_dropout* to 0.1 – settings similar to the best system in the Restricted Track but without edited MLE;

2. the *transformer_clean_big_tpu* configuration – this configuration uses no internal dropouts;

3. the *transformer_clean_big_tpu* configuration with *input_word_dropout* 0.2 and *target_word_dropout* 0.1;

4. the *transformer_clean_big_tpu* configuration trained on sentences extracted from Wikipedia revisions without introducing additional spelling errors and infillment marker.

Checkpointing	Dev combined $F_{0.5}$
No checkpointing	41.55
Averaging 4 checkpoints	43.00
Averaging 6 checkpoints	43.13
Averaging 8 checkpoints	43.29

Table 5: Maximum development combined $F_{0.5}$ score achieved by averaging the given number of checkpoints.

ID	Model	Dev combined $F_{0.5}$
1	*transformer_big* 0.2 src drop, 0.1 tgt drop	22.03
2	*transformer_clean_big_tpu* no src drop, no tgt drop	26.05
3	*transformer_clean_big_tpu* 0.2 src drop, 0.1 tgt drop	24.80
4	*transformer_clean_big_tpu* no spelling or infillment errors	21.16

Table 6: Development combined $F_{0.5}$ score achieved with different models in the Low-Resource Track.

All but the fourth model use the training data as described in Section 3.2 and the training scheme is in all models identical. The results of all models are presented in Table 6.

The best results are achieved with the second model which performs no dropouts. When we incorporate source and target word dropouts in the third experiment, the performance deteriorates by more than 1%. When we also add Transformer in-

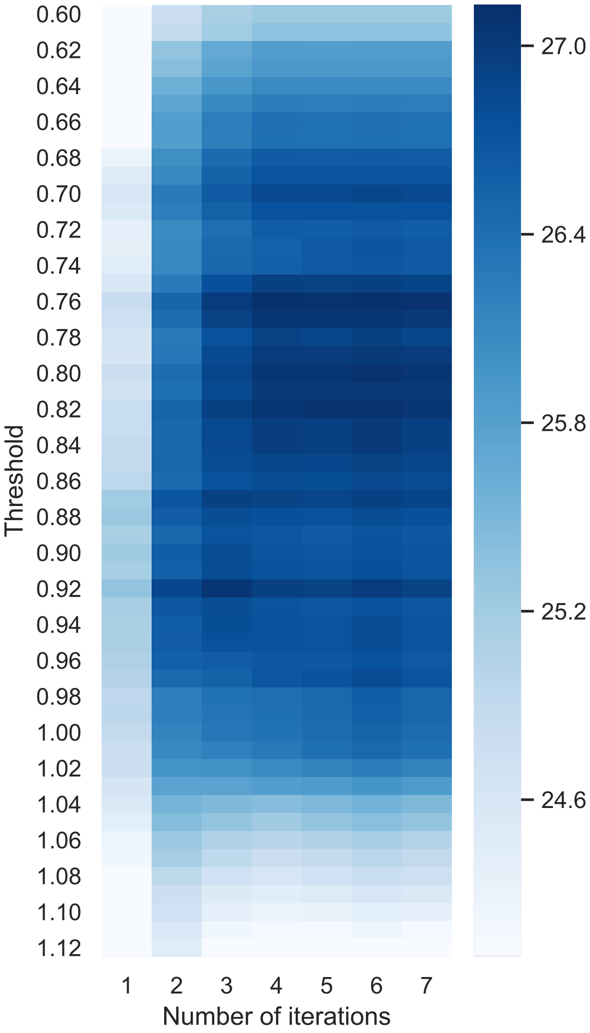


Figure 2: Performance of iterative decoding depending on number of iterations and threshold parameters.

ternal dropouts in the first experiment, the performance drops by additional 2.8%. This confirms our assumption that the enormous amount of data is strong enough regularizer and the usage of additional regularizers leads to worse performance.

The results of the fourth model, which was trained on data without additional spelling and infillment noise, are almost 5% worse than when training on data with this noise. It would be an interesting experiment to evaluate the effect of spelling and infillment noise separately, but this was not done in this paper.

We also run an exhaustive grid search to find optimal values of *threshold* and *max_iters* in iterative decoding. As we can see in Figure 2, the optimal value of *threshold* is now below 1 indicating that precision is now increased at the expense of recall. A performance gain in using more than one iteration is clearly visible.

4.4 Unrestricted Track

In the Unrestricted Track, we tried finetuning the pretrained system with two different learning rate schedules:

- linearly increase learning rate from 0 to 0.011 over the first 8000 steps, then decrease it proportionally to the number of steps after that – exactly same as while training system from scratch in the Restricted Track (see Section 3.1.7);
- linearly increase learning rate from 0 to 3e-4 then keep the learning rate constant as proposed by Lichtarge et al. (2018).

All other hyper-parameters and the training process remain the same as described in Section 3.3.

The first finetuning scheme overfitted the training corpus quite quickly while reaching score of 48.33. The second scheme converged slower and reached a higher score of 48.82.

5 Conclusion

We have presented our three systems submitted to the BEA 2019 Shared Tasks. By employing larger architecture, source and target word dropout, edited MLE, dataset extension, checkpoint averaging, and iterative decoding, our system reached 59.39 $F_{0.5}$ score in the Restricted Track, finishing 10th out of 21 participants.

In the Low Resource Track, we utilized Wikipedia revision edits as a training data, reaching 44.14 $F_{0.5}$ score. Finally, we finetuned this model using the annotated training data, obtaining 65.55 $F_{0.5}$ score in the Unrestricted Track, ranking 3rd out of 7 submissions.

As future work, we would like to explore iterative decoding algorithm more thoroughly. Specifically, we hope that allowing *threshold* parameter to change in each iteration might provide gains. We would also like to train systems on Wikipedia revisions in other languages.

Acknowledgements

The work described herein has been supported by OP VVV VI LINDAT/CLARIN project (CZ.02.1.01/0.0/0.0/16_013/0001781) and it has been supported and has been using language resources developed by the LINDAT/CLARIN project (LM2015071) of the Ministry of Education, Youth and Sports of the Czech Republic.

This research was also partially supported by SVV project number 260 453 and GAUK 578218 of the Charles University.

References

Chris Brockett, William B Dolan, and Michael Gamon. 2006. Correcting esl errors using phrasal smt techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 249–256. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 Shared Task on Grammatical Error Correction. In *Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The nus corpus of learner english. In *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, pages 22–31.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*.

Sylviane Granger. 1998. The computer learner corpus: A versatile new source of data for SLA research. In Sylviane Granger, editor, *Learner English on Computer*, pages 3–18. Addison Wesley Longman, London and New York.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. *arXiv preprint arXiv:1605.06353*.

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*.

Jared Lichtarge, Christopher Alberti, Shankar Kumar, Noam Shazeer, and Niki Parmar. 2018. Weakly supervised grammatical error correction using iterative decoding. *arXiv preprint arXiv:1811.01710*.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning sns for automated japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110(1):43–70.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *arXiv preprint arXiv:1804.04235*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for esl learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 198–202. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 180–189. Association for Computational Linguistics.

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*.

Grammatical Error Correction in Low-Resource Scenarios

Jakub Náplava and **Milan Straka**
Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics
{naplava,straka}@ufal.mff.cuni.cz

Abstract

Grammatical error correction in English is a long studied problem with many existing systems and datasets. However, there has been only a limited research on error correction of other languages. In this paper, we present a new dataset AKCES-GEC on grammatical error correction for Czech. We then make experiments on Czech, German and Russian and show that when utilizing synthetic parallel corpus, Transformer neural machine translation model can reach new state-of-the-art results on these datasets. AKCES-GEC is published under CC BY-NC-SA 4.0 license at http://hdl.handle.net/11234/1-3057, and the source code of the GEC model is available at https://github.com/ufal/low-resource-gec-wnut2019.

1 Introduction

A great progress has been recently achieved in grammatical error correction (GEC) in English. The performance of systems has since CoNLL 2014 shared task (Ng et al., 2014) increased by more than 60% on its test set (Bryant et al., 2019) and also a variety of new datasets appeared. Both rule-based models, single error-type classifiers and their combinations were due to larger amount of data surpassed by statistical and later by neural machine translation systems. These address GEC as a translation problem from a language of ungrammatical sentences to a grammatically correct ones.

Machine translation systems require large amount of data for training. To cope with this issue, different approaches were explored, from acquiring additional corpora (e.g. from Wikipedia edits) to building a synthetic corpus from clean monolingual data. This was apparent on recent Building Educational Applications (BEA) 2019 Shared Task on GEC (Bryant et al., 2019) when top scoring teams extensively utilized synthetic corpora.

The majority of research has been done in English. Unfortunately, there is a limited progress on other languages. Namely, Boyd (2018) created a dataset and presented a GEC system for German, Rozovskaya and Roth (2019) for Russian, Náplava (2017) for Czech and efforts to create annotated learner corpora were also done for Chinese (Yu et al., 2014), Japanese (Mizumoto et al., 2011) and Arabic (Zaghouani et al., 2015).

Our contributions are as follows:

- We introduce a new Czech dataset for GEC. In comparison to dataset of Šebesta et al. (2017) it contains separated edits together with their type annotations in M2 format (Dahlmeier and Ng, 2012) and also has two times more sentences.
- We extend the GEC model of Náplava and Straka (2019) by utilizing synthetic training data, and evaluate it on Czech, German and Russian, achieving state-of-the-art results.

2 Related Work

There are several main approaches to GEC in *low-resource* scenarios. The first one is based on a noisy channel model and consists of three components: a candidate model to propose (word) alternatives, an error model to score their likelihood and a language model to score both candidate (word) probability and probability of a whole new sentence. Richter et al. (2012) consider for a given word all its small modifications (up to character edit distance 2) present in a morphological dictionary. The error model weights every character edit by a trained weight, and three language models (for word forms, lemmas and POS tags) are used to choose the most probable sequence of corrections. A candidate model of Bryant and Briscoe

Proceedings of the 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-generated Text, pages 346–356
Hong Kong, Nov 4, 2019. ©2019 Association for Computational Linguistics

(2018) contains for each word spell-checker proposals, its morphological variants (if found in Automatically Generated Inflection Database) and, if the word is either preposition or article, also a set of predefined alternatives. They assign uniform probability to all changes, but use strong language model to re-rank all candidate sentences. Lacroix et al. (2019) also consider single word edits extracted from Wikipedia revisions.

Other popular approach is to extract parallel sentences from Wikipedia revision histories. A great advantage of such an approach is that the resulting corpus is, especially for English, of great size. However, as Wikipedia edits are not human curated specifically for GEC edits, the corpus is extremely noisy. Grundkiewicz and Junczys-Dowmunt (2014) filter this corpus by a set of regular expressions derived from NUCLE training data and report a performance boost in statistical machine translation approach. Grundkiewicz et al. (2019) filter Wikipedia edits by a simple language model trained on BEA 2019 development corpus. Lichtarge et al. (2019), on the other hand, reports that even without any sophisticated filtering, Transformer (Vaswani et al., 2017) can reach surprisingly good results when used iteratively.

The third approach is to create synthetic corpus from a clean monolingual corpus and use it as additional data for training. Noise is typically introduced either by rule-based substitutions or by using a subset of the following operations: token replacement, token deletion, token insertion, multi-token swap and spelling noise introduction. Yuan and Felice (2013) extract edits from NUCLE and apply them on a clean text. Choe et al. (2019) apply edits from W&I+Locness training set and also define manual noising scenarios for preposition, nouns and verbs. Zhao et al. (2019) use an unsupervised approach to synthesize noisy sentences and allow deleting a word, inserting a random word, replacing a word with random word and also shuffling (rather locally). Grundkiewicz et al. (2019) improve this approach and replace a token with one of its spell-checker suggestions. They also introduce additional spelling noise.

3 Data

In this Section, we present existing corpora for GEC, together with newly released corpus for Czech.

3.1 AKCES-GEC

The AKCES (Czech Language Acquisition Corpora; Šebesta, 2010) is an umbrella project comprising of several acquisition resources – CzeSL (learner corpus of Czech as a second language), ROMi (Romani ethnolect of Czech Romani children and teenagers) and SKRIPT and SCHOLA (written and spoken language collected from native Czech pupils, respectively).

We present the AKCES-GEC dataset, which is a grammar error correction corpus for Czech generated from a subset of AKCES resources. Concretely, the AKCES-GEC dataset is based on CzeSL-man corpus (Rosen, 2016) consisting of manually annotated transcripts of essays of non-native speakers of Czech. Apart from the released CzeSL-man, AKCES-GEC further utilizes additional unreleased parts of CzeSL-man and also essays of Romani pupils with Romani ethnolect of Czech as their first language.

The CzeSL-man annotation consists of three Tiers – Tier 0 are transcribed inputs, followed by the level of orthographic and morphemic corrections, where only word forms incorrect in any context are considered (Tier 1). Finally, the rest of errors is annotated at Tier 2. Forms at different Tiers are manually aligned and can be assigned one or more error types (Jelínek et al., 2012). An example of the annotation is presented in Figure 1, and the list of error types used in CzeSL-man annotation is listed in Table 1.

We generated AKCES-GEC dataset using the three Tier annotation of the underlying corpus. We employed Tier 0 as source texts, Tier 2 as corrected texts, and created error edits according to the manual alignments, keeping error annotations where available.[1] Considering that the M2 format (Dahlmeier and Ng, 2012) we wanted to use does not support non-local error edits and therefore cannot efficiently encode word transposition on long distances, we decided to consider word swaps over at most 2 correct words a single edit (with the constant 2 chosen according to the coverage of long-range transpositions in the data). For illustration, see Figure 2.

The AKCES-GEC dataset consists of an explicit train/development/test split, with each set divided into foreigner and Romani students; for de-

[1] The error annotations are unfortunately not available in the whole underlying corpus, and not all errors are annotated with at least one label.

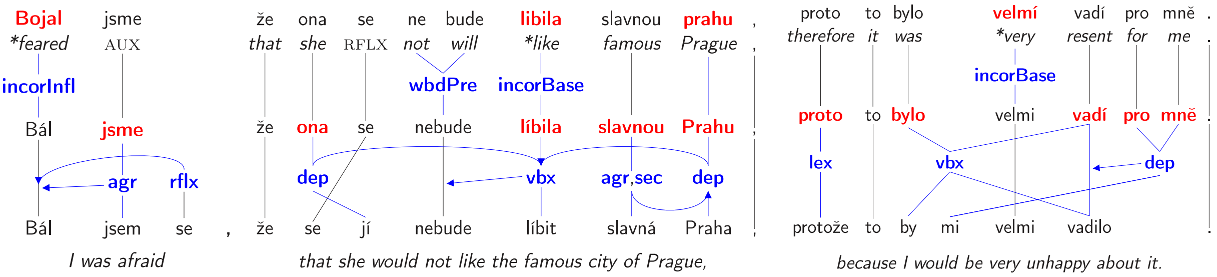

Bojal jsme že ona se ne bude libila slavnou prahu , proto to bylo velmí vadí pro mně .
*feared AUX that she RFLX not will *like famous Prague , therefore it was *very resent for me .
incorInfl wbdPre incorBase incorBase
Bál jsme že ona se nebude líbila slavnou Prahu , proto to bylo velmi vadí pro mně .
agr rflx dep vbx agr,sec dep lex vbx dep
Bál jsem se , že se jí nebude líbit slavná Praha , protože to by mi velmi vadilo .
I was afraid
that she would not like the famous city of Prague,
because I would be very unhappy about it.

Figure 1: Example of two-level annotation of a sentence in CzeSL corpus, reproduced from (Rosen, 2016).

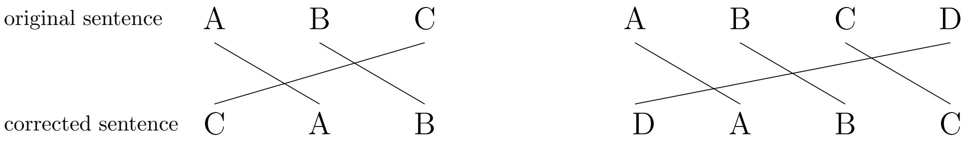


Figure 2: Word swap over one or two correct words (on the left) is considered a single edit (A B C→ C A B). Word swap over more than two correct words (on the right) is represented as two edits of deleting D and inserting D.

Error type	Description	Example	Occ
incorInfl	incorrect inflection	[**pracovají** → *pracují*] v továrně	8 986
incorBase	incorrect word base	musíš to [**posvětlit** → *posvětit*]	20 334
fwFab	non-emendable, „fabricated“ word	pokud nechceš slyšet [**smášky**]	78
fwNC	foreign word	váza je na [**Tisch** → *stole*]	166
flex	supplementary flag used with fwFab and fwNC marking the presence of inflection	jdu do [**shopa** → *obchodu*]	34
wbdPre	prefix separated by a space or preposition w/o space	musím to [**při pravit** → *připravit*]	817
wbdComp	wrongly separated compound	[**český anglický** → *česko-anglický*] slovník	92
wbdOther	other word boundary error	[**mocdobře** → *moc dobře*]; [**atak** → *a tak*]	1326
stylColl	colloquial form	[**dobrej** → *dobrý*] film	3 533
stylOther	bookish, dialectal, slang, hyper-correct form	holka s [**hnědými očimi** → *hnědýma očima*]	156
agr	violated agreement rules	to jsou [**hezké** → *hezcí*] chlapci; Jana [**čtu** → *čte*]	5 162
dep	error in valency	bojí se [**pes** → *psa*]; otázka [**čas** → *času*]	6 733
ref	error in pronominal reference	dal jsem to jemu i [**jejího** → *jeho*] bratrovi	344
vbx	error in analytical verb form or compound predicate	musíš [**přijdeš** → *přijít*]; kluci [**jsou**] běhali	864
rflx	error in reflexive expression	dívá [∅ → *se*] na televizi; Pavel [**si** → *se*] raduje	915
neg	error in negation	[**půjdu ne** → *nepůjdu*] do školy	111
lex	error in lexicon or phraseology	dopadlo to [**přírodně** → *přirozeně*]	3 967
use	error in the use of a grammar category	pošta je [**nejvíc blízko** → *nejblíže*]	1 458
sec	secondary error (supplementary flag)	stará se o [**našich holčičkách** → *naše holčičky*]	866
stylColl	colloquial expression	viděli jsme [**hezký** → *hezké*] holky	3 533
stylOther	bookish, dialectal, slang, hyper-correct expression	rozbil se mi [**hadr**]	156
stylMark	redundant discourse marker	[**no**]; [**teda**]; [**jo**]	15
disr	disrupted construction	známe [**hodné spoustu** → *spoustu hodných*] lidí	64
problem	supplementary label for problematic cases		175
unspec	unspecified error type		69 123

Table 1: Error types used in CzeSL corpus taken from (Jelínek et al., 2012), including number of occurrences in the dataset being released. Tier 1 errors are in the upper part of the table, Tier 2 errors are in the lower part. The *stylColl* and *stylOther* are annotated on both Tiers, but we do not distinguish on which one in the AKCES-GEC.

velopment and test sets, the foreigners are further split into Slavic and non-Slavic speakers. Furthermore, the development and test sets were annotated by two annotators, so we provide two references if the annotators utilized the same sentence segmentation and produced different annotations.

The detailed statistics of the dataset are presented in Table 2. The AKCES-GEC dataset is released under the CC BY-NC-SA 4.0 license at http://hdl.handle.net/11234/1-3057.

We note that there already exists a CzeSL-GEC dataset (Šebesta et al., 2017). However, it consists only of a subset of data and does not contain error types nor M2 files with individual edits.

3.2 English

Probably the largest corpus for English GEC is the Lang-8 Corpus of Learner English (Mizumoto

		Train				Dev				Test			
		Doc	Sent	Word	Error r.	Doc	Sent	Word	Error r.	Doc	Sent	Word	Error r.
Foreign.	Slavic	1 816	27 242	289 439	22.2 %	70	1 161	14 243	21.8 %	69	1 255	14 984	18.8 %
	Other					45	804	8 331	23.8 %	45	879	9 624	20.5 %
Romani		1 937	14 968	157 342	20.4 %	80	520	5 481	21.0 %	74	542	5 831	17.8 %
Total		3 753	42 210	446 781	21.5 %	195	2 485	28 055	22.2 %	188	2 676	30 439	19.1 %

Table 2: Statistics of the AKCES-GEC dataset – number of documents, sentences, words and error rates.

et al., 2011; Tajiri et al., 2012). It comes from an online language learning website, where users are able to post texts in language they are learning. These texts then appear to native speakers for correction. The corpus has over 100 000 raw English entries comprising of more than 1M sentences. Due to the fact that texts are corrected by online users, this corpus is also quite noisy.

Other corpora are corrected by trained annotators making them much cleaner but also significantly smaller. NUCLE (Dahlmeier et al., 2013) has 57 151 sentences originating from 1 400 essays written by mainly Asian undergraduate students at the National University of Singapore. FCE (Yannakoudakis et al., 2011) is a subset of the Cambridge Learner Corpus (CLC) and has 33 236 sentences from 1 244 written answers to FCE exam questions. Recent Write & Improve (W&I) and LOCNESS v2.1 (Bryant et al., 2019; Granger, 1998) datasets were annotated for different English proficiency levels and a part of them also comes from texts written by native English speakers. Altogether, it has 43 169 sentences.

To evaluate system performance, CoNLL-2014 test set is most commonly used. It comprises of 1 312 sentences written by 25 South-East Asian undergraduates. The gold annotations are matched against system hypothesis using MaxMatch scorer outputting $F_{0.5}$ score. The other frequently used dataset is JFLEG (Napoles et al., 2017; Heilman et al., 2014), which also tests systems for how fluent they sound by utilizing the GLEU metric (Napoles et al., 2015). Finally, recent W&I and LOCNESS v2.1 test set allows to evaluate systems on different levels of proficiency and also against different error types (utilizing ERRANT scorer).

3.3 German

Boyd (2018) created GEC corpus for German from two German learner corpora: Falko and MERLIN (Boyd et al., 2014). The resulting dataset comprises of 24 077 sentences divided into training, development and test set in the ratio of 80:10:10. To evaluate system performance, MaxMatch scorer is used.

Apart from creating the dataset, Boyd (2018) also extended ERRANT for German. She defined 21 error types (15 based on POS tags) and extended spaCy[2] pipeline to classify them.

3.4 Russian

Rozovskaya and Roth (2019) introduced RULEC-GEC dataset for Russian GEC. To create this dataset, a subset of RULEC corpus with foreign and heritage speakers was corrected. The final dataset has 12 480 sentences annotated with 23 error tags. The training, development and test sets contain 4 980, 2 500 and 5 000 sentence pairs, respectively.

3.5 Corpora Statistics

Table 3 indicates that there is a variety of English datasets for GEC. As Náplava and Straka (2019) show, training Transformer solely using these annotated data gives solid results. On the other hand, there is only limited number of data for Czech, German and Russian and also the existing systems perform substantially worse. This motivates our research in these low-resource languages.

Table 3 also presents an average error rate of each corpus. It is computed using maximum alignment of original and annotated sentences as a ratio of non-matching alignment edges (insertion, deletion, and replacement). The highest error rate of 21.4 % is on Czech dataset. This implies that circa every fifth word contains an error. German is also quite noisy with an error rate of 16.8 %. The average error rate on English ranges from 6.6 % to 14.1 % and, finally, the Russian corpus contains the least errors with an average error rate of 6.4%.

[2] https://spacy.io/

Language	Corpus	Sentences	Err. r.
English	Lang-8	1 147 451	14.1%
	NUCLE	57 151	6.6%
	FCE	33 236	11.5%
	W&I+LOCNESS	43 169	11.8%
Czech	AKCES-GEC	42 210	21.4%
German	Falko-MERLIN	24 077	16.8%
Russian	RULEC-GEC	12 480	6.4%

Table 3: Statistics of available corpora for Grammatical Error Correction.

3.6 Tokenization

The most popular metric for benchmarking systems are MaxMatch scorer (Dahlmeier and Ng, 2012) and ERRANT scorer (Bryant et al., 2017). They both require data to be tokenized; therefore, most of the GEC datasets are tokenized.

To tokenize monolingual English and German data, we use spaCy v1.9.0 tokenizer utilizing *en_core_web_sm-1.2.0* and *de* model. We use custom tokenizers for Czech[3] and Russian[4].

4 System Overview

We use neural machine translation approach to GEC. Specifically, we utilize Transformer model (Vaswani et al., 2017) to translate ungrammatical sentences to grammatically correct ones. We further follow Náplava and Straka (2019) and employ source and target word dropouts, edit-weighted MLE and checkpoint averaging. We do not use iterative decoding in this work, because it substantially slows down decoding. Our models are implemented in Tensor2Tensor framework version 1.12.0.[5]

4.1 Pretraining on Synthetic Dataset

Due to the limited number of annotated data in Czech, German and Russian we decided to create a corpus of synthetic parallel sentences. We were also motivated by the fact that such approach was shown to improve performance even in English with substantially more annotated training data.

We follow Grundkiewicz et al. (2019), who use an unsupervised approach to create noisy input sentences. Given a clean sentence, they sample a probability p_{err_word} from a normal distribution with a predefined mean and a standard deviation. After multiplying p_{err_word} by a number of words in the sentence, as many sentence words are selected for modification. For each chosen word, one of the following operations is performed with a predefined probability: substituting the word with one of its ASpell[6] proposals, deleting it, swapping it with its right-adjacent neighbour or inserting a random word from dictionary after the current word. To make the system more robust to spelling errors, same operations are also used on individual characters with p_{err_char} sampled from a normal distribution with a different mean and standard deviation than p_{err_word} and (potentially) different probabilities of character operations.

When we inspected the results of a model trained on such dataset in Czech, we observed that the model often fails to correct casing errors and sometimes also errors in diacritics. Therefore, we extend word-level operations to also contain operation to change casing of a word. If a word is chosen for modification, it is with 50% probability whole converted to lower-case, or several individual characters are chosen and their casing is inverted. To increase the number of errors in diacritics, we add a new character-level noising operation, which for a selected character either generates one of its possible diacritized variants or removes diacritics. Note that this operation is performed only in Czech.

We generate synthetic corpus for each language from WMT News Crawl monolingual training data (Bojar et al., 2017). We set p_{err_word} to 0.15, p_{err_char} to 0.02 and estimate error distributions of individual operations from development sets of each language. The constants used are presented in Table 4. We limited amount of synthetic sentences to 10M in each language.

4.2 Finetuning

A model is (pre-)trained on a synthetic dataset until convergence. Afterwards, we finetune the model on a mix of original language training data and synthetic data. When finetuning the model, we preserve all hyperparameters (e.g., learning rate and optimizer moments). In other words, the training continues and only the data are replaced.

When finetuning, we found that it is crucial to preserve some portion of synthetic data in the training corpus. Finetuning with original training

[3] A slight modification of MorphoDiTa tokenizer.

[4] https://github.com/aatimofeev/spacy_russian_tokenizer

[5] https://github.com/tensorflow/tensor2tensor

[6] http://aspell.net/

Language	Token-level operations					Character-level operations				
	sub	ins	del	swap	recase	sub	ins	del	recase	toggle diacritics
English	0.6	0.2	0.1	0.05	0.05	0.25	0.25	0.25	0.25	0
Czech	0.7	0.1	0.05	0.1	0.05	0.2	0.2	0.2	0.2	0.2
German	0.64	0.2	0.1	0.01	0.05	0.25	0.25	0.25	0.25	0
Russian	0.65	0.1	0.1	0.1	0.05	0.25	0.25	0.25	0.25	0

Table 4: Language specific constants for token- and character-level noising operations.

data leads to fast overfitting with worse results on all of Czech, German and Russian. We also found out that it also slightly helps on English.

