# Natural Language Correction - Thesis Proposal

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

## Abstract

Tools that correct errors in human written texts are an important part of many existing systems used by countless people. Although the rise of deep neural networks in recent years made it possible to handle more complex errors including even grammatical and fluency ones, current models still suffer from several problems. In this thesis proposal, we describe issues we find are the most crucial in the area of the natural language correction and discuss what has been done so far to solve them. We further describe the work we performed until now, and finally we provide an overview of our future research plans.

## 1 Introduction

Natural Language Correction (commonly referred to as Grammatical Error Correction or simply GEC) is the task of building systems for correcting all errors appearing in human written texts. Given a noisy input text, the corrected text should preserve the semantic meaning, while being well-formed without any errors. For texts that do not contain any errors, no changes shall be done.

The variety of error types ranges from simple local errors such as bad word spelling or missing punctuation up to complex grammar errors such as subject-verb agreement or even fluency errors, that require a whole segment of text to be rewritten. Two examples taken from an English GEC dataset are presented below.

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 our ancestors if ~~old people could not find~~ did not develop sciences and technologies ~~sciences and tecnologies and they did not developped~~ .

Figure 1: Two examples from an English GEC dataset.

One of the reasons why natural language correction is difficult is that broad context is often needed to determine correct replacement – for certain error types, such as wrong article or tense errors, not even whole paragraph may be enough. Furthermore, as there are often multiple errors appearing in text and the variety of error types is large, it is nearly impossible to manually write down correcting rules with enough coverage.

The natural language correction systems gradually evolved from simple rule-based systems, through combination of single error classifiers, to recent statistical and neural machine translation approaches. These handle natural language correction task as a translation problem from a noisy input text into a well-formed text. As the recent Building Educational Applications Shared Task on Grammatical Error Correction (Bryant et al., 2019) showed, slightly modified Transformer architecture is the current state-of-the-art architecture for English.

To evaluate system performance, multiple metrics were proposed. Probably the most common ones are MaxMatch M2 scorer (Dahlmeier and Ng, 2012b) and ERRANT scorer (Bryant et al., 2017). They evaluate system performance by means of correctly performed *edits*, where an edit consists of a part of the noisy text and its correction. These edits are extracted automatically. Because omitting a correction is usually not as bad as proposing a bad one, $F_{0.5}$ score emphasizing precision over recall is used. There are also other metrics such as GLEU (Napoles et al., 2015) or I-measure (Felice and Briscoe, 2015), but they are used in rather specific scenarios.

In certain domains and languages, human-level performance was already matched. Specifically, Ge et al. (2018) claim to reach human-level performance on correcting essays of East-Asian second learners of English. Although the results are

great, they were not repeated on any other domain or language, as the second-learner English is the largest dataset by a wide margin, with majority of conducted research.

While there are numerous datasets for training and evaluating correction systems on English, there is only a limited number of datasets for other languages. Similarly, the reported performance of these systems is currently worse than performance of the current state-of-the art systems on English. Specifically, we are aware of only several attempts to perform GEC in Czech, none of them having comparable performance to systems on English.

Except for its most straightforward usage in all kinds of editing tools such as Microsoft Word or LibreOffice Writer, language correcting systems may also be used in a pre-processing pipeline to other systems that work with user-generated data. These are nowadays frequently trained solely on clean data and their performance may deteriorate substantially when processing noisy data.

This proposal is structured as follows. In Section 2, we present a short history of language correction systems. In Section 3, we describe the work we have done so far. Section 4 describes our future research plans and finally Section 5 summarizes this proposal.

## 2 History of Natural Language Correction

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 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 rule 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.

In 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; 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, multiple of them must be combined to allow more realistic usage. Dahlmeier et al. (2012) used a pipeline system comprising of several sequential steps. Dahlmeier and Ng (2012a) 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 classifier for certain more complex error types might be complicated.

The Czech system for context sensitive spelling correction of Richter et al. (2012) does not use any specific classifiers, but its approach can be seen as a generalisation of the approach of Dahlmeier and Ng (2012a). It uses 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

utilize linguistic features. To find an optimal correction, Viterbi algorithm (Forney, 1973) is used.

Brockett et al. (2006) propose to consider natural language processing to be a machine translation problem of translating grammatically incorrect sentences into correct ones. Initially, statistical machine translation systems were employed, and two out three top performing systems of CoNLL 2014 Shared Task on Grammatical Error Correction (Ng et al., 2014) were using statistical machine translation approaches.

