Swedish Grammar Error Correction Using Neural Networks and Data Supplementation Methods

Jae Eun Hong
Department of Linguistics and Philology
Uppsala University

jaeeun.hong.7225@student.uu.se

Abstract

Grammar Error Correction (GEC) refers to an automatic process of correcting grammatical errors in given sentences. GEC is being actively studied as a sister field of machine translation(MT), but there exists a fatal limitation, namely the paucity of the parallel dataset for most of the languages including Swedish. This paper, therefore, presents a neural baseline for Swedish GEC, using two neural models- bi-directional Gated Recurrent Unit (GRU) and the pre-trained T5 model, which are proven to be effective on text-to-text setting. We further experimented with two data supplementation methods to bolster the small-scale Swedish error-annotated dataset. Our baseline consists of a bi-directional GRU model trained on the seed corpus ($F_{0.5}$ score = 40.37).The GRU model on the train data combined with synthetic data with a proportion of 1:10 achieved the best result ($F_{0.5}$ score = **72.95**).

1 Introduction

Grammar Error Correction (GEC) aims for automatically revising grammatical errors in given sentences. Errors include from spelling errors to the violation of linguistic rules, such as morphological/syntactic level errors. Recently, GEC is being actively studied along with neural machine translation (NMT), one of the NLP fields that has made great strides through an application of neural sequence-to-sequence (seq2seq) models (Cho et al., 2014; Sutskever et al., 2014). Following the framework of NMT, GEC regards the erroneous sentence as a source sentence and the grammatically correct sentence as a target sentence. The source-target pair sentences are fed into the encoder-decoder architecture..

However, there are two fatal challenges in GEC with seq2seq architecture - the scarcity of parallel corpus (erroneous-clean sentence pairs) and skewness of data towards few languages. Neural seq2seq models, in general, function properly with a large training dataset to sufficiently train millions of parameters, which is a rare scenario on GEC task, especially for languages other than English. One of the largest datasets for GEC is the NAIST Lang-8 Learner Corpora (Lang-8 corpus)[1], which contains 80 languages. The distribution of the dataset size, however, is highly lopsided towards high resource languages, such as English and Japanese, while most of the other languages hardly reach 1000 sentence pairs each. In order to overcome the challenges on neural GEC models, recent studies deliberate on various data supplementation methods, mainly generating artificial errors to a large monolingual corpus (Xie et al., 2018; Lichtarge et al., 2018; Zhao et al., 2019).

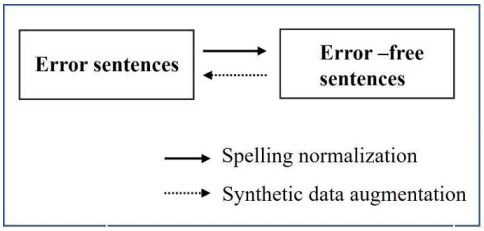

Figure 1: The flow of data supplementation methods

In this study, we choose Swedish to explore neural GEC and two data supplementation methods- spelling normalization and synthetic data augmentation using pre-defined rules. The flow of our supplementation techniques are shown in Figure 1. To promote diverse follow-up studies related to Swedish GEC, this paper suggests a baseline model of Swedish GEC using a bi-directional Gated Recurrent Unit (GRU) and a pre-trained T5.

In the data supplementation procedures, we use SWEGRAM(Näsman et al., 2017), the linguistic

[1]https://sites.google.com/site/naistlang8corpora/

annotation tool for Swedish to normalize spelling on erroneous sentences. For the synthetic data augmentation, we create pre-defined rules to generate character, morphological, and word-level errors.

Experiments demonstrate that neural GEC for Swedish with the proposed data supplementation methods outperform the baseline model which only uses the original dataset.

In short, this paper aims to make the following contributions:

- We suggest a neural baseline for Swedish GEC using SweLL corpora. To the best of our knowledge, it is the first work to experiment on neural GEC using SweLL corpora, which contain various Swedish learner's essay texts.
- We implement two simple yet effective data supplementation methods to strengthen our total training data, which enhance the performance of seq2seq-based GEC model.

2 Related Work

This section describes prior studies/tasks related to GEC with neural networks as well as data augmentation methods.

2.1 Neural GEC

The importance of GEC started to receive considerable attention when The CoNLL-2014 shared task was introduced by Ng et al. (2014). The task was devoted to the correction of grammatical errors of all types using English essay sentences tagged with error annotations. When a big leap has been made in the development of deep learning, the field of MT also made great progress with the emergence of encoder-decoder based recurrent neural network architecture (seq2seq) (Cho et al., 2014; Sutskever et al., 2014).As the neural seq2seq models contributed greatly to the performance of machine translation, recent GEC studies began to view the task from the perspective of NMT, started by Yuan and Briscoe (2016).

As attention mechanism (Bahdanau et al., 2016) proved its high performance by selectively focusing on the most relevant context in the translation task, Transformer, the novel architecture based on attention was introduced (Vaswani et al., 2017). The recent studies of GEC are applying Transformer (Lichtarge et al., 2018; Zhao et al., 2019), instead of the traditional statistical machine translation (SMT) or recurrent neural network (RNN)-based models. Zhao et al. (2019) further applied copying mechanism (Gulcehre et al., 2016) on GEC and proved its high performance. Influenced by the upward trend to use pre-trained models in the field of NLP such as BERT (Devlin et al., 2019), GEC also applied them and achieved promising results (Kaneko et al., 2020).

