UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Agnes Luhtaru

Low-resource Grammatical Error Correction via Synthetic Pre-training and Monolingual Zero-shot Translation

Master's Thesis (30 ECTS)

Supervisor: Mark Fišel, PhD

Tartu 2022

Low-resource Grammatical Error Correction via Synthetic Pre-training and Monolingual Zero-shot Translation

Abstract:
State-of-the-art neural grammatical error correction (GEC) systems are valuable for correcting various grammatical mistakes in texts. However, training neural models requires a lot of error correction examples, which is a scarce resource for less common languages. We study two methods that work without human-annotated data and see how a small GEC corpus improves the performance of both models. The first method we explore is pre-training using mainly language-independent synthetic data. The second one is correcting errors with multilingual neural machine translation (NMT) via monolingual zero-shot translation. We found that the model trained using only synthetic data corrects few mistakes but rarely proposes incorrect edits. On the contrary, the NMT model corrects many different mistakes but adds numerous unnecessary changes. Training with the GEC data decreases the differences between the models - the synthetic model starts to correct more errors, and the NMT model is less creative with changing the text.

Keywords:
natural language processing, neural machine translation, grammatical error correction

CERCS: P176 Artificial intelligence

Grammatiliste vigade parandamise meetodid väheste ressurssidega keeltele — sünteetilistel andmetel treenimine ja ühekeelne *zero-shot* tõlge

Lühikokkuvõte:
Närvivõrkudel põhinevad grammatiliste vigade parandamise meetodid on edukad tekstides eri grammatikavigade parandamises. Selliste süsteemide treenimine nõuab palju veaparandusnäiteid, mida väiksema kõnelejaskonnaga keeltel napib. Me võrdleme kahte lähenemist, mis ei vaja märgendatud andmeid ja vaatame, kuidas väikese veaparanduskorpusega edasi treenimine mudelite käitumist mõjutab. Esimene meetod keskendub sünteetilistel andmetel eeltreenimisele, teine kasutab mitmekeelset masintõlget, et parandada vigu ühekeelse *zero-shot* tõlkega. Analüüs näitab, et sünteetiliste andmetega treenitud mudel parandab vähe vigu, kuid pakub harva vääraid muudatusi. Mitmekeelne masintõlke mudel parandab palju vigu, kuid teeb ka mitmeid ebavajalikke asendusi. Süsteemide veaparandusnäidetega edasi treenimine vähendab erinevusi kahe mudeli vahel: sünteetiliste andmetega treenitud mudel hakkab parandama rohkem vigu ja masintõlkemudel asendab vähem õigeid sõnu.

Võtmesõnad:
loomuliku keele töötlus, neuromasintõlge, grammatiliste vigade parandus

CERCS: P176 Tehisintellekt

Contents

1 Introduction

The grammatical error correction (GEC) task involves correcting various errors native speakers or language learners make while writing texts. It includes, for example, simple typos, subject-verb agreement errors, complex lexical decisions etc. The desired performance of a grammar checker is correcting erroneous parts of the text without changing the original meaning and wording.

Before neural networks became dominant in different natural language processing (NLP) tasks, especially machine translation, grammatical error correction approaches were mainly based on language-specific rules (Clément et al., 2009) or statistical methods (Yuan and Felice, 2013). When Junczys-Dowmunt et al. (2018) showed that we could approach GEC as a translation task, neural networks took over the GEC research. Based on the method proposed by Junczys-Dowmunt et al. (2018), one could view GEC as translation from erroneous text to correct text and use similar systems that work for neural machine translation (NMT) for training GEC models.

The approaches based on neural networks are resource-heavy. For training proper NMT models, it's vital to have a reasonable amount of error correction data. For English, researchers have created multiple corpora like The First Certificate in English (FCE) (Yannakoudakis et al., 2011), Lang-8 Corpus of Learner English (Mizumoto et al., 2011; Tajiri et al., 2012) and Write and Improve (W&I) (Yannakoudakis et al., 2018), etc. So, there are over million publicly available erroneous sentences with corrections, and training NMT models is not a problem. For Estonian, quite the opposite; we have less than 10 thousand corrected sentences in the University of Tartu's Language Learner's corpus (Rummo and Praakli, 2017).

Building a comparable NMT system out of the box requires more data. As part of this master's thesis, we find ways to use the other available data to cope with data scarcity and incorporate the limited GEC corpora. We compare two existing approaches, one that uses monolingual data and the other solely learning from parallel data. We see how well these approaches perform without error correction examples and how can we continue training with a small GEC corpus.

The first approach was proposed for the BEA-2019 Shared Task on GEC (Bryant et al., 2019). Competition organisers introduced the Low Resource Track with limited annotated data to develop less resource-demanding methods. They aimed to encourage researching GEC methods suitable for languages lacking the training data. Grundkiewicz et al. (2019) as the most successful team used monolingual data for generating synthetic error correction datasets for pre-training the model. They show that it is possible to fine-tune a model trained with artificial data with a small GEC corpus. Náplava and Straka (2019) applied this approach to other languages as well. We will partly re-produce the work of Grundkiewicz et al. (2019) in English and train similar models for Estonian.

The second approach we are analysing uses parallel translation data for grammatical error correction. Korotkova et al. (2019) proposed the method of using the multilingual

NMT model to achieve the correction via monolingual zero-shot translation. The main disadvantage of this method is low precision; the model changes correct text (Korotkova et al., 2019; Luhtaru, 2020). We are evaluating the GEC performance of the multilingual NMT model containing significantly more data and languages than the ones Korotkova et al. (2019) and Luhtaru (2020) trained, and we are building a model with modifications proposed by Luhtaru (2020) on top of an already existing model. Also, we continue training the final model with GEC data; there are no published attempts to do so.

Our main aim is to study methods we could use for low-resource languages. We train models in Estonian and English. Estonian is an example of a language with limited GEC data, and English models give a better comparison with other works and a more detailed evaluation. Because of that, for Estonian models, we use the available GEC resources, but we limit the data used for English to keep the models comparable. We analyse both approaches to understand how these work and find possibilities for further improvements. We are searching for answers to the following questions.

- How does the GEC performance of the synthetic pretraining and NMT models differ? Which one is better?
- How does the performance change when we continue training the models with limited GEC data?
- Are English and Estonian behaving similarly? Does synthetic pretraining work for Estonian?

In the next section, we briefly explain other methods used for GEC. The methodology gives a detailed overview of the approaches compared in this work and discusses the main evaluation methods. The experiments section explains the data we used, preprocessing pipeline, the training process and evaluation details. The results section shows the scores and error category analyses for all the main models. Finally, we discuss how the approaches differ and propose ideas for future work.

2 Related Work

This section briefly discusses the main trends in grammatical error correction (GEC) research and points out some other works with novel approaches from recent years.

Recent research has interpreted GEC as a sequence generation task similar to low-resource NMT. Since the GEC data is insufficient to train a robust translation system without additional modification, most recent works have focused on finding ways to cope with data scarcity. The use of synthetic data for pre-training has been tremendously popular. We use the method proposed by Grundkiewicz et al. (2019), but there are many other works on that topic (Lichtarge et al., 2019; Xu et al., 2019; Choe et al., 2019; Takahashi et al., 2020).

One of the recent methods for adding artificial mistakes is introduced by Stahlberg and Kumar (2021) and generates a language-specific synthetic corpus using the Seq2Edits model created for sequence transduction problems, where few changes are needed (Stahlberg and Kumar, 2020). They predict edits as tags with the Seq2Edits model for corrupting the text. Their method allows to consider error type frequency similar to the development set and add various realistic errors.

Generating multiple hypotheses to choose the best one is also a successful technique (Grundkiewicz et al., 2019). One of the more distinctive ways to determine which sentence to pick is using grammatical error detection and comparing how much the detection system's predictions match the error correction system's changes (Yuan et al., 2021). They also incorporate a detection system in their GEC model's architecture.

There have also been some different approaches for GEC that are not solely NMT systems. One of the recent multilingual works on GEC (Rothe et al., 2021) builds their model on top of the multilingual text-to-text transfer transformer (mT5) (Xue et al., 2020), which has been trained to solve various natural language processing (NLP) tasks. They add GEC pre-training, which corrupts text with different simple synthetic errors, like swapping and inserting characters but leave behind the language-specific parameters and other more sophisticated approaches for data generation.

It is also possible to correct errors only using language models (Bryant and Briscoe, 2018; Alikaniotis and Raheja, 2019). The main idea is to create confusion sets for different categories, like pronouns, and score sentences with various options to determine if the original token is remarkably worse than other candidates.

Researchers from Grammarly (Omelianchuk et al., 2020) contrastingly doubt if NMT-based methods are the best. They argue that these models have low inference speed, require a lot of data and are not easily interpretable. They proposed we should see GEC as a sequence tagging task instead of a sequence generation task. We could correct mistakes via training the existing transformer encoders, like BERT (Devlin et al., 2019), to tag the type of changes needed to fix the sentence. They tag transformations such as "transform case capital", "delete", and "append_the" iteratively and correct the sentence until it is correct.

3 Methodology

In this section, we first explain how neural machine translation (NMT) works. As already mentioned above, at their core the GEC approaches we are exploring are NMT systems. Then we describe the synthetic data generation method briefly and discuss how correction via monolingual zero-shot translation works with multilingual NMT. Finally, we give an overview of the evaluation methods.

3.1 Neural Machine Translation

Current state-of-the-art NMT systems use an encoder-decoder structure. The system's encoder compresses the input text into representations, and the decoder generates a sequence using the encoder's output. NMT systems generate text autoregressively; at each step, the decoder generates the probability of being the next token for every item in the vocabulary based on encoded representations. The decoder uses the already constructed sequence and encoded representations during the generation, as shown in Equation 1.

$$p(x_i^{(output)}|x_{1\ldots i-1}^{(output)}, x_{1..J}^{(input)}) = \ldots \tag{1}$$

Since Vaswani et al. (2017) introduced the transformer, it has been the leading architecture for machine translation and other tasks as well. Before transformers, common architectures for sequence to sequence tasks used convolution (Gehring et al., 2017) or recurrence (Bahdanau et al., 2014). Instead, transformers rely only on the attention mechanism that Bahdanau et al. (2014) first used in addition to recurrent neural networks. The attention mechanism allows the system to pay attention to some of the tokens in the sequence more than others. Self-attention in the transformer makes it possible for the system to consider the rest of the sentence while encoding a word. In the decoder, the attention mechanism allows paying attention to different parts of encoded representations and already constructed sequence. Both transformer's encoder and decoder consist of blocks containing multi-head attention and feed-forward layers.

Since languages are constantly changing, it is impossible to create a vocabulary containing every existing word. In order to deal with unknown tokens, we can use a method proposed by Sennrich et al. (2016). They suggest using the adapted Byte Pair Encoding (BPE) (Gage, 1994) algorithm to divide less common words into subword units. The algorithm analyses text and creates a vocabulary with a specified size based on which character sequences often appear together in the text (Sennrich et al., 2016). With a well-chosen vocabulary size parameter common words stay together and rare words appear as pieces.