From 2018, top performing systems in natural language correction employ neural machine translation approaches (Ge et al., 2018; Chollampatt et al., 2019; Grundkiewicz et al., 2019). These will be described in later sections.

## 3 Work done so far

Our work in the area of natural language correction started with development of systems for automatic diacritics generation. While it is very common that people in certain languages (e.g. Czech or Vietnamese) and on certain devices write without diacritics, we saw that existing systems based on traditional statistical approaches do not sometimes perform satisfactory – especially for words that need large context to determine which of its variants with diacritics is correct, and also for out-of-vocabulary words. As recurrent neural networks started to show amazing results on many other natural language processing tasks, we decided to investigate how would they perform on this particular task. Given the relative simplicity of this task, it was also a reasonable first step to evaluate what are neural networks capable of.

After achieving state-of-the-art results on the diacritics generation task, we moved our focus to developing a system that could correct more errors. A natural choice was extending our system with a spelling correction, as it is a common part of many current tools. After spending non-trivial amount of time on developing the most appropriate neural model, we started asking ourselves what spelling correction actually is. Many papers define spelling error correction as locating misspelled words that are not in the vocabulary and replacing them with correct ones. Other papers extend this definition by also correcting words that, despite being in vocabulary, are in fact spelling errors. We found the latter definition to be more accurate as accidentally misclicking computer key may result in a valid word, which is not covered by the former definition. However, many grammatical errors, such as subject verb-agreement (*he appear*) or certain cases of verb-tense error (*he disliked* vs *he dislikes*), would fit into this category as well.

By the time we were at a loss about how to acquire real-world dataset consisting of spelling errors only, Building Educational Applications Workshop announced a new shared task on grammatical error correction on English. It was a great incentive to change our focus and, instead of spending time on an intermediate task of vaguely defined spelling error correction, start developing general system capable of correcting any grammatical errors. The shared task came with 3 tracks: *restricted*, where only provided data could be used for training; *unrestricted*, where any data could be used to train a system; and *low-resource* track, where no annotated data could be used for training. We participated in all three of them.

One of the shared task outcomes was that utilizing synthetic data for training helps a great deal even in English, in which numerous annotated datasets already exist. Inspired by this, we asked ourselves, how much would a synthetic noise help in case of low-resource languages, i.e. languages that have only thousands of annotated sentences. When we automatically generated artificial synthetic data and combined them with authentic data, we reached new state-of-the-art results on all 3 tested low-resource languages: Czech, German and Russian.

### 3.1 Diacritics Generation

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. č with acute or caron) by the underlying basic character without diacritics. Practically speaking, they write in ASCII. Diacritics Generation (also known as Automatic Accent Insertion, Diacritics Restoration or even Diacritization) is a subtask of general grammatical error correction which aims to correct all missing diacritics in the text.

One of the first papers to describe systems for automatic diacritics generation is a seminal work by Yarowsky (1999), who compares several statistical algorithms for restoration of diacritics in French and Spanish. They used word based approach and their final system combines decision

lists with morphological and collocational information. The word level approaches were further explored in Crandall (2005); Šantić et al. (2009); Richter et al. (2012). The latter one builds a Hidden Markov Model with vertices being word diacritics variants, the transition costs are composed of morphological lemma feature, morphological tag feature and word forms feature. The emission probabilities are based on an error model and Viterbi algorithm is used to find an optimal diacritization. There are also approaches that operate on character level (Mihalcea, 2002; Mihalcea and Nastase, 2002; Scannell, 2011) and claim that they work better on low resource languages. With the recent advent of neural networks, Belinkov and Glass (2015) employed recurrent neural networks and reached new state-of-the-art results on Arabic. Note that although the bidirectional LSTM (Hochreiter and Schmidhuber, 1997) backbone is the same as in our work, they do not utilize residual connections to allow building deeper models and also do not incorporate language model in decoding.