We ran a small grid-search to estimate the ratio of synthetic versus original sentences in the finetuning phase. Although the ratio of 1:2 (5M original oversampled training pairs and 10M synthetic pairs) still overfits, we found it to work best for English, Czech and German, and stop training when the performance on the development set starts deteriorating. For Russian, the ratio of 1:20 (0.5M oversampled training pairs and 10M synthetic pairs) works the best.

The original sentences for English finetuning are concatenated sentences from Lang-8 Corpus of Learner English, FCE, NUCLE and W&I and LOCNESS. To better match domain of test data, we oversampled training set by adding W&I training data 10 times, FCE data 5 times and NUCLE corpus 5 times to the training set. The original sentences in Czech, German and Russian are the training data of the corresponding languages.

4.3 Implementation Details

When running grid search for hyperparameter tuning, we use *transformer_base_single_gpu* configuration, which uses only 1 GPU to train *Transformer Base* model. After we select all hyperparameter, we train *Transformer Big* architecture on 4 GPUs. Hyperparameters described in following paragraphs belong to both architectures.

We use Adafactor optimizer (Shazeer and Stern, 2018), linearly increasing the learning rate from 0 to 0.011 over the first 8000 steps, then decrease it proportionally to the number of steps after that (using the `rsqrt_decay` schedule). Note that this only applies to the pre-training phase.

All systems are trained on Nvidia P5000 GPUs. The vocabulary consists of approximately 32k most common word-pieces, the batch size is 2000 word-pieces per each GPU and all sentences with more than 150 word-pieces are discarded during training. Model checkpoints are saved every hour.

At evaluation time, we decode using a beam size of 4. Beam-search length-balance decoding hyperparameter alpha is set to 0.6.

5 Results

We present results of our model when trained on English, Czech, German and Russian in this Section. As we are aware of only one system in German, Czech and Russian to compare with, we start with English model discussion. We show that our model is on par or even slightly better than current state-of-the-art systems in English when no ensembles are allowed. We then discuss our results on other languages, where our system exceeds all existing systems by a large margin.

In all experiments, we report results of three systems: *synthetic pretrain*, which is based on Transformer Big and is trained using synthetic data only, and *finetuned* and *finetuned base single GPU*, which are based on Transformer Big and Base, respectively, and are both pretrained and finetuned. Note that even if the *finetuned base* system has 3 times less parameters than *finetuned*, its results on some languages are nearly identical.

We also tried training the system using annotated data only. With our model architecture, all but English experiments (which contain substantially more data) starts overfitting quickly, yielding poor performance. The overfitting problem could be possibly addressed as proposed by Sennrich and Zhang (2019). Nevertheless, given that our best system on English is by circa 10 points in $F_{0.5}$ score better than the system trained solely on annotated data, we focused primarily on the synthetic data experiments.

Apart from the W&I+L development and test sets, which are evaluated using ERRANT scorer, we use MaxMatch scorer in all experiments.

System	W&I+L test	W&I+L dev	CoNLL 14 test	
			No W&I+L	With W&I+L
including ensembles				
Lichtarge et al. (2019)	–	–	60.40	–
Zhao et al. (2019)	–	–	61.15	–
Xu et al. (2019)	67.21	55.37	–	63.20
Choe et al. (2019)	69.06	52.79	57.50	–
Grundkiewicz et al. (2019)	**69.47**	53.00	**61.30**	**64.16**
no ensembles				
Lichtarge et al. (2019)	–	–	**56.80**	–
Xu et al. (2019)	**63.94**	52.29	–	60.90
Choe et al. (2019)	63.05	47.75	–	–
Grundkiewicz et al. (2019)	–	50.01	–	–
no ensembles				
Our work – synthetic pretrain	51.16	32.76	41.85	44.12
Our work – finetuned base single GPU	67.18	52.80	59.87	–
Our work – finetuned	69.00	53.30	60.76	63.40

Table 5: Comparison of systems on two English GEC datasets. CoNLL 2014 Test Set is divided into two system groups (columns): those who do not train on W&I+L training data and those who do.

System	P	R	$F_{0.5}$
Boyd (2018)	51.99	29.73	45.22
Our work – synthetic pretrain	67.45	26.35	51.41
Our work – finetuned base single GPU	78.11	59.13	73.40
Our work – finetuned	78.21	59.94	73.71

Table 6: Results on on Falko-Merlin Test Set (German).

5.1 English

We provide comparison between our model and existing systems on W&I+L test and development sets and on CoNLL 14 test set in Table 5. Even if the results on the W&I+L development set are only partially indicative of system performance, we report them due to the W&I+L test set being blind. All mentioned papers do not train their systems on the development set, but use it only for model selection. Also note that we split the results on CoNLL 14 test set into two groups: those who do not use the W&I+L data for training, and those who do. This is to allow a fair comparison, given that the W&I+L data were not available before the BEA 2019 Shared Task on GEC.

The best performing systems are utilizing ensembles. Table 5 shows an evident performance boost (3.27-6.01 points) when combining multiple models into an ensemble. The best performing system on English is an ensemble system of Grundkiewicz et al. (2019).

The aim of this paper is to concentrate on low-resource languages rather than on English. Therefore, we report results of our single model. Despite that our best system reaches 69.0 $F_{0.5}$ score, which is comparable to the performance of best systems that employ ensembles. Although Grundkiewicz et al. (2019) do not report their single system score, we can hypothesise that given development set scores, our system is on par with theirs or even performs slightly better.

Note that there is a significant difference between results reported on W&I+L dev and W&I+L test sets. This is caused by the fact that each sentence in the W&I+L test set was annotated by 10 annotators, while there is only a single annotator for each sentence in the development set.

5.2 German

Boyd (2018) developed a GEC system for German based on multilayer convolutional encoder-decoder neural network (Chollampatt and Ng, 2018). To account for the lack of annotated

System	Test Subset	P	R	$F_{0.5}$
Richter et al. (2012)	All	68.72	36.75	58.54
Our work – synthetic pretrain	All	80.32	39.55	66.59
Our work – finetuned base single GPU	All	84.21	66.67	80.00
Our work – finetuned	Foreigners – Slavic	84.34	71.55	81.43
	Foreigners – Other	81.03	62.36	76.45
	Romani	86.61	71.13	83.00
	All	83.75	68.48	80.17

Table 7: Results on on AKCES-GEC Test Set (Czech).

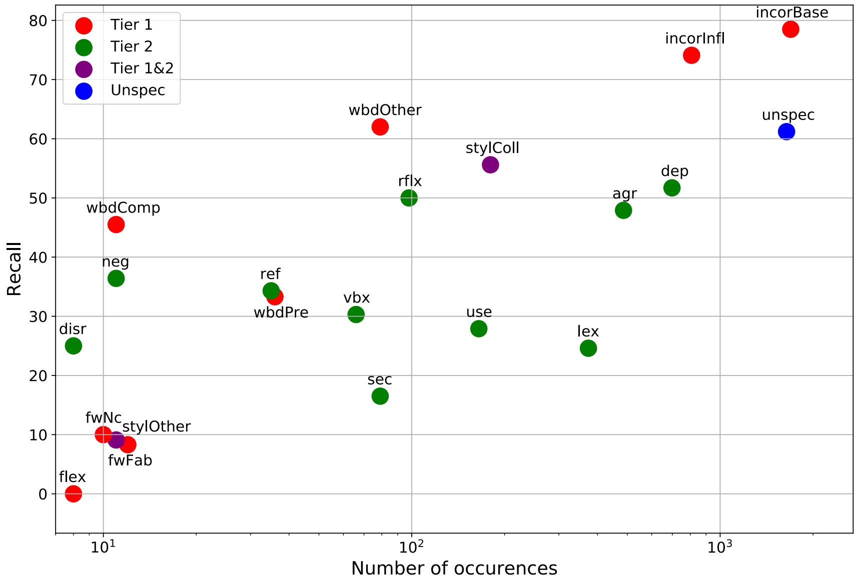


Figure 3: Recall for each error type in the test set of AKCES-GEC, computed using the first annotator (ID 0).

data, she generated additional training data from Wikipedia edits, which she filtered to match the distribution of the original error types. As Table 6 shows, her best system reaches 45.22 $F_{0.5}$ score on Falko-Merlin test set. All our three systems outperform it.

Compared to Boyd (2018), our system trained solely on synthetic data has lower recall, but substantially higher precision. The main reason behind the lower recall is the unsupervised approach to synthetic data generation. Both our finetuned models outperform Boyd (2018) system by a large margin.

5.3 Czech

We compare our system with Richter et al. (2012), who developed a statistical spelling corrector for Czech. Although their system can only make local changes (e.g., cannot insert a new word or swap two nearby words), it achieves surprisingly solid results. Nevertheless, all our three system perform better in both precision, recall and $F_{0.5}$ score. Possibly due to already quite high precision of the pretrained model, the finetuning stage improves mainly model recall.

We also evaluate performance of our best system on three subsets of the AKCES-GEC test set: Foreigners–Slavic, Foreigners–Other and Romani. As the name suggests, the first of them is a part of AKCES-GEC collected from essays of non-Czech Slavic people, the second from essays of non-Czech non-Slavic people and finally Romani comes from essays of Romani pupils with Romani ethnolect of Czech as their first language. The best result is reached on Romani subset, while on Foreigners–Other the $F_{0.5}$ score is by more than 6 points lower. We hypothesize this effect is caused by the fact, that Czech is the primary language of Romani pupils. Furthermore, we presume that foreigners with Slavic background should learn Czech faster than non-Slavic foreigners, because of the similarity between their mother tongue and

System	P	R	$F_{0.5}$
Rozovskaya and Roth (2019)	38.0	7.5	21.0
Our work – synthetic pretrain	47.76	26.08	40.96
Our work – finetuned base single GPU	59.13	26.05	47.15
Our work – finetuned	63.26	27.50	50.20

Table 8: Results on on RULEC-GEC Test Set (Russian).

Czech. This fact is supported by Table 2, which shows that the average error rate of Romani development set is 21.0%, Foreigners–Slavic 21.8% and the Foreigners–Other 23.8%.

Finally, we report recall of the best system on each error type annotated by the first annotator (ID 0) in Figure 3. Generally, our system performs better on errors annotated on Tier 1 than on errors annotated on Tier 2. Furthermore, a natural hypothesis is that the more occurrences there are for an error type, the better the recall of the system on the particular error type. Figure 3 suggests that this hypothesis seems plausible on Tier 1 errors, but its validity is unclear on Tier 2.

5.4 Russian

As Table 8 indicates, GEC in Russian currently seems to be the most challenging task. Although our system outperforms the system of Rozovskaya and Roth (2019) by more than 100% in $F_{0.5}$ score, its performance is still quite poor when compared to all previously described languages. Because the result of our system trained solely on synthetic data is comparable with the similar system for English, we hypothesise that the main reason behind these poor results is the small amount of annotated training data – while Czech has 42 210 and German 19 237 training sentence pairs, there are only 4 980 sentences in the Russian training set. To validate this hypothesis, we extended the original training set by 2 000 sentences from the development set, resulting in an increase of 3 percent points in $F_{0.5}$ score.

6 Conclusion

We presented a new dataset for grammatical error correction in Czech. It contains almost twice as much sentences as existing German dataset and more than three times as RULEC-GEC for Russian. The dataset is published in M2 format containing both separated edits and their error types.

Furthermore, we performed experiments on three low-resource languages: German, Russian and Czech. For each language, we pretrained Transformer model on synthetic data and finetuned it with a mixture of synthetic and authentic data. On all three languages, the performance of our system is substantially higher than results of the existing reported systems. Moreover, all our models supersede reported systems even if only pretrained on unsupervised synthetic data.

The performance of our system could be even higher if we trained multiple models and combined them into an ensemble. We plan to do that in future work. We also plan to extend our synthetic corpora with data modified by supervisedly extracted rules. We hope that this could help especially in case of Russian, which has the lowest amount of training data.

Acknowledgments

The work described herein has been supported by OP VVV VI LINDAT/CLARIN project (CZ.02.1.01/0.0/0.0/16_013/0001781) and it has been supported and has been using language resources developed by the LINDAT/CLARIN project (LM2015071) of the Ministry of Education, Youth and Sports of the Czech Republic. This research was also partially supported by SVV project number 260 453, GAUK 578218 of the Charles University and FP7-ICT-2010-6-257528 (MŠMT 7E11042).

References

Ondřej Bojar, Chatterjee Rajen, Christian Federmann, Yvette Graham, Barry Haddow, Huck Matthias, Koehn Philipp, Liu Qun, Logacheva Varvara, Monz Christof, et al. 2017. Findings of the 2017 conference on machine translation (wmt17). In *Second Conference onMachine Translation*, pages 169–214. The Association for Computational Linguistics.

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.

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

Christopher Bryant and Ted Briscoe. 2018. Language model based grammatical error correction without annotated training data. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 247–253.

Christopher Bryant, Mariano Felice, Øistein E Andersen, and Ted Briscoe. 2019. The bea-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75.

Christopher Bryant, Mariano Felice, and Edward John Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. 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*.

Shamil Chollampatt and Hwee Tou Ng. 2018. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572. Association for Computational Linguistics.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The nus corpus of learner english. In *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, pages 22–31.

Sylviane Granger. 1998. The computer learner corpus: A versatile new source of data for SLA research. In Sylviane Granger, editor, *Learner English on Computer*, pages 3–18. Addison Wesley Longman, London and New York.

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.

Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. 2014. Predicting grammaticality on an ordinal scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 174–180, Baltimore, Maryland. Association for Computational Linguistics.

Tomáš Jelínek, Barbora Štindlová, Alexandr Rosen, and Jirka Hana. 2012. Combining manual and automatic annotation of a learner corpus. In *Text, Speech and Dialogue*, pages 127–134, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ophélie Lacroix, Simon Flachs, and Anders Søgaard. 2019. Noisy channel for low resource grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 191–196.

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*.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. 2011. Mining revision log of language learning sns for automated japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155.

Jakub Náplava. 2017. Natural language correction. Diploma thesis, Univerzita Karlova, Matematicko-fyzikální fakulta.

Jakub Náplava and Milan Straka. 2019. Cuni system for the building educational applications 2019 shared task: Grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 183–190.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. 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)*, pages 588–593.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. 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*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14.

Michal Richter, Pavel Straňák, and Alexandr Rosen. 2012. Korektor–a system for contextual spellchecking and diacritics completion. In *Proceedings of COLING 2012: Posters*, pages 1019–1028.

Alexandr Rosen. 2016. Building and using corpora of non-native Czech. In *Proceedings of the 16th ITAT: Slovenskočeský NLP workshop (SloNLP 2016)*, volume 1649 of *CEUR Workshop Proceedings*, pages 80–87, Bratislava, Slovakia. Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, CreateSpace Independent Publishing Platform.

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.

Karel Šebesta, Zuzanna Bedřichová, Kateřina Šormová, Barbora Štindlová, Milan Hrdlička, Tereza Hrdličková, Jiří Hana, Vladimír Petkevič, Tomáš Jelínek, Svatava Škodová, Petr Janeš, Kateřina Lundáková, Hana Skoumalová, Šimon Sládek, Piotr Pierscieniak, Dagmar Toufarová, Milan Straka, Alexandr Rosen, Jakub Náplava, and Marie Poláčková. 2017. CzeSL grammatical error correction dataset (CzeSL-GEC). http://hdl.handle.net/11234/1-2143 LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. *arXiv preprint arXiv:1905.11901*.

Noam Shazeer and Mitchell Stern. 2018. Adafactor: Adaptive learning rates with sublinear memory cost. *arXiv preprint arXiv:1804.04235*.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. Tense and aspect error correction for esl learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 198–202. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Karel Šebesta. 2010. Korpusy češtiny a osvojování jazyka [Corpora of Czech and language acquisition]. In *Studie z aplikované lingvistiky [Studies in Applied Linguistics]*, volume 2010(2), pages 11–33.

Shuyao Xu, Jiehao Zhang, Jin Chen, and Long Qin. 2019. Erroneous data generation for grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 180–189. Association for Computational Linguistics.

Liang-Chih Yu, Lung-Hao Lee, and Li-Ping Chang. 2014. Overview of grammatical error diagnosis for learning chinese as a foreign language. In *Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications*, pages 42–47.

Zheng Yuan and Mariano Felice. 2013. Constrained grammatical error correction using statistical machine translation. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 52–61.

Wajdi Zaghouani, Behrang Mohit, Nizar Habash, Ossama Obeid, Nadi Tomeh, Alla Rozovskaya, Noura Farra, Sarah Alkuhlani, and Kemal Oflazer. 2015. Large scale arabic error annotation: Guidelines and framework.

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*.

Czech Grammar Error Correction with a Large and Diverse Corpus

Jakub Náplava[†] **Milan Straka**[†] **Jana Straková**[†] **Alexandr Rosen**[‡]

[†]Charles University, Faculty of Mathematics and Physics
Institute of Formal and Applied Linguistics
{naplava,straka,strakova}@ufal.mff.cuni.cz

[‡]Charles University, Faculty of Arts
Institute of Theoretical and Computational Linguistics
alexandr.rosen@ff.cuni.cz

Abstract

We introduce a large and diverse Czech corpus annotated for grammatical error correction (GEC) with the aim to contribute to the still scarce data resources in this domain for languages other than English. The *Grammar Error Correction Corpus for Czech* (*GECCC*) offers a variety of four domains, covering error distributions ranging from high error density essays written by non-native speakers, to website texts, where errors are expected to be much less common. We compare several Czech GEC systems, including several Transformer-based ones, setting a strong baseline to future research. Finally, we meta-evaluate common GEC metrics against human judgements on our data. We make the new Czech GEC corpus publicly available under the CC BY-SA 4.0 license at http://hdl.handle.net/11234/1-4639.

1 Introduction

Representative data both in terms of size and domain coverage are vital for NLP systems development. However, in the field of grammar error correction (GEC), most GEC corpora are limited to corrections of mistakes made by foreign or second language learners even in the case of English (Tajiri et al., 2012; Dahlmeier et al., 2013; Yannakoudakis et al., 2011, 2018; Ng et al., 2014; Napoles et al., 2017). At the same time, as recently pointed out by Flachs et al. (2020), learner corpora are only a part of the full spectrum of GEC applications. To alleviate the skewed perspective, the authors released a corpus of website texts.

Despite recent efforts aimed to mitigate the notorious shortage of national GEC-annotated corpora (Boyd, 2018; Rozovskaya and Roth, 2019; Davidson et al., 2020; Syvokon and Nahorna, 2021; Cotet et al., 2020; Náplava and Straka, 2019), the lack of adequate data is even more acute in languages other than English. We aim to address both the issue of scarcity of non-English data and the ubiquitous need for broad domain coverage by presenting a new, large and diverse Czech corpus, expertly annotated for GEC.

Grammar Error Correction Corpus for Czech (*GECCC*) includes texts from multiple domains in a total of 83 058 sentences, being, to our knowledge, the largest non-English GEC corpus, as well as being one of the largest GEC corpora overall.

In order to represent a diversity of writing styles and origins, besides essays of both native and non-native speakers from Czech learner corpora, we also scraped website texts to complement the learner domain with supposedly lower error density texts, encompassing a representation of the following four domains:

- *Natives Formal* – essays written by native students of elementary and secondary schools
- *Natives Web Informal* – informal website discussions
- *Romani* – essays written by children and teenagers of the Romani ethnic minority
- *Second Learners* – essays written by non-native learners

Using the presented data, we compare several state-of-the-art Czech GEC systems, including some Transformer-based.

Finally, we conduct a meta-evaluation of GEC metrics against human judgements to select the most appropriate metric for evaluating corrections on the new dataset. The analysis is performed across domains, in line with Napoles et al. (2019).

Our contributions include (i) a **large and diverse Czech GEC corpus**, covering learner corpora and website texts, with unified and, in some domains, completely new GEC annotations, (ii) a **comparison of Czech GEC systems**, and (iii) a **meta-evaluation of common GEC metrics** against human judgement on the released corpus.

2 Related Work

2.1 Grammar Error Correction Corpora

Until recently, attention has been focused mostly on English, while GEC data resources for other languages were in short supply. Here we list a few examples of English GEC corpora, collected mostly within an English-as-a-second-language (ESL) paradigm. For a comparison of their relevant statistics see Table 1.

Lang-8 Corpus of Learner English (Tajiri et al., 2012) is a corpus of English language learner texts from the Lang-8 social networking system.

NUCLE (Dahlmeier et al., 2013) consists of essays written by undergraduate students of the National University of Singapore.

FCE (Yannakoudakis et al., 2011) includes short essays written by non-native learners for the Cambridge ESOL First Certificate in English.

W&I+LOCNESS is a union of two datasets, the *W&I (Write & Improve)* dataset (Yannakoudakis et al., 2018) of non-native learners' essays, complemented by the *LOCNESS* corpus (Granger, 1998), a collection of essays written by native English students.

The GEC error annotations for the learner corpora above were distributed with the BEA-2019 Shared Task on Grammatical Error Correction (Bryant et al., 2019).

The *CoNLL-2014* shared task test set (Ng et al., 2014) is often used for GEC systems evaluation. This small corpus consists of 50 essays written by 25 South-East Asian undergraduates.

JFLEG (Napoles et al., 2017) is another frequently used GEC corpus with fluency edits in addition to usual grammatical edits.

To broaden the restricted variety of domains, focused primarily on learner essays, a *CWEB* collection (Flachs et al., 2020) of website texts was recently released, aiming at contributing lower error density data.

AESW (Daudaravicius et al., 2016) is a large corpus of scientific writing (over 1M sentences), edited by professional editors.

Finally, Napoles et al. (2019) recently released *GMEG*, a corpus for the evaluation of GEC metrics across domains.

Grammatical error correction corpora for languages other than English are less common and – if available – usually limited in size and domain: German *Falko-MERLIN* (Boyd, 2018), Russian *RULEC-GEC* (Rozovskaya and Roth, 2019), Spanish *COWS-L2H* (Davidson et al., 2020), Ukrainian *UA-GEC* (Syvokon and Nahorna, 2021) and Romanian *RONACC* (Cotet et al., 2020).

To better account for multiple correction options, datasets often contain several reference sentences for each original noisy sentence in the test set, proposed by multiple annotators. As we can see in Table 1, the number of annotations typically ranges between 1 and 5 with an exception of the CoNLL14 test set, which – on top of the official 2 reference corrections – later received 10 annotations from Bryant and Ng (2015) and 8 alternative annotations from Sakaguchi et al. (2016).

2.2 Czech Learner Corpora

By the early 2010s, Czech was one of a few languages other than English to boast a series of learner corpora, compiled under the umbrella project *AKCES*, evoking the concept of *acquisition corpora* (Šebesta, 2010).

The native section includes transcripts of handwritten essays (*SKRIPT 2012*) and classroom conversation (*SCHOLA 2010*) from elementary and secondary schools. Both have their counterparts documenting the Roma ethnolect of Czech:[1] essays (*ROMi 2013*) and recordings and transcripts of dialogues (*ROMi 1.0*).[2]

The non-native section goes by the name of *CzeSL*, the acronym of *Czech as the Second Language*. *CzeSL* consists of transcripts of short handwritten essays collected from non-native learners with various levels of proficiency and native languages, mostly students attending Czech language courses before or during their studies at a Czech

[1] The Romani ethnolect of Czech is the result of contact with Romani as the linguistic substrate. To a lesser (and weakening) extent the ethnolect shows some influence of Slovak or even Hungarian, because most of its speakers have roots in Slovakia. The ethnolect can exhibit various specifics across all linguistic levels. However, nearly all of them are complementary with their colloquial or standard Czech counterparts. A short written text, devoid of phonological properties, may be hard to distinguish from texts written by learners without the Romani backround. The only striking exception are misspellings in contexts where the latter benefit from more exposure to written Czech. The typical example is the omission of word boundaries within phonological words, e.g. between a clitic and its host. In other respects, the pattern of error distribution in texts produced by ethnolect speakers is closer to native rather than foreign learners (Bořkovcová, 2007, 2017).

[2] A more recent release *SKRIPT 2015* includes a balanced mix of essays from *SKRIPT 2012* and *ROMi 2013*. For more details and links see `http://utkl.ff.cuni.cz/akces/`.

Language	Corpus	Sentences	Err. r.	Domain	# Refs.
English	*Lang-8*	1 147 451	14.1%	SL	1
	NUCLE	57 151	6.6%	SL	1
	FCE	33 236	11.5%	SL	1
	W&I+LOCNESS	43 169	11.8%	SL, native students	5
	CoNLL-2014 test	1 312	8.2%	SL	2,10,8
	JFLEG	1 511	—	SL	4
	GMEG	6 000	—	web, formal articles, SL	4
	AESW	over 1M	—	scientific writing	1
	CWEB	13 574	∼2%	web	2
Czech	*AKCES-GEC*	47 371	21.4%	SL essays, Romani ethnolect of Czech	2
German	*Falko-MERLIN*	24 077	16.8%	SL essays	1
Russian	*RULEC-GEC*	12 480	6.4%	SL, heritage speakers	1
Spanish	*COWS-L2H*	12 336	—	SL, heritage speakers	2
Ukrainian	*UA-GEC*	20 715	7.1%	natives/SL, translations and personal texts	2
Romanian	*RONACC*	10 119	—	native speakers transcriptions	1

Table 1: Comparison of GEC corpora in size, token error rate, domain and number of reference annotations in the test portion. SL = second language learners.

university. There are several releases of *CzeSL*, which differ mainly to what extent and how the texts are annotated (Rosen et al., 2020).[3]

More recently, hand-written essays have been transcribed and annotated in *TEITOK* (Janssen, 2016),[4] a tool combining a number of corpus compilation, annotation and exploitation functionalities.

Learner Czech is also represented in *MERLIN*, a multilingual (German, Italian and Czech) corpus built in 2012–2014 from texts submitted as a part of tests for language proficiency levels (Boyd et al., 2014).[5]

Finally, *AKCES-GEC* (Náplava and Straka, 2019) is a GEC corpus for Czech created from the subset of the above mentioned *AKCES* resources (Šebesta, 2010): the *CzeSL-man* corpus (non-native Czech learners with manual annotation) and a part of the *ROMi* corpus (speakers of the Romani ethnolect).

Compared to the *AKCES-GEC*, the new *GECCC* corpus contains much more data (47 371 sentences vs. 83 058 sentences, respectively), by extending data in the existing domains and also adding two new domains: essays written by native learners and website texts, making it the largest non-English GEC corpus and one of the largest GEC corpora overall.

3 Annotation

3.1 Data Selection

We draw the original uncorrected data from the following Czech learner corpora or Czech websites:

- *Natives Formal* – essays written by native students of elementary and secondary schools from the *SKRIPT 2012* learner corpus, compiled in the *AKCES* project
- *Natives Web Informal* – newly annotated informal website discussions from Czech Facebook Dataset (Habernal et al., 2013a,b) and Czech news site `novinky.cz`.
- *Romani* – essays written by children and teenagers of the Romani ethnic minority from the *ROMi* corpus of the *AKCES* project and the *ROMi* section of the *AKCES-GEC* corpus
- *Second Learners* – essays written by non-native learners, from the *Foreigners* section of the *AKCES-GEC* corpus, and the *MERLIN* corpus

Since we draw our data from several Czech corpora originally created in different tools with different annotation schemes and instructions, we re-annotated the errors in a unified manner for the entire development and test set and partially also

[3] For a list of *CzeSL* corpora with their sizes and annotation details see `http://utkl.ff.cuni.cz/learncorp/`.

[4] `http://www.teitok.org`

[5] `https://www.merlin-platform.eu`

Dataset	Documents	Selected
AKCES-GEC-test	188	188
AKCES-GEC-dev	195	195
MERLIN	441	385
Novinky.cz	—	2 695
Facebook	10 000	3 850
SKRIPT 2012	394	167
ROMi	1 529	218

Table 2: Data resources for the new Czech GEC corpus. The second column (Selected) shows the size of the selected subset from all available documents (first column, Documents).

for the training set.