3.2 Using Monolingual and Parallel Data

In this subsection, we explain how we train the GEC model with a synthetic dataset using the methodology proposed by Grundkiewicz et al. (2019) and use a similar multilingual NMT model as Korotkova et al. (2019) to correct grammatical errors.

3.2.1 Pretraining with Synthetic Data

We are following the method proposed by Grundkiewicz et al. (2019), which makes use of monolingual data by creating a synthetic dataset via adding simple errors through random edit operations. The model is a regular NMT model with a single direction in which the source language is erroneous text and the target language is correct text. The model follows a similar NMT architecture with some of the GEC modifications, like increased dropout, as Junczys-Dowmunt et al. (2018) specified.

As discussed in the previous paragraph, synthetic data has been used widely for GEC. We can randomly substitute, insert, delete and swap words or use more advanced logic. The most novel part of the synthetic data generation method proposed by Grundkiewicz et al. (2019) is the reverse speller idea. They generate the confusion set for substitution from the spellchecker's suggestion list. This method ensures we replace the token with a similar one. These errors are not context-aware, but it is possible to apply this method for every language with a suitable spellchecker without knowing the language or having GEC corpora.

After training the model separately on synthetic data, one can fine-tune it with GEC corpora. For the models trained by Grundkiewicz et al. (2019), the fine-tuning set contains a larger GEC corpus, or the WikEd Error Corpus (Grundkiewicz and Junczys-Dowmunt, 2014) for Low Resource Track. Unlike them, we are using only a tiny subset of the GEC corpus without additional resources to mimic the available data for Estonian. Grundkiewicz et al. (2019) found that the best strategy for continuing training with the GEC data is keeping the state of the learning rate scheduler and optimiser, that means only the training data changes. Another approach would be initialising the weights from the synthetic model resetting the learning rate scheduler's and optimiser's parameters. We follow the first method, which Grundkiewicz et al. (2019) call fine-tuning.

3.2.2 Correction via Monolingual Zero-shot Translation

The multilingual NMT model incorporates multiple languages into one model and learns to translate with sampling data from different directions. Johnson et al. (2017) show that in such a system zero-shot translation is possible. When the model has seen examples from Portuguese to English and English to Spanish but has not trained on Portuguese to Spanish sentences, the model can still generate that translation.

Korotkova et al. (2019) show that it is possible to achieve monolingual translation in the same way. For example, the model trained only on parallel data but has Estonian on

both the source and target sides and it can produce a zero-shot translation from Estonian to Estonian (see Figure 1). The model has trained to encode the text in one language and produce a sentence with the same meaning in another specified language. It has never seen monolingual examples and is trained only on grammatically correct text. When we ask for the monolingual translation, the model encodes the sentence as it would when translating and decodes output based on that representation. Instead of copying the input, the model aims to generate grammatically correct text, which enables the model to correct errors when producing a monolingual translation. As a downside, monolingual translation suffers from low precision because it is not trained to keep the wording and structure of the sentence and changes words to synonyms, etc. (Korotkova et al., 2019).

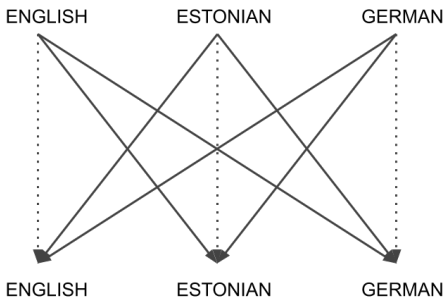

Figure 1. Schematic of grammatical error correction via zero-shot translation. Full lines show language pairs in training data, dotted lines represent zero-shot translations.

As opposed to Korotkova et al. (2019) and Luhtaru (2020), we use a slightly different multilingual NMT architecture. Multilingual NMT models can have separate encoders and decoders for each language in the system or share one encoder and decoder between all languages (Lyu et al., 2020). We use the architecture where all languages have separate encoders and decoders, which is called modular multilingual NMT architecture.

For our experiments, we use the modular model trained by Purason (2021) containing more languages and data than previously reported work on error correction via multilingual NMT (Korotkova et al., 2019; Luhtaru, 2020). Luhtaru (2020) showed that adding noise to the translation data's source sentences improves the GEC performance. Training massive NMT models from scratch is time-consuming and expensive, so we initialise the weights from the big NMT model trained by Purason (2021) and train the model with fewer sentences and language pairs.

We can try to continue training the multilingual NMT model with GEC data by adding a new translation direction. With language-specific encoders and decoders we can use, for example, an English encoder for encoding the erroneous sentences and an English decoder for decoding the correct output (adding from English to English

direction). In that way, the model which was previously trained to translate starts training using GEC data. The translation direction added is new to the model.

3.3 Evaluation

There are two main automatic evaluation approaches for GEC – extracting and comparing edits versus calculating n-gram overlap. The first one is proposed as a method called MaxMatch (M^2) (Dahlmeier and Ng, 2012) and improved in Error Annotation Toolkit (ERRANT) (Bryant et al., 2017). The second one is used in A Fluency Corpus and Benchmark for Grammatical Error (JFLEG) GLEU scorer (Napoles et al., 2017).

We can view the workflow of M^2 and ERRANT scorers in two steps. At first, it is necessary to extract edits. The scorer compares the GEC system's input and output sentences. M^2 scorer extracts the edits using classical Levenshtein distance (Levenshtein, 1966). After extracting the edits, the script merges the token level edits, choosing the maximally matching sequence with the annotators' corrections. Merging is necessary because phrases can count as one edit and annotators mark errors differently (Dahlmeier and Ng, 2012). Unlike M^2, the ERRANT scorer uses the set of language-dependent rules to merge edits (Felice et al., 2016) and also classifies edits with the rule-based algorithm.

The second step is comparing edits extracted by the scorer with human-created gold annotations (Dahlmeier and Ng, 2012). Scorers compute precision, recall and F-score. Before The CoNLL-2014 Shared Task on Grammatical Error Correction, authors reported a regular F1 score, but now the primary evaluation metric is $F_{0.5}$-score. It counts precision twice as important as recall. According to Ng et al. (2014), $F_{0.5}$-score is better because grammar checker users expect reliable edits but do not mind if some errors are not corrected.

Both M^2 and ERRANT scorers require gold annotations. The M^2 scorer is more language-independent because, in addition to annotations, the ERRANT scorer needs language-dependent merging and error classification rules. For Estonian, we have resources for neither of the scorers. For the M^2 scorer, we lack the gold annotations, and for the ERRANT scorer, the merging and classification rules are also not implemented.

The GLEU score is similar to the BLEU score widely used to evaluate machine translation (Papineni et al., 2002). It compares n-grams in system output and reference sentences. The only difference between GLEU and BLEU is that GLEU adds a penalty to n-grams that contain corrections in human reference but are left unchanged in the system output (Napoles et al., 2017). The GLEU scorer is more reliable when there are multiple different corrections for one incorrect sentence. So, the GLEU scorer only needs parallel sentences to calculate the score and is readily usable for English and Estonian.

4 Experiments

In this section, we first give a short overview of all the models we train or evaluate. After that, we list which data we use and preprocessing schemes we follow. Then, we explain how we generate synthetic mistakes and specify model architecture and training details. Finally, we state the evaluation method and test sets used.

4.1 Models

We are focusing on two approaches, synthetic pre-training and multilingual NMT experiments. Since this work contains various models trained with different data, we first give a brief overview of the experiments. Table 1 includes information on all of the main models. We will explain data and architecture more in detail in following sections.

We train models with synthetic monolingual data separately for English and Estonian. We also fine-tune these using GEC corpora. It means we have two pre-trained models and two fine-tuned models. In addition to the main models, we train three English pre-train models with different data volumes (2.5, 5, and 10 million sentences) to assess the importance of an enormous corpus.

Model	Languages	Data volume	Data type	Param init
synthetic 2x	EN/ET	20M/20M	M	-
synthetic + GEC 2x	EN/ET	10k/7k	G	synthetic
NMT (Purason, 2021)	7 lang	∼300M	P	-
noisy NMT 1x	EN, ET, DE	30M	P	NMT
noisy NMT + GEC 2x	EN/ET	10k/7k	G	noisy NMT

Table 1. The synthetic pre-training and multilingual neural machine translation (NMT) models we train (or evaluated) as part of this work's main experiments, the data type corresponds to monolingual (M), parallel (P) or GEC corpus (G).

For multilingual NMT, we evaluate the big modular model by Purason (2021) and train three additional models. We train the noisy NMT model by initialising the weights from the big NMT model. The noisy NMT contains three languages, the minimum number of languages needed for the zero-shot effect.[1] Although it is possible to continue training the multilingual NMT model using multiple languages, we decided to train separate models for English and Estonian.

[1]Luhtaru (2020) noted that the NMT model with two directions does not produce a decent zero-shot translation. We tried to train one from scratch with modular NMT architecture but failed. It might be possible that the zero-shot translation is still good when we initialise the weights from the model that already has that capability.

4.2 Data

As already briefly mentioned, we use three different types of data: monolingual corpora, parallel machine translation corpora and GEC corpora. In this section, we explain all three.

4.2.1 Monolingual Data

Generating synthetic corpora for pre-training requires decent-sized monolingual corpora. For English, we use News Crawl, open-source monolingual corpus collected from online newspapers (Barrault et al., 2019). We uniformly sample sentences from the years 2007 to 2020. The main synthetic model contains 20 million sentences. We also created training sets of 2.5, 5 and 10 million sentences for data volume experiments.

For Estonian, we use all available data from News Crawl 2014-2021 (around 8.4 million sentences) and sample additional data (approximately 11.6 million sentences) from Estonian National Corpus 2019 (Koppel and Kallas, 2020) containing texts from Wikipedia, web, Open Access Journals, etc.

4.2.2 Parallel Data

As stated above, we use the septilingual model trained by (Purason, 2021) and train the noisy NMT model with English, Estonian and German. We use a subset of data used for training the septilingual model for the noisy model.

We randomly sample five million sentences for each direction (en-et, en-de, et-en, et-de, de-en and de-et) and add noise on the input side. The method for noise generation is the same we use for synthetic pre-train.

4.2.3 Error Correction Data

For English, we use a subset of combined Cambridge English Write & Improve (W&I) (Yannakoudakis et al., 2018), and the LOCNESS (Granger, 1998) corpus. This jointly called W&I+LOCNESS corpus was introduced during Building Educational Applications (BEA) 2019 shared task on grammatical error correction (Bryant et al., 2019) and contains texts from language learners (levels A, B, C) and also from native English speakers. We chose that dataset for fine-tuning because it allows separate language-level evaluation and is often used in recent literature (Grundkiewicz et al., 2019; Omelianchuk et al., 2020; Xue et al., 2020). We sampled the subset of 10 000 sentences to mimic the low-resource scenario for English.[2]

[2]We did not follow the BEA Shared Task Low Resource Track's requirement of only using the W&I+LOCNESS development set because we preferred to mimic the number of sentences we could realistically have for Estonian.