Majority of papers we have found evaluated their models on a restricted number of languages (usually one to three). Therefore, we started our work by analyzing the set of languages for which the task of diacritics generation may be relevant (difficult). For each language contained within UD 2.0 (Nivre et al., 2017), we measured the ratio of words with diacritics. As high occurrence of words with diacritics does not naturally imply that generating diacritics is an ambiguous task for the language, we also evaluated word error rate of a simple dictionary baseline: according to a large raw text corpora we constructed a dictionary of the most frequent variant with diacritics for a given word without diacritics, and used the dictionary to perform the diacritics restoration. The results of this experiment are presented in Table 1. For simplicity, only top 12 languages sorted w.r.t. to the dictionary baseline are presented.

| 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% |

Table 1: Analysis of percentage of words with diacritics and the word error rate of a dictionary baseline. Measured on UD 2.0 data and CoNLL 17 UD shared task raw data for dictionary. Only words containing at least one alphabetical character are considered. This table displays top 15 languages with worst results on dictionary baseline.

Table 1 shows that for nine languages, the word error rate is larger than 2%, and the rest still have word error rate still above 1%. We concluded that even with a very large dictionary, the diacritics restoration is a challenging task for these languages. Because there was no consistent approach to obtaining datasets for diacritics restoration and also there were no datasets covering majority of languages from Table 1, we proposed a new pipeline utilizing both clean data from Wikipedia and also not that clean data from general web (utilizing CommonCrawl corpus). The generated dataset together with scripts that were run to generate it were made freely available (Náplava et al., 2018).

Further, we implemented a novel model for diacritics restoration. It consists of a character level recurrent neural network, which for each input character outputs its correct variant with diacritics. We combined the network with an external word level language model, which is incorporated during beam search decoding. As you can see in Figure 2, the recurrent neural network is bidirectional, thus each character has information both from its preceding and following context. For simplicity, the visualisation does not show that there are actually multiple stacked RNNs with residual connections.

We evaluated the proposed model on two existing datasets and on our new dataset. The first existing dataset consisted of Croatian, Serbian and Slovenian data and we reduce the error of the previous state-of-the-art system by more than 30% on Wikipedia part of the dataset and by more than 20% on the Twitter part of the dataset. The second dataset is for diacritics generation in Czech and we reduce the error of the best previous state-of-

| 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 2: Results obtained on new multilingual dataset for diacritics generation. 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 error reduction of our model combined with language model compared to the *corpus* method.

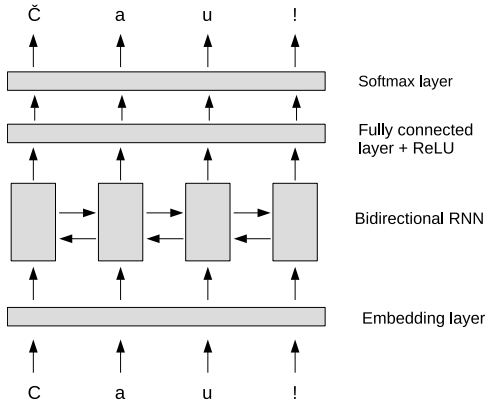


Figure 2: Visualisation of the recurrent neural model we used for diacritics generation.

the-art results of Korektor by more than 60%. On the new dataset, we compare the performance our model with dictionary baseline as described earlier and so called *corpus* method, which extends dictionary baseline via log-linear model with context probability. The results, together with dataset statistics are for 12 languages presented in Table 2.

To conclude, we developed a new method for diacritics generation and shown that it outperforms existing methods on two existing datasets. Moreover, we proposed a new pipeline for obtaining consistent diacritics restoration datasets and made them freely available. The full paper presentend on LREC 2018 conference is available at https://www.aclweb.org/anthology/L18-1247.pdf, the code of our model is available at https://github.com/arahusky/diacritics_restoration and the published dataset can be found at http://hdl.handle.net/11234/1-2607 (Náplava et al., 2018).

### 3.2 GEC

Grammatical error correction (GEC) is the Holy Grail of natural language correction. As it requires systems to correct all types of errors, it has been a goal impossible to achieve for a long time. Because the majority of research has been conducted on English and we are not aware of any breakthrough paper on any other language, all papers described in this Section are trained and tested for English.

Brockett et al. (2006) was the first system to employ statistical machine translation (SMT) to GEC. Although they used it only for correcting 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 than, 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., 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.

In the following years, the prevailing number of papers utilized machine translation approach. 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., 2014) 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 area of GEC.

Yuan and Briscoe (2016) proposed a first GEC system based on neural machine translation. Its backbone was classical encoder-decoder word-level 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 model comprising of a character level pyramidal encoder and a character decoder with an attention mechanism. Although the use of characters as basic units eliminated 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 (2018). 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.