The data split was carefully designed to maintain representativeness, coverage and backwards compatibility. Specifically, (i) test and development data contain roughly the same amount of annotated data from all domains, (ii) original *AKCES-GEC* dataset splits remain unchanged, (iii) additional available detailed annotations such as user proficiency level in *MERLIN* were leveraged to support the split balance. Overall, the main objective was to achieve a representative cover over development and testing data. Table 2 presents the sizes of data resources in the number of documents. The first column (Documents) shows the number of all available documents collected in an initial scan. The second column (Selected) is a selected subset from the available documents, due to budgetary constraints and to achieve a representative sample over all domains and data portions. The relatively higher number of documents selected for the *Natives Web Informal* domain is due to its substantially shorter texts, yielding fewer sentences; also, we needed to populate this part of the corpus as a completely new domain with no previously annotated data.

To achieve more fine-grained balancing of the splits, we used additional metadata where available: user's proficiency levels and origin language from *MERLIN* and the age group from *AKCES*.

3.2 Preprocessing

De/tokenization is an important part of data preprocessing in grammar error correction. Some formats, such as the M^2 format (Dahlmeier and Ng, 2012), require tokenized formats to track and evaluate correction edits. On the other hand, detokenized text in its natural form is required for other applications. We therefore release our corpus in two formats: a tokenized M^2 format and detokenized format aligned at sentence, paragraph and document level. As part of our data is drawn from earlier, tokenized GEC corpora *AKCES-GEC* and *MERLIN*, this data had to be detokenized. A slightly modified Moses detokenizer[6] is attached to the corpus. To tokenize the data for the M^2 format, we use the UDPipe tokenizer (Straka et al., 2016).

3.3 Annotation

The test and development sets in all domains were annotated from scratch by five in-house expert annotators,[7] including re-annotations of the development and test data of the earlier GEC corpora to achieve a unified annotation style. All the test sentences were annotated by two annotators; one half of the development sentences received two annotations and the second half one annotation. The annotation process took about 350 hours in total.

The annotation instructions were unified across all domains: The corrected text must not contain any grammatical or spelling errors and should sound fluent. Fluency edits are allowed if the original is incoherent. The entire document was given as a context for the annotation. Annotators were instructed to remove too incomprehensible documents or those containing private information.

To keep the annotation process simple for the annotators, the sentences were annotated (corrected) in a text editor and postprocessed automatically to retrieve and categorize the GEC edits by the ERRor ANnotation Toolkit (ERRANT) (Bryant et al., 2017).

3.4 Train Data

The first source for the training data are the data from the *SKRIPT 2012*; the *MERLIN* corpus and the *AKCES-GEC* train set that were not annotated, thus containing original annotations. These data cover the *Natives Formal*, the *Romani* and the *Second Learners* domain. The second part of the training data are newly annotated data. Specifically, these are all *Natives Web Informal* data and also a small part in the *Second Learners* domain.

[6] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/tokenizer/detokenizer.perl

[7] Our annotators are senior undergraduate students of humanities, regularly employed for various annotation efforts at our institute.

All data in the training set were annotated with one annotation.

3.5 Corpus Alignment

The majority of models proposed for grammatical error correction operates over sentences. However, preliminary studies on document-level grammatical error correction recently appeared (Chollampatt et al., 2019; Yuan and Bryant, 2021). The models were shown to benefit from larger context as certain errors such as errors in articles or tense choice do require larger context. To simplify future work with our dataset, we release three alignment levels: (i) sentence-level, (ii) paragraph-level and (iii) document-level. Given that the state-of-the-art grammatical error correction systems still operate on sentence level despite the initial attempts with document-level systems, we perform model training and evaluation at the usual sentence level.[8]

3.6 Inter-Annotator Agreement

As suggested by Rozovskaya and Roth (2010), followed later by Rozovskaya and Roth (2019) and Syvokon and Nahorna (2021), we evaluate inter-annotator agreement by asking a second annotator to judge the need for a correction in a sentence already annotated by someone else, in a single-blind setting as to the status of the sentence (corrected/uncorrected).[9] Five annotators annotated the first pass and three annotators judged the sentence correctness in the second pass. In the second pass, each of the three annotators judged a disjoint set of 120 sentences. Table 3 summarizes the inter-annotator agreement based on second-pass judgements: the numbers represent the percentage of sentences judged correct in the second pass.

Both the average and the standard deviation (82.96 ± 12.12) of our inter-annotator agreement are similar to inter-annotator agreement measured on English (63 ± 18.46, Rozovskaya and Roth 2010), Russian (80 ± 16.26, Rozovskaya and Roth 2019) and Ukrainian (69.5 ± 7.78 Syvokon and Nahorna 2021).

First → Second ↓	A1	A2	A3	A4	A5
A1	—	93.39	97.96	89.63	72.50
A2	84.43	—	95.91	90.18	78.15
A3	68.80	87.68	—	79.39	57.50

Table 3: Inter-annotator agreement based on second-pass judgements: numbers represent percentage of sentences judged correct in second-pass proofreading. Five annotators annotated the first pass, three annotators judged the sentence correctness in the second pass.

3.7 Error Type Analysis

To retrieve and categorize the correction edits from the erroneous-corrected sentence pairs, ERRor ANnotation Toolkit (ERRANT) (Bryant et al., 2017) was used. Inspired by Boyd (2018), we adapted the original English error types to the Czech language. For the resulting set see Table 4. The POS error types are based on the UD POS tags (Nivre et al., 2020) and may contain an optional *:INFL* subtype when the original and the corrected words share a common lemma. The word-order error type was extended by an optional *:SPELL* subtype to allow for capturing word order errors including words with minor spelling errors. The original orthography error type *ORTH* covering both errors in casing and whitespaces is now subtyped with *:WSPACE* and *:CASING* to better distinguish between the two phenomena. Finally, we add two error types specific to Czech: *DIACR* for errors in either missing or redundant diacritics and *QUOTATION* for wrongly used quotation marks. Two original error types remain unchanged: *MORPH*, indicating replacement of a token by another with the same lemma but different POS, and *SPELL*, indicating incorrect spelling.

For part-of-speech tagging and lemmatization we rely on UDPipe (Straka et al., 2016).[10] The word list for detecting spelling errors comes from MorfFlex (Hajič et al., 2020).[11]

We release the Czech ERRANT at `https://github.com/ufal/errant_czech`. We assume that it is applicable to other languages with similar set of errors, especially Slavic languages, if lemmatizer, tagger and morphological dictionary are available.

[8] Note that even if human evaluation in Section 5 is performed on sentence-aligned data, human annotators process whole documents, and thus take the full context into account.

[9] A sentence-level agreement on sentence correctness is generally preferred in GEC annotations to an exact inter-annotator match on token edits, since different series of corrections may possibly lead to a correct sentence (Bryant and Ng, 2015).

[10] Using the czech-pdt-ud-2.5-191206.udpipe model.

[11] We also use the *aggresive* variant of the stemmer from `https://research.variancia.com/czech_stemmer/`.

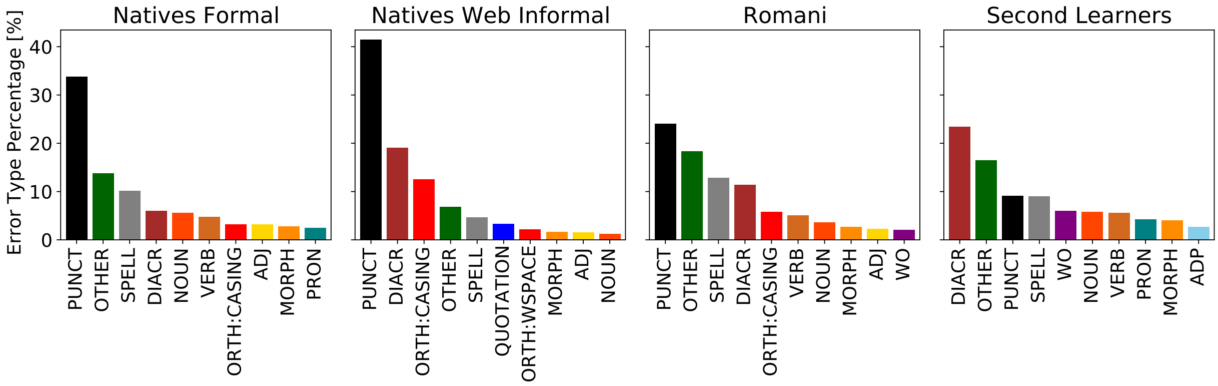


Figure 1: Distribution of top-10 ERRANT error types per domain in the development set.

Error Type	Subtype	Example
POS (15)		*tažené* → *řízené*
	:INFL	*manželka* → *manželkou*
MORPH		*maj* → *mají*
ORTH	:CASING	*usa* → *USA*
	:WSPACE	*přes to* → *přesto*
SPELL		*ochtnat* → *ochutnat*
WO		*plná jsou* → *jsou plná*
	:SPELL	*blískají zeleně* → *zeleně blýskají*
QUOTATION		*"* → *„*
DIACR		*tiskarna* → *tiskárna*
OTHER		*sem* → *jsem ho*

Table 4: Czech ERRANT Error Types.

3.8 Final Dataset

The final corpus consists of 83 058 sentences and is distributed in two formats: the tokenized M^2 format (Dahlmeier and Ng, 2012) and the detokenized format with alignments at the sentence, paragraph and document levels. Although the detokenized format does not include correction edits, it does retain full information about the original spacing.

The statistics of the final dataset are presented in Table 5. The individual domains are balanced on the sentence level in the development and testing sets, each of them containing about 8 000 sentences. The number of paragraphs and documents varies: on average, the *Natives Web Informal* domain contains less than 2 sentences per document, while the *Natives Formal* domain more than 20.

As expected, the domains differ also in the error rate, i.e. the proportion of erroneous tokens (see Table 5). The students' essays in the *Natives Formal* domain are almost 3 times less erroneous than any other domain, while in the *Romani* and *Second Learners* domain, approximately each 4-th token is incorrect.

Furthermore, the prevalence of error types differs for each individual domain. The 10 most common error types in each domain are presented in Figure 1. Overall, errors in punctuation (*PUNCT*) constitute the most common error type. They are the most common error in three domains, although their relative frequency varies. We further estimated that of these errors, 9% (*Natives Formal*) – 27% (*Natives Web Informal*) are uninteresting from the linguistic perspective, as they are only omissions of the sentence formal ending, probably purposeful in case of *Natives Web Informal*. The rest (75–91%) appears in a sentence, most of which (35–68% *Natives Formal*) is a misplaced comma: In Czech, syntactic status of finite clauses strictly determine the use of commas in the sentence. Finally, in 5–7% cases of all punctuation errors, a correction included joining two sentences or splitting a sentence into two sentences. Errors in either missing or wrongly used diacritics (*DIACR*), spelling errors (*SPELL*) and errors in orthography (*ORTH*) are also common, with varying frequency across domains.

Compared to the *AKCES-GEC* corpus, the *Grammar Error Correction Corpus for Czech* contains more than 3 times as many sentences in the development and test sets, more than 50% sentences in the training set and also two new domains.

To the best of our knowledge, the newly introduced *GECCC* dataset is the largest among GEC corpora in languages other than English and it is surpassed in size only by the English *Lang-8* and *AESW* datasets. With the exclusion of these two datasets, the *GECCC* dataset contains more sentences than any other GEC corpus currently known to us.

	Sentence-aligned			Paragraph-aligned			Doc-aligned			Error Rate
	#sentences			#paragraphs			#docs			
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test	
Natives Formal	4 060	1 952	1 684	1 618	859	669	227	87	76	5.81%
Natives Web Informal	6 977	2 465	2 166	3 622	1 294	1 256	3 619	1 291	1 256	15.61%
Romani	24 824	1 254	1 260	9 723	574	561	3 247	173	169	26.21%
Second Learners	30 812	2 807	2 797	8 781	865	756	2 050	167	170	25.16%
Total	66 673	8 478	7 907	23 744	3 592	3 242	9 143	1 718	1 671	18.19%

Table 5: Corpus statistics at three alignment levels: sentence-aligned, paragraph-aligned and doc-aligned. Average Error rate was computed on the concatenation of development and test data in all three alignment levels.

4 Model

In this section, we describe five systems for automatic error correction in Czech and analyze their performance on the new dataset. Four of these systems represent previously published Czech work (Richter et al., 2012; Náplava and Straka, 2019; Náplava et al., 2021) and one is our new implementation. The first system is a pre-neural approach, published and available for Czech (Richter et al., 2012), included for historical reasons as a previously known and available Czech GEC tool; the following four systems represent the current state of the art in GEC: they are all neural network architectures based on Transformers, differing in the training procedure, training data or training objective. A comparison of systems, trained and evaluated on English, Czech, German and Russian, with state of the art is given in Table 6.

4.1 Models

We experiment with the following models:

Korektor (Richter et al., 2012) is a pre-neural statistical spellchecker and (occasional) grammar checker. It uses the noisy channel approach with a candidate model that for each word suggests its variants up to a predefined edit distance. Internally, a Hidden Markov Model (Baum and Petrie, 1966) is built. Its hidden states are the variants of words proposed by the candidate model, and the transition costs are determined from three N-gram language models built over word forms, lemmas and part-of-speech-tags. To find an optimal correction, Viterbi algorithm (Forney, 1973) is used.

Synthetic trained (Náplava and Straka, 2019) is a neural-based Transformer model that is trained to translate the original ungrammatical text to a well formed text. The original Transformer model (Vaswani et al., 2017) is regularised with an additional source and target word dropout and the training objective is modified to focus on tokens that should change (Grundkiewicz and Junczys-Dowmunt, 2019). As the amount of existing annotated data is small, an unsupervised approach with a spelling dictionary is used to generate a large amount of synthetic training data. The model is trained solely on these synthetic data.

AKCES-GEC (AG) finetuned (Náplava and Straka, 2019) is based on *Synthetic trained*, but finetunes its weights on a mixture of synthetic and authentic data from the *AKCES-GEC* corpus, i.e., on data from the *Romani* and *Second Learners* domains. See Table 6 for comparison with state of the art in English, Czech, German and Russian.

GECCC finetuned uses the same architecture as *Synthetic trained*, but we finetune its weights on a mixture of synthetic and (much larger) authentic data from the newly released GECCC corpus. We use the official code of Náplava and Straka (2019) with the default settings and mix the synthetic and new authentic data in a ratio of 2:1.

Joint GEC+NMT (Náplava et al., 2021) is a Transformer model trained in a multi-task setting. It pursues two objectives: (i) to correct Czech and English texts; (ii) to translate the noised Czech texts into English texts and the noised English texts into Czech texts. The source data come from the *CzEng v2.0* corpus (Kocmi et al., 2020) and were noised using a statistical system KaziText (Náplava et al., 2021) that tries to model several most frequently occurring errors such as diacritics, spelling or word ordering. The statistics of the Czech noise were estimated on the new training set, therefore, the system was indirectly trained also on data from *Natives Formal* and *Natives Web Informal* domains, unlike the *AG finetuned* sys-

System	Params	English		Czech	German	Russian
		W&I+L	CoNLL 14	AKCES-GEC	Falko-Merlin	RULEC-GEC
Boyd (2018)	–	–	–	–	45.22	–
Choe et al. (2019)	–	63.05	–	–	–	–
Lichtarge et al. (2019)	–	–	56.8	–	–	
Lichtarge et al. (2020)	–	66.5	62.1	–	–	–
Omelianchuk et al. (2020)	–	72.4	65.3	–	–	–
Rothe et al. (2021) *base*	580M	60.2	54.10	71.88	69.21	26.24
Rothe et al. (2021) *xxl*	13B	69.83	65.65	83.15	75.96	51.62
Rozovskaya and Roth (2019)	–	–	–	–	–	21.00
Xu et al. (2019)	–	63.94	60.90	–	–	–
AG finetuned	210M	69.00	63.40	80.17	73.71	50.20

Table 6: Comparison of selected single-model systems on English (W&I+L, CoNLL-2014), Czech (AKCES-GEC), German (Falko-Merlin GEC) and Russian (RULEC-GEC) datasets. Our reimplementation of the *AG finetuned* model is from Náplava and Straka (2019). Note that models vastly differ in training/fine-tuning data and size (e.g., Rothe et al. (2021) *xxl* is 50 times larger than *AG finetuned*).

System	$M^2_{0.5}$-score					Mean human score				
	NF	NWI	R	SL	Σ	NF	NWI	R	SL	Σ
Original	—	—	—	—	—	8.47	7.99	7.76	7.18	7.61
Korektor	28.99	31.51	46.77	55.93	45.09	8.26	7.60	7.90	7.55	7.63
Synthetic trained	46.83	38.63	46.36	62.20	53.07	8.55	7.99	8.10	7.88	7.98
AG finetuned	65.77	55.20	69.71	71.41	68.08	8.97	8.22	**8.91**	8.35	8.38
GECCC finetuned	**72.50**	**71.09**	**72.23**	**73.21**	**72.96**	**9.19**	**8.72**	**8.91**	**8.67**	**8.74**
Joint GEC+NMT	68.14	66.64	65.21	70.43	67.40	9.06	8.37	8.69	8.19	8.35
Reference	—	—	—	—	—	9.58	9.48	9.60	9.63	9.57

Table 7: Mean score of human judgements and $M^2_{0.5}$ score for each system in domains (NF = *Natives Formal*, NWI = *Natives Web Informal*, R = *Romani*, SL = *Second Learners*, Σ = whole dataset). All results in the whole dataset (the Σ column) are statistically significant with p-value < 0.001, except for the *AG finetuned* and *Joint GEC+NMT* systems, where the p-value is less than 6.2% for $M^2_{0.5}$ score and less than 4.3% for human score, using the Monte Carlo permutation test with 10M samples and probability of error at most 10^{-6} (Fay and Follmann, 2002; Gandy, 2009).

tem. The statistics of the English noise were estimated on *NUCLE* (Dahlmeier et al., 2013), *FCE* (Yannakoudakis et al., 2011) and *W&I+LOCNESS* (Yannakoudakis et al., 2018; Granger, 1998).

4.2 Results and Analysis

Table 7 summarizes the evaluation of the five grammar error correction systems (described in the previous Section 4.1), evaluated with highest-correlating and widely used metric, the M^2 score with $\beta = 0.5$, denoted as $M^2_{0.5}$ (left); and with human judgements (right). For the meta-evaluation of GEC metrics against human judgements, see the following Section 5.

Clearly, learning on GEC annotated data improves performance significantly, as evidenced by a giant leap between the systems without GEC data (*Korektor*, *Synthetic trained*) and the systems trained on GEC data (*AG finetuned*, *GECCC finetuned* and *Joint GEC+NMT*). Further addition of GEC data volume and domains is statistically significantly better ($p < 0.001$), as the only difference between *AG finetuned* and *GECCC finetuned* systems is that the former uses the *AKCES-GEC* corpus, while the latter is trained on larger and domain-richer *GECCC*. Access to larger data and more domains in the multi-task setting is useful (compare *Joint GEC+NMT* and *AG finetuned* on newly added *Natives Formal* and *Natives Web Informal* domains), although direct training seems superior (*GECCC finetuned* over *Joint GEC+NMT*).

Error Type	#	P	R	$F_{0.5}$
DIACR	3 617	86.84	88.77	87.22
MORPH	610	73.58	55.91	69.20
ORTH:CASING	1 058	81.60	55.15	74.46
ORTH:WSPACE	385	64.44	74.36	66.21
OTHER	3 719	23.59	20.04	22.78
POS	2 735	56.50	22.12	43.10
POS:INFL	1 276	74.47	48.22	67.16
PUNCT	4 709	71.42	61.17	69.10
QUOTATION	223	89.44	61.06	81.83
SPELL	1 816	77.27	75.76	76.96
WO	662	60.00	29.89	49.94

Table 8: Analysis of *GECCC finetuned* model performance on individual error types. For this analysis, all POS-error types were merged into a single error type POS.

We further analyse the best model (*GECCC finetuned*) and inspect its performance with respect to individual error types. For simpler analysis, we grouped all POS-related errors into two error types: *POS* and *POS:INFL* for words which are erroneous only in inflection and share the same lemma with their correction.

As we can see in Table 8, the model is very good at correcting local errors in diacritics (*DIACR*), quotation (*QUOTATION*), spelling (*SPELL*) and casing (*ORTH:CASING*). Unsurprisingly, small changes are easier than longer edits: similarly, the system is better in inflection corrections (*POS:INFL*, words with the same lemma) than on *POS* (correction involves finding a word with a different lemma).

Should the word be split or joined with an adjacent word, the model does so with a relatively high success rate (*ORTH:WSPACE*). The model is also able to correctly reorder words (*WO*), but here its recall is rather low. The model performs worst on errors categorized as *OTHER*, which includes edits that often require rewriting larger pieces of text. Generally, the model has higher precision than recall, which suits the needs of standard GEC, where proposing a bad correction for a good text is worse than being inert to an existing error.

5 Meta-evaluation of Metrics

There are several automatic metrics used for evaluating system performance on GEC dataset, although it is not clear which of them is preferable in terms of high correlation with human judgements on our dataset.

The most popular GEC metrics are the MaxMatch (M^2) scorer (Dahlmeier and Ng, 2012) and the ERRANT scorer (Bryant et al., 2017).

The MaxMatch (M^2) scorer reports the F-score over the optimal phrasal alignment between a source sentence and a system hypothesis reaching the highest overlap with the gold standard annotation. It was used as the official metric for the CoNLL 2013 and 2014 Shared Tasks (Ng et al., 2013, 2014) and is also used on various other datasets such as the German *Falko-MERLIN GEC* (Boyd, 2018) or Russian *RULEC-GEC* (Rozovskaya and Roth, 2019).

The ERRANT scorer was used as the official metric of the recent Building Educational Application 2019 Shared Task on GEC (Bryant et al., 2019). The ERRANT scorer also contains a set of rules operating over a set of linguistic annotations to construct the alignment and extract individual edits.

Other popular automatic metrics are the General Language Evaluation Understanding (GLEU) metric (Napoles et al., 2015), that additionally measures text fluency, and I-Measure (Felice and Briscoe, 2015), that calculates weighted accuracy of both error detection and correction.

5.1 Human Judgements Annotation

In order to evaluate the correlation of several GEC metrics with human judgements, we collected annotations of the original erroneous sentences, the manually corrected gold references and automatic corrections made by five GEC systems described in Section 4. We used the hybrid *partial ranking with scalars* (Sakaguchi and Van Durme, 2018), in which the annotators judged the sentences on a scale 0–10 (from ungrammatical to correct).[12] The sentences were evaluated with respect to the context of the document. In total, three annotators judged 1 100 documents, sampled from the test set comprising about 4 300 original sentences and about 15 500 unique corrected variants and gold references of the sentences. The annotators annotated 127 documents jointly and the rest was annotated by a single annotator. This annotation process took about 170 hours. Together with the

[12] Recent works (Sakaguchi and Van Durme, 2018; Novikova et al., 2018) both found partial ranking with scalars to be more reliable than direct assessment framework used by WMT (Bojar et al., 2016) and earlier GEC evaluation approaches (Grundkiewicz et al., 2015; Napoles et al., 2015).

Domain	Sentence level r	Sentence level ρ	System level r	System level ρ
Natives Formal	87.13	88.76	92.01	92.52
Natives Web Inf.	80.23	81.47	95.33	91.80
Romani	86.57	86.57	88.73	85.90
Second Learners	78.50	79.97	96.50	97.23
Whole Dataset	79.07	80.40	96.11	95.54

Table 9: Human judgements agreement: Pearson (r) and Spearman (ρ) mean correlation between 3 human judgements of 5 sentence versions at sentence- and system-level.

model training, data preparation and management of the annotation process, our rough estimation is about 300+ man-hours for the correlation analysis per corpus (language).

5.2 Agreement in Human Judgements

For the agreement in human judgements, we report the *Pearson correlation* and *Spearman's rank correlation coefficient* between 3 human judgements of 5 automatic sentence corrections at the system- and sentence-level. At the *sentence level*, the correlation of the judgements about the 5 sentence corrections is calculated for each sentence and each pair of the three annotators. The final sentence-level annotator agreement is the mean of these values over all sentences.

At the *system level*, the annotators' judgements for each system are averaged over the sentences, and the correlation of these averaged judgements is computed for each pair of the three annotators. In order to obtain smoother estimates (especially for Spearman's ρ), we utilize bootstrap resampling with 100 samples of a test set.

The human judgements agreement across domains is shown in Table 9. On the sentence level, the human judgements correlation is high on the least erroneous domain *Natives Formal*, implying that it is easier to judge the corrections in a low error density setting, and it is more difficult in high error density domains, such as *Romani* and *Second Learners* (compare error rates in Table 5).

5.3 Metrics Correlations with Judgements

Following Napoles et al. (2019), we provide a meta-evaluation of the following common GEC metrics robustness on our corpus:

- MaxMatch (M^2) (Dahlmeier and Ng, 2012)
- ERRANT (Bryant et al., 2017)
- GLEU (Napoles et al., 2015)
- I-measure (Felice and Briscoe, 2015)

Moreover, we vary the proportion of recall and precision, ranging from 0 to 2.0 for M^2-scorer and ERRANT, as Grundkiewicz et al. (2015) report that the standard choice of considering precision two times as important as recall may be suboptimal.

While we considered both sentence-level and system-level evaluation in Section 5.2, the automatic metrics should by design be used on a whole corpus, leaving us with only system-level evaluation. Given that the GEC systems perform differently on the individual domains (as indicated by Table 7), we perform the correlation computation on each domain separately and report the average.

For a given domain and metric, we compute the correlation between the automatic metric evaluations of the five systems on one side and the (average of) human judgements on the other side. In order to obtain a smoother estimate of Spearman's ρ and also to estimate standard deviations, we employ bootstrap resampling again, with 100 samples.

The results are presented in Table 10. While Spearman's ρ has more straightforward interpretation, it also has a much higher variance, because it harshly penalizes the differences in the ranking of systems with similar performance (namely *AG finetuned* and *Joint GEC+NMT* in our case). This fact has previously been observed by Macháček and Bojar (2013).

Therefore, we choose the most suitable GEC metric for our *GECCC* dataset according to Pearson r, which implies that M$^2_{0.5}$ and ERRANT$_{0.5}$ are the metrics most correlating with human judgements. Of those two, we prefer the M$^2_{0.5}$ score, not due to its marginal superiority in correlation (Table 10), but rather because it is much more language-agnostic compared to ERRANT, which requires a POS tagger, lemmatizer, morphological dictionary and language-specific rules.

Our results confirm that both M^2-scorer and ERRANT with $\beta = 0.5$ (chosen only by intuition for the CoNLL 2014 Shared task; Ng et al., 2014) correlate much better with human judgements, compared to $\beta = 0.2$ and $\beta = 1$. The detailed plots of correlations of M$^2_\beta$ score and ERRANT$_\beta$ score with human judgements for β ranging between 0 and 2, presented in Figure 2,

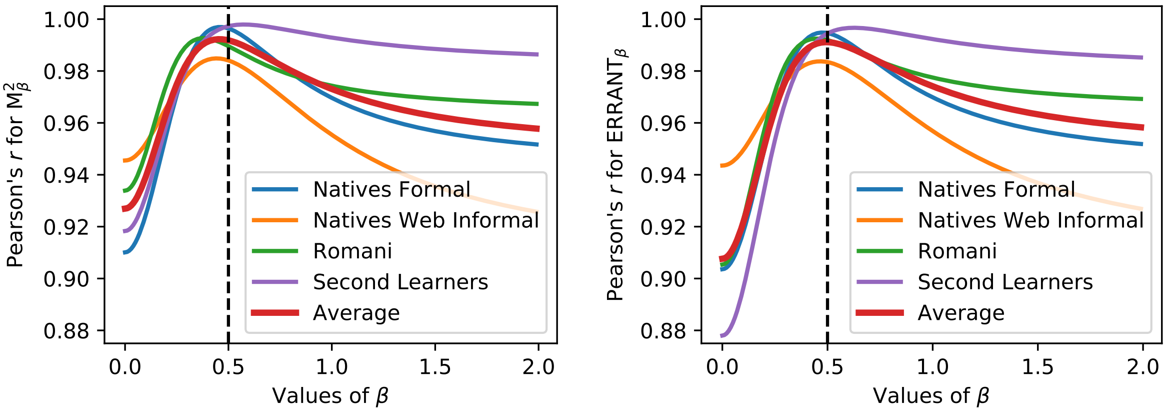


Figure 2: **Left:** System-level Pearson correlation coefficient r between human annotation and M^2_β-scorer for various values of β. **Right:** The same correlation for ERRANT$_\beta$.