For Estonian, we use all available 7121 sentences from the UT GEC corpus (Rummo and Praakli, 2017) training set. This corpus contains sentences from language learners.

4.3 Preprocessing

We use two different preprocessing pipelines, one for synthetic pretrain experiments and the other for NMT experiments.

For NMT experiments, we follow the preprocessing scheme used by Purason (2021). It has one step, splitting text into subword units with the BPE segmentation (Sennrich et al., 2016) algorithm using SentencePiece (Kudo and Richardson, 2018). Each language has its own model and vocabulary containing 16 000 subwords.

For synthetic pre-train experiments, we use the preprocessing scheme used in BEA Shared Task (Bryant et al., 2019) and by Grundkiewicz et al. (2019) with some modifications. We first normalise the punctuation using the script from the Moses toolkit (Koehn et al., 2007). Then we tokenize the text using spaCy[3] for English and EstNLTK (Laur et al., 2020) for Estonian. We truecase text using TartuNLP truecaser[4] and split the text into subwords using the SentencePiece models trained for septilingual NMT model by Purason (2021). The reasoning behind reusing the SentencePiece model is having the same vocabulary for both approaches, which can be useful in future work.

4.4 Artificial Errors

We generate artificial errors for both the synthetic pre-train and noisy NMT models. The methodology is the same, but for synthetic models, the errors are a substitution for GEC data, whereas in the case of the NMT model, they are effectively noise, making the encoder more familiar with the broken text. We also generate errors in German for the NMT model.

We generate errors using the same principles as Grundkiewicz et al. (2019). For every sentence, we sample the error probability from the normal distribution. We use the mean of 0.15 and a standard deviation of 0.2 for every language. We multiply this probability by the sentence length and use four types of operations: substitute, insert, delete and swap. For substitution, we use Aspell speller[5] generated confusion set. Other operations are with a random word.

For English, we choose the operation type probabilities from previous work (Grundkiewicz et al., 2019). For German, we modify previously used probabilities to match our operation types (Náplava and Straka, 2019). For Estonian, we arbitrarily pick probabilities as a mixture of probabilities used for two previous languages (see table 2). In addition, like Grundkiewicz et al. (2019), we also perturb characters in 10% of the words.

[3]https://spacy.io/

[4]https://github.com/TartuNLP/truecaser

[5]http://aspell.net/

Language	Substitute	Insert	Delete	Swap
English	0.7	0.1	0.1	0.1
German	0.65	0.2	0.1	0.05
Estonian	0.65	0.15	0.1	0.1

Table 2. Probabilities for choosing the error category for three languages (German probabilities for the noisy NMT model only).

Subword	Language	Example substitutions
_relating	en	_ref l ating, _rel ay ing, _rel o ading
_signific	en	_sign ified, _sign ifies, _sign ify
ified	en	ivied, offed, effed
_teadsin	et	_sead sin, _teadsid, _tea ksin
se	et	sae, sea, see
_kunst	et	_ku nts, _kun ste, _kunsti

Table 3. Examples of the substitutions generated.

We add synthetic errors to subword segmented text, so we generate substitutions with subwords and consider if it is at the beginning of the word (starting with "_") or not. All substitution examples are also split into subwords if they do not appear in the vocabulary (see Table 3).

4.5 Model Architecture and Parameters

We use the Fairseq library (Ott et al., 2019) for training all the models. We chose Fairseq because it is widely used, well-documented and implemented in Python using PyTorch[6] making it convenient to use.

All models have transformer architecture (Vaswani et al., 2017) where the encoder and decoder have six blocks with 8-head self-attention. The embedding vectors are of a size 512 and feed-forward layers of size 2048.

All models use Adam optimiser (Kingma and Ba, 2014) with inverse square root learning rate scheduler. We train synthetic models with 8000 warm-up updates and a learning rate of 2e-4. Purason (2021) trained the modular NMT model with 4000 warm-up steps and a learning rate of 8e-4. We used the same parameters when training the noisy NMT model. Label smoothing with the weight of 0.1 is used for all of the models. For synthetic models, we use a dropout of 0.3, attention and activation dropout of 0.1, and for NMT models, all dropout parameters are 0.1.

[6]https://pytorch.org/

4.6 Training

We train all the models at the University of Tartu HPC Center (University of Tartu, 2018) on 1 NVIDIA Tesla V100 GPU accumulating gradients for 12 iterations for synthetic models and 40 for NMT models. We use a batch size of 15000 tokens.

Synthetic models are validated on BLEU (Papineni et al., 2002) after every 5000 updates and we use early stopping with the patience of 10.[7] English model with 20 million sentences trained for 60 epochs (final learning rate 3.6e-05), Estonian 96 (final learning rate 3.2e-05). We start fine-tuning from the last epoch's weights, keep the state of learning rate scheduler and use the same ending criteria. The English model with 10k sentences trained for 28 epochs and the Estonian model with around 7k sentences trained for 42.

We validate the noisy NMT model on BLEU and train for 30 epochs (final learning rate 2.9e-4). As for fine-tuning synthetic models, we keep the state of the learning rate scheduler. English model trained for 54 epochs and Estonian 71.

4.7 Evaluation

We use automatic metrics to evaluate the models. We generate hypotheses for development and test sets with default beam size of 5. We evaluate English mainly with ERRANT (Bryant et al., 2017) and report scores on five different sets:

- W&I+LOCNESS development (Yannakoudakis et al., 2018; Granger, 1998; Bryant et al., 2019) (4384 sentences),
- W&I+LOCNESS test (Yannakoudakis et al., 2018; Granger, 1998; Bryant et al., 2019) (4477 sentences),
- FCE test (Yannakoudakis et al., 2011) (2695 sentences),
- CoNLL-2014 test (Ng et al., 2014) (1312 sentences),
- JFLEG test (Napoles et al., 2017) (747 sentences).

We use W&I+LOCNESS as the primary evaluation corpus. The development set is public but Bryant et al. (2019) decided to withhold the gold annotations for the test set to encourage fairer GEC research. They also added 5 extra annotation alternatives for the test set, so it better accounts for different but still valid corrections (Bryant et al., 2019). The W&I+LOCNESS sets contain texts written by authors with different language levels (A, B, C and native), and it is possible to evaluate these levels separately. In addition,

[7] We use BLEU instead of the methods used for evaluating GEC because it is already available in the library.

the W&I+LOCNESS sets have compatible error categories with ERRANT, and is the official evaluation corpus of BEA 2019 Shared Task (Bryant et al., 2019).

Since we also evaluate the models on other test sets, we can see if there is a gap in performance between W&I+LOCNESS and the other corpora. Except for W&I+LOCNESS, the most trustworthy is the FCE test set because it also has ERRANT compatible annotations. The CoNLL-2014 test set is developed for MaxMatch scorer (Dahlmeier and Ng, 2012), which is similar but has a different approach to edit extraction, so we still calculate ERRANT on that set but trust other scores more.

We also calculate GLEU for the JFLEG test set because Napoles et al. (2017) created the test set for the GLEU scorer. It has four different corrections for every sentence to provide variety in ways to correct one sentence (Napoles et al., 2017).

There are not so many evaluation sets and metrics for Estonian. Since MaxMatch (Dahlmeier and Ng, 2012) and ERRANT (Bryant et al., 2017) scorers require test sets in M^2-format. Still, the University of Tartu's Language Learner's corpus (Rummo and Praakli, 2017) currently only has parallel sentences, so we only evaluate the language learner's corpus development and test sets (1000 and 800 sentences) with the GLEU scorer.

5 Results

This section discusses the evaluation results for both approaches and how the error correction data changes the models' performance. We show ERRANT and GLEU scores for different test sets, error category analyses and error correction capability among different language levels.

5.1 Pre-training with Synthetic Data

We pre-trained four English models using 2.5, 5, 10 and 20 million synthetic sentences, an Estonian model using 20 million sentences and fine-tuned models separately with small GEC corpora for both languages. We discuss the results in the following sections.

5.1.1 Data Volume

Grundkiewicz et al. (2019) pre-trained their system using 100 million synthetic sentences. Training large models is quite a time consuming and expensive task. We wanted to see how the model's performance changes when increasing the data volume starting from 2.5 million sentences.

We trained the models until convergence. As we can see from Figure 2, when we increase that data quantity, the model converges later and keeps improving longer. Training the model with 2.5 million sentences took 22 hours. A model with 20 million sentences took 235 hours.

Data increase	P↑	R↑	$F_{0.5}$ ↑
$2.5 \rightarrow 5$	+1.37	+0.81	+1.42
$5 \rightarrow 10$	+2.31	+0.45	+1.29
$10 \rightarrow 20$	+0.50	+0.26	+0.49

Table 4. The change of ERRANT scores when increasing the data quantity on the W&I+LOCNESS development set.

Training models using more data requires more resources. Still, when we look at the increase in performance (see Table 4), we can see that precision and recall rise with every iteration. We get a considerable benefit even after using 10 million sentences. Although we can suggest that it is good to train models using more than 20 million sentences for maximum performance, as part of this work, we will not further increase the data quantity.

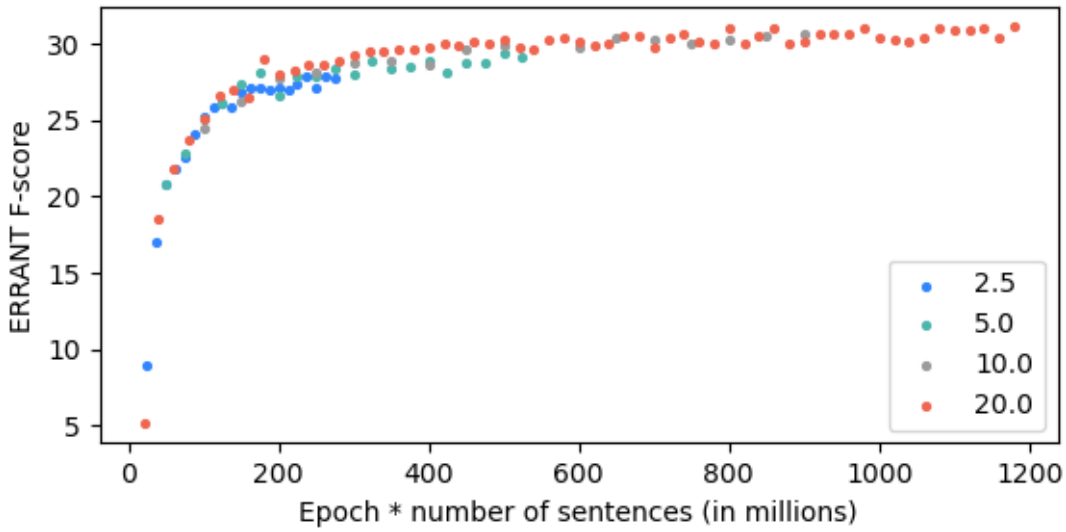

Figure 2. The increase of ERRANT $F_{0.5}$ score with training the models with 2.5, 5, 10 and 20 million sentences, evaluated on the W&I+LOCNESS development set. The horizontal axis corresponds to the epoch multiplied by the number of sentences to compare the models after training on the same amount of sentences.