Junczys-Dowmunt et al. (2018) used Transformer architecture with some tweaks adapted from low resource machine translation: 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. To overcome the issue with the lack of annotated data, Lichtarge et al. (2018) trained the Transformer model on a large corpus of Wikipedia edits and, because edits concerning single article may be split into multiple revisions, they used their model incrementally. In other words, the model could repeatedly translate its current output as long as the translation is more probable than 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. This is achieved by dynamically extending the training corpus with so called "fluency sentence pairs". These are corrections of a partially trained model for which it holds that the correction has higher fluency score than the original input. To compute the fluency score, authors use an external language model.

Although this is only a short excerpt of research in GEC, we hope that it covers the most influential papers and ideas before the start of another shared task on grammatical error correction in 2019, which we participated in.

In 2019, Building Educational Applications Workshop announced another shared task on grammatical error correction. With the hope of replacing the most commonly used CoNLL-2014 test set, which has a large bias to Asian second learners of English and is also quite small, it came with a new dataset. This dataset is much more diverse, contains also a subset of native speakers and has annotated English language levels. The shared task came with 3 tracks that aimed to test systems under different data restrictions. The Restricted track precisely defined what annotated training

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

data could be used. In the Unrestricted Track, participants could use any data to train the systems, and finally in the Low Resource track, no annotated data could be used. Errant $F_{0.5}$ scorer was used as the official metric mainly due to its abilities to report statistics on individual error types.

We started our work in GEC by implementing the model proposed by Junczys-Dowmunt et al. (2018). Specifically, we extended standard Transformer model by source word dropout 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. We used this model in all three tracks of the shared task. Moreover, we implemented iterative decoding as proposed by Lichtarge et al. (2018) and use the trained system incrementally as long as the cost of the correction is less then the cost of the identity translation times a learned constant. We used the model log-likelihood as the cost function. Finally, to reduce variance during training and hopefully also achieve better results, we used checkpoint averaging as described by Popel and Bojar (2018).

In the Restricted track, we used all allowed data for training the model. Because majority of data came from the noisiest Lang-8 corpus, we oversampled other datasets to make the difference smaller. We further experimented with the value of source and target word dropout, weight for non-identity words in modified objective, constant used in iterative decoding and the differences between lighter Transformer BASE and heavier Transformer BIG architecture. The chosen hyperparameters improved the performance of baseline Transformer model by almost 12 points in Errant $F_{0.5}$ score.

Because no annotated data were allowed in the Low Resource track, we decided to incorporate Wikipedia revisions. We followed an approach of Lichtarge et al. (2018) and downloaded Wikipedia XML revision dumps, extracted individual pages, removed all non-text elements, downsampled the snapshots and with a small probability injected spelling noise. With the same low probability, a random text substring (up to 8 characters) was replaced with a marker, which should force the model to learn infilling. Finally, the texts from two consecutive snapshots were aligned and sequences between matching segments were extracted to form a training pair. Only 4% of identical samples was preserved. This resulted in over 190M segment pairs, which were used to train the system.

Finally in the Unrestricted track, we used the best system from the Low Resource track and finetuned it on the oversampled 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. The major outcome for our future research was to learn to generate and use synthetic (artificial) data, which boosted the performance of top performing models a lot. Combining multiple models into an ensemble was the other key factor that distinguished top performing systems from others. The presented system description paper can be found on https://www.aclweb.org/anthology/W19-4419.pdf.

### 3.3 Low-Resource GEC

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. Namely, Boyd (2018) created a dataset and presented a GEC system for German, Rozovskaya and Roth (2019) for Russian, Náplava (2017)[1] 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., 2014). The reason why the majority of research has been conducted on English is the availability of data. While we are aware of at least 6 datasets for GEC

[1]This author's diploma thesis.

| System | P | R | $F_{0.5}$ |
|---|---|---|---|
| Rozovskaya and Roth (2019) | 38.0 | 7.5 | 21.0 |
| Our work – pretrain | 47.76 | 26.08 | 40.96 |
| Our work – finetuned | 63.26 | 27.50 | 50.20 |

Table 4: Comparison of our systems on Russian GEC test set (RULEC-GEC).

| System | P | R | $F_{0.5}$ |
|---|---|---|---|
| Boyd (2018) | 51.99 | 29.73 | 45.22 |
| Our work – pretrain | 67.45 | 26.35 | 51.41 |
| Our work – finetuned | 78.21 | 59.94 | 73.71 |

Table 5: Comparison of our systems on German GEC test set (Falko-Merlin Test Set).

in English with millions of sentences altogether, we know of only a single dataset in the rest of the languages, each having at most tens of thousands of sentence pairs. This is naturally an issue as the current machine translation approaches require large corpora to train properly.