Metric	System level	
	r	ρ
GLEU	97.37 ± 1.52	92.28 ± 6.19
I-measure	95.37 ± 2.16	98.66 ± 3.21
$M^2_{0.2}$	96.25 ± 1.71	93.27 ± 9.45
$M^2_{0.5}$	98.28 ± 1.03	97.77 ± 4.27
$M^2_{1.0}$	95.62 ± 1.81	93.22 ± 4.30
ERRANT$_{0.2}$	94.66 ± 2.44	91.19 ± 4.76
ERRANT$_{0.5}$	98.28 ± 1.04	98.35 ± 4.81
ERRANT$_{1.0}$	95.70 ± 1.80	93.61 ± 4.47

Table 10: System-level Pearson (r) and Spearman (ρ) correlation between the automatic metric scores and human annotations.

show that optimal β in our case lies between 0.4 and 0.5. However, we opt to employ the widely used $\beta = 0.5$ because of its prevalence and because the difference to the optimal β is marginal.

Our results are distinct from the results of Grundkiewicz et al. (2015), where $\beta = 0.18$ correlates best on the *CoNLL 14 test set*. Nevertheless, Napoles et al. (2019) demonstrate that $\beta = 0.5$ correlates slightly better than $\beta = 0.2$ on the *FCE* dataset, but that $\beta = 0.2$ correlates substantially better than $\beta = 0.5$ on *Wikipedia* and also on *Yahoo* discussions (a dataset containing paragraphs of Yahoo! Answers, which are informal user answers to other users' questions).

In the latter work, Napoles et al. (2019) propose that larger $\beta = 0.5$ correlate better on datasets with higher error rate and vice versa, given that the *FCE* dataset has 20.2% token error rate, compared to the error rates of 9.8% and 10.5% of *Wikipedia* and *Yahoo*, respectively. The hypothesis seems to extend to our results and the results of Grundkiewicz et al. (2015), considering that the *GECCC* dataset and the *CoNLL 14 test set* have token error rates of 18.2% and 8.2%, respectively.

5.4 GEC Systems Results

Table 7 presents both human scores for the GEC systems described in Section 4 and also results obtained by the chosen $M^2_{0.5}$ metric. The results are presented both on the individual domains and the entire dataset. Measuring over the entire dataset, human judgements and the M^2-scorer rank the systems in accordance.

Judged by the human annotators, all systems are better than the "do nothing" baseline (the *Original*) measured over the entire dataset, although *Korektor* makes harmful changes in two domains: *Natives Formal* and *Natives Web Informal*. These two domains contain frequent named entities, which upon an eager change disturb the meaning of a sentence, leading to severe penalization by human annotators. *Korektor* is also not capable of deleting, inserting, splitting or joining tokens. The fact that *Korektor* sometimes performs detrimental changes cannot be revealed by the M^2-scorer as it assigns zero score to the *Original* baseline and does not allow negative scores.

The human judgements confirm that there is still a large gap between the optimal *Reference* score and the best performing models. Regarding the domains, the neural models in the finetuned mode that had access to data from all domains seemed to improve the results consistently across each domain. However, given the fact that the source sentences in the *Second Learners* domain received the worst scores by human annotators, this domain seems to hold the greatest potential for future improvements.

6 Conclusions

We release a new Czech GEC corpus, the *Grammar Error Correction Corpus for Czech* (*GECCC*). This large corpus with 83 058 sentences covers four diverse domains, including essays written by native students, informal website texts, essays written by Romani ethnic minority children and teenagers and essays written by non-native speakers. All domains are professionally annotated for GEC errors in a unified manner, and errors were automatically categorized with a Czech-specific version of ERRANT released at https://github.com/ufal/errant_czech. We compare several strong Czech GEC systems, and finally, we provide a meta-evaluation of common GEC metrics across domains in our data. We conclude that M^2 and ERRANT scores with $\beta = 0.5$ are the measures most correlating with human judgements on our dataset, and we choose the $M^2_{0.5}$ as the preferred metric for the *GECCC* dataset. The corpus is publicly available under the CC BY-SA 4.0 license at http://hdl.handle.net/11234/1-4639.

Acknowledgements

This work has been supported by the Grant Agency of the Czech Republic, project EXPRO LUSyD (GX20-16819X). This research was also partially supported by SVV project number 260 575 and GAUK 578218 of the Charles University. The work described herein has been supported by and has been using language resources stored by the LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz) of the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2018101). This work was supported by the European Regional Development Fund project “Creativity and Adaptability as Conditions of the Success of Europe in an Interrelated World” (reg. no.: CZ.02.1.01/0.0/0.0/16_019/0000734).

We would also like to thank the reviewers and the TACL action editor for their thoughtful comments, which helped to improve this work.

References

Leonard E Baum and Ted Petrie. 1966. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *The annals of mathematical statistics*, 37(6):1554–1563.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Máša Bořkovcová. 2007. *Romský etnolekt češtiny*. Signeta, Praha.

Máša Bořkovcová. 2017. Romský etnolekt češtiny. In Petr Karlík, Marek Nekula, and Jana Pleskalová, editors, *Nový encyklopedický slovník češtiny*. Nakladatelství Lidové Noviny.

Adriane Boyd. 2018. Using Wikipedia Edits in Low Resource Grammatical Error Correction. In *Proceedings of the 4th Workshop on Noisy User-generated Text*. Association for Computational Linguistics.

Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Štindlová, and Chiara Vettori. 2014. The MERLIN corpus: Learner Language and the CEFR. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1281–1288, Reykjavik, Iceland. European Language Resources Association (ELRA).

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 2019. The BEA-2019 Shared Task on Grammatical Error Correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 2017. Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Christopher Bryant and Hwee Tou Ng. 2015. 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)*, pages 697–707, Beijing, China. 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. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, Florence, Italy. Association for Computational Linguistics.

Shamil Chollampatt, Weiqi Wang, and Hwee Tou Ng. 2019. Cross-Sentence Grammatical Error Correction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 435–445.

Teodor-Mihai Cotet, Stefan Ruseti, and Mihai Dascalu. 2020. Neural Grammatical Error Correction for Romanian. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 625–631.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better Evaluation for Grammatical Error Correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. 2016. A Report on the Automatic Evaluation of Scientific Writing Shared Task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–62, San Diego, CA. Association for Computational Linguistics.

Sam Davidson, Aaron Yamada, Paloma Fernandez Mira, Agustina Carando, Claudia H. Sanchez Gutierrez, and Kenji Sagae. 2020. Developing NLP Tools with a New Corpus of Learner Spanish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7238–7243, Marseille, France. European Language Resources Association.

Michael P Fay and Dean A Follmann. 2002. Designing Monte Carlo Implementations of Permutation or Bootstrap Hypothesis Tests. *The American Statistician*, 56(1):63–70.

Mariano Felice and Ted Briscoe. 2015. Towards a standard evaluation method for grammatical error detection and correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 578–587, Denver, Colorado. Association for Computational Linguistics.

Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. 2020. Grammatical Error Correction in Low Error Density Domains: A New Benchmark and Analyses. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478, Online. Association for Computational Linguistics.

G David Forney. 1973. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Axel Gandy. 2009. Sequential Implementation of Monte Carlo Tests With Uniformly Bounded Resampling Risk. *Journal of the American Statistical Association*, 104(488):1504–1511.

Sylvianne Granger. 1998. *Learner English on Computer*, chapter The computer learner corpus: A versatile new source of data for SLA research. Addison Wesley Longman, London & New York.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 2019. Minimally-Augmented Grammatical Error Correction. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 357–363, Hong Kong, China. Association for Computational Linguistics.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Edward Gillian. 2015. Human Evaluation of Grammatical Error Correction Systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 461–470, Lisbon, Portugal. Association for Computational Linguistics.

Ivan Habernal, Tomáš Ptáček, and Josef Steinberger. 2013a. Facebook Data for Sentiment Analysis. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Ivan Habernal, Tomáš Ptáček, and Josef Steinberger. 2013b. Sentiment Analysis in Czech Social Media Using Supervised Machine Learning. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 65–74, Atlanta, Georgia. Association for Computational Linguistics.

Jan Hajič, Jaroslava Hlaváčová, Marie Mikulová, Milan Straka, and Barbora Štěpánková. 2020. MorfFlex CZ 2.0. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Maarten Janssen. 2016. TEITOK: Text-Faithful Annotated Corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 4037–4043, Paris, France. European Language Resources Association (ELRA).

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020. Announcing CzEng 2.0 Parallel Corpus with over 2 Gigawords. *CoRR*, abs/2007.03006v1.

Jared Lichtarge, Chris Alberti, and Shankar Kumar. 2020. Data Weighted Training Strategies for Grammatical Error Correction. *Transactions of the Association for Computational Linguistics*, 8:634–646.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. 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)*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.

Matouš Macháček and Ondřej Bojar. 2013. Results of the WMT13 Metrics Shared Task. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 45–51, Sofia, Bulgaria. Association for Computational Linguistics.

Jakub Náplava, Martin Popel, Milan Straka, and Jana Straková. 2021. Understanding Model Robustness to User-generated Noisy Texts. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 340–350, Online. Association for Computational Linguistics.

Jakub Náplava and Milan Straka. 2019. Grammatical Error Correction in Low-Resource Scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Stroudsburg, PA, USA. Association for Computational Linguistics.

Courtney Napoles, Maria Nădejde, and Joel Tetreault. 2019. Enabling Robust Grammatical Error Correction in New Domains: Data Sets, Metrics, and Analyses. *Transactions of the Association for Computational Linguistics*, 7:551–566.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. 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)*, pages 588–593, Beijing, China. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 2017. 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*, pages 229–234, Valencia, Spain. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. 2013. The CoNLL-2013 Shared Task on Grammatical Error Correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria. Association for Computational Linguistics.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. 2020. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France. European Language Resources Association.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. 2018. RankME: Reliable Human Ratings for Natural Language Generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 72–78, New Orleans, Louisiana. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – Grammatical Error Correction: Tag, Not Rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA, Online. Association for Computational Linguistics.

Michal Richter, Pavel Straňák, and Alexandr Rosen. 2012. Korektor – A System for Contextual Spell-Checking and Diacritics Completion. In *Proceedings of COLING 2012: Posters*, pages 1019–1028, Mumbai, India. The COLING 2012 Organizing Committee.

Alexandr Rosen, Jiří Hana, Barbora Hladká, Tomáš Jelínek, Svatava Škodová, and Barbora Štindlová. 2020. *Compiling and annotating a learner corpus for a morphologically rich language – CzeSL, a corpus of non-native Czech*. Karolinum, Charles University Press, Praha.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. 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)*, pages 702–707, Online. Association for Computational Linguistics.

Alla Rozovskaya and Dan Roth. 2010. Annotating ESL Errors: Challenges and Rewards. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 28–36, Los Angeles, California. Association for Computational Linguistics.

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.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. 2016. Reassessing the Goals of Grammatical Error Correction: Fluency Instead of Grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182.

Keisuke Sakaguchi and Benjamin Van Durme. 2018. Efficient Online Scalar Annotation with

Bounded Support. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 208–218, Melbourne, Australia. Association for Computational Linguistics.

Karel Šebesta. 2010. Korpusy češtiny a osvojování jazyka. *Studie z aplikované lingvistiky/Studies in Applied Linguistics*, 1:11–34.

Milan Straka, Jan Hajič, and Jana Straková. 2016. UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia. European Language Resources Association (ELRA).

Oleksiy Syvokon and Olena Nahorna. 2021. UA-GEC: Grammatical Error Correction and Fluency Corpus for the Ukrainian Language. *CoRR*, abs/2103.16997v1.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 2012. 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)*, pages 198–202, Jeju Island, Korea. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shuyao Xu, Jiehao Zhang, Jin Chen, and Long Qin. 2019. Erroneous data generation for Grammatical Error Correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158, Florence, Italy. Association for Computational Linguistics.

Helen Yannakoudakis, Øistein Andersen, Ardeshir Geranpayeh, Ted Briscoe, and Diane Nicholls. 2018. Developing an automated writing placement system for ESL learners. *Applied Measurement in Education*, 31.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Zheng Yuan and Christopher Bryant. 2021. Document-level Grammatical Error Correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 75–84.

The Prague Bulletin of Mathematical Linguistics
NUMBER 116 APRIL 2021 27–42

Diacritics Restoration using BERT with Analysis on Czech language

Jakub Náplava, Milan Straka, Jana Straková

Institute of Formal and Applied Linguistics Charles University, Czech Republic Faculty of Mathematics and Physics

Abstract

We propose a new architecture for diacritics restoration based on contextualized embeddings, namely BERT, and we evaluate it on 12 languages with diacritics. Furthermore, we conduct a detailed error analysis on Czech, a morphologically rich language with a high level of diacritization. Notably, we manually annotate all mispredictions, showing that roughly 44% of them are actually not errors, but either plausible variants (19%), or the system corrections of erroneous data (25%). Finally, we categorize the real errors in detail. We release the code at `https://github.com/ufal/bert-diacritics-restoration`.

1. Introduction

Diacritics Restoration, also known as Diacritics Generation or Accent Restoration, is a task of correctly restoring diacritics in a text without any diacritics. Its main difficulty stems from ambiguity where context needs to be taken into account to select the most appropriate word variant, because diacritization removal creates new groups of homonymy.

Current state-of-the-art algorithms for diacritics restoration are mostly based on either recurrent neural networks combined with an external language model (Náplava et al., 2018; AlKhamissi et al., 2020) or Transformer (Mubarak et al., 2019). Recently, BERT (Devlin et al., 2019) was shown to outperform many models on many tasks while being much faster due to the fact that it uses simple parallelizable classification head instead of a slow auto-regressive approach.

In this work, we first describe a model for diacritics restoration based on BERT and evaluate it on multilingual dataset comprising of 12 languages (Náplava et al., 2018).

Corresponding author: naplava@ufal.mff.cuni.cz

Cite as: Jakub Náplava, Milan Straka, Jana Straková. Diacritics Restoration using BERT with Analysis on Czech language. The Prague Bulletin of Mathematical Linguistics No. 116, 2021, pp. 27–42.
doi: 10.14712/00326585.013.

We show that the proposed model outperforms the previous state-of-the-art system (Náplava et al., 2018) in 9 languages significantly.

We further provide an extensive analysis of our model performance in Czech, a language with rich morphology and a high level of diacritization. In addition to clean data from Wikipedia (Náplava et al., 2018), the model was evaluated on data collected from other domains, including noisy data, and we show that stable performance holds even if the text contains spelling and other grammatical errors.

Sometimes, multiple plausible diacritization variants are possible, while only one gold reference exists, which comes from the original text before diacritization was automatically stripped to create test data. To assess the extent of these cases, we employed annotators to manually annotate all mispredictions and we found that 19% of errors are plausible variants and 25% of errors are system corrections of errors in data.

Finally, we further analyse the remaining errors by analysing characteristics of plausible variants.

2. Related Work

Diacritics Restoration is an active area of research in many languages: Vietnamese (Nga et al., 2019), Romanian (Nuţu et al., 2019), Czech (Náplava et al., 2018), Turkish (Adali and Eryiğit, 2014), Arabic (Madhfar and Qamar, 2020; AlKhamissi et al., 2020) and many others.

There are three main architectures currently used in diacritics restoration: convolutional neural networks (Alqahtani et al., 2019), recurrent neural networks often combined with an external language model (Belinkov and Glass, 2015; Náplava et al., 2018; AlKhamissi et al., 2020) and Transformer-based models (Orife, 2018; Mubarak et al., 2019). The convolutional neural networks are fast to train and also to infer. However, compared to the recurrent and Transformer-based architectures, they do generally achieve slightly worse results due to the fact that they model long-range dependencies worse. On the other hand, recurrent- and Transformer-based architectures are much slower.

Recently, the BERT model (Devlin et al., 2019) comprising of self-attention layers, was proposed and shown to reach remarkable results on a variety of tasks. As it uses no recurrent layers, its inference time is much shorter. We expect BERT to significantly improve the performance over current state-of-the-art diacritization architectures.

3. Model Architecture

The core of our system is a pre-trained multilingual BERT model that uses self-attention layers to create contextualized embeddings for tokenized text without diacritics. The contextual embeddings are fed into a fully-connected feed-forward neural network followed by a softmax layer. This outputs a vector with a distribution over a set of instructions that define diacritization operation over individual characters of

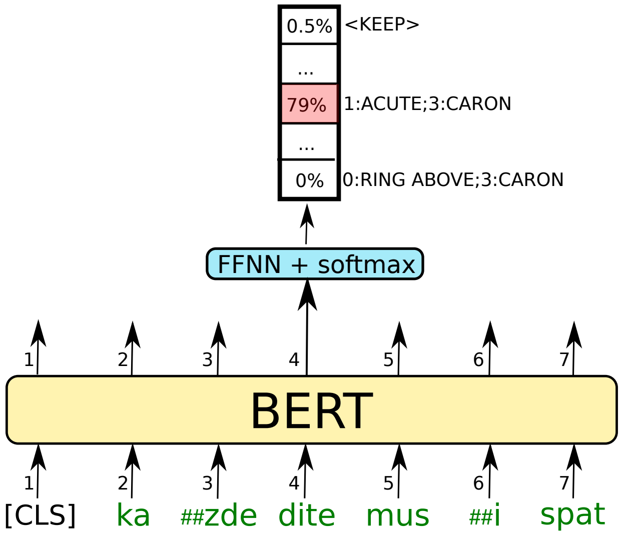


Figure 1. Model architecture. Text without diacritics, tokenized into subwords, is fed to BERT and for each of its outputs, fully-connected network followed by softmax is applied to obtain the most probable instruction for diacritization. ##-prefixes of some subwords are added by the BERT tokenizer.

each input token. We select the instruction with maximum probability. The model is illustrated in Figure 1.

3.1. Diacritization Instruction Set

To decrease the size of the final softmax layer, the output labels are not the diacritized variants of input subwords, as one would expect, but they are a set of instructions that provide prescription on how to restore diacritics. Specifically, one such instruction consists of index-diacritical mark tuples that define on what index of input subword a particular diacritical mark should be added.

An example of a diacritization instructions set can be seen in Figure 2. Given an input subword *dite* (*dítě*), with four characters indexed from 0 to 3, the appropriate diacritization instruction is *1:ACUTE;3:CARON*, in which acute is to be added to *i* and caron is to be added to *e* resulting in a properly diacritized word *dítě*. Obviously, the network can choose to leave the (sub)word unchanged, for which a special instruction *<KEEP>* is reserved. Should the network accidentally select an impossible instruction, no operation is carried out and the input (sub)word is also left unchanged.

To construct the set of possible diacritization instructions, we tokenize the undiacritized text of the particular training set and align each input token to the corresponding token in the diacritized text variant. The diacritical mark in each instruction is obtained from the Unicode name of the diacritized character. We keep only those

input	instruction	result	note
dite	1:CARON;3:ACUTE	dítě	optimal instruction
dite	1:CARON	díte	
dite	3:ACUTE	ditě	
dite	<KEEP>	dite	no change
dite	2:RING ABOVE	dite	impossible instruction ignored

Figure 2. Diacritization instructions examples for input "dite (dítě)" with 4 characters, indexed from 0 to 3. Index-Instruction tuples generate diacritics for given input.

instructions that occurred at least twice in a training set to filter out extremely rare instructions that originate for example from foreign words or bad spelling.

3.2. Training Details

We train both the fully-connected network and BERT with AdamW optimizer which minimizes the negative log-likelihood. The learning rate linearly increases from 0 to 5e-5 over the first 10000 steps and then remains the same. We use HuggingFace implementation of *BertForTokenClassification* and initialize *BERT-base* values from *bert-base-multilingual-uncased* model.

We use the batch size of 2048 sentences and clip each training sentence on 128 tokens. We train each model for circa 14 days on Nvidia P5000 GPU and select the best checkpoint according to development set.

4. Automatic Evaluation on Diacritization Corpus with 12 Languages

We evaluate our approach on the dataset of Náplava et al. (2018). This dataset contains training and evaluation data for 12 languages: Vietnamese, Romanian, Latvian, Czech, Polish, Slovak, Irish, Hungarian, French, Turkish, Spanish and Croatian.

We evaluate the model performance using a standard metric, the *alpha-word accuracy*. This metric omits words composed of non-alphabetical characters (e.g., punctuation).

For each language, we compute an independent set of operations and train a separate model. We use the concatenation of the Wiki and the Web training data of (Náplava et al., 2018) both for computing a set of instructions and also as the training data for our model.[1] The size of each instruction set and our results in comparison

[1] In Romanian Web data, ş (LATIN SMALL LETTER S WITH CEDILLA) is for historical reasons often used instead of ș (LATIN SMALL LETTER S WITH COMMA BELOW) and similarly ţ (LATIN SMALL LETTER T WITH CEDILLA) is often used instead of ț (LATIN SMALL LETTER T WITH COMMA BELOW). We replace the occurrences of the previously-used characters (the former ones) with their standard versions (the latter ones).

Language	Instruction Set Size	Náplava et al. (2018)	Ours	Error Reduction
Czech	1005	99.06	**99.22 ±0.046**	17 %
Vietnamese	2018	97.73	**98.53 ±0.037**	35 %
Latvian	720	97.49	**98.63 ±0.045**	45 %
Polish	1005	99.55	**99.66 ±0.041**	24 %
Slovak	785	99.09	**99.32 ±0.030**	25 %
French	681	**99.71**	**99.71 ±0.016**	0 %
Irish	189	98.71	**98.88 ±0.040**	13 %
Spanish	492	**99.65**	99.62 ±0.018	− 9 %
Croatian	541	99.67	**99.73 ±0.018**	18 %
Hungarian	767	99.29	**99.41 ±0.038**	17 %
Turkish	1005	**99.28**	98.95 ±0.046	− 46 %
Romanian	1677	98.37	**98.64 ±0.056**	17 %

Table 1. Comparison of alpha-word accuracy of our model including 95% confidential intervals to previous state-of-the-art on 12 languages.

with the previous state-of-the-art-results of Náplava et al. (2018) are presented in Table 1. Apart for alpha-word accuracy itself, we also report 95% confidential intervals computed using bootstrap resampling method.

On 9 of 12 languages, our approach significantly outperforms previous state-of-the-art combined recurrent neural networks with an external language model. The most significant improvements are achieved on Vietnamese and Latvian.

5. Detailed Analysis on Czech

We further provide a detailed analysis of our model performance in Czech, a language with rich morphology and a high diacritization level: Of the 26 English alphabet letters, a half of them can have one or two kinds of diacritization marks (Zeman, 2016). Czech is also the 4-th most diacritized language of the 12 languages found in the diacritization corpus of Náplava et al. (2018).

Particularly, we are interested in the three following questions:

- How would our system perform outside the very clean Wiki domain? (Section 5.1)
- Is it possible that some of the labeled mispredictions are actually plausible variants? (Section 5.2)
- Is there an observable characteristics in the real errors made by the system? (Section 5.3)

Domain	Sentences	Words	Evaluated Words
Natives Formal	1 743	19 973	19 138
Natives Informal	7 223	99 352	86 720
Romi	1 490	15 971	13 080
Second Learners	5 117	63 859	50 630

Table 2. Basic statistics of new data for testing diacritics restoration in Czech.

5.1. Additional Domains

The testing dataset of Náplava et al. (2018) is composed of clean sentences originating from Wikipedia. It is, however, a well-known fact that the performance of the (deep neural) models may deteriorate substantially when the input domain is changed (Belinkov and Bisk, 2017; Rychalska et al., 2019). To test our system in other, more challenging domains, we used data from a new Czech dataset (unpublished, in annotation process) for grammatical-error-correction that contains data collected from 4 sources:

- Natives Formal – Essays of elementary school Czech pupils (decent Czech proficiency)
- Natives Informal – texts collected from web discussions
- Second Learners – essays of Czech second learners
- Romi – texts of Czech pupils with Romani ethnolect (low Czech proficiency)

The dataset covers a wide range of Czech domains. It contains texts annotated in M2 format, a standard annotation format for grammar-error-correction corpora. In this format, each document contains original sentences with potential errors (e.g. spelling, grammatical or errors in diacritics) and a set of annotations describing what operations should be performed in order to fix each error.

To create target data for diacritics restoration, we apply all correcting edits that fix errors in diacritics and casing. We leave other errors intact, but do not evaluate on words that contain these errors, because they are not directly relevant to diacritics and in many cases, the errors are so severe that evaluation would be controversial. To rule out such words, we create a binary mask that distinguishes between evaluated and omitted words. Although the severely perturbed words are omitted from evaluation, they still remain in the sentence context and may still confuse the diacritization system, making the task potentially more difficult. See examples of such misleading sentence contexts in Figure 3.

The basic statistics of the new dataset are presented in Table 2. We display the number of sentences, the number of all words and the number of evaluated (unmasked) words. Compared to the Wikipedia dataset (Náplava et al., 2018), our new dataset has half the number of sentences and one third of its number of words.

Potřebujeme nové idea i novych **lidi**/lidí* , ktery je přinesou .

Na ulicích vidíme často nekterý lidi , kteří nosí **barevné**/barevně* oblečeny , které jsou snad hezké , ale určitě nejsou elegantní .

English translation (without ambiguities)

*We need new ideas and also **people** to come up with them.*

*In the streets, we can see some people wearing **colourful** clothes, which may be nice but certainly not elegant.*

Figure 3. Examples of misleading contexts in noisy texts. Correct diacritization (bold) can only be achieved by grammar corrections of the surrounding words (underlined).

We evaluate our model on all the above introduced Czech domains and present the results in Table 3. Despite our initial concern that the model would perform worse on these domains due to the noisy nature of the data, the results show that the model performance remains roughly stable on all domains. We suppose that although the writers produced quite noisy texts, they at the same time avoided foreign words that are generally harder to correctly diacritize.

5.2. Error Annotation

Clearly, removing diacritics creates new groups of homonymy (*dal/dál*, *krize/kříže*). In most cases, the correct diacritization variant can be inferred by a method which takes the sentence context into consideration. However, there are cases, in which more plausible variants are available, e.g., *šachu/šachů*, *pradlena/přadlena*, *podána/podaná*, as illustrated in Figure 4. Furthermore, some variants can only be disambiguated in the context of the whole document, such as in: *K nejvýznamnějším patří zmiňované vily/víly.* (more examples in Figure 6), not to mention other examples that can be only disambiguated by real-world knowledge such as in *Povrch satelitu/satelitů Země už zkoumalo několik sond*.

However, all our evaluation data are limited only to a single gold reference for each word without diacritics, given by the fact that the gold reference comes from the original text with diacritics. To explore both phenomena among the mispredictions, we hired annotators to examine: a) whether a word is correctly diacritized given the context of current sentence; and b) whether it is correct given a context of two previous sentences, current sentence and two following sentences (thus ruling out the words with even longer document dependencies).

While the evaluation of the clear Wiki data (Náplava et al., 2018) is straightforward, some of our newly introduced noisy data may become controversial to evaluate

Nebo záměna kapitol a jejich časová posloupnost v knize je pak ve filmu **podána/podaná** rozdílně .