5.1.2 Final Pre-trained Models

We evaluate the final pre-trained models on all test sets (see Table 5 and 6). Grundkiewicz et al. (2019) show results for their model pre-trained using synthetic data on W&I+LOCNESS development set (precision 43.2, recall 10.6 and $F_{0.5}$-score 26.76) and FCE test set ($F_{0.5}$-score 34.00). We achieve higher scores than they reported. An increase in scores can be due to a slightly different training framework or their early stopping.

Evaluation set	P	R	$F_{0.5}$
W&I+LOCNESS dev	48.04	12.84	31.03
W&I+LOCNESS test	59.44	26.87	47.8
FCE test	54.69	16.90	37.80
CoNLL test	46.87	15.97	33.79
JFLEG test	46.75	21.35	37.77

Table 5. ERRANT scores (precision, recall and $F_{0.5}$-score) for the model pre-trained on 20 million synthetic sentences.

The precision for every test set precision is between 45-60% but the recall is modest. The recall is higher for W&I+LOCNESS and JFLEG sets. That can be due to the

number of annotators. As we can see, the W&I+LOCNESS development set's recall is significantly lower than the test set's. The BEA 2019 Shared Task organisers added additional alternative corrections to the test set (Bryant et al., 2019). Also, the JFLEG test set has more ways to correct one sentence than the CoNLL and FCE test sets.

When talking about GLEU, it is also important to look at the score for completely unchanged text. GLEU evaluates the n-gram overlap between hypothesis and human-corrected references; the erroneous input sentences have correct words, which means there is an overlap even before correcting, and the score for input text is not zero. In Table 6, we show the original test set's GLEU score and our system's hypothesis score. As we can see English test set's score increases by around 13 points but Estonian only by about 5. We can predict that this is due to mainly leaving the text unchanged, but further evaluation is necessary to say anything more certain.

Evaluation set	Input	Pre-trained model
JFLEG test (EN)	40.51	53.4
UT test (ET)	27.14	32.16

Table 6. GLEU scores for unchanged input text and hypothesis of synthetic model pre-trained on 20 million sentences.

A closer look into the precision and recall per error category (see Figure 3) shows that compared to the other types, the model is very good at correcting spelling errors. The precision is over 70% and recall over 85%. The other category with decent performance is orthography (case or whitespace errors), for which the precision is over 60% and recall 40%. For other error types, the recall is mostly lower than 40%, except for some categories with few occurrences. The recall is especially low for preposition and determiner errors (9% and 17%), verb and verb tense mistakes (13% and 8%) and for a category other (11%), which contains errors not fitting into the other categories, like paraphrasing.

Based on the category analyses, we can suggest two changes for synthetic error generation. Many word order errors are not corrected and there are many false positives in that category. The recall is less than 20% and the precision slightly over 20%. Lowering the frequency for swapping words could result in higher precision. Adding more punctuation errors during artificial error generation could also be beneficial, as they are not often corrected.

It is difficult to say anything conclusive about language levels. The performance seems better when more experienced language users write texts, except for level B, in which precision and recall are lower than in other categories (see Table 7). The number of errors in text and types of mistakes language learners (A, B and C language levels) versus native speakers make affect scores, but more evaluation is necessary to highlight anything.

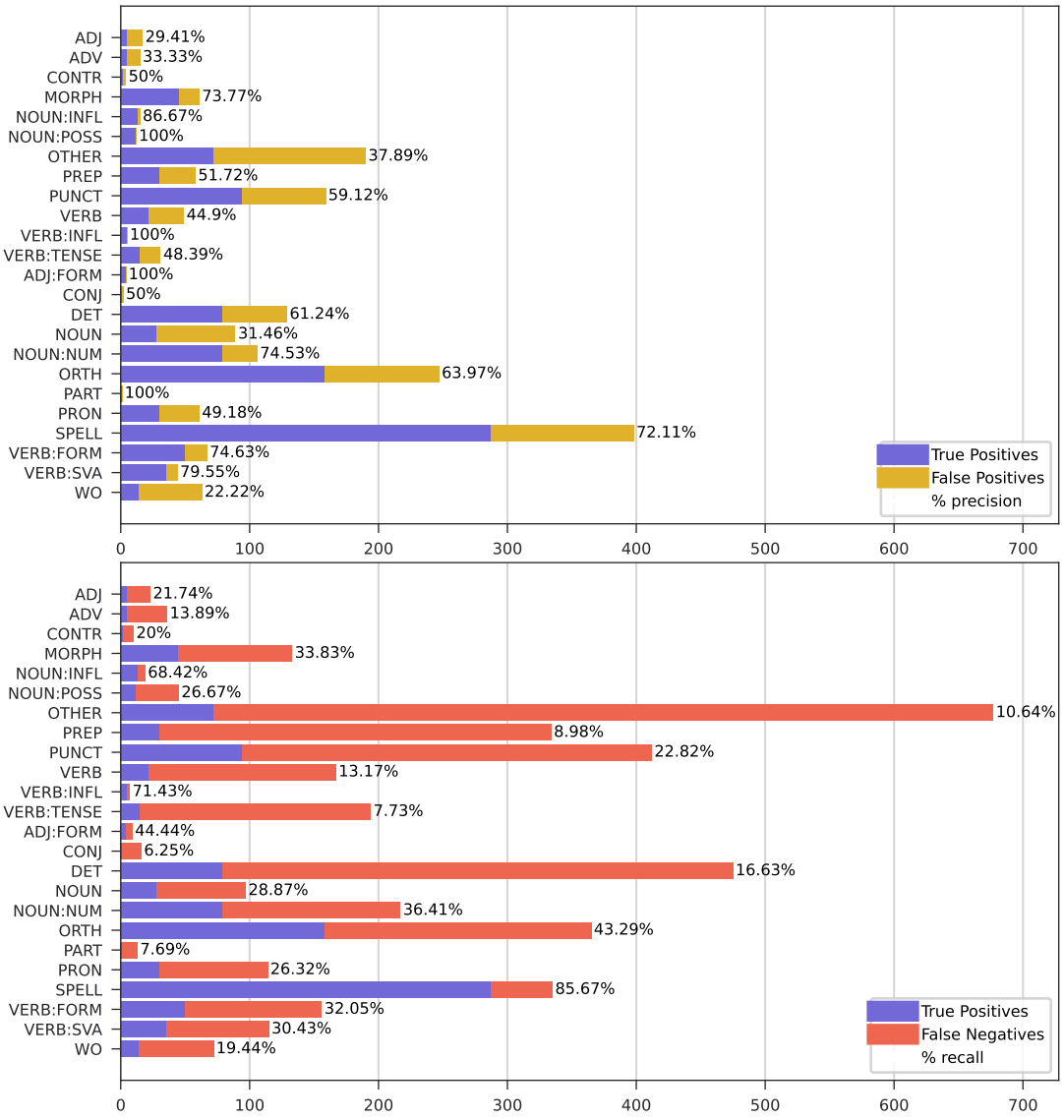

Figure 3. Synthetic pre-train true positives, false negatives, false positives, recall and precision per error category for the W&I+LOCNESS test set.

Level	P	R	$F_{0.5}$
A	60.49	25.04	47.14
B	55.21	20.87	41.54
C	57.87	28.75	48.12
Native	64.35	52.08	61.45

Table 7. ERRANT scores for different language levels for synthetic pre-train model, evaluated on W&I+LOCNESS test set.

5.1.3 Fine-tuned

We report the results for models fine-tuned only using GEC data. Grundkiewicz et al. (2019) and Náplava and Straka (2019) emphasised the importance of keeping some portion of synthetic data while fine-tuning. We tried blending the synthetic sentences with the training set, but the initial results were worse.[8] Further investigation needs to be conducted to study the effect of keeping synthetic data for fine-tuning.

	Fine-tuned model			Increase from pre-train		
Evaluation set	P	R	$F_{0.5}$	P↑	R↑	$F_{0.5}$ ↑
W&I+LOCNESS dev	49.79	31.77	44.72	+1.75	+18.93	+13.69
W&I+LOCNESS test	61.25	51.26	58.96	+1.81	+24.39	+11.11
FCE test	51.3	31.55	45.59	-3.39	+14.65	+7.79
CoNLL test	53.15	37.88	49.19	+6.28	+21.91	+15.4
JFLEG test	50.09	34.78	46.04	+3.34	+13.43	+8.27

Table 8. ERRANT scores (precision, recall and $F_{0.5}$-score) for the synthetic model fine-tuned with 10k sentences from the W&I+LOCNESS training set and comparison with the pre-train model.

Evaluation set	Fine-tuned model	Increase from pre-train
JFLEG test (EN)	58.16	+4.76
UT test (ET)	51.38	+19.22

Table 9. GLEU scores for the synthetic model fine-tuned with 10k/7k sentences from the W&I+LOCNESS/UT training set and comparison with the pre-train model.

As we can see from Table 8 recall increases substantially with fine-tuning, but precision stays in the same range. It means more mistakes are corrected, but fine-tuning also increases the number of false positives. The fine-tuned model generally adds more

[8]We trained models with 1-1, 1-2 and 1-5 splits where the amount of GEC data was always 10k sentences. Precision and recall were lower for all models on the W&I+LOCNESS and FCE test sets.

edits to the text. This behaviour is expected because the pre-trained model has good precision, but the recall is relatively low and mainly comes from correcting spelling errors. We also get substantial increase in GLEU for Estonian set (see Table 9).