The issue with the lack of the annotated data might be mitigated by utilizing additional artificially generated synthetic data. In recent years, several approaches to create them have been proposed. The first of them, the so called back-translation model, consists of training another machine translation model in the opposite direction, i.e. to learn to translate from correct into incorrect sentences (Náplava, 2017; Rei et al., 2017; Kasewa et al., 2018). While this approach might generate high quality synthetic data, it once again requires large volumes of training data, which is in the case of low resource languages intractable.

The second approach is to extract large volume of parallel data from Wikipedia revisions and pre-train the GEC system on them (Lichtarge et al., 2018). The problem with this approach is that even Wikipedia might not be big enough for certain languages.

The third approach is to create synthetic data 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. 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. This simple approach works surprisingly well on on English as Grundkiewicz et al. (2019) was one of two winning teams of BEA 2019 Shared Task on GEC.

The most prominent issue with the unsupervised approach of Grundkiewicz et al. (2019) might be the language specific spell-checker. However, because the latest ASpell[2] has dictionaries for 61 languages, we do not consider that an issue for now and decided to use this approach in our work. Specifically, given a clean sentence, we first sample 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 dictionary after the current word. To make the system more robust to spelling errors, the same operations are also used on individual characters. Because the model after training often failed to correct errors in casing and diacritics, we extended the word level operations by changing word casing and the character level operations by changing character diacritics.

We employed the described pipeline to generate large synthetic data for English, Russian and German from clean WMT News Crawl monolingual training data (Ondřej et al., 2017). We used these data to pre-train our modified Transformer model as described in Section 3.2 (for each language one model). Although these models never saw any authentic data, they were already better than previous state-of-the-art systems on German (Boyd, 2018) and Russian (Rozovskaya and Roth, 2019). We further fine-tuned the models with a mixture of authentic and synthetic data, which increased the performance even further.

The final model on English was slightly worse than the current state-of-the-art system (*69.47* vs *69.00*). However, given that Grundkiewicz et al. (2019) use an ensemble of multiple models, which are known to boost system performance considerably, we hypothesise, that our results are at least

[2] http://aspell.net/

on par with theirs. As can be seen in Table 4 and Table 5, the main outcome of our work is that our model works well in low resource scenarios, outperforming previous Russian and German state-of-the-art system by a large margin. Note that we report both results of our pre-trained model only (*Our work – pretrain*) and the final fine-tuned system (*Our work – finetuned*).

The Czech GEC dataset from Náplava (2017) does not allow systems to be evaluated using standard Errant $F_{0.5}$ scorer or M2 scorer, because it is published only as aligned sentences. We therefore decided to improve it and created a new dataset AKCES-GEC with separated edits together with their type annotations in M2 format (Dahlmeier and Ng, 2012b). To create the dataset, we used CzeSL-man corpus (Rosen and Matliare, 2016) consisting of manually annotated transcripts of essays of nonnative 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 edits were created according to the manual alignments. The newly generated AKCES-GEC dataset consists of an explicit train/development/test split, with each set divided into foreigner and Romani students; for development 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 6.

With the new Czech AKCES-GEC dataset, we could reproduce our experiments from other languages on it. Specifically, we generated a synthetic corpus on which we pre-trained our system. We then fine-tuned the system on a mixture of synthetic and authentic data from the corpus. The results of our system compared to the previous state-of-the-art of Richter et al. (2012) are presented in Table 7.

To summarize our work in GEC with low amount of annotated authentic data, we first showed that when utilizing synthetic corpus, we can reach surprisingly good results. Although we did not explicitly mentioned it before, we also trained systems using authentic data only and the results were disappointing. We also showed that fine-tuning the model on a mixture of authentic and synthetic data boosts the performance even more. Moreover, we created a new dataset for GEC in Czech. We published the dataset at http://hdl.handle.net/11234/1-3057. The paper presented on WNUT 2019 can be found at https://www.aclweb.org/anthology/D19-55.pdf#page=366 and the code for synthesizing the data and training the models is available at https://github.com/ufal/low-resource-gec-wnut2019.