Hraní **šachu/šachů** , ale především karetních her , kritizoval také Petr Chelčický .

Jeho matka byla **přadlena/pradlena** , která ke sklonku života propadla alkoholu .

Hororová hudba slouží především pro dokreslení **filmů/filmu** .

English translation

The chapters and their chronological order in the book are then ***presented/given*** *differently in the film.*

Playing ***a game of chess/games of chess*** *, but especially card games was criticized by Petr Chelčický .*

His mother was a ***washerwoman/laundress*** *who fell into alcoholism towards the end of her life .*

Horror music is mainly used to complete ***a movie/movies*** *.*

Figure 4. Examples of ambiguities, each illustrating two diacritization variants (bold), both valid in a given context.

due to erroneous words. Therefore, such words were also marked by the annotators and subsequently removed from our analysis.

An example of a final annotation item presented to an annotator is illustrated in Figure 5.

To create the annotation items, we concatenated data from all domains, both the original Wikipedia data (Náplava et al., 2018) and other domains (Section 5.1) and we further considered those words in which the results of our system did not match target word. Before annotation, we automatically filtered out some cases:

- Predictions, in which the system and the target words are variants (as marked by MorphoDita (Straková et al., 2014)) were automatically marked correct.
- Predictions, in which the target word was marked as non-existing by MorphoDiTa, while the system word was marked as Czech, were considered dubious and removed from our analysis.

For the remaining 4702 words, two annotation items were created: one with the predicted word and one with the gold reference word in the position of the annotated *Current Word*. The annotation process took circa 70 hours.

The basic analysis of the annotated system errors is the following: There are 4702 wrongly diacritized words in the all our data concatenated. Annotations revealed that 960 of the mispredicted words contain a non-diacritical error and we do not consider

Předpřechozí věta	Popisujeme sítě , které nepoužívají sdílený přenosový prostředek .
Předchozí věta	Přenosové rychlosti se velmi liší podle typu sítě .
Začátek aktuální věty	Začínají na desítkách kilobitů za sekundu , ale dosahují i
Aktuální slovo	**rychlosti**
Konec aktuální věty	řádu několik gigabitů za sekundu .
Následující věta	Příkladem takové sítě může být internet .
Věta po následující větě	Mezi rozlehlé sítě patří :
Je správně vůči aktuální větě:	Ano
Je správně vůči cel. kontextu:	Ne
Obsahuje překlep:	Ne

English translation

Before Previous Sentence:	*We describe networks that do not use a shared transmission medium .*
Previous sentence:	*Transmission speeds vary greatly depending on the type of network .*
Current Sentence Start:	*They start at tens of kilobits per second , but also reach*
Current Word:	***speeds***
Current Sentence End	*of the order of a few gigabits per second .*
Next Sentence:	*An example of such a network is the Internet.*
After Next Sentence:	*Large networks include :*
Is Correct w.r.t. Cur. Sentence:	*True*
Is Correct w.r.t. Whole Context:	*False*
Contains Spelling Typo:	*False*

Figure 5. Annotation item example. The annotator marks whether the word ”rychlosti” is correct given a context of the current sentence, whether it is still correct in the context of two previous and two following sentences and whether it contains a typo.

them further, as mentioned above. The remaining 3742 mispredicted words can be categorized as follows:

- System correct, Gold correct: 19% (694 of 3742) – plausible variants
- System correct, Gold wrong: 25% (964 of 3742) – system corrects data error
- System wrong, Gold wrong 1% (31 of 3742) – uncorrected error in data
- System wrong, Gold correct 55% (2 084 of 3742) – real errors

Interestingly, the annotations revealed that about 44% of errors are not errors at all. In 694 cases (19%) both the system word and the gold word are correct, which is justified by the plausible variants. In 964 cases (25%) the original gold annotation was wrong whereas the system annotation was correct, which means that the system effectively corrected some of the errors in the original data. The remaining 31 cases are for neither the system nor the gold word being correct. Finally, the annotations confirmed 2084 real system errors, which we postpone for a more detailed analysis in the following Section 5.3.

Plausible variants, which constitute 19% of the annotated errors, are the most interesting item. Please note that our criterion for plausible variant was strict: only

Domain	Original	Annotated	Annotated w/o annotated typos
Wiki	99.22	99.49	99.66
Natives Formal	99.50	99.75	99.75
Natives Informal	99.12	99.53	99.62
Romi	99.11	99.46	99.54
Second Learners	99.18	99.73	99.79

Table 3. Alpha-word accuracy of Czech model on 5 datasets from various domains.

cases ambiguous both in the sentence and document context were marked as plausible variants. Circa 72% percent of these words share a common lemma. As Table 4.a and Table 5.a show, singular/plural ambiguities by far most often arise in inanimate masc. genitive (*programu/programů*, *šachu/šachů*). Another common ambiguity is passive participle vs. adjective (*založena/založená*), generally known to be difficult for diacritization disambiguation (Zeman, 2016). More interesting examples are given in Table 4.a and Table 5.a.

To conclude, we use the collected annotations to refine our previous results, which we display in Table 3. When considering all annotated words, including those preprocessed with MorphoDiTa, we achieve 35% to 67% error reduction. When omitting words newly marked by human annotators as containing another (non-diacritical) error, the error rate gets additionally reduced by up to 33%.

5.3. Analysis of Real Errors

We follow with a morphological analysis of the remaining confirmed errors, which constitute 55% of the annotated mispredictions. To determine the morphological categories of the erroneously predicted words, we use UDPipe (Straka et al., 2019) to generate morphological annotations for all words in model hypotheses and gold sentences. We then inspect the most frequent confusions between the system and the gold morphological annotations of words, using the Universal POS tags and Universal features (Nivre et al., 2020).

The annotations confirmed an interesting discourse phenomenon: a word can be correctly diacritized in multiple ways given the context of its sentence, however only a single correct diacritization variant exists if a wider context is taken into account. There are 50 such annotated cases; two examples are displayed in Figure 6. Although this phenomenon is interesting from a discourse perspective, its low proportion to actual errors (50 of 2084) indicates that it is quite rare. This implies that training models on longer texts (we currently train our model on examples comprising maximally 128 subwords – see Section 3.2) does not promise potential for overall improvement. Finally, we offer a categorization of such ambiguities by means of the Universal POS tags and Universal features (Nivre et al., 2020) in Table 4.b and Table 5.b, respectively.

The remaining errors are a mix of complicated disambiguation cases or rare named entities. The most frequent errors bear similarity to plausible variants (compare Table 5.a and Table 5.c), only with a different order of appearance. Unlike plausible variants (Table 5.a), most frequent mismatches occur already at the level of lemmas (*stát*/*stať*, *že*/*ze*, see Table 5.c). Second most frequent cases are rare named entities (*Sokrates*/*Sókratés*, *Aristoteles*/*Aristotelés*, *Diogenés*/*Díogenés*). Number is again often hard to disambiguate in inanimate masc. genitive (*milionu*/*milionů*, *reproduktoru*/*reproduktorů*, *dokumentu*/*dokumentů*), followed by fem. case (*ji*/*jí*, *ni*/*ní*, *zemi*/*zemí*).

6. Conclusion

We implemented a model for diacritics restoration based on BERT that outperforms previous state-of-the-art models. Further analysis on Czech data collected from additional, noisy domains shown that the model exhibits strong performance regardless the domain of the data.

We further annotated all reported mispredictions in Czech and found out that more than one correct variant is sometimes possible. Rarely, disambiguation on document level is necessary to distinguish between variants correct within the sentence context. We elaborated on these phenomena using morphological annotations and utilized them to further analyse real confirmed errors of the systems.

As for future work, we propose experimenting with a single joint model for a subset of languages, despite our initial unsuccessful attempts at training a single model for all languages, including an introduction of a larger XLM-Roberta model (Conneau et al., 2020).

Acknowledgements

The work described herein has been supported by and has been using language resources stored by LINDAT/CLARIAH-CZ project of the Ministry of Education, Youth and Sports of the Czech Republic (project No. LM2018101) and also by OP VVV LINDAT/CLARIAH-CZ EXTENSION project of the Ministry of Education, Youth and Sports of the Czech Republic (project No. CZ.02.1.01/0.0/0.0/18_046/0015782). It has further been supported by the Grant Agency of the Czech Republic, project EXPRO LUSyD (GX20-16819X), and Mellon Foundation project No. G-1901-06505, by SVV project number 260 575 and by GAUK 578218 of the Charles University.

We are very grateful to our anonymous reviewers for their valuable comments and corrections.

Tento motiv může být ovlivněn sibiřským šamanismem a průvodce pak má funkci psychopompa .

Kromě bohů znali pohanští Slované i celou řadu nižších bytostí , nazývány byly většinou slovem běs či div , které souvisí s indickým déva .

K nejvýznamnějším patří zmiňované **víly/vily** .

V různých podáních existují víly lesní , vzdušné , horské a také víly zlé .

Existují další ženské bytosti jim podobné , patří mezi ně především rusalky , divé ženy nebo divoženky doprovázené divými muži .

Další dokumenty týkající se Jana Žižky z Kalichu jsou dva listy odeslané z kláštera ve Vilémově datované k 16. březnu a 1. dubnu 1423 .

Slepý vojevůdce v nich vyzývá své straníky z orebského svazu k poradě naplánované na 7. či 8. dubna do Německého Brodu .

Z **dopisů/dopisu** je patrné , že se pokoušel dokonaleji zorganizovat husitskou vojenskou moc , pro boj s domácím i zahraničním nepřítelem .

O čtrnáct dní později Žižka spolu s orebity vedl válku se spojenci krále Zikmunda , zejména na Bydžovsku s panem Čeňkem z Vartenberka .

Tohoto šlechtice s jeho leníky a spojenci porazil 20. nebo 23. dubna v bitvě u Hořic , načež dál pokračoval v plenění jeho zboží .

English translation

This motif can be influenced by Siberian shamanism , and the guide then has the function of a psychopomp .

Apart from the gods, the pagan Slavs knew a number of lower beings , mostly called Raver or Wonder , which is related to Indian deva .

*Among the most important are the mentioned **fairies/villas**.*

There are wood fairies, air fairies , mountain fairies , and also evil fairies in various forms .

There are other female beings similar to them , they include mainly mermaids , wild women or witches accompanied by wild men .

Other documents concerning Jan Žižka of the Kalich are two letters sent from the monastery in Vilémov dated March 16 and April 1 , 1423 .

In them , the blind military leader invites his party members from the Orebic Union to a meeting scheduled for April 7 or 8 in Německý Brod .

*The **letter shows/letters show** that he has tried to better organize Hussite military power , to fight both domestic and foreign enemies.*

Fourteen days later , Žižka , together with the Orebits , waged war with King Zikmund's allies , especially in the Bydžov region with Mr. Čeněk of Vartenberk .

He defeated this nobleman with his feoffees and allies on April 20 or 23 at the Battle of Hořice , after which he continued to plunder his goods .

Figure 6. Two examples of ambiguous diacritization determined by document context.

Type	Count	Examples
NOUN ↔ NOUN	406	program[uů], šach[uů], text[uů]
ADJ ↔ ADJ	162	znám[áa], založen[aá], schopn[ií]
ADV ↔ ADJ	59	stejn[ěé], krásn[ěé], běžn[ěé]
PROPN ↔ PROPN	31	Aristotel[eé]s, Sokrates/Sókratés, J[aá]n
VERB ↔ VERB	20	zamýšlím/zamyslím, odráží/odrazí, os[ií]dlují
ADJ ↔ VERB	3	vznikl[áa], rádi/radí, splaskl[áa]
NOUN ↔ ADJ	2	přesvědčen[íi], očištěn[íi]
ADJ ↔ NOUN	2	veden[ií], považován[ií]
DET ↔ DET	2	jej[íi]ch, svoj[íi]

(a) Plausible variants.

Type	Count	Examples
NOUN → NOUN	32	stát/stať, objekt[uů], pulsar[uů]
VERB → VERB	4	narazí/naráží, řekn[ěe]te, žij[íi]
DET → DET	3	jej[ií]ch
ADJ → ADV	3	současn[éě], pravé/právě, praktick[ýy]
ADJ → ADJ	2	znám[áa], žádanou/zadanou
ADV → ADJ	2	stejn[ě/é]
NOUN → VERB	1	mysl[ií]

(b) Disambiguation from document context.

Type	Count	Examples
NOUN → NOUN	1596	stát/stať, lid[íi], program[uů]
PROPN → PROPN	587	Aristotel[eé]s, Sokrates/Sókratés, Kast[ií]lie
ADJ → ADJ	521	znám[aá], založen[aá], říd[ií]cí
VERB → VERB	193	m[ůu]že, M[aá]m, m[aá]
ADJ → ADV	134	krásn[éě], hezk[ýy], dobré/dobře
PRON → PRON	129	j[íi], n[íi], n[íi]ž
ADV → ADJ	112	stejn[ěé], pěkn[ěé], Obvykl[eé]
DET → DET	59	jej[íi]ch, svoj[ií], naš[ií]
NOUN → ADJ	47	mobiln[ií], brány/braný, češka/česká

(c) Real errors.

Table 4. Error categorization with universal POS. The context-dependent morphological annotations were obtained automatically using UDPipe.

Type	Count	Examples
Number	325	program[uů], šach[uů], objekt[uů]
Passive participle / adjective + more features	116	založen[aá], vzdálen[aá], nazývan[aá]
Lemma	82	l[eé]ty, mas[ií]vu, p[ée]rových
Adj ↔ Adv	59	stejn[éě], krásn[éě]
Variant + more features	31	znám[áa], schopn[ií], spokojen[íi]
Case	25	dr[aá]hami, dr[aá]hách, č[aá]rou
Lemma + more features	21	zamýšlím/zamyslím, ná[sš], pacht[uů]
Lemma, NameType	20	Aristotel[eé]s, Sokrates/Sókratés, [ÍI]lias
Case, Number	8	boh[ůu], násobk[uů], funkc[íi]
Number, Person	5	považuj[íi], věnuj[ií], kupuj[ií]

(a) Plausible variants.

Type	Count	Examples
Lemma + more features	15	stát/stať, tvář/tvar, pravé/právě
Number	15	objekt[ůu], pulsar[uů], muzikál[ůu]
Lemma	6	řazení/ražení, v[ií]ly
Adj ↔ Adv	4	stejn[ěé], současn[éě], praktick[ýy]
Case, Gender, Number	3	jej[ií]ch
Number, Person	2	narazí/naráží

(b) Disambiguation from document context.

Type	Count	Examples
Lemma + more features	924	stát/stať, [čc], [žz]e
Lemma, named entity + more features	382	D[ií]ogenés, Hal/Ħal, Dvořák/Dvorak
Number	226	milion[uů], reproduktor[ůu], dokument[ůu]
Case	149	j[ií], n[íi], zem[íi]
Adj ↔ Adv	132	pěkn[éě], česk[ýy], současn[éě]
Passive participle / adjective + more features	37	spojen[aá], pojmenovan[áa], prodaný/prodány
Case, Number	27	referent[uů], Dvořák[ůu], akademi[íi]
Case, Gender, Number	16	jej[íi]ch, j[íi]m
Number, Person	15	píš[ií], pracuj[ií], žij[íi]
Variant + more features	8	znám[áa], schopn[áa], hodn[áa]

(c) Real errors.

Table 5. Error categorization with extended Universal Features. The first column (Type) is the (primary) difference between the context-dependent feature sets of the system word and the gold word.

Bibliography

Adali, Kübra and Gülşen Eryiğit. Vowel and diacritic restoration for social media texts. In *Proceedings of the 5th Workshop on Language Analysis for Social Media (LASM)*, pages 53–61, 2014. doi: 10.3115/v1/W14-1307.

AlKhamissi, Badr, Muhammad N ElNokrashy, and Mohamed Gabr. Deep Diacritization: Efficient Hierarchical Recurrence for Improved Arabic Diacritization. *arXiv preprint arXiv:2011.00538*, 2020.

Alqahtani, Sawsan, Ajay Mishra, and Mona Diab. Efficient Convolutional Neural Networks for Diacritic Restoration. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1442–1448, 2019. doi: 10.18653/v1/D19-1151.

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

Belinkov, Yonatan and James Glass. Arabic diacritization with recurrent neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2281–2285, 2015. doi: 10.18653/v1/D15-1274.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 2020.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 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)*, pages 4171–4186, 2019.

Madhfar, Mokhtar and Ali Mustafa Qamar. Effective Deep Learning Models for Automatic Diacritization of Arabic Text. *IEEE Access*, 2020.

Mubarak, Hamdy, Ahmed Abdelali, Hassan Sajjad, Younes Samih, and Kareem Darwish. Highly effective Arabic diacritization using sequence to sequence modeling. 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)*, pages 2390–2395, 2019.

Náplava, Jakub, Milan Straka, Pavel Straňák, and Jan Hajic. Diacritics restoration using neural networks. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Nga, Cao Hong, Nguyen Khai Thinh, Pao-Chi Chang, and Jia-Ching Wang. Deep Learning Based Vietnamese Diacritics Restoration. In *2019 IEEE International Symposium on Multimedia (ISM)*, pages 331–3313. IEEE, 2019. doi: 10.1109/ISM46123.2019.00074.

Nivre, Joakim, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. Universal Dependencies v2: An Evergrowing Multilingual Treebank Collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May 2020.

European Language Resources Association. URL https://www.aclweb.org/anthology/2020.lrec-1.497.

Nuţu, Maria, Beáta Lőrincz, and Adriana Stan. Deep learning for automatic diacritics restoration in romanian. In *2019 IEEE 15th International Conference on Intelligent Computer Communication and Processing (ICCP)*, pages 235–240. IEEE, 2019. doi: 10.1109/ICCP48234.2019.8959557.

Orife, Iroro. Attentive Sequence-to-Sequence Learning for Diacritic Restoration of YorùBá Language Text. *Proc. Interspeech 2018*, pages 2848–2852, 2018. doi: 10.21437/Interspeech.2018-42.

Rychalska, Barbara, Dominika Basaj, Alicja Gosiewska, and Przemysław Biecek. Models in the Wild: On Corruption Robustness of Neural NLP Systems. In *International Conference on Neural Information Processing*, pages 235–247. Springer, 2019. doi: 10.1007/978-3-030-36718-3_20.

Straka, Milan, Jana Straková, and Jan Hajič. Czech Text Processing with Contextual Embeddings: POS Tagging, Lemmatization, Parsing and NER. In Ekštein, Kamil, editor, *Text, Speech, and Dialogue*, pages 137–150, Cham, 2019. Springer International Publishing. doi: 10.1007/978-3-030-27947-9_12.

Straková, Jana, Milan Straka, and Jan Hajič. Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 13–18, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-5003. URL http://www.aclweb.org/anthology/P/P14/P14-5003.pdf.

Zeman, Dan. DIAKRITIZACE TEXTU. In Petr Karlík, Marek Nekula, Jana Pleskalová (eds.), editor, *CzechEncy - Nový encyklopedický slovník češtiny*. Nakladatelství Lidové noviny, Praha, Czech Republic, 2016.

Address for correspondence:
Jakub Náplava
naplava@ufal.mff.cuni.cz
Malostranské náměstí 25
118 00 Praha
Czech Republic

Character Transformations for Non-Autoregressive GEC Tagging

Milan Straka and **Jakub Náplava** and **Jana Straková**
Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics
{straka,naplava,strakova}@ufal.mff.cuni.cz

Abstract

We propose a character-based non-autoregressive GEC approach, with automatically generated character transformations. Recently, per-word classification of correction edits has proven an efficient, parallelizable alternative to current encoder-decoder GEC systems. We show that word replacement edits may be suboptimal and lead to explosion of rules for spelling, diacritization and errors in morphologically rich languages, and propose a method for generating character transformations from GEC corpus. Finally, we train character transformation models for Czech, German and Russian, reaching solid results and dramatic speedup compared to autoregressive systems. The source code is released at https://github.com/ufal/wnut2021_character_transformations_gec.

1 Introduction

The current state of the art for grammatical error correction (GEC) is achieved with encoder-decoder architectures, leveraging large models with enormous computational demands (Grundkiewicz et al., 2019; Rothe et al., 2021). As such autoregressive approach is slow on inference and is impossible to parallelize, it has recently been suggested to replace autoregressive sequence-to-sequence decoding with per-token tagging to enable parallel decoding, achieving a dramatic speedup by a factor of 10 in NMT (Gu et al., 2018) and very recently, also in GEC (Omelianchuk et al., 2020).

Omelianchuk et al. (2020) approaches GEC as a tagging task, discriminating between a set of word-level transformations. The designed set is efficient for English corpora, which rarely contain spelling errors, and for English language, which does not have diacritization marks and its morphology is very modest compared to morphologically rich languages such as Czech or Russian. Using a set of word-level transformations designed for English, all character-level corrections would have to be handled by generic word-for-word REPLACE rule, leading to an explosion of rules.

We therefore suggest character transformations on subword level. Moreover, our transformations are automatically inferred from the corpus as opposed to being manually designed. Our approach has the following advantages:

- character-level errors, such as diacritics, spelling and morphology are handled,
- the transformations can be shared between subwords, preventing an explosion of rules,
- the transformations are generated automatically from corpus for each language.

We present an oracle analysis of various transformations sets at different levels, in English and three other languages: Czech, German and Russian. We find that the word-level set of rules may be suboptimal for morphologically rich languages and corpora with spelling errors and diacritics.

Finally, we train models with character transformations for non-autoregressive grammatical error correction in Czech, German and Russian, reaching solid results and dramatic speedup compared to autoregressive systems.

2 Related Work

Awasthi et al. (2019) propose an alternative to popular encoder-decoder architecture for GEC: a sequence-to-edit model which labels words with edits. Its advantage is parallel decoding while keeping competitive results. Mallinson et al. (2020) introduce a framework consisting of two tasks: *tagging*, which chooses and arbitrarily reorders a subset of input tokens to keep, and *insertion*, which in-fills the missing tokens with another pretrained masked language model. Omelianchuk et al. (2020) develop custom, manually designed, per-token *g-transformations*.

We further improve the sequence-to-edit model

Proceedings of the 2021 EMNLP Workshop W-NUT: The Seventh Workshop on Noisy User-generated Text, pages 417–422
November 11, 2021. ©2021 Association for Computational Linguistics

Algorithm 1: Extended LCS Alignment

```
Input: input subwords $\boldsymbol{s}$, gold characters $\boldsymbol{g}$
$\boldsymbol{w}[:,:] \leftarrow \mathbf{0}$
for $i \leftarrow |\boldsymbol{s}|$ to $1$ do
    for $j \leftarrow |\boldsymbol{g}|$ to $1$ do
        $\boldsymbol{w}[i,j] \leftarrow \boldsymbol{w}[i+1,j]$
        for $l \leftarrow 1$ to $\max(|\boldsymbol{g}|-j, 8+3|\boldsymbol{s}[i]|)$ do
            $g \leftarrow \boldsymbol{g}[j:j+l]$
            if $g$.isspace() then continue
            $c \leftarrow \begin{cases} 1 & \textbf{if } \boldsymbol{s}[i]=g \\ 0.75 & \textbf{if } \boldsymbol{s}[i].\mathit{strip()}=g.\mathit{strip()} \\ 0.5 \cdot \text{LevenshSimilarity}(\boldsymbol{s}[i],g) \end{cases}$
            $\boldsymbol{w}[i,j] \leftarrow \max(\boldsymbol{w}[i,j], c+\boldsymbol{w}[i,j+1])$
        end
    end
end
return alignment with weight $\boldsymbol{w}[0,0]$, as in LCS
```

Input	gatherin		leafes	
Correct	Gathering		leaves	
Subwords	␣gathe	rin	␣lea	fes
string-at-word	␣Gathering		␣leaves	
string-at-subword	␣Gathe	APP. g	KEEP	ves
char-at-word	APPEND g, UPPERC. 2^{nd}		REPL. 3^{rd} *from end* *with* v	
char-at-subword	UPPERC. 2^{nd}	APP. g	KEEP	REPL. 1^{st} *with* v

Figure 1: Example of the four types of transformations.

with an attempt at non-autoregressive grammatical error correction for languages other than English, with *character* transformations applied at *subwords*, inferred *automatically* from parallel GEC corpus.

3 Transformations

Given that we encode an input sentence using BERT (Devlin et al., 2019), it is natural to represent it using a sequence of *subwords*. We prepend a space to every first subword in a word and use no special marker for other subwords. Note that the subwords might not correspond directly to parts of input, because the *bert uncased* model strip input casing and diacritics.

3.1 Alignment

In order to *automatically* encode the gold data via *character* transformations, we first align the input subwords and the corrected sentence. We compute the alignment with Algorithm 1, which is an extended version of LCS, where each subword is aligned with a *sequence* of gold characters. We ignore casing, diacritics and consider all punctuation equal during the alignment, and bound the maximum length of a correction (number of characters aligned to a single input subword) by $8+3\cdot$*input subword character length* for efficiency.

3.2 Transformations

We consider four kinds of transformations, differing in two dimensions – the granularity of the transformation and the unit it is applied on:

- character transformations applied on each subword separately (*char-at-subword*),
- character transformations applied on each complete word (*char-at-word*),
- string transformations applied on each subword (*string-at-subword*),
- string transformations applied on each complete word (*string-at-word*).

In such terminology, the transformations proposed by Awasthi et al. (2019) and Omelianchuk et al. (2020) can be referred to as *string-at-word*. An example of the described transformation types is illustrated in Figure 1.

To apply a transformation on a complete word, we concatenate the corresponding subwords and aim to produce the concatenation of the subwords' corrections.

A string transformation can be one of *keep*, *replace* by given string or *append* a given string before/after.

A character transformation consists of multiple character edits, which we construct as follows:

1. We start by computing the standard smallest *edit script* between an input subword and a correction. The edit script is a sequence of *inserts*, *replaces* and *deletes*, and we index each edit operation either from the beginning of the input subword (if it involves the first half of it) or from the end of it (otherwise). The edit script is computed on lowercased strings, and in case of *bert uncased* models, also on undiacritized strings.
2. Afterwards, the unmodified input subword (i.e., including casing in case of *bert cased* models) is processed by the edit script, obtaining a correction with possibly incorrect casing. If some incorrectly lowercased charac-

ters are indeed present, *uppercase* operations are added, each indexing a single character either from the beginning of the correction (if the character is in the first half of it) or from the end of it (otherwise).

3. Finally, for *bert uncased* models, we still need to handle missing diacritics. We achieve it analogously to step 2, adding missing diacritical marks with operations indexing single characters again either from the beginning of the correction (if the character is in the first half) or from the end of it.

The reason for special handling of casing (and diacritization for *bert uncased* models) is that the proposed rules are more general, allowing to capture for example corrections *go→Going* and *walk→Walking* with a single rule *append "-ing", uppercase first*.

4 Transformations Upper-bound F-score

To assess the effect of number and type of transformations, we compute the potential maximum $F_{0.5}$ score with the MaxMatch M^2 scorer (Dahlmeier and Ng, 2012).

We generate transformation dictionary from the training portion of the following GEC corpora: CoNLL-2014 shared task for English (Ng et al., 2014), AKCES-GEC (Náplava and Straka, 2019) for Czech, FALKO-MERLIN GEC (Boyd, 2018) for German and RULEC-GEC (Rozovskaya and Roth, 2019) for Russian. The sizes of the datasets are quantified in Table 1.

Language	Dataset	Sentences
Czech	AKCES-GEC	47 371
German	Falko-MERLIN	24 077
Russian	RULEC-GEC	12 480

Table 1: GEC datasets used for constructing rules and for evaluation, including their size.

We also generate transformations from synthetic data augmentation used for training (Section 5); to prevent the explosion of the transformation dictionary, we consider only 1000 synthetic sentences, except for Russian, which employs 5000 synthetic sentences because of very small authentic data. Finally, we add a special *uncorrectable error* dictionary element, indicating an error that cannot be corrected by any dictionary transformation.