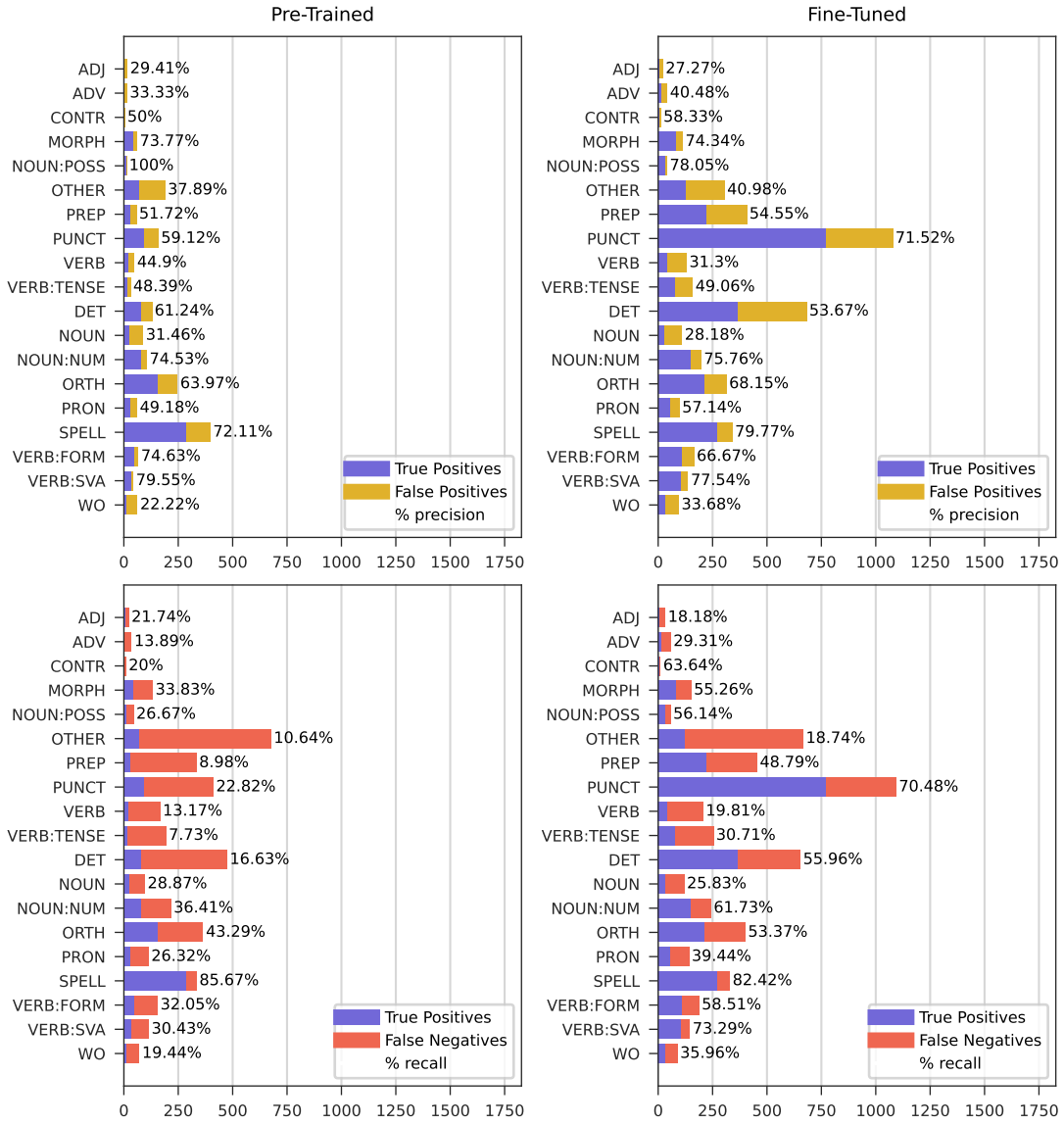

Figure 4. The synthetic pre-train and fine-tuned model's true positives, false negatives, false positives, recall and precision per error category, evaluated on the W&I+LOCNESS test set.

The most significant increase with fine-tuning is in correcting punctuation errors (see Figure 4). Recall jumps from 23% to 70%. We can also see a significant change in the recall of determiners (17% → 56%) and prepositions (4% → 49%). Also, subject-verb agreement (30% → 73%), verb form (32% → 59%) and noun number (36% → 62%) categories increase significantly. Fine-tuning improves error correction significantly in categories that are not isolated and require more sentence context.

	Fine-tuned model			Increase from pre-train		
Level	P	R	$F_{0.5}$	P↑	R↑	$F_{0.5}$ ↑
A	62.21	46.76	58.35	+1.72	+21.72	+11.21
B	62.79	47.39	58.96	+7.58	+26.52	+17.42
C	63.38	58.64	62.37	+5.51	+29.89	+14.25
Native	55.07	68.75	57.35	-9.28	+16.67	-4.10

Table 10. ERRANT scores for different language levels (W&I+LOCNESS test set) for the synthetic model fine-tuned with 10k sentences from the W&I+LOCNESS training set and comparison with the pre-train model, evaluated on the W&I+LOCNESS test set.

When we look at how the scores change for different language levels (see Table 10), we can see that for every level, all the scores increase except for precision in the case of native speakers. The training data contains only texts from language learners (A, B and C language levels) but not from native speakers, which could explain the drop in precision. The model may expect a more erroneous text and try to correct mistakes where there are none.

5.2 Multilingual Modular Neural Machine Translation

In the following sections, we first evaluate the septilingual modular neural machine translation (NMT) model's (Purason, 2021) GEC performance, compare it with the noisy multilingual NMT model and finally see how the performance changes when we continue training with GEC data.

5.2.1 Septilingual Neural Machine Translation

Other works with monolingual zero-shot translation (Korotkova et al., 2019; Luhtaru, 2020) use different NMT architecture and fewer languages and training sentences. We wanted to see if and how well the model trained by Purason (2021) corrects errors, as shown in Tables 11 and 12, the septilingual model can give a decent monolingual zero-shot translation and correct errors.

Compared to Luhtaru (2020), the septilingual modular NMT model achieves better scores without adding noise to the input. Luhtaru (2020) report ERRANT scores on

Evaluation set	P	R	$F_{0.5}$
W&I+LOCNESS dev	24.04	29.16	24.92
W&I+LOCNESS test	33.55	50.03	35.91
FCE test	28.37	26.01	27.86
CoNLL test	35.60	36.50	35.78
JFLEG test	36.69	25.69	33.79

Table 11. ERRANT scores of the septilingual modular NMT model (Purason, 2021).

Evaluation set	Input	NMT
JFLEG test (EN)	40.51	49.80
UT test (ET)	27.14	39.56

Table 12. GLEU scores of the septilingual modular NMT model (Purason, 2021).

FCE, CoNLL and JFLEG test sets, and the septilingual model is better on each. For example, on the FCE test set, their $F_{0.5}$ score is 22.29, and the Purason (2021) model's score is 27.86. The GLEU score for the UT test set is also higher, septilingual NMT model achieves 39.56, Luhtaru (2020) highest score is 38.14. The overall behaviour is similar to Korotkova et al. (2019) and Luhtaru (2020). Compared to the state-of-the-art unsupervised approaches, the precision is low and recall high.

We will not discuss this model's error category and language level analysis because we use the noisy NMT model for training with the GEC data. However, we still found it necessary to report the scores for comparing the system with previous works.

5.2.2 Noisy Multilingual Neural Machine Translation

Since Luhtaru (2020) showed adding synthetic errors to the input can help the performance, we decided to use that method. Training a model with the same architecture and data quantity as Purason (2021) with different data is expensive and time-consuming, which is why we did not train the noisy multilingual model from scratch. Instead, we initialised the weights from the big septilingual model discussed in the last section.

Luhtaru (2020) showed that adding noise increases the recall of the model, probably due to the increased ability to correct spelling errors, but the model's precision remains about the same. With our approach, we get a similar result (see Table 13). The recall increases up to 10 points. The precision decreases a few points for CoNLL and JFLEG test, increases for FCE, and stays the same for W&I+LOCNESS sets. Estonian GLEU score drops just like Luhtaru (2020) noted (see Table 14). We keep the better performance from the larger model but get a similar improvement with added noise. That means we do not have to train noisy models from scratch in the future. Instead, we can use larger NMT models trained for other purposes for initialising the weights of noisy models.

	Noisy NMT			Increase from noisy data		
Evaluation set	P	R	$F_{0.5}$	P↑	R↑	$F_{0.5}$ ↑
W&I+LOCNESS dev	24.19	31.87	25.42	+0.15	+2.71	+0.5
W&I+LOCNESS test	33.67	54.11	36.42	-0.12	+4.08	+0.51
FCE test	29.38	35.41	30.42	+1.01	+9.40	+2.56
CoNLL test	32.9	42.68	34.4	-2.7	+6.18	-1.38
JFLEG test	35.06	33.25	34.68	-1.63	+7.56	+0.86

Table 13. ERRANT scores (precision, recall and $F_{0.5}$-score) of the noisy NMT model and comparison with the NMT model not trained on noisy data.

Evaluation set	Hypothesis	Increase from noisy data
JFLEG test (EN)	53.58	+3.78
UT test (ET)	38.87	-0.69

Table 14. GLEU scores of the noisy NMT model and comparison with the NMT model.

Based on error categories (see Figure 5), the model is very good at correcting subject-verb agreement (precision 72% and recall 80%), punctuation (67% and 65%) and orthography (67% and 60%) errors. Additionally, recall is high among verb form, spelling, noun number and morphology categories, where precision is not remarkably high. Based on that, we can say the model finds and corrects complex errors, which require sentence context and are not found with a simple spellchecker, but also corrects spelling and orthography errors.

The downside of this approach is that precision is very low in multiple categories. The model is the worst at correcting noun (precision 8% and recall 34%) and verb (16% and 34%) choice errors and a category other (11% and 23%), which contains errors not fitting into the other categories. In addition, the precision is especially low among contractions, adjectives and adverbs, where the recall is reasonable. Low precision in these general word choice categories illustrates that the model is rephrasing and changing the text. This issue has also been pointed out by Luhtaru (2020) and Korotkova et al. (2019). For example, when we think about the contraction errors, the model has not learned to keep the original vocabulary, so it can change "do not" to "don't" and vice versa depending on the text it has seen during training.

For different language levels, as we can see from Table 15, the recall increases as the language levels advance, but precision decreases. The improvement in recall could be expalained by the model's ability to correct spelling and complex grammatical errors that more advanced speakers make. The drop in recall could be caused by the model adding many unnecessary changes. We can assume more advanced speakers make fewer mistakes, and their text needs less correcting than beginners' sentences, but the model still changes the wording, thereby adding unnecessary edits.