## 4 Future Work

### 4.1 GEC in Czech

One of the main focus of our future research is grammatical error correction in Czech. We present main points of our concrete research plans:

- **Multi-domain dataset** A big issue of many machine learning models is their domain specificity. Therefore, new dataset for Czech GEC evaluation must cover broad domain area.
- **Metrics** Although Errant $F_{0.5}$ scorer and MaxMatch M2 scorer work well for many English domains, there are certain domains, where other metrics correlate with human scores better.
- **Models** Compare domain specific models to models that work well across multiple domains. Also, as multiple errors require larger context than a single sentence, analyze the paragraph or even document level models.
- **Deployability** Despite that large architectures with millions of parameters work well, it would be great to find out the smallest possible architecture that works well enough.

We have already assembled a team of annotators who are about to annotate texts from 5 different sources: essays of Czech pupils, essays of Czech pupils with Romi background, essays of foreigners learning Czech as a second language, user comments from several Czech Facebook pages and comments from discussion forum of Czech online news Novinky.cz. In comparison to our former dataset AKCES-GEC, this dataset already contains texts from Czech natives with both formal and informal language. Due to a limited budget, only the testing and development sets will be annotated.

| | | 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 6: Statistics of the AKCES-GEC dataset – number of documents, sentences, words and error rates.

| System | P | R | $F_{0.5}$ |
|---|---|---|---|
| Richter et al. (2012) | 68.72 | 36.75 | 58.54 |
| Our work – pretrain | 80.32 | 39.55 | 66.59 |
| Our work – finetuned | 83.75 | 68.48 | 80.17 |

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

Once the dataset is annotated, we plan to train multiple models and use them to correct noisy input sentences from the new dataset. Similarly to Napoles et al. (2019), we also want the annotators to assign quality of corrections and use these annotations together with a set of metrics to either select the single best metric or train a new metric. Because there may be too large differences between certain domains (e.g. Czech pupils with Romi background tend to create texts with poor quality that need large fluency edits), it is possible that different metrics will be chosen for different domains.

The sentence level GEC models usually work quite well on English. However, some errors can only be corrected reliably using cross-sentence context. An example of such an error in English are articles or verb tense errors and in Czech errors in subject-verb agreement. The document level machine translation already seems to work decently well when trained with large batches as described in Junczys-Dowmunt (2019). There has also been some research conducted in GEC. Specifically, Chollampatt et al. (2019) extend convolutional translation model of Gehring et al. (2017) with an auxiliary encoder that encodes previous sentences and incorporate the encoding in the decoder via attention. They claim to reach better results with this cross sentence model than with a model operating on single sentences.

It will be an interesting research to evaluate the performance of a single model working well across all 5 domains compared to models that are trained only on a single domain. Because some domains are closer to each other (e.g. comments from Czech Facebook and comments from Czech online news), it is also possible that model trained on a subset of all domains may show superior results.

Finally, it would be great to have a working GEC system deployed so that everyone could test and use it. Therefore, plan to distill small enough model that works well and deploy it to LINDAT.

### 4.2 ML-Noise

Nowadays, many models used in natural language processing are trained and tested on clean data. However, they are often used on noisy user generated content. We thus plan to investigate, how much does natural noise in data hurt model's performance and also how to make systems robust again such noise.

The fact that data with natural noise deteriorate performance of current systems is by no means a novel idea. Considering recent work on noisy data, Belinkov and Bisk (2017) found that natural noise such as misspellings and typos cause significant drops in BLEU scores of current state-of-the-art character-level machine translation models. They explored multiple strategies to increase the model's robustness and found out that when the model is trained on a mixture of original and noisy input, it can learn to address certain amount of errors.

Adversarial attacks to deep classification models were explored in Liang et al. (2017). They identified text items important for the classification and perturbed them to generate adversarial inputs that fool the classifier. Although their approach was able to successively find small perturbations that make the classifier perform wrong predictions, it is not clear to what extent would a human make such errors. The adversarial black box generation was further explored in Gao et al. (2018), who also tried to minimize edit distance when generating the adversarial examples.

Rychalska et al. (2019) implemented a framework capable of introducing multiple noise types into text such as removing or swapping articles, rewriting digit numbers into words or introducing errors in spelling. They tested their framework on 4 natural language processing tasks and found out that even recent state-of-the-art systems based on BERT (Devlin et al., 2018) or ELMO (Peters et al., 2018) embeddings are not completely robust against such natural noise. They also re-trained the systems on adversarial noisy data and observed improvement for certain error types.