To encode a gold correction with a transformation, we first try looking it up in the dictionary. If it is not present, we go through all dictionary transformations in random order, accepting the first one producing the correct output. If no transformation match, we resort to the *uncorrectable error* (during prediction, it will be replaced by the input token).

We show all combinations (*character/string* at *words/subwords*), using cased and uncased mBERT, in Figure 2. Clearly, character transformations applied at subwords (*char-at-subword*, green) have the highest potential in terms of upper-bound $\mathrm{F}_{0.5}$ in all four languages. At the same time, word replacements (*string-at-word*, red) do not scale well. This effect is emphasized in morphologically-rich Czech and Russian, for which the upper-bound string replacement $\mathrm{F}_{0.5}$ (*string-at-word*, red) falls below the current GEC systems state-of-the-art $\mathrm{F}_{0.5}$ (horizontal dotted line).

5 Experiments

We train the character subword GEC tagging model using the *char-at-subword* transformation, which have achieved the best upper-bound score.[1] We train the character subword GEC tagging model (*char-at-subword*) for Czech, German and Russian, in two stages: First, models are trained on a large synthetic corpus, generated by a reimplementation of Náplava and Straka (2019). Then, the models are finetuned on a mixture of synthetic and authentic data in ratio 1:2. The authentic data used in the second stage are AKCES-GEC (Náplava and Straka, 2019) for Czech, FALKO-MERLIN GEC (Boyd, 2018) for German and RULEC-GEC (Rozovskaya and Roth, 2019) for Russian.

The model is based on a pretrained BERT encoder (Devlin et al., 2019), specifically *bert-base-multilingual* (*uncased* for Czech and German, *cased* for Russian). After encoding the tokens, we add a simple softmax classifier that projects embeddings for each subword into a distribution over a set of transformations (Section 3) generated from authentic data with a limited addition from synthetic data (Section 4). We generate 7.7k transformations for Czech, 4.3k transformations for German and 3.1k transformations for Russian.

GEC models based on Transformer and BERT-encoder were shown to perform better when applied iteratively (Lichtarge et al., 2018;

[1] We also performed preliminary experiments with *char-at-word* GEC tagging model, and it performed worse than using the *char-at-subword* transformations.

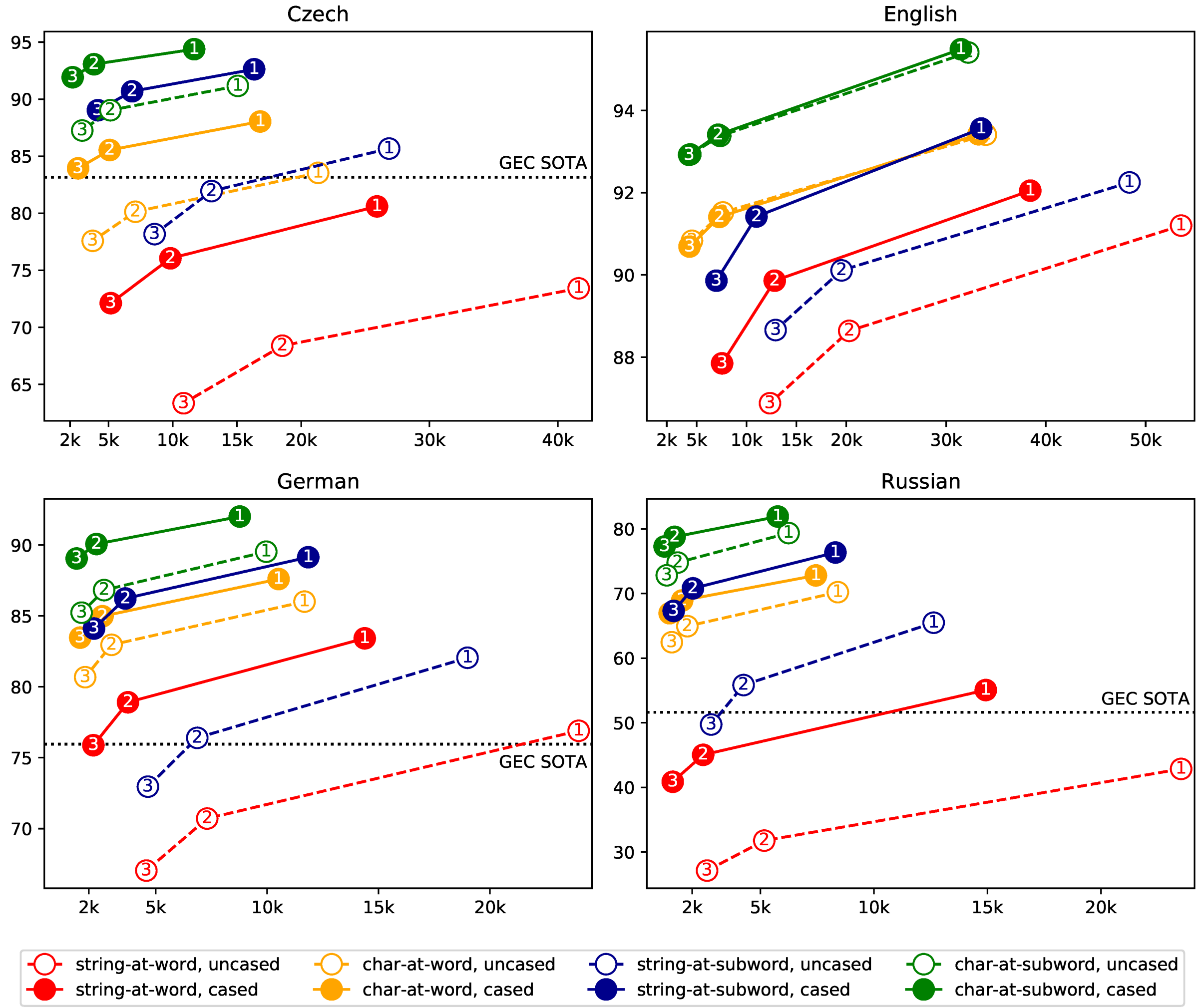


Figure 2: $F_{0.5}$ depending on number and type of transformations, if all transformations were correctly predicted (upper-bound). Up and right is better (higher $F_{0.5}$, fewer rules), down and left is worse (lower $F_{0.5}$, more rules). Circled numbers ①, ② and ③ denote that we kept transformations present at least once, twice or three times in the training data, respectively (larger means less transformations).

Omelianchuk et al., 2020). Therefore, we experiment with multiple iterations and report results both for single iteration and four iteration phases after which we did not observe significant improvements.

We train both the fully-connected network and BERT with AdamW optimizer (Loshchilov and Hutter, 2019) which minimizes the negative log-likelihood. Both for pretraining and finetuning, the learning rate linearly increases from 0 to $5 \cdot 10^{-5}$ over the first 10000 steps and linearly decreases to 0 over 20 epochs. We use the batch size of 2048 sentences and clip each training sentence to 128 tokens. We pretrain each model for circa 14 days and finetune it for circa 2 days on Nvidia P5000 GPU and select the best checkpoint according to development set.

We experimented with weighting all classes different from the KEEP instruction by a factor of 3. It turned out effective only for pretraining Russian.

We present the results of our models in Table 2. Compared to autoregressive models of similar size (Náplava and Straka, 2019), our models achieve solid results with large speedup due to the non-autoregressive tagging approach. Obviously, the inflation of model size (Rothe et al., 2021) to enormous size (13B parameters) leads to further improvements at the cost of increased computational demands.

5.1 Runtime Performance

To evaluate the speed-up of the non-autoregressive decoding, we compare the runtime performance of our system to the autoregressive Transformer

Model	Params	$F_{0.5}$
Richter et al. (2012)		58.54
Náplava and Straka (2019)^{synt}	210M	66.59
Náplava and Straka (2019)^{fine}	210M	80.17
Rothe et al. (2021) *base*	580M	71.88
Rothe et al. (2021) *xxl*	13B	83.15
Ours synthetic	172M	64.29
Ours finetuned	172M	72.86
Ours finetuned 4 iterations	172M	75.06

(a) Czech

Model	Params	$F_{0.5}$
Boyd (2018)		45.22
Náplava and Straka (2019)^{synt}	210M	51.41
Náplava and Straka (2019)^{fine}	210M	73.71
Rothe et al. (2021) *base*	580M	69.21
Rothe et al. (2021) *xxl*	13B	75.96
Ours synthetic	170M	44.29
Ours finetuned	170M	62.92
Ours finetuned 4 iterations	170M	65.95

(b) German

Model	Params	$F_{0.5}$
Rozovskaya and Roth (2019)		21.00
Náplava and Straka (2019)^{synt}	210M	40.96
Náplava and Straka (2019)^{fine}	210M	50.20
Rothe et al. (2021) *base*	580M	26.24
Rothe et al. (2021) *xxl*	13B	51.62
Ours synthetic	180M	25.36
Ours finetuned	180M	36.62
Ours finetuned 4 iterations	180M	38.68

(c) Russian

Table 2: Model results

encoder-decoder architecture from Náplava and Straka (2019), which is of comparable size. The measurements are performed using both a CPU-only decoding (performed on a dedicated 32-core Intel Xeon E5-2630) and GPU decoding (measured on an Nvidia Quadro P5000). The results presented in Table 3 show that the non-autoregressive system is four times faster.

Model	Time Per Sentence
T2T	162.34
BERT-GEC	41.26

(a) CPU decoding on a 32-core Intel Xeon

Model	Time Per Sentence
T2T	22.36
BERT-GEC	5.09

(b) GPU decoding on Nvidia Quadro P5000

Table 3: Average time in milliseconds required to process a single sentence in the Czech test set, measured using both (a) CPU decoding and (b) GPU decoding.

6 Conclusion And Future Work

We proposed a character-based method to generate target transformation instructions for GEC tagging models, as an alternative to autoregressive models. We compared the character transformations to previously used word-level transformation instructions and have shown that character-based rules have better coverage and scale better in Czech, German and Russian. Moreover, we trained character-based GEC tagging models for these languages. The source code is available at https://github.com/ufal/wnut2021_character_transformations_gec.

For future work, we propose to investigate ways to generate synthetic data to achieve better coverage of the target transformation set, since the current process for generating synthetic errors is well suited for encoder-decoder models, but may fail to cover certain transformations.

Acknowledgements

The research described herein has been supported by the Ministry of Education, Youths and Sports of the Czech Republic, under the project LINDAT/CLARIAH-CZ (LM2018101). This research was also partially supported by SVV project number 260 575 and GAUK 578218 of the Charles University.

References

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. In *Proceedings of the 2019 Conference on Empirical*

Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4260–4270, Hong Kong, China. Association for Computational Linguistics.

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, Brussels, Belgium. Association for Computational Linguistics.

Daniel Dahlmeier and Hwee Tou Ng. 2012. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jared Lichtarge, Christopher Alberti, Shankar Kumar, Noam Shazeer, and Niki Parmar. 2018. Weakly supervised grammatical error correction using iterative decoding. *arXiv preprint arXiv:1811.01710*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. FELIX: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. Association for Computational Linguistics.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. 2014. The CoNLL-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online. Association for Computational Linguistics.

Michal Richter, Pavel Straňák, and Alexandr Rosen. 2012. Korektor–a system for contextual spell-checking and diacritics completion. In *Proceedings of COLING 2012: Posters*, pages 1019–1028.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. *CoRR*, abs/2106.03830.

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.

Understanding Model Robustness to User-generated Noisy Texts

Jakub Náplava and **Martin Popel** and **Milan Straka** and **Jana Straková**
Charles University,
Faculty of Mathematics and Physics,
Institute of Formal and Applied Linguistics
{naplava,popel,straka,strakova}@ufal.mff.cuni.cz

Abstract

Sensitivity of deep-neural models to input noise is known to be a challenging problem. In NLP, model performance often deteriorates with naturally occurring noise, such as spelling errors. To mitigate this issue, models may leverage artificially noised data. However, the amount and type of generated noise has so far been determined arbitrarily. We therefore propose to model the errors statistically from grammatical-error-correction corpora. We present a thorough evaluation of several state-of-the-art NLP systems' robustness in multiple languages, with tasks including morpho-syntactic analysis, named entity recognition, neural machine translation, a subset of the GLUE benchmark and reading comprehension. We also compare two approaches to address the performance drop: a) training the NLP models with noised data generated by our framework; and b) reducing the input noise with external system for natural language correction. The code is released at https://github.com/ufal/kazitext.

1 Introduction

Although there has recently been an amazing progress in variety of NLP tasks (Vaswani et al., 2017; Devlin et al., 2019) with some models even reaching performance comparable to humans on certain domains (Ge et al., 2018; Popel et al., 2020), it has been shown that the models are very sensitive to noise in data (Belinkov and Bisk, 2017; Rychalska et al., 2019).

Multiple areas of NLP have been studied to evaluate the effect of noise in data (Belinkov and Bisk, 2017; Heigold et al., 2018; Ribeiro et al., 2018; Glockner et al., 2018) and a framework for text corruption to test NLP models robustness is also available (Rychalska et al., 2019). However, all these systems introduce noise in a custom-defined, arbitrary level and typically for a single language.

We suggest modeling natural noise statistically from corpora and we propose a framework with the following distinctive features:

- The error probabilities are estimated on real-world grammatical-error-correction corpora.
- The intended noisiness can be scaled to a desired level and various aspects (types) of errors can be turned on/off to test the NLP systems robustness to specific error types.

Furthermore, we also present a thorough evaluation of several current state-of-the-art NLP systems' with varying level of data noisiness and a selection of error aspects in multiple languages. The NLP tasks include morpho-syntactic analysis, named entity recognition, neural machine translation, a subset of GLUE benchmark and reading comprehension. We conclude that:

- The amount of noise is far more important than the distribution of error types.
- Sensitivity to noise differs greatly among NLP tasks. While tasks such as lemmatization require correcting the input text, only an approximate understanding is sufficient for others.

We also compare two approaches for increasing models robustness to noise: training with noise and external grammatical-error-correction (GEC) preprocessing. Our findings suggest that training with noise is beneficial for models with large capacity and large training data (neural machine translation), while the preprocessing with grammatical-error-correction is more suitable for limited-data classification tasks, such as morpho-syntactic analysis.

Finally, we also offer an evaluation on authentic noise: We assembled a new dataset with authentic Czech noisy sentences translated into English and we evaluate the noise-mitigating strategies in the neural machine translation task on this dataset.

Proceedings of the 2021 EMNLP Workshop W-NUT: The Seventh Workshop on Noisy User-generated Text, pages 340–350
November 11, 2021. ©2021 Association for Computational Linguistics

2 Related Work

Many empirical findings have shown the fact that data with natural noise deteriorate NLP systems performance. Belinkov and Bisk (2017) found that natural noise such as misspellings and typos cause significant drops in BLEU scores of character-level machine translation models. To increase the model's robustness, they trained the model on a mixture of original and noisy input and found out that it learnt to address certain amount of errors. Similar findings were observed by Heigold et al. (2018) who tested machine translation and morphological tagging under three types of word-level errors.

Ribeiro et al. (2018) defined a set of substitution rules that produce semantically equivalent text variants. They used them to test systems in machine comprehension, visual question answering and sentiment analysis. Glockner et al. (2018) created a new test set for natural language inference and showed that current systems do not generalize well even for a single-word replacements by synonyms and antonyms.

Rychalska et al. (2019) implemented a framework for introducing multiple noise types into text such as removing or swapping articles, rewriting digit numbers into words or introducing errors in spelling. They found out in four NLP tasks that even recent state-of-the-art systems based on contextualized word embeddings are not completely robust against such natural noise. They also retrained the systems on noisy data and observed improvements for certain error types.

Similarly to Rychalska et al. (2019), we also developed a general framework that allows to test a variety of NLP tasks. The difference is that we estimate the probabilities of individual error types from real-world error corpora. This makes the generated sentences more similar to what humans would do. Moreover, since we defined the individual error types with no language-specific rules, we can apply it to multiple languages with an available annotated grammatical-error corpus.

Grammatical-error corpora are typically used as training data for estimating error statistics in GEC systems. In a setting similar to ours, Choe et al. (2019); Rozovskaya et al. (2017) also estimated error statistics and used them to generate additional training data for GEC systems. However, compared to our approach, they defined only a small set of predefined error categories and used it specifically for training GEC systems whereas we also use it to asses model performance in noisy scenarios.

Authentic Noise Evaluation The growing interest in developing production-ready machine translation models that are robust to natural noise resulted in the First Shared Task on Machine Translation Robustness (Li et al., 2019). The shared task used the MTNT dataset (Michel and Neubig, 2018), which consists of noisy texts collected from Reddit and their translations between English and French and English and Japanese.

Improving Model Robustness Using Noisy Data Majority of research on improving model robustness is dedicated to training on a mixture of original and noisy data. The same procedure is usually used for generating both the test corpus and training data (Belinkov and Bisk, 2017; Heigold et al., 2018; Ribeiro et al., 2018; Rychalska et al., 2019).

To generate synthetic training data, researchers in machine translation and GEC often use so called back-translation (Sennrich et al., 2016). A reverse model translating in the opposite direction (i.e. from the target language to the source language or from the clean sentence into noisy sentence, respectively) is trained (Rei et al., 2017; Náplava, 2017; Kasewa et al., 2018; Xie et al., 2018). It is then used on the corpus of clean sentences to generate noisy input data. While this approach might generate high-quality synthetic data, it requires large volumes of training data.

We evaluate two approaches to alleviate performance drop on noisy data: We either train the system on a mixture of synthetic (generated statistically from real error corpora) and original authentic data; or we use an external grammatical-error-correction system to correct the noisy data before inputting them to the system itself. We are not aware of any other work that compares these two approaches and we believe that both approaches may be beneficial under certain conditions.

3 Modeling Natural Noise from Corpora

Robustness of NLP models to natural noise would ideally be evaluated on texts with authentic noise, with error corrections annotated by humans. (We present such authentic data evaluation in Section 7.) This perfect-world setting, however, requires an immense annotation effort, as multiple target domains have to be covered by well-educated human annotators for multiple NLP tasks in a range of languages. To ease the annotation burden, we propose a new framework, named KaziText, for introducing natural-like errors in a text.

The core of KaziText is a set of several common error type classes, *aspects* (following naming convention of Rychalska et al., 2019). The *aspects* are composable (can be combined) and the probability of the aspect manifestation as well as the aspect's internal probabilities are estimated from grammatical-error-correction corpora.

3.1 Noising Aspects

One of the main objectives of our error aspects' design was to avoid manually designed rules, especially those derived from a single language. An ideal approach, automatically inferring the aspects themselves, is however limited by the amount of available data. Therefore, we defined a rich set of aspects which can be estimated from the data:

1. **Diacritics** Strip diacritics either from a whole sentence or randomly from individual characters.
2. **Casing** Change casing of a word, distinguishing between changing the first letter and other.
3. **Spelling** Insert, remove, replace or swap individual characters (*wrong* → *worng*) or use ASpell[1] to transform a word to other existing word (*break* → *brake*).
4. **Suffix/Prefix** Replace common suffix (*do* → *doing*) and prefix (*bid* → *forbid*).
5. **Punctuation** Insert, remove or replace punctuation.
6. **Whitespace** Remove or insert spaces in text.
7. **Word Order** Reorder several adjacent words.
8. **Common Other** Insert, replace or substitute common phrases as seen in data (*the* → *a*, *a lot of* → *many*). This is the aspect which should learn language specific rules.

The natural errors found in real-world texts rarely fall into mutually exclusive categories. Casing errors are also spelling errors; common other aspect covers all other aspects. Therefore, some of the aspects naturally overlap. We therefore opted for evaluating the aspects in a *cumulative* manner in the designed order.

When designing the order of the aspects, our goal was to respect the natural inclusion of aspects and also error severity. We therefore start with diacritical-only changes, given that for example in Czech, users may deliberately write without diacritics. We then add casing changes, spelling errors and then suffix/prefix changes (the latter being morphologically motivated spelling errors). The first four aspects do not modify tokenization, making them suitable for tokenization-dependent tasks like POS tagging or lemmatization.

The remaining aspects change the number of tokens or token boundaries. The punctuation, whitespace and word order aspects are relatively independent, with the common other aspect covering all of them and thus being the last one.

3.2 Estimating Noising Aspects Probabilities

We use grammatical-error-correction (GEC) datasets to estimate probabilities of individual aspects. The GEC datasets are distributed in M2 format,[2] which for a tokenized input noisy sentence contains a set of correcting edits. Each correcting edit contains the corresponding input sentence span, the correction itself and the error type. The noising aspect probabilities are estimated by frequency analysis.[3] To accurately model the distribution of amount of errors in different sentences, we also measure the standard deviation of the token edit probability per sentence.

We collected M2 files from various grammatical-error-correction corpora in 4 languages: English, Czech, Russian and German. The majority of annotated content comes from Second Learners of the particular language and in addition, more speaker groups are available in English and Czech:

- English
  - Natives: LOCNESS v2.1 (Granger, 1998)
  - Second Learners: NUCLE (Dahlmeier et al., 2013), FCE (Yannakoudakis et al., 2011), Write & Improve (Yannakoudakis et al., 2018)
- Czech
  - Natives: essays of Czech primary schools students, in submission process
  - Natives Informal: web discussions data, in submission process
  - Second Learners: AKCES-GEC (Šebesta et al., 2019)
  - Romani: AKCES-GEC (Šebesta et al., 2019) – Romani ethnic minority children and teenagers using Czech

[1] http://aspell.net/

[2] GEC file format since the CoNLL-2013 shared task

[3] We refer to the published source code for details.

Language	Corpus	Sentences	Error rate	Domain
English	*NUCLE* (Dahlmeier et al., 2013)	57 151	6.6%	SL
	FCE (Yannakoudakis et al., 2011)	33 236	11.5%	SL
	W&I (Yannakoudakis et al., 2018)	37 704	11.7%	SL
	LOCNESS (Granger, 1998)	988	4.7%	native students
Czech	Romani part of *AKCES-GEC* (Šebesta et al., 2019)	16 030	20.3%	Romani heritage speakers
	SL part of *AKCES-GEC* (Šebesta et al., 2019)	31 341	22.1%	SL essays
	Natives Informal	11 608	15.6%	web discussions
	Natives	7 696	5.8%	native students
German	*Falko-MERLIN* (Boyd, 2018)	24 077	16.8%	SL essays
Russian	*RULEC-GEC* (Rozovskaya and Roth, 2019)	12 480	6.4%	SL, heritage speakers

Table 1: Comparison of used GEC corpora in size, token error rate and domain. SL = second language learners.

- German (Second Learners): Falko-MERLIN GEC Corpus (Boyd, 2018)
- Russian (Second Learners): RULEC-GEC (Rozovskaya and Roth, 2019)

An overview of the sizes and error rates of the datasets above is presented in Table 1.

We call the resulting single file containing all aspect probabilities for one group of speakers a *profile*. The *profile* therefore describes the grammatical style of a particular given group of users, derived from M2 file annotations.

Each profile has a development and test version originating from the respective M2 development and test files. The test profiles are used for synthesising data intended directly for assessing models' performance in noisy setting while the development profile is intended for creating data for training the models.

3.3 Adjusting the Percentage of Token Edits

In order to reach an intended percentage of *token edits*, which directly corresponds to the amount of noise in the generated data, we correspondingly scale the aspects' probabilities. We refer to the percentage of token edits in the original corpus as a *corpus error level*.

3.4 Noising the Data

When noising an input sentence, we first sample a token edit probability from the error amount distribution, scaled according to the required number of token edits. We then introduce the desired aspects with the chosen error level.

We allowed the framework to generate any noising aspect, including adding new tokens, in test sets without token-level gold annotations: neural machine translation, GLUE benchmark, tokens outside named entities in NER and tokens outside the answer in reading comprehension.

When introducing errors into classification test sets with token-level gold annotations, we need to maintain the original tokenization. For this reason, we allowed only the first 4 aspects for the following data: morpho-syntactic analysis, tokens inside named entity spans in NER and tokens inside answers in the reading comprehension task.

All experiments are repeated with 5 different random seeds and we report means with standard deviations.

4 Evaluated Tasks

4.1 Morpho-syntactic Analysis

Model We employed UDPipe (Straka et al., 2019), a tool for morpho-syntactic analysis.

Dataset We used the Universal Dependencies 2.3 (Nivre et al., 2018) corpus (UD 2.3).[4]

Metrics We utilized the following metrics (Zeman et al., 2018) – **UPOS**: coarse POS tags accuracy, **UFeats**: fine-grained morphological features accuracy, **Lemmas**: lemmatization accuracy, **LAS**: labeled attachment score and **MLAS**: combination of morphological tags and syntactic relations.

4.2 Named Entity Recognition

Model Recently published architecture (Straková et al., 2019) was used for NER evaluation.

Dataset For English and German, we evaluated on the standard CoNLL-2003 shared task data (Tjong

[4] Many English UD test set tokens contain casing or spelling errors, propagated into lemmas, rendering such data unsuitable for analysis. We try to use only error-free English test documents and we therefore drop all test documents containing a sentence starting with a lowercase character, keeping more than half of the data. Apart from the lemmatization accuracy, the results for full test set are nearly identical.

Kim Sang and De Meulder, 2003); for Czech, we used a fine-grained Czech Named Entity Corpus 2.0 (Ševčíková et al., 2007) with 46 types of nested entities.

Metric The evaluation metric is F1 score.

4.3 Neural Machine Translation

Model We chose a state-of-the-art Czech-to-English NMT system CUBBITT (Popel et al., 2020), but we trained it on the newest version (2.0) of the CzEng parallel corpus (Kocmi et al., 2020). We trained with batch size of ca. 23k tokens for 550k steps, saved a checkpoint each hour (ca. 4600 steps) and selected the checkpoint with the highest dev-set BLEU (which was at 547k steps).

Dataset We use WMT17 (newstest2017, 3005 sentences)[5] as our development set. Our test set is a concatenation of WMT13, WMT16 and WMT18 (8982 sentences in total).

Metric We evaluate the translation quality with case-insensitive BLEU score.[6]

4.4 GLUE Benchmark

We select a subset of GLUE (Wang et al., 2018) tasks, namely Microsoft Research Paraphrase Corpus (MRPC), Semantic Textual Similarity Benchmark (STS-B), Quora Question Pairs (QQP) and The Stanford Sentiment Treebank (SST-2). We finetune BERT on each of these tasks and evaluate them on various levels of noise.

Model We finetune pretrained BERT with an additional feed-forward neural network with one hidden layer predicting score on particular task's data. We use *bert-base-cased* configuration and HuggingFace's Transformers (Wolf et al., 2019) implementation.

Dataset We use official GLUE datasets as provided by `https://gluebenchmark.com/tasks`.

Metric We report following metrics: F1 for MRPC and QQP, Pearson-Spearman Corr for STS-B and accuracy for SST-2.

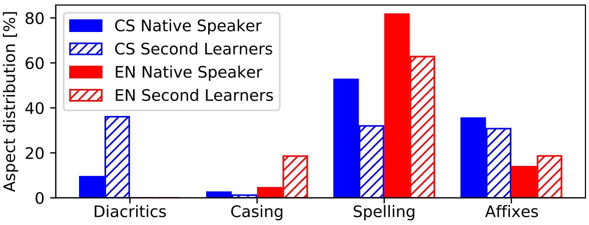


Figure 1: Proportional distribution of the first 4 aspects (diacritics, casing, spelling, affixes) in Czech and English.

4.5 Reading Comprehension

Model We utilize a BERT base architecture with a standard SQuAD classifier on top (Devlin et al., 2019).

Dataset We employ English SQuAD 2 (Rajpurkar et al., 2018) and its Czech translation (Macková and Straka, 2020).

Metric Our experiments are evaluated using F1 score.