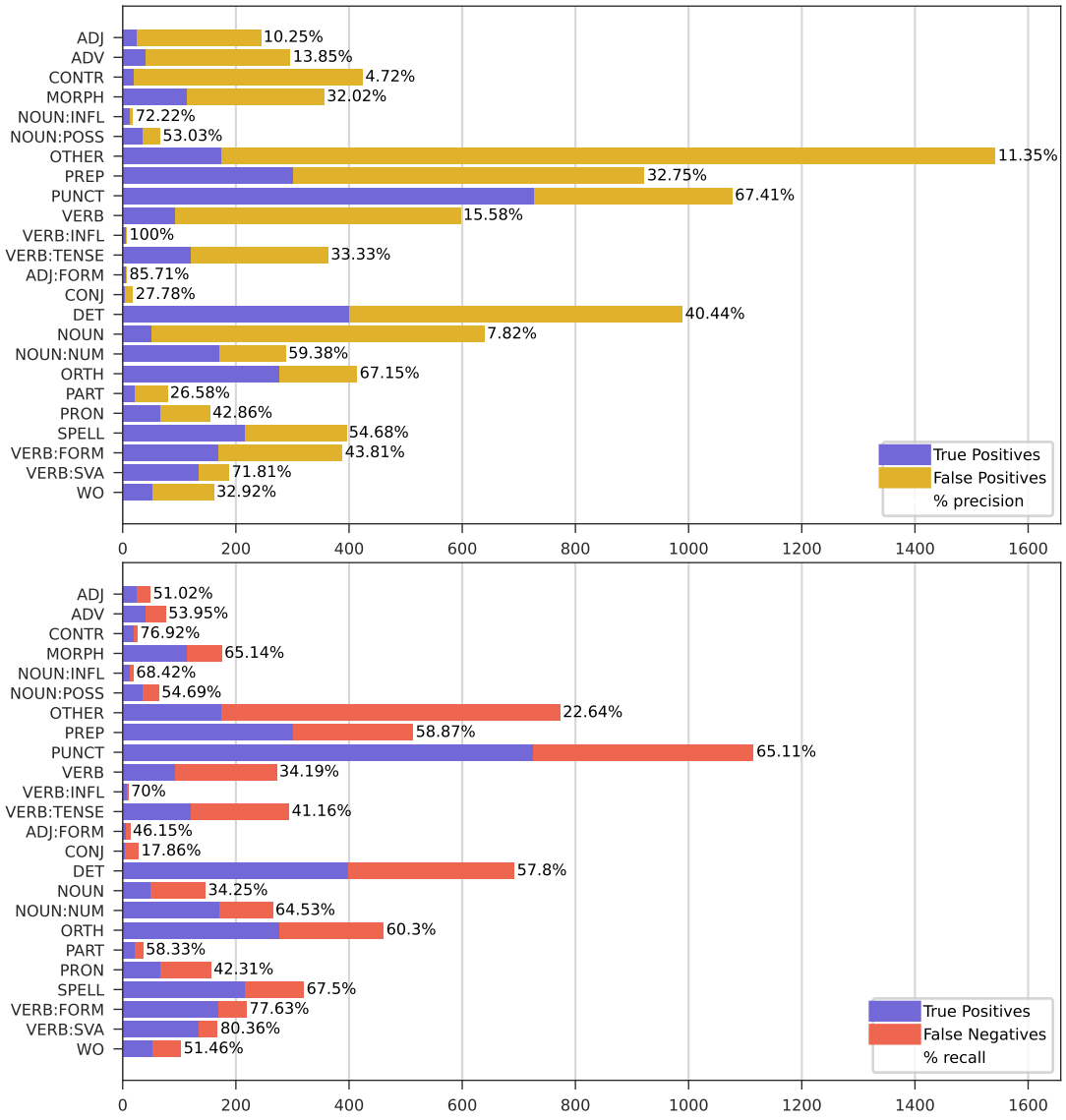

Figure 5. Noisy multilingual NMT true positives, false negatives, false positives, recall and precision per error category for W&I+LOCNESS test set.

Level	P	R	$F_{0.5}$
A	41.99	48.17	43.10
B	36.24	51.07	38.47
C	29.08	63.62	32.62
Native	24.47	68.52	28.08

Table 15. ERRANT scores of the noisy NMT model for different language levels, evaluated on the W&I+LOCNESS test set.

5.2.3 With Added Monolingual Direction

In this section, we discuss how continuing to train the noisy NMT model with GEC corpus changes the performance.[9] The original model heavily struggles with precision but shows remarkable recall in many categories.

We can say that GEC data is beneficial for the model's precision (see Table 16). For the W&I+LOCNESS test set, the precision increases by over 20 points, and we can also see significant improvement for every other test set. At the same time, recall raises when we evaluate scores on W&I+LOCNESS sets but drops slightly on other corpora. We trained the model using a subset of the W&I+LOCNESS training set. We can assume it contains similar errors as the development and test set, which means the model can improve its correcting skills based on errors relevant to that corpus.

From Table 17, we can also see that the Estonian UT test set's GLEU improves significantly, but the English JFLEG test set's score does not rise as much. Based on JFLEG ERRANT scores, we can assume the slight improvement is due to fewer incorrect edits. The Estonian test set's GLEU score increases over four times more than JFLEG's. We also trained the model using UT's training set. The significant improvement of the GLEU score suggests that continuing to train with GEC data works similarly for the two languages, which means it is likely that training on GEC data reduces the number of incorrect edits for the Estonian model as it did for English.

As we can see from Figure 6, the precision increases in every category except for punctuation and orthography and recall drops. The most significant increase in precision is for contraction (5% $\rightarrow$ 53%), determiner (40% $\rightarrow$ 62%) and verb tense (33% $\rightarrow$ 51%) errors. Compared to other categories, the precision of the two last ones was initially already moderately high. On the other hand, the contraction category had the lowest precision before, but after training using GEC examples, the model corrects these errors relatively well. Since the original model has many false positives in that category, we suggested that zero-shot translation mixes shortened and uncontracted forms (like "isn't" versus "is not") without taking the form used in the original sentence into consideration. Based on the improvement in precision, we can see that this problem

[9] We also fine-tuned the base model trained by Purason (2021), but it still struggled with correcting spelling errors.

	Trained with GEC data			Increase from noisy NMT		
Evaluation set	P	R	$F_{0.5}$	P↑	R↑	$F_{0.5}$ ↑
W&I+LOCNESS dev	42.5	35.41	40.87	+18.31	+3.54	+15.45
W&I+LOCNESS test	54.19	55.80	54.50	+20.52	+1.69	+18.08
FCE test	46.13	34.36	43.17	+16.75	-1.05	+12.75
CoNLL test	46.56	40.01	45.08	+13.66	-2.67	+10.6
JFLEG test	44.78	32.81	41.74	+9.72	-0.44	+7.06

Table 16. ERRANT scores (precision, recall and $F_{0.5}$-score) for the noisy NMT model continued to train with the GEC data and comparison with the noisy NMT.

Evaluation set	Trained with GEC data	Increase from noisy NMT
JFLEG test (EN)	57.1	+3.52
UT test (ET)	54.22	+15.35

Table 17. GLEU scores for the noisy NMT model continued to train with the GEC data and comparison with the noisy NMT.

has reduced significantly. We can suggest that because the usage of contractions in the English language is straightforward, the model learns to keep the form used in the input sentence without needing many examples.

Before training with GEC data, the precision was also very low for the noun, verb, adjective and adverb errors and mistakes labelled into category other. The precision of all these categories improves but not as impressively as we saw for contraction errors. The precision of category other rises the most (11% $\rightarrow$ 27%), but overall we see precision increase by around 10 points and a drop in recall. These categories also remain the most problematic. Precision stays below 30%, and the recall is not higher than 50%. Based on this, we can conclude that training on the GEC data helps reduce unnecessary edits, but it does not entirely solve the problem. Using more data might help solve the issue since the model learns not to mix contraction forms as much as before, but fully keeping the wording and structure of the original text may be more complex task.

The model's recall meaningfully increases only in punctuation (65% $\rightarrow$ 73%) and orthography (60% $\rightarrow$ 67%) categories. Based on the number of true positives and false negatives, these are also more frequent categories in the test set, so we can suggest that the training set also contains more examples in these categories, which could explain the most significant change. These categories were also among the categories with the highest recall before. All error types that the original model corrected well, like subject-verb agreement, verb form, spelling noun number and morphology, are still the categories the model corrects the best. Most of these types gained precision and lost recall during training with the GEC data.

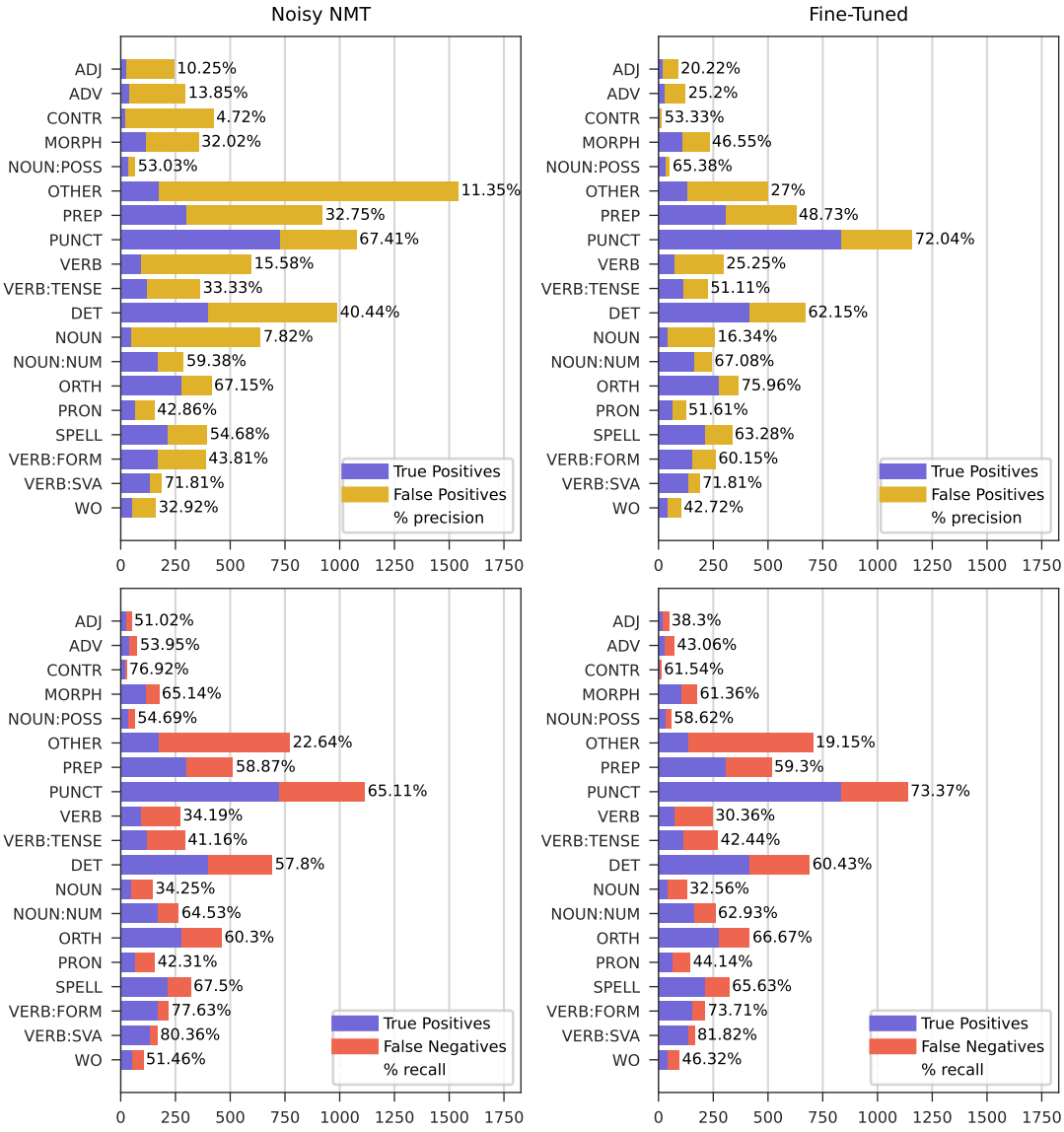

Figure 6. True positives, false negatives, false positives, recall and precision per error category for the noisy NMT model and the noisy NMT model continued to train on the GEC data, evaluated on the W&I+LOCNESS test set.