The main goal of our work is to evaluate performance of the current systems under natural noisy data scenario and propose methods to mitigate the potential drops in their performance. Ideally, annotators would create as realistic data as possible. However, because this approach would be too costly, especially when considering multiple systems in multiple languages, we decided to propose an automated method introducing errors in texts, which corresponds to human errors as much as possible.

To meet the requirement that the noised data are realistic, we decided to infer the error probabilities from M2 files used in GEC datasets. These datasets contain for each noisy sentence a set of correcting edits and their type annotations. Given these annotations, we define the following set of error types and estimate their probabilities:

- **Diacritics** Strip diacritics either from a whole sentence or randomly from individual characters.
- **Spelling** Use ASpell to transform a word to other existing word (*break* → *brake*) or introduce the usual perturbations on individual characters (*wrong* → *worng*)
- **Suffix/Prefix** Change common suffix/prefix (*do* → *doing*).
- **Casing** Change casing of a word.
- **Punctutation** Insert, remove or replace certain punctuation in text.
- **Whitespace** Remove or insert spaces in text.
- **Word Order** Reorder several adjacent words.
- **Common Other** Insert, replace or substitute common errors as seen in data (*the* → *a*, *on* → *in*).

We plan to infer the operation probabilities for all 4 languages (English, German, Russian, Czech) for which the M2 GEC files exist. Moreover, as English and Czech have M2 files divided into multiple categories, such as natives or second learners, we could create more detailed user profiles.

We will use these profiles to test multiple systems. At the moment, we have chosen the following tasks: morphosyntactic analysis of Universal Dependencies (Nivre et al., 2020), neural machine translation, reading comprehension on the SQuAD dataset (Rajpurkar et al., 2018), several text classification tasks from the GLUE benchmark (Wang et al., 2018) and dialog system slot filling on Czech Public Transport data[3].

Once we evaluate all proposed tasks, we plan to implement two methods for handling the input noise. Firstly, we want to employ a trained GEC system to correct the noisy texts and only then pass them to the models themselves. Secondly, for all systems where possible, we will re-train the systems with a mixture of the original and the noisy data. Ideally, the second approach would not hurt the model's performance on clean data while improving its performance on the noisy data.

## 5 Summary

In this thesis proposal we described the task of natural language correction, together with its current state and a brief history. We presented our work on automatic diacritics restoration and we also discussed our system for grammatical error correction that participated in Building Educational Applications 2019 Shared Task on Grammatical Error Correction and our following work on training correction systems in low resource scenarios.

The main focus of our future work will be on automatic language correction in Czech. To cover broad range of its possible users, we plan to assemble dataset from multiple domains. Apart from the standard sentence-level systems, we also plan to try systems operating on multiple sentences. Besides this research, we also plan to test current systems for various natural language processing tasks with noisy user generated data and explore

[3] http://gitlab.com/ufal/dsg/dialmonkey

methods to make the systems more robust to the noise.

## References


Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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

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

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.

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 Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75.

Christopher Bryant, Mariano Felice, and Edward Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. Association for Computational Linguistics.

Flora Ramírez Bustamante and Fernando Sánchez León. 1996. Gramcheck: A grammar and style checker. In *Proceedings of the 16th conference on Computational linguistics-Volume 1*, pages 175–181. Association for Computational Linguistics.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

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 *Thirty-Second AAAI Conference on Artificial Intelligence*.

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.

David Crandall. 2005. Automatic accent restoration in spanish text. *Indiana University Bloomington*.

Daniel Dahlmeier and Hwee Tou Ng. 2012a. 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.

Daniel Dahlmeier and Hwee Tou Ng. 2012b. 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 Eric Jun Feng Ng. 2012. Nus at the hoo 2012 shared task. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 216–224.

Robert Dale, Ilya Anisimoff, and George Narroway. 2012. 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.

Robert Dale and Adam Kilgarriff. 2011. 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.

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

Mariano Felice and 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.

Mariano Felice, Zheng Yuan, Øistein E Andersen, Helen Yannakoudakis, and Ekaterina Kochmar. 2014. Grammatical error correction using hybrid systems and type filtering. Association for Computational Linguistics.

G David Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE.

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*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org.