5 Robustness to Noise

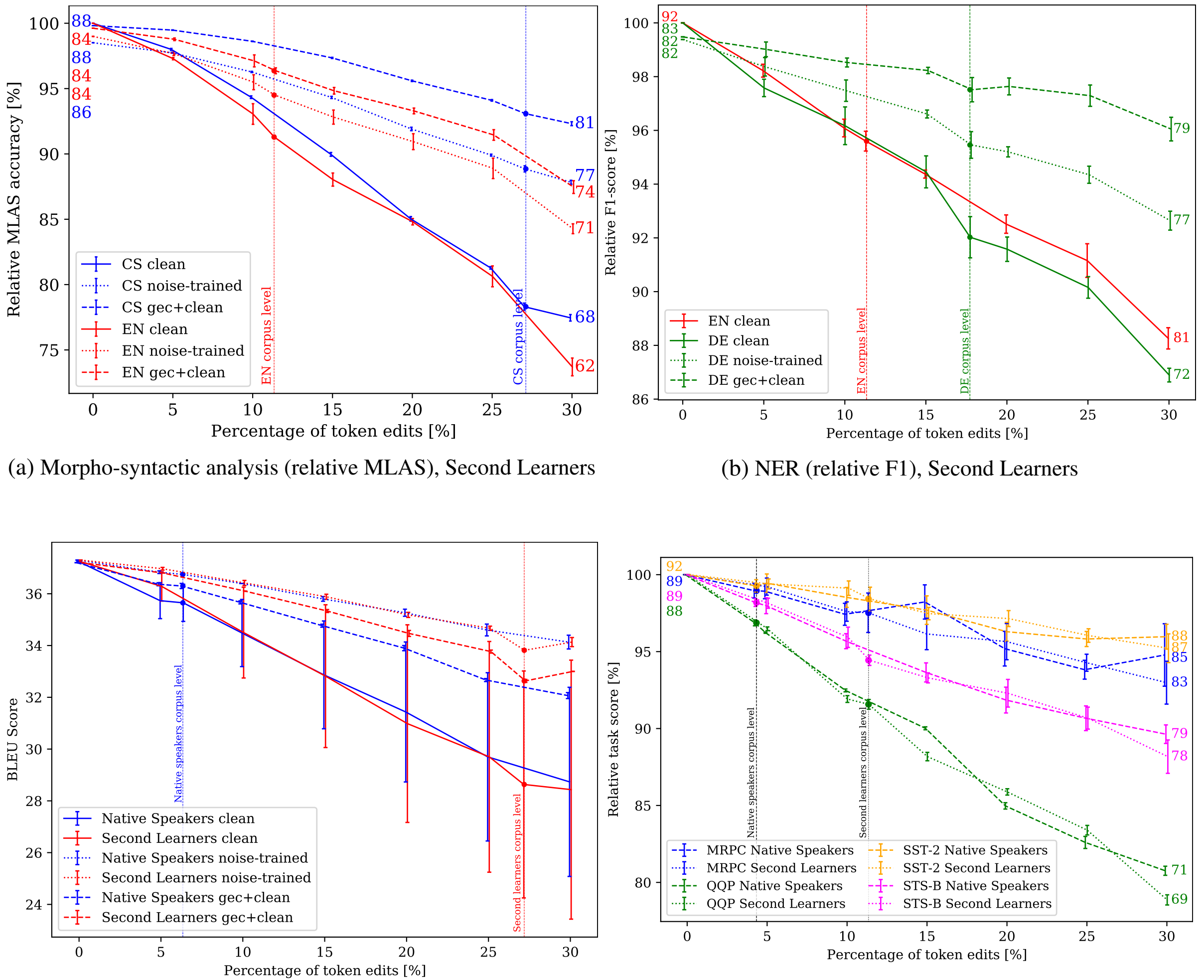
We evaluated the models robustness both to the amount of noise (Figure 2) and to error types (Figures 3 and 4).

A unifying trend can be observed in models performance with respect to increasing percentage of token edits. Solid lines in Figure 2 display the morpho-syntactic MLAS, NER F1 and NMT BLEU on texts with up to 30% of token edits. The relative performance decreases roughly linearly with the amount of token edits, in accordance with previous findings (Rychalska et al., 2019). The tendency is consistent across tasks, languages and profiles: For example, compare the Czech and English Second Learners profiles in morpho-syntactic analysis (Figure 2a) or Czech Native Speakers and Czech Second Learners profiles in the NMT clean model (Figure 2c), which exhibit similar behaviour despite their differing distributions of aspects (Figure 1). This consistency implies that it is the sheer amount of noise rather than the distribution of aspects, that contributes to the model performance deterioration. More results are available in Supplementary Material (Figures S2 and S3).

Estimating the amount of noise is important, as the *corpus error level* differs greatly across languages and profiles. For example, compare the Second Learners profile in English (11.3% token edits) and Czech (27.1%) in Figure 2a, or in Czech, see Native Speakers (6.4%) and Second Learners

[5] `http://statmt.org/wmt17`

[6] We use SacreBLEU (Post, 2018) with signature BLEU+case.lc+numrefs.1+smooth.exp+tok.intl+version.1.4.14. When using the case-sensitive version, the results show similar trends, except for the Casing aspect, which causes more harm to the score, as could be expected. However, it is questionable if copying the "wrong" casing to the translation (e.g. not capitalizing the first word in a sentence or using all-uppercase) should be considered a translation error. We thus opted for case-insensitive BLEU as our primary metric.



(a) Morpho-syntactic analysis (relative MLAS), Second Learners

(b) NER (relative F1), Second Learners

(c) NMT (BLEU), Czech Native Speakers and Second Learners

(d) GLUE (relative task score), English Native Speakers and Second Learners

Figure 2: Increasing percentage of token edits with clean model, noise-trained model and grammatical-error-correction. Numbers near lines are absolute values.

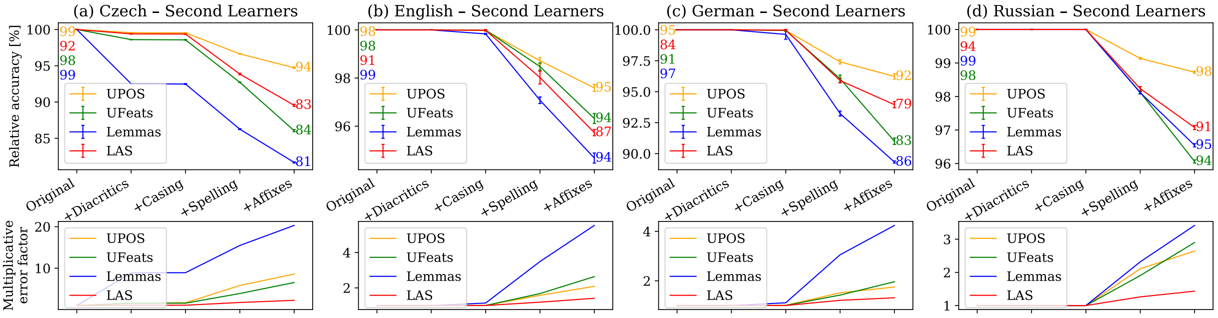
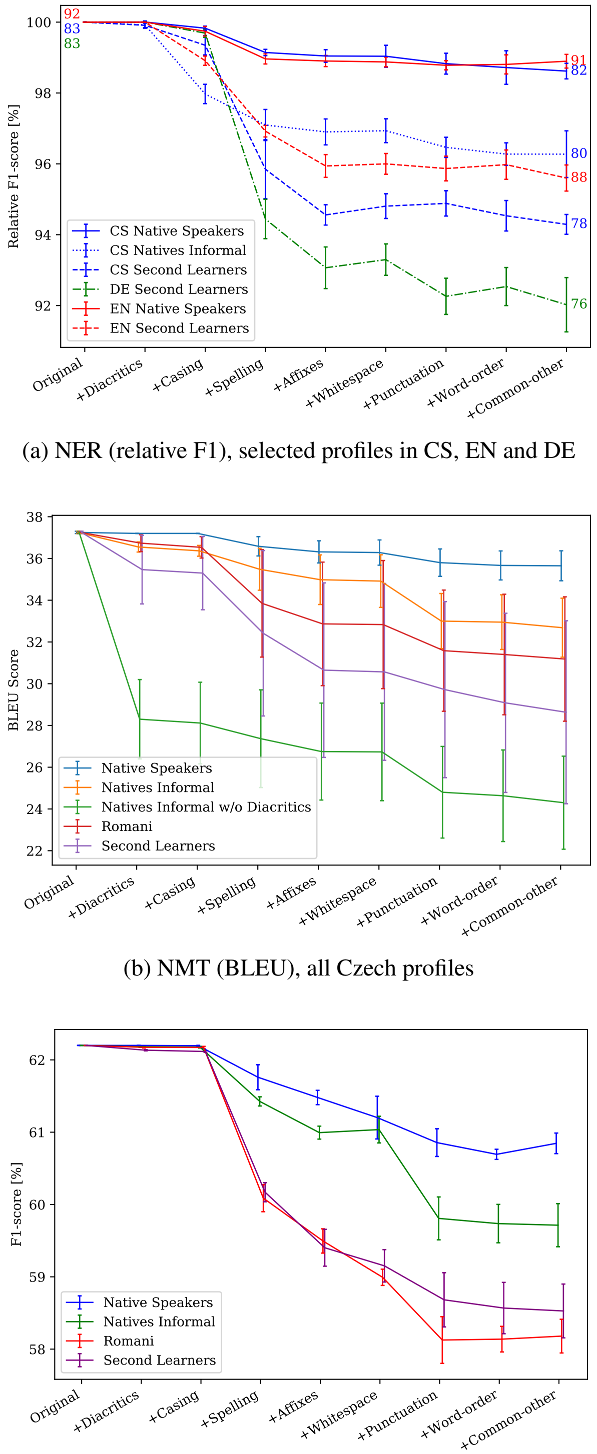


Figure 3: Morpho-syntactic tasks with additive noising aspects in Second Learners profile across (a) Czech, (b) English, (c) German and (d) Russian. The amount of introduced errors is the corpus error level for each aspect. **Upper row** Accuracy relative to original data accuracy, numbers near lines are absolute values. **Lower row** Multiplicative factor of errors to original data error rate.



(a) NER (relative F1), selected profiles in CS, EN and DE

(b) NMT (BLEU), all Czech profiles

(c) Reading comprehension (relative F1), all Czech profiles

Figure 4: Evaluation with additive noising aspects. The amount of introduced errors is the corpus error level for each aspect. Numbers near lines are absolute values.

(27.1%) in Figure 2c. Testing near the estimated noisiness level provides more accurate evaluation of the models' performance.

From a qualitative point of view, spelling and affixes make for the major performance drop in morpho-syntactic analysis (Figure 3), NER (Figure 4a) and NMT (Figure 4b).

Some tasks are more sensitive to certain aspects: Casing is a crucial aspect for NER. This is clearly shown in the Czech Natives Informal profile, which contains text scraped from the internet discussions and contains nontrivial amount of casing errors (Figure 4a). We further elaborate the casing aspect effect on NER in Section S2 in Supplementary Material. In NMT and reading comprehension, errors in punctuation seem to decrease the model performance consistently across all profiles (Figures 4b and 4c, respectively).

For Czech as a language with diacritic marks, diacritics is an interesting aspect. We can see that when it is introduced at a *corpus error level*, the Czech model's performance on Lemmas drops by circa 7 percent. Figure S1 in Supplementary Material further illustrates that performance significantly deteriorates when all diacritics is stripped, which is quite common in informal Web texts. Similarly, to emphasize the effect of the diacritization aspect on NMT, we created a new profile *Natives Informal w/o Diacritics* from the Natives Informal profile by stripping all diacritization. Figure 4b shows that not using diacritics at all results in a performance drop of ca. 10 BLEU points.

Some tasks are more sensitive to noise than others. Lemmatization is the most sensitive to errors (20 times more errors when processing Czech Second Learners texts with a clean model, see Figure 3), which is understandable, given that all lemma characters must be generated correctly from a corrupted surface token. The effect on POS tagging is the least pronounced (Figure 3), although 8 times as many errors in Czech (when processing noisy texts with a clean model) makes the POS tags much less reliable.

6 Noise-coping Strategies

We implemented and evaluated two strategies to alleviate the performance drop on noisy inputs: *external* and *internal* correction. In the *external* correction approach, we use a separately trained grammatical-error-correction model to denoise texts before inputting them to the model itself. In the *internal* correction approach, we instead directly train the model on a combination of noisy and authentic texts.

We hypothesise that the external approach may

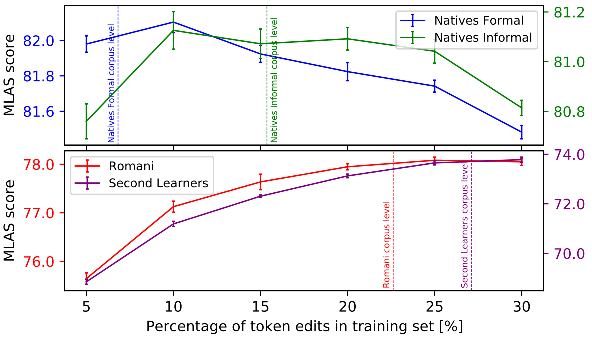


Figure 5: Morpho-syntactic analysis: Training data increasingly noised with each single profile, evaluation with the corresponding profile corpus error level.

be better in scenarios with small amount of annotated data. In such cases, only few iterations over training data are typically performed to prevent overfitting, and we suppose that learning the task itself and denoising at the same time would harm its performance a lot. Contrarily, with enough data and appropriate model capacity, learning the denoising and the task jointly may reduce the amount of potential false positives that might be otherwise proposed by the external language corrector.

6.1 External Correction Model

We use the grammatical-error-correction system of Náplava and Straka (2019) in our experiments. Their models trained on Czech, German and Russian achieve state-of-the-art results and slightly below state-of-the-art results on English. We use their "pretrained" version.

We modified the pipeline of Náplava and Straka (2019) to train on detokenized text. Furthermore, we also trained new grammatical-error-correction models which only make corrections that strictly keep the given tokenization (important in morpho-syntactic annotations). To sum up, we trained two types of grammatical-error-correction models: 1. detokenized error correction model (for NMT) 2. tokenization-preserving error-correction model (for morpho-syntactic tasks and NER).

6.2 Training on Noisy Data

In the *internal* approach to increase model robustness, we train the systems on a mixture of original and noisy data, while keeping the number of training steps unchanged. The noisy data are generated using the KaziText framework operating on development profiles and concatenated to original data.

We noise the training data with appropriately estimated corpus error levels in all our experiments.

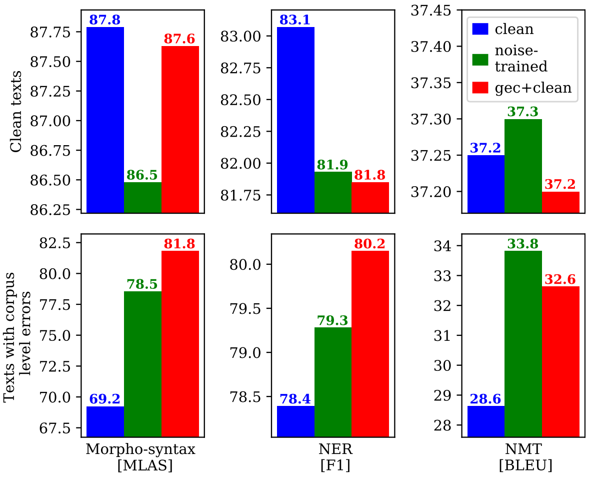


Figure 6: Comparison of three models (clean, noise-trained and GEC-preprocessed) on three tasks in Czech Second Learners profile. **Upper row** Original clean test data. **Lower row** Test data with corpus error level noise.

To illustrate the effect of noise level introduced into training data, we trained the UDPipe on variably noised morpho-syntactic data for all four Czech profiles. In each single profile, we increasingly noised the morpho-syntactic training data and evaluated on the testing data noised with the corresponding profile *corpus error level*. In all cases, the best performance is found near the corpus error level (Figure 5).

When training the NMT model, the best checkpoint on a development set consisting of concatenated standard WMT17 and WMT17 noised with our framework is selected.

We train a single model for each language on a concatenation of noisy data generated by all profiles of the particular language. This makes the final model generalize well across all profiles, although training a single model for each profile could make sense for other scenarios.

6.3 Evaluation

We present the effect of both the *internal* and *external* noise-coping strategies in Figure 2. There are two main points of interest in the graphs: the first one showing performance of models on clean texts and the second one showing model performance on texts with corpus level errors. Additionally, an excerpt showing performance of Czech Second Learners on these two levels is presented in Figure 6.

It is not a surprise that the model trained on clean training data surpasses the noise-coping models on

System	BLEU on Faust	
	Noisy	Cleaned
clean	43.3	50.9
noise-trained	47.0	50.5
gec+clean	44.1	50.4

Table 2: NMT results on authentic user noisy texts. We report BLEU on the Faust-Noisy test set with noisy input sentences and also on Faust-Cleaned that has manually corrected sentences on input.

the clean test data. Adapting to noise clearly comes with a cost. Surprisingly though, the clean model head start is only marginal in the NMT task.

The clean models perform substantially worse than either of the two proposed methods in all three tasks when errors are introduced in the same amount as the corpus error level (marked with vertical lines in Figure 2). Therefore, whenever noisy inputs of particular domain are expected, it is beneficial to adapt to noise using either of the two methods.

With increasing noise, the gap between the clean model and the *external* and *internal* model grows in all three tasks (Figure 2). There is a threshold at which the noise-coping models surpass the clean model for each task. Interestingly, the threshold oscillates around relatively low noise levels up to 5% of token edits.

Finally, we confirm our initial hypothesis that *external* approach with GEC model works better than *internal* approach on low resource tasks: morpho-syntactic analysis and named-entity recognition. The *internal* approach then outperforms *external* approach on machine translation task for which there is a large amount of training data and a model with greater capacity.

7 Evaluating on Authentic User Text

We assembled a new dataset for MT evaluation consisting of 2223 authentic Czech noisy input sentences translated into English, which we release at http://hdl.handle.net/11234/1-3775. The sentences originate from the project FAUST[7] where they were collected from various users of reverso.net. The advantage of this dataset is that in addition to the original Czech noisy sentences, there are manually corrected Czech sentences and manual translations to English.

On this dataset, we evaluate our neural machine translation models from Section 4.3 and Section 6, specifically the *clean* model trained on clean data, *noise-trained* model trained on a mixture of authentic and noised data and their combination with external grammatical-error-correction system. The results of these systems on authentic noisy texts are presented in Table 2. It is evident that noise-trained model outperforms clean model by a large margin on Faust-Noisy data while not losing much precision on Faust-Cleaned data. Similarly to our conclusions in Section 6, the external grammatical-error-correction system helps the clean model on noisy data, however is inferior to noise-trained model.

8 Conclusions

We estimated natural error probabilities statistically from real-world grammatical-error-correction corpora in order to model and generate noisy inputs for machine learning tasks. We extensively evaluated several state-of-the-art NLP downstream systems with respect to their robustness to input noise, both in increasing level of text noisiness and in variations of error types. We confirmed that the noise hurts the model performance substantially and we compared two coping strategies: training with noise and preprocessing with GEC, concluding that each strategy is beneficial in different scenarios. Finally, we also presented authentic noisy data evaluation using a newly assembled dataset for machine translation with authentic Czech noisy sentences translated to English. We release both the new framework (under MPL 2.0) at https://github.com/ufal/kazitext and the newly assembled dataset (under CC BY-NC-SA license) at http://hdl.handle.net/11234/1-3775.

Acknowledgements

This work has been supported by the Grant Agency of the Czech Republic, project EXPRO LUSyD (GX20-16819X). This research was also partially supported by SVV project number 260 575 and GAUK 578218 of the Charles University. It has has also been using data provided by the LINDAT/CLARIAH-CZ Research Infrastructure (https://lindat.cz), supported by the Ministry of Education, Youth and Sports of the Czech Republic (Project No. LM2018101).

[7] https://ufal.mff.cuni.cz/grants/faust

References

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

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.

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. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. 2013. Building a large annotated corpus of learner english: The nus corpus of learner english. In *Proceedings of the eighth workshop on innovative use of NLP for building educational applications*, pages 22–31.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186.

Tao Ge, Furu Wei, and Ming Zhou. 2018. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655.

Sylviane Granger. 1998. The computer learner corpus: A versatile new source of data for SLA research. In Sylviane Granger, editor, *Learner English on Computer*, pages 3–18. Addison Wesley Longman, London and New York.

Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef van Genabith. 2018. How robust are character-based word embeddings in tagging and mt against wrod scramlbing or randdm nouse? In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 68–80.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a right: Generating better errors to improve grammatical error detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4977–4983.

Tom Kocmi, Martin Popel, and Ondrej Bojar. 2020. Announcing CzEng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan Pino, and Hassan Sajjad. 2019. Findings of the first shared task on machine translation robustness. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 91–102.

Kateřina Macková and Milan Straka. 2020. Reading comprehension in czech via machine translation and cross-lingual transfer. In *Text, Speech, and Dialogue*, pages 171–179, Cham. Springer International Publishing.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 543–553.

Jakub Náplava. 2017. Natural language correction. *Diploma Thesis*.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356.

Joakim Nivre et al. 2018. Universal dependencies 2.3. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtskỳ. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(1):1–15.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. 2017. Artificial error generation with machine translation and syntactic patterns. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287–292.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865.

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.

Alla Rozovskaya, Dan Roth, and Mark Sammons. 2017. Adapting to learner errors with minimal supervision. *Computational Linguistics*, 43(4):723–760.

Barbara Rychalska, Dominika Basaj, Alicja Gosiewska, and Przemysław Biecek. 2019. Models in the wild: On corruption robustness of neural nlp systems. In *International Conference on Neural Information Processing*, pages 235–247. Springer.

Karel Šebesta, Zuzanna Bedřichová, Kateřina Šormová, Barbora Štindlová, Milan Hrdlička, Tereza Hrdličková, Jiří Hana, Vladimír Petkevič, Tomáš Jelínek, Svatava Škodová, Petr Janeš, Kateřina Lundáková, Hana Skoumalová, Šimon Sládek, Piotr Pierscieniak, Dagmar Toufarová, Milan Straka, Alexandr Rosen, Jakub Náplava, and Marie Poláčková. 2019. AKCES-GEC grammatical error correction dataset for czech. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Magda Ševčíková, Zdeněk Žabokrtský, and Oldřich Krůza. 2007. Named entities in czech: Annotating data and developing ne tagger. In *Text, Speech and Dialogue*, pages 188–195, Berlin, Heidelberg. Springer Berlin Heidelberg.

Milan Straka, Jana Straková, and Jan Hajič. 2019. Evaluating contextualized embeddings on 54 languages in POS tagging, lemmatization and dependency parsing.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested NER through linearization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5326–5331, Florence, Italy. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 180–189. Association for Computational Linguistics.

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.

Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–21, Brussels, Belgium. Association for Computational Linguistics.

Bibliography

A. Alfaifi and Eric Atwell. Arabic learner corpus v1: A new resource for arabic language research. 07 2013. doi: 10.13140/2.1.3599.9688.

Hiroki Asano, Tomoya Mizumoto, and Kentaro Inui. Reference-based metrics can be replaced with reference-less metrics in evaluating grammatical error correction systems. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 343–348, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I17-2058.

Kevin Atkinson. Gnu aspell 0.60. 4, 2006.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

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

Birkbeck spelling error corpus / Roger Mitton. Birkbeck spelling error corpus / roger mitton, 1980. URL http://hdl.handle.net/20.500.12024/0643. Oxford Text Archive.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-2301. URL https://aclanthology.org/W16-2301.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-4717. URL https://aclanthology.org/W17-4717.

Marcel Bollmann. A large-scale comparison of historical text normalization systems. 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)*, pages 3885–3898, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1389. URL https://aclanthology.org/N19-1389.

Adriane Boyd. 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, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6111. URL https://aclanthology.org/W18-6111.

Adriane Boyd, Jirka Hana, Lionel Nicolas, Detmar Meurers, Katrin Wisniewski, Andrea Abel, Karin Schöne, Barbora Štindlová, and Chiara Vettori. The MERLIN corpus: Learner language and the CEFR. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1281–1288, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/606_Paper.pdf.

Eric Brill and Robert C. Moore. An improved error model for noisy channel spelling correction. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 286–293, Hong Kong, October 2000. Association for Computational Linguistics. doi: 10.3115/1075218.1075255. URL https://aclanthology.org/P00-1037.

Chris Brockett, William B Dolan, and Michael Gamon. Correcting esl errors using phrasal smt techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, 2006.

Christopher Bryant and Hwee Tou Ng. 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)*, pages 697–707, Beijing, China, July 2015. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P15-1068.

Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1074. URL https://aclanthology.org/P17-1074.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. The BEA-2019 shared task on grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4406. URL https://aclanthology.org/W19-4406.

Christian Buck, Kenneth Heafield, and Bas van Ooyen. N-gram counts and language models from the Common Crawl. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages

3579–3584, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/1097_Paper.pdf.

Flora Ramírez Bustamante and Fernando Sánchez León. Gramcheck: A grammar and style checker. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 175–181. Association for Computational Linguistics, 1996.

Hsun-wen Chiu, Jian-cheng Wu, and Jason S. Chang. Chinese spelling checker based on statistical machine translation. In *Proceedings of the Seventh SIGHAN Workshop on Chinese Language Processing*, pages 49–53, Nagoya, Japan, October 2013. Asian Federation of Natural Language Processing. URL https://aclanthology.org/W13-4408.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014.

Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. A neural grammatical error correction system built on better pre-training and sequential transfer learning. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, 2019.

Shamil Chollampatt and Hwee Tou Ng. A reassessment of reference-based grammatical error correction metrics. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2730–2741, Santa Fe, New Mexico, USA, August 2018a. Association for Computational Linguistics. URL https://aclanthology.org/C18-1231.

Shamil Chollampatt and Hwee Tou Ng. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018b.

Shamil Chollampatt, Weiqi Wang, and Hwee Tou Ng. Cross-sentence grammatical error correction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 435–445, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1042. URL https://aclanthology.org/P19-1042.

Shamil Chollampatt, Weiqi Wang, and Hwee Tou Ng. Cross-sentence grammatical error correction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 435–445, 2019b.

Leshem Choshen and Omri Abend. Reference-less measure of faithfulness for grammatical error correction. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 124–129, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2020. URL https://aclanthology.org/N18-2020.

Teodor-Mihai Cotet, Stefan Ruseti, and Mihai Dascalu. Neural grammatical error correction for romanian. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 625–631, 2020a.

Teodor-Mihai Cotet, Stefan Ruseti, and Mihai Dascalu. Neural grammatical error correction for romanian. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 625–631. IEEE, 2020b.

Daniel Dahlmeier and Hwee Tou Ng. Better evaluation for grammatical error correction. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 568–572, Montréal, Canada, June 2012a. Association for Computational Linguistics. URL https://aclanthology.org/N12-1067.

Daniel Dahlmeier and Hwee Tou Ng. A beam-search decoder for grammatical error correction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 568–578. Association for Computational Linguistics, 2012b.

Daniel Dahlmeier, Hwee Tou Ng, and Eric Jun Feng Ng. Nus at the hoo 2012 shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 216–224, 2012.

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu. Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W13-1703.

Robert Dale and Adam Kilgarriff. Helping our own: The hoo 2011 pilot shared task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249. Association for Computational Linguistics, 2011.

Robert Dale, Ilya Anisimoff, and George Narroway. Hoo 2012: A report on the preposition and determiner error correction shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 54–62. Association for Computational Linguistics, 2012.

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–62, San Diego, CA, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/W16-0506. URL https://www.aclweb.org/anthology/W16-0506.