	Trained with GEC data			Increase from noisy NMT		
Level	P	R	$F_{0.5}$	P↑	R↑	$F_{0.5}$ ↑
A	60.67	50.83	58.41	+18.68	+2.66	+15.31
B	59.05	54.18	58.01	+22.81	+3.11	+19.54
C	53.96	62.09	55.41	+24.88	-1.53	+22.79
Native	38.53	68.81	42.25	+14.06	+0.29	+14.17

Table 18. ERRANT scores for different language levels for the noisy NMT model continued to train on GEC data and comparison with the noisy NMT model, evaluated on the W&I+LOCNESS test set.

When we look at the language levels (see Table 18), we can see that the recall improves mainly for beginner levels A and B, suggesting that these sentences may contain more punctuation and orthography errors. These were the categories that increased the most. The precision raised the most for levels B and C. It is difficult to draw any conclusions from the language level scores.

6 Comparison and Discussion

The previous section introduced the results for both approaches with and without including grammatical error correction (GEC) data. This section compares the models as unsupervised GEC methods and models we can use for continuing training. We compare the primary scores we trust for evaluation and compare precision and recall per error category. We also discuss how we can answer the questions stated in the introduction.

6.1 Without Annotated Data

At first, both of these approaches use no error correction data and need no knowledge about the language. Synthetic data generation uses the frequencies in which language users make mistakes based on error corpora. However, the importance of using corpus specific numbers for synthetic data generation has not been researched or reported very thoroughly yet. The monolingual zero-shot translation is entirely independent of all GEC datasets.

From Table 19, we can see the English model pre-trained with artificial data achieves around two times better precision, but the noisy NMT model's recall is about twice as high as the synthetic model's recall. The $F_{0.5}$ score that combines both recall and precision is higher for synthetic pre-train. The noisy NMT model's recall is superior in every category except spelling, but its precision is only higher for punctuation, orthography and word order errors (see Figure 7).

And yet, the Estonian test set's GLEU is higher for noisy NMT (see Table 19). This could be explained by the fact that because GLEU adds a penalty to the unchanged parts, it might correlate more with recall than precision. Another reason might be that Estonian is more challenging to correct via simple synthetic errors. The Estonian language is more morphologically complex than English and it could potentially contain more mistakes with cases that require further sentence context for finding errors. This means that adapting other evaluation methods like ERRANT or performing systematic qualitative analysis is crucial for researching Estonian GEC further.

Evaluation set	Synthetic pre-train			Noisy NMT		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
W&I+LOCNESS test (EN)	59.44	26.87	47.8	33.67	54.11	36.42
	GLEU			GLEU		
UT test (ET)	32.16			38.87		

Table 19. The ERRANT scores on the W&I+LOCNESS test set and GLEU score on the UT test set for the model pre-trained on synthetic data and the noisy NMT model.

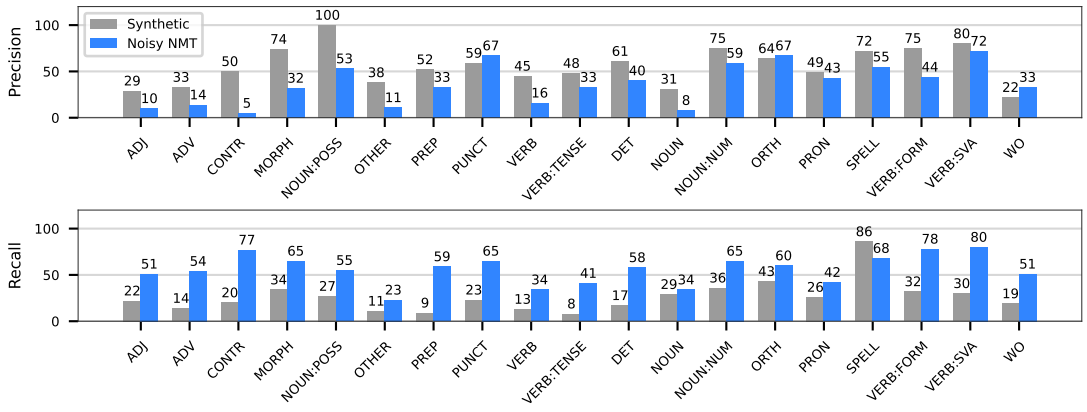

Figure 7. The error category analyses for the model pre-trained on synthetic data and the noisy NMT model, evaluated on the W&I+LOCNESS test set.

The contrast in models' behaviour comes from the fundamental differences. The model that learns error correction using synthetic data sees examples where the input and output text are very similar. There are edits in the synthetic data to encourage the model not to train to copy the input sentence but to also search and correct the errors it sees. It has learned to keep the wording and to fix the errors, but since the errors in synthetic data are primitive and not context-aware, the model cannot correct complex grammatical errors. The multilingual NMT model, on the other hand, learns to encode the text in one language and decode it in another. It trains to generate grammatically correct text in specified language based on encoded representations. The input and output sentences have the same meaning during the training process, but the wording and vocabulary used are different. Due to the nature of this approach, the model has no way of learning that changing the correct parts of an input sentence is not the desired behaviour.

In the introduction, we set a goal to determine the best approach. However, the experimental results have proved that it is difficult to determine the leading system. We get higher final scores for English with the synthetic data approach because the GEC research community has accepted using the $F_{0.5}$ score instead of F1, so precision is generally preferred. When the recall is as low as 27%, and there is a considerable gap between the ability to correct spelling and other errors, one might argue whether a grammar checker, which fixes so few mistakes, is more beneficial than simple spellcheckers.

It raises again the dilemma if we prefer very low precision or poor recall. With the synthetic data approach, we get fewer incorrect edit suggestions but have to accept that our model fails to correct most errors, except for spelling. On the contrary, with the NMT model, we get many edits, including relevant ones about complex grammatical errors, but it is much more eager to change our wording and propose incorrect suggestions.

When the grammar checker is in use without the user checking its edits or the user is not an advanced language speaker, preferring precision seems reasonable. When the grammar checker proposes edits live for the user who can assess if the suggestions are relevant but wishes to get more variety or improve fluency, preferring the model with higher recall and very low precision seems to be the better option. Both models have significant disadvantages, making real-life use tricky without further improvements.

6.2 With Limited Amount of Annotated Data

We trained both models with the GEC data similarly. The synthetic pre-train model continues learning error correction with one monolingual translation direction. The multilingual NMT model stops training on translation data and starts seeing monolingual examples. It is the first time the model sees examples between the same language. For example, the first time English encoder and English decoder are used together in training settings is during training with GEC data.

As we can see, continuing training the synthetic pre-train model increased the recall, and the precision rose for the noisy NMT model (see Table 20). After using the GEC data, there is not as significant gap between models' performances as before. The precision of the synthetic pre-train model is around seven points higher, and the recall of noisy NMT is about five points higher. It is significantly less than the two times difference before fine-tuning. The Estonian GLEU score is still better for noisy NMT, but the difference has decreased.

Regarding different error types, the precision is still higher in most of the categories for synthetic pre-train and recall better for noisy NMT (see Figure 8). Before training with GEC data, the noisy NMT model's precision was higher only for orthography and word order errors. Determiner and verb tense errors now also have slightly higher precision. Before seeing GEC examples, synthetic pre-train model's recall was only higher for spelling errors; now, it is also better for contraction errors. Overall, the differences between the ability to correct mistakes for different types have reduced, and the models behave somewhat similarly. However, in some categories, like morphology, the precision of synthetic pre-train is still significantly lower.

Evaluation set	Synthetic fine-tuned			Noisy NMT + GEC		
	P	R	$F_{0.5}$	P	R	$F_{0.5}$
W&I+LOCNESS test (EN)	61.25	51.26	58.96	54.19	55.80	54.50
	GLEU			GLEU		
UT test (ET)	51.38			54.22		

Table 20. The ERRANT scores on the W&I+LOCNESS test set and GLEU score on the UT test set for the fine-tuned synthetic model and the noisy NMT model continued to train on the GEC data.

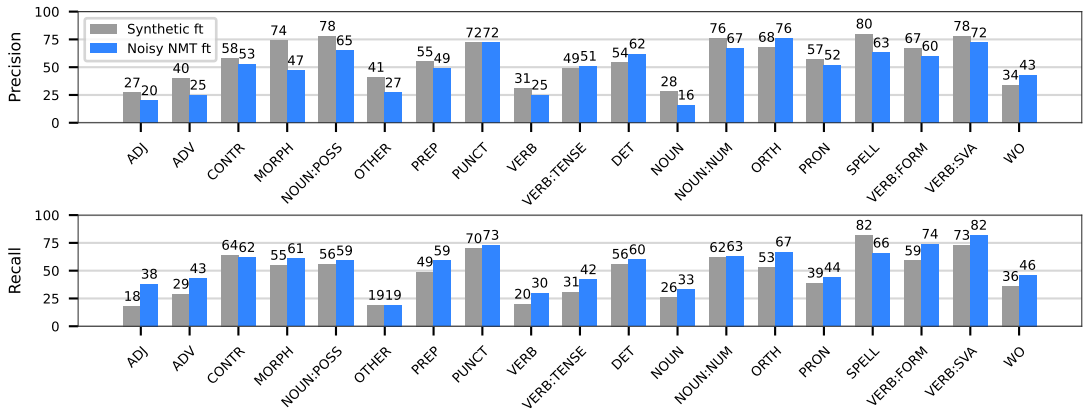

Figure 8. The error category analyses for the fine-tuned synthetic model and the noisy NMT model continued to train on the GEC data.

The synthetic pre-train model gains slightly more recall with fine-tuning than how much the noisy NMT model's precision improves. It is also easier to train using the GEC data. With synthetic pre-train, if the model is not seriously overfitting, the fine-tuning can only enhance the performance. It should not lose any performance learned during pre-training. On the other hand, training the noisy NMT model longer or with more GEC data can be tricky because we are not training the translation directions further. Hence, the model starts to forget that task, which could lead to losing corrections coming from the zero-shot effect. If we were to train translation directions further, the model would still lose the zero-shot effect to some degree. We are adding a monolingual direction, which can overpower the zero-shot translation. The model's performance would probably be better because the recall is still a major bottleneck, but further experiments are needed to confirm that.

In conclusion, when comparing the final models, it seems that at least for English, the synthetic pre-train approach appears to be more effective. Compared to the noisy NMT model, it has a higher precision, but the recall is not that much lower. For Estonian, the evaluation metrics are not clear enough to determine which model is preferrable.