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 and Marcin Junczys-Dowmunt. 2018. Near human-level performance in grammatical error correction with hybrid machine translation. *arXiv preprint arXiv:1804.05945*.

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.

Na-Rae Han, Martin Chodorow, and Claudia Leacock. 2004. Detecting errors in english article usage with a maximum entropy classifier trained on a large, diverse corpus. In *LREC*.

George E. Heidorn, Karen Jensen, Lance A. Miller, Roy J. Byrd, and Martin S Chodorow. 1982. The epistle text-critiquing system. *IBM Systems Journal*, 21(3):305–326.

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

Jianshu Ji, Qinlong Wang, Kristina Toutanova, Yongen Gong, Steven Truong, and Jianfeng Gao. 2017. A nested attention neural hybrid model for grammatical error correction. *arXiv preprint arXiv:1707.02026*.

Marcin Junczys-Dowmunt. 2019. Microsoft translator at wmt 2019: Towards large-scale document-level neural machine translation. *arXiv preprint arXiv:1907.06170*.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2014. 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.

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*.

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

Kevin Knight and Ishwar Chander. Automated postediting of documents.

Karen Kukich. 1992. Techniques for automatically correcting words in text. *Acm Computing Surveys (CSUR)*, 24(4):377–439.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2010. Automated grammatical error detection for language learners. *Synthesis lectures on human language technologies*, 3(1):1–134.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *arXiv preprint arXiv:1704.08006*.

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*.

Nina Macdonald, Lawrence Frase, P Gingrich, and Stacey Keenan. 1982. The writer's workbench: Computer aids for text analysis. *IEEE Transactions on Communications*, 30(1):105–110.

M McIlroy. 1982. Development of a spelling list. *IEEE Transactions on Communications*, 30(1):91–99.

Rada Mihalcea and Vivi Nastase. 2002. Letter level learning for language independent diacritics restoration. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–7. Association for Computational Linguistics.

Rada F Mihalcea. 2002. Diacritics restoration: Learning from letters versus learning from words. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 339–348. Springer.

Guido Minnen, Francis Bond, and Ann Copestake. 2000. 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.

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.

Ryo Nagata, Takahiro Wakana, Fumito Masui, Atsuo Kawai, and Naoki Isu. 2005. Detecting article errors based on the mass count distinction. In *International Conference on Natural Language Processing*, pages 815–826. Springer.

Jakub Náplava. 2017. Natural language correction.

Jakub Náplava, Milan Straka, Jan Hajič, and Pavel Straňák. 2018. Corpus for training and evaluating diacritics restoration systems. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University, LINDAT/CLARIN PID http://hdl.handle.net/11234/1-2607.

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.

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.

Joakim Nivre, Željko Agić, Lars Ahrenberg, et al. 2017. Universal dependencies 2.0. lindat/clarin digital library at the institute of formal and applied linguistics, charles university, prague.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Jan Hajic, 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 4027–4036, Marseille, France. European Language Resources Association.

Bojar Ondřej, Rajen Chatterjee, Federmann Christian, Graham Yvette, Haddow Barry, 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.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

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

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. *arXiv preprint arXiv:1707.05236*.

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.

Alexandr Rosen and Tatranské Matliare. 2016. Building and using corpora of non-native czech. In *ITAT*, pages 80–87.

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.

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.

Nikola Šantić, Jan Šnajder, and Bojana Dalbelo Bašić. 2009. Automatic diacritics restoration in croatian texts. *INFuture2009: Digital Resources and Knowledge Sharing*, pages 309–318.

Kevin P Scannell. 2011. Statistical unicodification of african languages. *Language resources and evaluation*, 45(3):375.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

WD Taylor. 1981. Grope—a spelling error correction tool. *AT& T Bell Labs Tech. Mem.*

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, Brussels, Belgium. Association for Computational Linguistics.

Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y Ng. 2016. Neural language correction with character-based attention. *arXiv preprint arXiv:1603.09727*.

David Yarowsky. 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.

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. 2017. Grammatical error correction in non-native english. Technical report, University of Cambridge, Computer Laboratory.

Zheng Yuan and Ted Briscoe. 2016. 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.

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, Sofia, Bulgaria. Association for Computational Linguistics.

Wajdi Zaghouani, Behrang Mohit, Nizar Habash, Ossama Obeid, Nadi Tomeh, Alla Rozovskaya, Noura Farra, Sarah Alkuhlani, and Kemal Oflazer. 2014. 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*.