Sam Davidson, Aaron Yamada, Paloma Fernandez Mira, Agustina Carando, Claudia H. Sanchez Gutierrez, and Kenji Sagae. Developing NLP tools with a new corpus of learner Spanish. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 7238–7243, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://www.aclweb.org/anthology/2020.lrec-1.894.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Mona Diab, Mahmoud Ghoneim, and Nizar Habash. Arabic diacritization in the context of statistical machine translation. In *Proceedings of MT-Summit*. Citeseer, 2007.

Anne Dirkson, Suzan Verberne, and Wessel Kraaij. Lexical normalization of user-generated medical text. In *Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*, pages 11–20, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-3202. URL https://aclanthology.org/W19-3202.

Michael P Fay and Dean A Follmann. Designing monte carlo implementations of permutation or bootstrap hypothesis tests. *The American Statistician*, 56(1): 63–70, 2002. doi: 10.1198/000313002753631385. URL https://doi.org/10.1198/000313002753631385.

Mariano Felice and Ted Briscoe. Towards a standard evaluation method for grammatical error detection and correction. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 578–587, Denver, Colorado, May–June 2015. Association for Computational Linguistics. doi: 10.3115/v1/N15-1060. URL https://aclanthology.org/N15-1060.

Mariano Felice, Zheng Yuan, Øistein E Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. Grammatical error correction using hybrid systems and type filtering. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 15–24, 2014.

Mariano Felice, Christopher Bryant, and Ted Briscoe. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/C16-1079.

Simon Flachs, Ophélie Lacroix, Helen Yannakoudakis, Marek Rei, and Anders Søgaard. Grammatical error correction in low error density domains: A new benchmark and analyses. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8467–8478, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.680. URL https://www.aclweb.org/anthology/2020.emnlp-main.680.

Simon Flachs, Felix Stahlberg, and Shankar Kumar. Data strategies for low-resource grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 117–122, Online, April 2021a. Association for Computational Linguistics. URL https://aclanthology.org/2021.bea-1.12.

Simon Flachs, Felix Stahlberg, and Shankar Kumar. Data strategies for low-resource grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 117–122, 2021b.

G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.

Axel Gandy. Sequential implementation of monte carlo tests with uniformly bounded resampling risk. *Journal of the American Statistical Association*, 104 (488):1504–1511, 2009. doi: 10.1198/jasa.2009.tm08368. URL https://doi.org/10.1198/jasa.2009.tm08368.

Tao Ge, Furu Wei, and Ming Zhou. Reaching human-level performance in automatic grammatical error correction: An empirical study. *arXiv preprint arXiv:1807.01270*, 2018.

Filip Ginter, Jan Hajič, Juhani Luotolahti, Milan Straka, and Daniel Zeman. CoNLL 2017 shared task - automatically annotated raw texts and word embeddings, 2017. URL http://hdl.handle.net/11234/1-1989. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Max Glockner, Vered Shwartz, and Yoav Goldberg. Breaking nli systems with sentences that require simple lexical inferences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 650–655, 2018.

Takumi Gotou, Ryo Nagata, Masato Mita, and Kazuaki Hanawa. Taking the correction difficulty into account in grammatical error correction evaluation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2085–2095, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.188. URL https://aclanthology.org/2020.coling-main.188.

Sylviane Granger. *The computer learner corpus: a versatile new source of data for SLA research*. Routledge, 2014.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. 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, 2014.

Roman Grundkiewicz and Marcin Junczys-Dowmunt. Near human-level performance in grammatical error correction with hybrid machine translation. In

Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), pages 284–290, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2046. URL https://aclanthology.org/N18-2046.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Edward Gillian. Human evaluation of grammatical error correction systems. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 461–470, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1052. URL https://aclanthology.org/D15-1052.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 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, 2019.

Ivan Habernal, Tomáš Ptáček, and Josef Steinberger. Facebook data for sentiment analysis, 2013. URL http://hdl.handle.net/11858/00-097C-0000-0022-FE82-7. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Masato Hagiwara and Masato Mita. GitHub typo corpus: A large-scale multilingual dataset of misspellings and grammatical errors. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 6761–6768, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.835.

Jan Hajič, Eduard Bejček, Alevtina Bémová, Eva Buráňová, Eva Hajičová, Jiří Havelka, Petr Homola, Jiří Kárník, Václava Kettnerová, Natalia Klyueva, Veronika Kolářová, Lucie Kučová, Markéta Lopatková, Marie Mikulová, Jiří Mírovský, Anna Nedoluzhko, Petr Pajas, Jarmila Panevová, Lucie Poláková, Magdaléna Rysová, Petr Sgall, Johanka Spoustová, Pavel Straňák, Pavlína Synková, Magda Ševčíková, Jan Štěpánek, Zdeňka Urešová, Barbora Vidová Hladká, Daniel Zeman, Šárka Zikánová, and Zdeněk Žabokrtský. Prague dependency treebank 3.5, 2018. URL http://hdl.handle.net/11234/1-2621. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Jan Hajič, Jaroslava Hlaváčová, Marie Mikulová, Milan Straka, and Barbora Štěpánková. Morfflex cz 2.0, 2020. URL http://hdl.handle.net/11234/1-3186. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Bo Han, Paul Cook, and Timothy Baldwin. Lexical normalization for social media text. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(1): 1–27, 2013.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. Detecting errors in english article usage with a maximum entropy classifier trained on a large, diverse corpus. In *LREC*, 2004.

Saša Hasan, Carmen Heger, and Saab Mansour. Spelling correction of user search queries through statistical machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 451–460, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1051. URL https://aclanthology.org/D15-1051.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90. URL https://doi.org/10.1109/CVPR.2016.90.

George E. Heidorn, Karen Jensen, Lance A. Miller, Roy J. Byrd, and Martin S Chodorow. The epistle text-critiquing system. *IBM Systems Journal*, 21(3): 305–326, 1982.

Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef van Genabith. How robust are character-based word embeddings in tagging and mt against wrod scramlbing or randdm nouse? In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 68–80, 2018.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

Tomáš Holan, Vladislav Kubon, and Martin Plátek. A prototype of a grammar checker for czech. In *Fifth Conference on Applied Natural Language Processing*, pages 147–154, 1997.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015. URL http://arxiv.org/abs/1508.01991.

Miloš Jakubíček and Aleš Horák. Punctuation detection with full syntactic parsing. *Natural Language Processing and its Applications*, page 335, 2010.

Maarten Janssen. TEITOK: Text-faithful annotated corpora. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4037–4043, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1637.

Tomáš Jelínek, Barbora Štindlová, Alexandr Rosen, and Jirka Hana. Combining manual and automatic annotation of a learner corpus. In Petr Sojka, Aleš Horák, Ivan Kopeček, and Karel Pala, editors, *Text, Speech and Dialogue*, pages 127–134, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-32790-2.

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. A nested attention neural hybrid model for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 753–762, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1070. URL https://aclanthology.org/P17-1070.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. The amu system in the conll-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 25–33, 2014.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. Phrase-based machine translation is state-of-the-art for automatic grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1546–1556, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1161. URL https://aclanthology.org/D16-1161.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. Approaching neural grammatical error correction as a low-resource machine translation task. 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 595–606, 2018.

Daniel Jurafsky and James H. Martin. *Speech and language processing - an introduction to natural language processing, computational linguistics, and speech recognition.* Prentice Hall series in artificial intelligence. Prentice Hall, 2000. ISBN 978-0-13-095069-7.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.391. URL https://aclanthology.org/2020.acl-main.391.

Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, and Kentaro Inui. An empirical study of incorporating pseudo data into grammatical error correction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1236–1242, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1119. URL https://aclanthology.org/D19-1119.

Kevin Knight and Ishwar Chander. Automated postediting of documents. In *AAAI*, volume 94, pages 779–784, 1994.

Tom Kocmi, Martin Popel, and Ondrej Bojar. Announcing czeng 2.0 parallel corpus with over 2 gigawords. *arXiv preprint arXiv:2007.03006*, 2020.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-3204. URL https://aclanthology.org/W17-3204.

Dan Kondratyuk and Milan Straka. 75 languages, 1 model: Parsing Universal Dependencies universally. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2779–2795, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1279. URL https://aclanthology.org/D19-1279.

Vojtěch Kovář. Partial grammar checking for czech using the set parser. In *International Conference on Text, Speech, and Dialogue*, pages 308–314. Springer, 2014.

Vojtěch Kovář, Jakub Machura, Kristỳna Zemková, and Michal Rott. Evaluation and improvements in punctuation detection for czech. In *International Conference on Text, Speech, and Dialogue*, pages 287–294. Springer, 2016.

Aomi Koyama, Tomoshige Kiyuna, Kenji Kobayashi, Mio Arai, and Mamoru Komachi. Construction of an evaluation corpus for grammatical error correction for learners of Japanese as a second language. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 204–211, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.26.

Michal Křen, Tomáš Bartoň, Václav Cvrček, Milena Hnátková, Tomáš Jelínek, Jan Kocek, Renata Novotná, Vladimír Petkevič, Pavel Procházka, Věra Schmiedtová, et al. Syn2010: žánrově vyvážený korpus psané češtiny. *Ústav Českého národního korpusu FF UK, Praha*, 2010.

Vladislav Kubon and Martin Platek. A grammar based approach to a grammar checking of free word order languages. In *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*, 1994.

Karen Kukich. Techniques for automatically correcting words in text. *Acm Computing Surveys (CSUR)*, 24(4):377–439, 1992.

Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. Adversarial attacks and defences competition. *The NIPS'17 Competition: Building Intelligent Systems*, page 195, 2018.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1030. URL https://aclanthology.org/N16-1030.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. Automated grammatical error detection for language learners. *Synthesis lectures on human language technologies*, 3(1):1–134, 2010.

Patrick Lewis, Barlas Oğuz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. Mlqa: Evaluating cross-lingual extractive question answering. *arXiv preprint arXiv:1910.07475*, 2019.

Hao Li, Yang Wang, Xinyu Liu, Zhichao Sheng, and Si Wei. Spelling error correction using a nested rnn model and pseudo training data. *arXiv preprint arXiv:1811.00238*, 2018.

Jared Lichtarge, Christopher Alberti, Shankar Kumar, Noam Shazeer, and Niki Parmar. Weakly supervised grammatical error correction using iterative decoding. *arXiv preprint arXiv:1811.01710*, 2018.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 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)*, pages 3291–3301, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1333. URL https://aclanthology.org/N19-1333.

Wang Ling, Chris Dyer, Alan W Black, Isabel Trancoso, Ramón Fermandez, Silvio Amir, Luís Marujo, and Tiago Luís. Finding function in form: Compositional character models for open vocabulary word representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1520–1530, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1176. URL https://aclanthology.org/D15-1176.

Nikola Ljubešić, Tomaž Erjavec, and Darja Fišer. Corpus-based diacritic restoration for south slavic languages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3612–3616, 2016.

Nina Macdonald, Lawrence Frase, P Gingrich, and Stacey Keenan. The writer's workbench: Computer aids for text analysis. *IEEE Transactions on Communications*, 30(1):105–110, 1982.

Martin Majliš. W2C – web to corpus – corpora, 2011. URL http://hdl.handle.net/11858/00-097C-0000-0022-6133-9. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

M McIlroy. Development of a spelling list. *IEEE Transactions on Communications*, 30(1):91–99, 1982.

Guido Minnen, Francis Bond, and Ann Copestake. Memory-based learning for article generation. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, pages 43–48. Association for Computational Linguistics, 2000.

Tomoya Mizumoto, Mamoru Komachi, Masaaki Nagata, and Yuji Matsumoto. Mining revision log of language learning sns for automated japanese error correction of second language learners. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 147–155, 2011.

Ryo Nagata, Takahiro Wakana, Fumito Masui, Atsuo Kawai, and Naoki Isu. Detecting article errors based on the mass count distinction. In *International Conference on Natural Language Processing*, pages 815–826. Springer, 2005.

Jakub Náplava. Natural language correction. Diploma thesis, Univerzita Karlova, Matematicko-fyzikální fakulta, 2017.

Jakub Náplava and Milan Straka. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-5545. URL https://aclanthology.org/D19-5545.

Jakub Náplava and Milan Straka. Cuni system for the building educational applications 2019 shared task: Grammatical error correction. *Bronze Sponsors*, page 183, 2019b.

Jakub Náplava, Milan Straka, Pavel Straňák, and Jan Hajic. Diacritics restoration using neural networks. In *Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018)*, 2018.

Jakub Náplava, Martin Popel, Milan Straka, and Jana Straková. Understanding model robustness to user-generated noisy texts. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 340–350, Online, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.wnut-1.38.

Jakub Náplava, Milan Straka, and Jana Straková. Diacritics Restoration using BERT with Analysis on Czech language. *The Prague Bulletin of Mathematical Linguistics*, 116:27–42, April 2021. ISSN 0032-6585. doi: 10.14712/00326585.013. URL https://ufal.mff.cuni.cz/pbml/116/art-naplava-straka-strakova.pdf.

Jakub Náplava, Milan Straka, Jana Straková, and Alexandr Rosen. Czech grammar error correction with a large and diverse corpus. *arXiv preprint arXiv:2201.05590*, 2022.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 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)*, pages 588–593, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-2097. URL https://aclanthology.org/P15-2097.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. There's no comparison: Reference-less evaluation metrics in grammatical error correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2109–2115, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1228. URL https://aclanthology.org/D16-1228.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 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*, pages 229–234, Valencia, Spain, April 2017a. Association for Computational Linguistics. URL https://aclanthology.org/E17-2037.

Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. 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*, pages 229–234, Valencia, Spain, April 2017b. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E17-2037.

Courtney Napoles, Maria Nădejde, and Joel Tetreault. Enabling robust grammatical error correction in new domains: Data sets, metrics, and analyses. *Transactions of the Association for Computational Linguistics*, 7:551–566, March 2019. doi: 10.1162/tacl_a_00282. URL https://aclanthology.org/Q19-1032.

Hwee Tou Ng, Siew Mei Wu, Yuanbin Wu, Christian Hadiwinoto, and Joel Tetreault. The CoNLL-2013 shared task on grammatical error correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–12, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/W13-3601.

Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant. The conll-2014 shared task on grammatical error correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 1–14, 2014.

Kiem-Hieu Nguyen and Cheol-Young Ock. Diacritics restoration in vietnamese: letter based vs. syllable based model. In *Pacific Rim International Conference on Artificial Intelligence*, pages 631–636. Springer, 2010.

Diane Nicholls. The cambridge learner corpus: Error coding and analysis for lexicography and elt. In *Proceedings of the Corpus Linguistics 2003 conference*, volume 16, pages 572–581, 2003.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Riyaz Ahmad Bhat, Eckhard Bick, Cristina Bosco, Gosse Bouma, Sam Bowman, Marie Candito, Gülşen Cebiroğlu Eryiğit, Giuseppe G. A. Celano, Fabricio Chalub, Jinho Choi, Çağrı Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Valeria de Paiva, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Cláudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökırmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Nizar Habash, Jan Hajič, Linh Hà Mỹ, Dag Haug, Barbora Hladká, Petter Hohle, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşıkara, Hiroshi Kanayama, Jenna Kanerva, Natalia Kotsyba, Simon Krek, Veronika Laippala, Phương Lê Hồng, Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Amir More, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Lương Nguyễn Thị, Huyền Nguyễn Thị Minh, Vitaly Nikolaev, Hanna Nurmi, Stina Ojala, Petya Osenova, Lilja Øvrelid, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Guy Perrier, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalniņa, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh, Manuela Sanguinetti, Baiba Saulīte, Sebastian Schuster, Djamé Seddah, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Dmitry Sichinava, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Dima Taji, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze, Jonathan North Washington, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. Universal dependencies 2.0, 2017. URL http://hdl.handle.net/11234/1-1983. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajič, Christopher D. Manning, Sampo Pyysalo, Sebastian Schuster, Francis Tyers, and Daniel Zeman. Universal Dependencies v2: An evergrowing multilingual treebank collection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4034–4043, Marseille, France, May 2020. European Language Resources Association. URL https://www.aclweb.org/anthology/2020.lrec-1.497.

Jekaterina Novikova, Ondřej Dušek, and Verena Rieser. RankME: Reliable human ratings for natural language generation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 72–78, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2012. URL https://aclanthology.org/N18-2012.

Marie Novotná. Automatická detekce chyb v gramatické shodě v češtině. Diploma thesis, Master's thesis, Masaryk University, Faculty of Arts, Brno, https://is.muni ..., 2018.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA → Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.bea-1.16. URL https://aclanthology.org/2020.bea-1.16.

Karel Pala, Pavel Rychlỳ, and Pavel Smrž. Text corpus with errors. In *International Conference on Text, Speech and Dialogue*, pages 90–97. Springer, 2003.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL https://aclanthology.org/N18-1202.

Vladimír Petkevič. Kontrola české gramatiky (český grammar checker), 2014. Univerzita Karlova, Filozofická fakulta.

Luan-Nghia Pham, Viet-Hong Tran, and Vinh-Van Nguyen. Vietnamese text accent restoration with statistical machine translation. In *Proceedings of the 27th Pacific Asia Conference on Language, Information, and Computation (PACLIC 27)*, pages 423–429, Taipei, Taiwan, November 2013. Department of English, National Chengchi University. URL https://aclanthology.org/Y13-1044.

Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1493. URL https://aclanthology.org/P19-1493.

Martin Popel and Ondřej Bojar. Training Tips for the Transformer Model. *The Prague Bulletin of Mathematical Linguistics*, 110:43–70, April 2018. ISSN 0032-6585. doi: 10.2478/pralin-2018-0002. URL https://ufal.mff.cuni.cz/pbml/110/art-popel-bojar.pdf.

Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. Artificial error generation with machine translation and syntactic patterns. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287–292, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-5032. URL https://aclanthology.org/W17-5032.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865, 2018.

Michal Richter, Pavel Straňák, and Alexandr Rosen. Korektor–a system for contextual spell-checking and diacritics completion. In *Proceedings of COLING 2012: Posters*, pages 1019–1028, 2012.

Alexandr Rosen. Building and using corpora of non-native Czech. In Broňa Brejová, editor, *Proceedings of the 16th ITAT: Slovenskočeský NLP workshop (SloNLP 2016)*, volume 1649 of *CEUR Workshop Proceedings*, pages 80–87, Bratislava, Slovakia, 2016. Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, CreateSpace Independent Publishing Platform. ISBN 978-1537016740. URL http://ceur-ws.org/Vol-1649/80.pdf.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. A simple recipe for multilingual grammatical error correction. *CoRR*, abs/2106.03830, 2021. URL https://arxiv.org/abs/2106.03830.

Alla Rozovskaya and Dan Roth. Annotating ESL errors: Challenges and rewards. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 28–36, Los Angeles, California, June 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W10-1004.

Alla Rozovskaya and Dan Roth. Grammar error correction in morphologically rich languages: The case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17, March 2019. doi: 10.1162/tacl_a_00251. URL https://aclanthology.org/Q19-1001.

Barbara Rychalska, Dominika Basaj, Alicja Gosiewska, and Przemysław Biecek. Models in the wild: On corruption robustness of neural nlp systems. In *International Conference on Neural Information Processing*, pages 235–247. Springer, 2019.

Pavel Rychlý. Czaccent–simple tool for restoring accents in czech texts. *RASLAN 2012 Recent Advances in Slavonic Natural Language Processing*, page 85, 2012.

Keisuke Sakaguchi and Benjamin Van Durme. Efficient online scalar annotation with bounded support. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 208–218, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1020. URL https://aclanthology.org/P18-1020.

Keisuke Sakaguchi, Courtney Napoles, Matt Post, and Joel Tetreault. Reassessing the goals of grammatical error correction: Fluency instead of grammaticality. *Transactions of the Association for Computational Linguistics*, 4:169–182, 2016.

David Samuel and Milan Straka. ÚFAL at MultiLexNorm 2021: Improving multilingual lexical normalization by fine-tuning ByT5. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 483–492, Online, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.wnut-1.54. URL https://aclanthology.org/2021.wnut-1.54.

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

Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

Karel Šebesta, Zuzanna Bedřichová, Kateřina Šormová, Barbora Štindlová, Milan Hrdlička, Tereza Hrdličková, Jiří Hana, Vladimír Petkevič, Tomáš Jelínek, Svatava Škodová, Petr Janeš, Kateřina Lundáková, Hana Skoumalová, Šimon Sládek, Piotr Pierscieniak, Dagmar Toufarová, Milan Straka, Alexandr Rosen, Jakub Náplava, and Marie Poláčková. CzeSL grammatical error correction dataset (CzeSL-GEC), 2017. URL http://hdl.handle.net/11234/1-2143. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, 2016.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Milan Straka, Jan Hajič, and Jana Straková. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4290–4297, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1680.

Milan Straka, Jakub Náplava, and Jana Straková. Character transformations for non-autoregressive GEC tagging. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 417–422, Online, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.wnut-1.46.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Oleksiy Syvokon and Olena Nahorna. UA-GEC: grammatical error correction and fluency corpus for the ukrainian language. *CoRR*, abs/2103.16997, 2021. URL https://arxiv.org/abs/2103.16997.

Toshikazu Tajiri, Mamoru Komachi, and Yuji Matsumoto. 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)*, pages 198–202, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P12-2039.

Yujin Takahashi, Satoru Katsumata, and Mamoru Komachi. Grammatical error correction using pseudo learner corpus considering learner's error tendency. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 27–32, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-srw.5. URL https://aclanthology.org/2020.acl-srw.5.

Gongbo Tang, Fabienne Cap, Eva Pettersson, and Joakim Nivre. An evaluation of neural machine translation models on historical spelling normalization. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1320–1331, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1112.

WD Taylor. Grope—a spelling error correction tool. *AT& T Bell Labs Tech. Mem*, 1981.

The Unicode Consortium. *The Unicode Standard, Version 10.0.0*. The Unicode Consortium, Mountain View, CA, 2017. ISBN 978-1-936213-16-0. URL http://www.unicode.org/versions/Unicode10.0.0/.

Rob van der Goot, Alan Ramponi, Arkaitz Zubiaga, Barbara Plank, Benjamin Muller, Iñaki San Vicente Roncal, Nikola Ljubešic, Özlem Çetinoglu, Rahmad Mahendra, Talha Çolakoglu, et al. Multilexnorm: A shared task on multilingual lexical normalization. In *Proceedings of the 7th Workshop on Noisy User-generated Text (W-NUT 2021), Punta Cana, Dominican Republic. Association for Computational Linguistics*, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Jonáš Vidra, Zdeněk Žabokrtský, Lukáš Kyjánek, Magda Ševčíková, Šárka Dohnalová, Emil Svoboda, and Jan Bodnár. DeriNet 2.1, 2021. URL http://hdl.handle.net/11234/1-3765. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Yuxuan Wang, Wanxiang Che, Jiang Guo, Yijia Liu, and Ting Liu. Cross-lingual BERT transformation for zero-shot dependency parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5721–5727, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1575. URL https://aclanthology.org/D19-1575.

Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1077. URL https://aclanthology.org/D19-1077.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. Neural language correction with character-based attention. *CoRR*, abs/1603.09727, 2016. URL http://arxiv.org/abs/1603.09727.

Shuyao Xu, Jiehao Zhang, Jin Chen, and Long Qin. Erroneous data generation for grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4415. URL https://aclanthology.org/W19-4415.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.41. URL https://aclanthology.org/2021.naacl-main.41.

Aaron Yamada, Sam Davidson, Paloma Fernández-Mira, Agustina Carando, Kenji Sagae, and Claudia Sánchez-Gutiérrez. Cows-l2h: A corpus of spanish learner writing. *Research in Corpus Linguistics*, 8(1):17–32, Mar. 2020. doi: 10.32714/ricl.08.01.02. URL https://ricl.aelinco.es/index.php/ricl/article/view/109.

Ikumi Yamashita, Satoru Katsumata, Masahiro Kaneko, Aizhan Imankulova, and Mamoru Komachi. Cross-lingual transfer learning for grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4704–4715, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.415. URL https://aclanthology.org/2020.coling-main.415.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 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*, pages 180–189, 2011.

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

Ryoma Yoshimura, Masahiro Kaneko, Tomoyuki Kajiwara, and Mamoru Komachi. SOME: Reference-less sub-metrics optimized for manual evaluations of grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6516–6522, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.573. URL https://aclanthology.org/2020.coling-main.573.

Zheng Yuan. Grammatical error correction in non-native english. Technical report, University of Cambridge, Computer Laboratory, 2017.

Zheng Yuan and Ted Briscoe. Grammatical error correction using neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, 2016.

Zheng Yuan and Christopher Bryant. Document-level grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 75–84, 2021.

Zheng Yuan and Mariano Felice. Constrained grammatical error correction using statistical machine translation. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 52–61, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/W13-3607.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. 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)*, pages 156–165, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1014. URL https://aclanthology.org/N19-1014.

Yuanyuan Zhao, Nan Jiang, Weiwei Sun, and Xiaojun Wan. Overview of the nlpcc 2018 shared task: Grammatical error correction. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 439–445. Springer, 2018.

Karel Šebesta. Korpusy češtiny a osvojování jazyka [Corpora of Czech and language acquistion]. *Studie z aplikované lingvistiky/Studies in Applied Linguistics*, 1:11–34, 2010.

List of Figures

List of Tables

List of Publications

Jakub Náplava, Milan Straka, Pavel Straňák, Jan Hajič: Diacritics Restoration Using Neural Networks. *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC 2018)*, pp. 1-10, European Language Resources Association, Paris, France, ISBN 979-10-95546-00-9
Cited by (excluding self-citations, according to Google Scholar): 25

Jakub Náplava, Milan Straka: CUNI System for the Building Educational Applications 2019 Shared Task: Grammatical Error Correction. *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 183-190, Association for Computational Linguistics, Stroudsburg, PA, USA, ISBN 978-1-950737-34-5
Cited by (excluding self-citations, according to Google Scholar): 4

Jakub Náplava, Milan Straka: Grammatical Error Correction in Low-Resource Scenarios. *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pp. 346-356, Association for Computational Linguistics, Stroudsburg, PA, USA, ISBN 978-1-950737-84-0
Cited by (excluding self-citations, according to Google Scholar): 25

Jakub Náplava, Milan Straka, Jana Straková: Diacritics Restoration using BERT with Analysis on Czech language. *The Prague Bulletin of Mathematical Linguistics*, ISSN 0032-6585, 116, pp. 27-42
Cited by (excluding self-citations, according to Google Scholar): 3

Milan Straka, Jakub Náplava, Jana Straková, David Samuel: RobeCzech: Czech RoBERTa, a Monolingual Contextualized Language Representation Model. *24th International Conference on Text, Speech and Dialogue*, pp. 197-209, Springer, Cham, Switzerland, ISBN 978-3-030-83526-2
Cited by (excluding self-citations, according to Google Scholar): 8

Milan Straka, Jakub Náplava, Jana Straková: Character Transformations for Non-Autoregressive GEC Tagging. *Proceedings of the 7th Workshop on Noisy User-generated Text (W-NUT 2021)*, pp. 417-422, Association for Computational Linguistics, Stroudsburg, PA, USA, ISBN 978-1-954085-90-9

Jakub Náplava, Martin Popel, Milan Straka, Jana Straková: Understanding Model Robustness to User-generated Noisy Texts. *Proceedings of the 7th Workshop on Noisy User-generated Text (W-NUT 2021)*, pp. 340-350, Association for Computational Linguistics, Stroudsburg, PA, USA, ISBN 978-1-954085-90-9

Matěj Kocián, Jakub Náplava, Daniel Štancl, Vladimír Kadlec: Siamese BERT-based Model for Web Search Relevance Ranking Evaluated on a New Czech Dataset. *Proceedings of the AAAI Conference on Artificial Intelligence (IAAI-22).* In print. arXiv:2112.01810

Jakub Náplava, Milan Straka, Jana Straková, Alexandr Rosen: Czech Grammar Error Correction with a Large and Diverse Corpus. *Transactions of the Association for Computational Linguistics (TACL 2022).* In print. arXiv:2201.05590