6.3 Further Directions

As part of this master's thesis, we did not try all the methods suggested by Grundkiewicz et al. (2019) and other works. Creating ensembles with multiple GEC models and a language model could improve the performance. Exploring different criteria for re-ranking the models' n-best hypothesis could also improve the performance of the models. In addition, we could use noisy GEC data, like edits from Wikipedia.

The results showed that the models lacked in opposing areas - the noisy NMT model had a low precision and the synthetic pre-train model had a low recall. In the future, we could attempt to combine the models to increase the performance of error correction. We cannot apply models one after another; the precision would stay low. Hence, the naive approach of combining the models without actually merging them during output generation would be based on re-ranking method, but instead of scoring one model's n-best hypotheses, we could generate hypotheses with both models, collect these into one pile and pick the best one from the extended set. There are also more complex ways we can try to combine the models. We could find options for considering both models' predictions while autoregressively generating the output sequence.

To improve the models independently without using more GEC data, we could enhance the synthetic error generation method for the synthetic pre-train model or find ways to limit the noisy NMT model's false positives. Artificial error generation is a more explored topic for different languages. We can try adapting the proposed language-specific methods for Estonian. Working with the NMT models can be a more novel topic. Finding ways to incorporate monolingual and GEC data together with parallel translation examples without losing the zero-shot effect could enhance the model's performance.

7 Conclusion

In this master's thesis, we researched two approaches for low-resource grammatical error correction (GEC) in two languages. As one method, we pre-trained models using synthetic data and fine-tuned the models with error correction examples. The second method uses multilingual neural machine translation (NMT) and corrects errors via monolingual zero-shot translation. We also continued training the NMT model with GEC data by adding a new monolingual translation direction.

We stated three main research questions in the introduction. Firstly, we wanted to see how the performance of the model pre-trained using synthetic data differs from correcting errors with the monolingual zero-shot translation. We found out that the synthetic model corrects spelling errors well and proposes some accurate predictions for other categories too. The recall of the model is relatively low, but it corrects errors quite precisely. The NMT, on the other hand, corrects a variety of different mistakes, and the recall is excellent in many error categories, however, it adds many unnecessary or incorrect changes to the text. The evaluation shows that both models have a significant disadvantage, meaning the real-life use is tricky with both, and the use case determines which model to prefer.

Secondly, we wanted to see how the GEC data changes the performance of both models. We found out that the GEC data reduces the difference between the two approaches. The synthetic model gains a lot in recall, but the NMT model's precision increases significantly. The overall tendencies stay the same, the synthetic model's precision is still higher, and the NMT model wins in recall. Continuing training improves the performance of both models, but after training with the GEC data, we should prefer the synthetic model since its recall is decent, but the precision is higher.

Finally, we wanted to see if the languages behave the same way. Based on our experiments and evaluation, we cannot give a precise answer to that question. We evaluate Estonian only on one score, making it difficult to draw any conclusions. The Estonian score is better for the NMT model, whereas the synthetic model performs better for English. The difference might be caused both by the language morphology and evaluation methods. We can say that it is possible to correct errors in Estonian with both methods, but evaluating the performance and behaviour requires further research.

In conclusion, both methods have potential and are usable for further GEC developments for low-resource languages. On their own, both approaches need to be improved. When trained with the GEC data, adding already developed methods like creating ensembles or re-ranking could enhance the performance.

References

D. Alikaniotis and V. Raheja. The unreasonable effectiveness of transformer language models in grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 127–133, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4412. URL https://aclanthology.org/W19-4412.

D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate, 2014. URL https://arxiv.org/abs/1409.0473.

L. Barrault, O. Bojar, M. R. Costa-jussà, C. Federmann, M. Fishel, Y. Graham, B. Haddow, M. Huck, P. Koehn, S. Malmasi, C. Monz, M. Müller, S. Pal, M. Post, and M. Zampieri. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5301. URL https://aclanthology.org/W19-5301.

C. Bryant and T. Briscoe. 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, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-0529. URL https://aclanthology.org/W18-0529.

C. Bryant, M. Felice, and T. 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.

C. Bryant, M. Felice, Ø. E. Andersen, and T. 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, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4406. URL https://aclanthology.org/W19-4406.

Y. J. Choe, J. Ham, K. Park, and Y. 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, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4423. URL https://aclanthology.org/W19-4423.

L. Clément, K. Gerdes, and R. Marlet. A grammar correction algorithm – deep parsing and minimal corrections for a grammar checker. 07 2009. ISBN 978-3-642-20168-4. doi: 10.1007/978-3-642-20169-1_4.

D. Dahlmeier and H. T. 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 2012. Association for Computational Linguistics. URL https://aclanthology.org/N12-1067.

J. Devlin, M.-W. Chang, K. Lee, and K. 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.

M. Felice, C. Bryant, and T. 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, Dec. 2016. The COLING 2016 Organizing Committee. URL https://aclanthology.org/C16-1079.

P. Gage. A new algorithm for data compression. *The C Users Journal archive*, 12:23–38, 1994.

J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning, 2017. URL https://arxiv.org/abs/1705.03122.

S. Granger. The computer learner corpus: a versatile new source of data for sla research. 1998.

R. Grundkiewicz and M. Junczys-Dowmunt. The wiked error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction. In A. Przepiórkowski and M. Ogrodniczuk, editors, *Advances in Natural Language Processing - 9th International Conference on NLP, PolTAL 2014, Warsaw, Poland, September 17-19, 2014. Proceedings*, volume 8686 of *Lecture Notes in Computer Science*, pages 478–490. Springer, 2014. ISBN 978-3-319-10887-2. doi: 10.1007/978-3-319-10888-9_47. URL http://dx.doi.org/10.1007/978-3-319-10888-9_47.

R. Grundkiewicz, M. Junczys-Dowmunt, and K. 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, Florence, Italy, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4427. URL https://aclanthology.org/W19-4427.

M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351, 2017. doi: 10.1162/tacl_a_00065. URL https://aclanthology.org/Q17-1024.

M. Junczys-Dowmunt, R. Grundkiewicz, S. Guha, and K. 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, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1055. URL https://aclanthology.org/N18-1055.

D. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/P07-2045.

K. Koppel and J. Kallas. Eesti keele ühendkorpus 2019, 2020.

E. Korotkova, A. Luhtaru, M. Del, K. Liin, D. Deksne, and M. Fishel. Grammatical error correction and style transfer via zero-shot monolingual translation, 2019.

T. Kudo and J. Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, Nov. 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-2012. URL https://aclanthology.org/D18-2012.

S. Laur, S. Orasmaa, D. Särg, and P. Tammo. Estnltk 1.6: Remastered estonian nlp pipeline. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 7154–7162, Marseille, France, May 2020. European Language Resources Association. URL https://www.aclweb.org/anthology/2020.lrec-1.884.

V. I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10(8), 1966. URL https://nymity.ch/sybilhunting/pdf/Levenshtein1966a.pdf.

J. Lichtarge, C. Alberti, S. Kumar, N. Shazeer, N. Parmar, and S. 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.

A. Luhtaru. Grammatical error correction via multilinual neural machine translation, 2020. URL https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=69750&year=2020.

S. Lyu, B. Son, K. Yang, and J. Bae. Revisiting modularized multilingual NMT to meet industrial demands. *CoRR*, abs/2010.09402, 2020. URL https://arxiv.org/abs/2010.09402.

T. Mizumoto, M. Komachi, M. Nagata, and Y. 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, Chiang Mai, Thailand, Nov. 2011. Asian Federation of Natural Language Processing. URL https://aclanthology.org/I11-1017.

J. Náplava and M. 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, Nov. 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5545. URL https://aclanthology.org/D19-5545.

C. Napoles, K. Sakaguchi, and J. 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, Apr. 2017. Association for Computational Linguistics. URL https://aclanthology.org/E17-2037.

H. T. Ng, S. M. Wu, T. Briscoe, C. Hadiwinoto, R. H. Susanto, and C. 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, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/W14-1701. URL https://aclanthology.org/W14-1701.

K. Omelianchuk, V. Atrasevych, A. Chernodub, and O. 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.

M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

K. Papineni, S. Roukos, T. Ward, and W.-J. 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.

T. Purason. Modular septilingual neural machine translation, 2021. URL https://comserv.cs.ut.ee/ati_thesis/datasheet.php?id=72049&year=2021.

S. Rothe, J. Mallinson, E. Malmi, S. Krause, and A. Severyn. 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, Aug. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.89. URL https://aclanthology.org/2021.acl-short.89.

I. Rummo and K. Praakli. TÜ eesti keele (võõrkeelena) osakonna õppijakeele tekstikorpus [the language learner's corpus of the department of estonian language of the university of tartu]. In *EAAL 2017: 16th annual conference Language as an ecosystem, 20-21 April 2017, Tallinn, Estonia: abstracts, 2017, p. 12-13*, 2017.

R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, Aug. 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1162. URL https://aclanthology.org/P16-1162.

F. Stahlberg and S. Kumar. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.418. URL https://aclanthology.org/2020.emnlp-main.418.

F. Stahlberg and S. Kumar. Synthetic data generation for grammatical error correction with tagged corruption models. In *Proceedings of the 16th Workshop on Innovative*

Use of NLP for Building Educational Applications, pages 37–47, Online, Apr. 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.bea-1.4.

T. Tajiri, M. Komachi, and Y. 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://aclanthology.org/P12-2039.

Y. Takahashi, S. Katsumata, and M. 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.

University of Tartu. Ut rocket, 2018. URL https://share.neic.no/#/marketplace-public-offering/c8107e145e0d41f7a016b72825072287/.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

S. Xu, J. Zhang, J. Chen, and L. 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, Aug. 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-4415. URL https://aclanthology.org/W19-4415.

L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel. mt5: A massively multilingual pre-trained text-to-text transformer. *CoRR*, abs/2010.11934, 2020. URL https://arxiv.org/abs/2010.11934.

H. Yannakoudakis, T. Briscoe, and B. 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, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL https://aclanthology.org/P11-1019.

H. Yannakoudakis, Ø. E. Andersen, A. Geranpayeh, T. Briscoe, and D. Nicholls. Developing an automated writing placement system for esl learners. *Applied Measurement in Education*, 31:251 – 267, 2018.

Z. Yuan and M. 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, Aug. 2013. Association for Computational Linguistics. URL https://aclanthology.org/W13-3607.

Z. Yuan, S. Taslimipoor, C. Davis, and C. Bryant. Multi-class grammatical error detection for correction: A tale of two systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8722–8736, Online and Punta Cana, Dominican Republic, Nov. 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.687. URL https://aclanthology.org/2021.emnlp-main.687.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Agnes Luhtaru**,
*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   TITLE,
   *(*title of thesis)

   supervised by Mark Fišel.
   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Agnes Luhtaru
17/05/2022