2.2 Data Supplementation in GEC

Application of state-of-the-art NMT mechanism in the GEC task falls few years behind than the field of MT (Junczys-Dowmunt et al., 2018). The most problematic factor in the field of GEC is the shortage of data on encoder-decoder based GEC model except for high resource languages, such as English. To address the challenge, back-translation (Sennrich et al., 2016), one of the frequently used synthetic data augmentation methods have also been used on GEC task (Xie et al., 2018) and reported high performance. Application of back translation, however, requires a large amount of data for model to successfully learn the errors in reverse, which again encounters GEC's fundamental problem- the scarcity of data.

Various data augmentation techniques are then widely studied with heuristics, which includes word/character insertion, deletion, or transposition (Lichtarge et al., 2019; Zhao et al., 2019; Xu et al., 2019; Choe et al., 2019; Wan et al., 2020). Choe et al. (2019), particularly uses a priori knowledge to generate grammatical errors based on token's part-of-speech types. Inspired by Choe et al. (2019); Wan et al. (2020), we make predefined rules based on character, morphological, and word-level errors which are frequently generated errors of Swedish L2 learners. We refer to the statistics of error types listed in the SweLL correction annotation guidelines [2]. Compared to Choe et al. (2019), we also explore the spelling normalization technique using SWEGRAM(Näsman et al., 2017) to generate grammatically correct sentences from erroneous sentences.

3 Data Supplementation Methods

This section describes two data supplementation methods applied in our experiments.

[2]SweLL correction annotation guidelines can be found here : http://hdl.handle.net/2077/69434

In this study, we compare the two methods to supplement the small-scale training data. First, we extract sentences from Swedish learner essays which are normalized through SWEGRAM (Näsman et al., 2017). Second, we generate synthetic data using pre-defined rules with heuristics using a large corpus that contains error-free sentences. The comparison between two data supplement methods are presented in the Section 6.

3.1 Spelling Normalization

The data used in this setting are three SweLL-pilot sub-corpora containing L2 Swedish learner essays(Volodina et al., 2016a). SWEGRAM, the normalization tool, is a free web-based tool for the linguistic analysis of Swedish texts contributed by Uppsala University (Näsman et al., 2017). The tool allows a user to use a pipeline from tokenization, spell checking and part-of-speech tagging to syntactic annotation. We used a basic spell checking function to normalize learners' sentences that contain lexical-level errors as shown in Example (1) below.

(1) Spelling normalization

a. ***peretar*** *med* → ***pratar*** *med*

b. *läser* ***sveniska*** → *läser* ***svenska***

In the training procedure, the original sentences written by students are fed into the neural models as a source text and the normalized sentences as a target text.

3.2 Synthetic Data Augmentation

To generate realistic grammatical errors that L2 Swedish learners would make, we propose four pre-defined rules - insertion, deletion, transposition, and lemmatization. To maximize the randomness of error generation, our proposed augmentation function randomly applies four rules with a probability of 0.05 respectively. The rules are further categorized into character-level, morphological-level, and word-level errors. Four rules are described in detail below.

Insertion Randomly insert a character inside a word with a probability of 0.05 as shown in the Example (2).

(2) *fort**s**ätta*→*fort**ss**ätta*.

Deletion Randomly delete a character inside a word with a probability of 0.05, e.g. *EU-länderna*→ *EU-läderna*. To generate more realistic errors, we added a rule for consonant doubling, such as *-ll-, -ss-, -mm-, or -nn-*. When the randomly chosen word contains the consonant doubling characters, delete one of the consonant. The example of consonant doubling deletion is listed in Example (3) below.

(3) *mä**nn**iskor*→*mä**n**iskor*

Additionally, we applied the morphological-level errors related to Swedish definite gender suffixes *-en/-et* and one of the Swedish plural suffixes *-na*. The example of definite gender suffix deletion is listed in Example (4) below.

(4) *sjukhus**et***→*sjukhus*

transposition Randomly substitute a character with its neighbouring character inside a word, with a probability of 0.05, as shown in the Example (5) below.

(5) *Polis**e**rna*→*Poli**e**srna*

To make commonly made errors, we also added a rule for the characters with the similar sounds, such as *å - o*, *e - i* or *k - c*. If the randomly chosen word contains *å, o, e, i, k* or *c*, the character is changed into its corresponding counterpart as shown in Example (6) below. This rule is ignored when the initiating character is capital letter to prevent changing the proper nouns.

(6) ***k**affe*→***c**affe*

In addition, we also applied both morphological and word-level transposition. For the morphological-level errors, we apply a transposition rule when the randomly chosen word contains definite suffixes such as *-en/ -et* or plural suffixes such as *-er/-ar*, to be substituted to its counterpart as shown in Example (7) and (8).

(7) *parti**et***→*parti**en***

(8) *häst**ar***→*häst**er***

For the word-level errors, we select pronoun *de-dem* and several auxiliary verbs that Swedish learners frequently makes an error, e.g. *får - måste, ville - ska, and vill - kommer att* to store in a list. When the randomly chosen word is in the list, it is substituted to its counterpart as shown in Example (9). This rule is inspired from the description of the wrong word or phrase (L-W) errors

on the SweLL correction annotation guide.

(9) *Vi **måste** stoppa det.→Vi **får** stoppa* det.

Lemmatization Lemmatize a word using the SweLL-gold corpus that contains linguistic information including each token's lemma form with a probability of 0.05 as shown in Example (10).

(10) *De **höga priserna**→De **hög pris** på hus*

4 GEC Model

In this study, we use neural GEC models using bi-directional GRU, the RNN-based neural network as a main model. Although the transformer-based model has achieved the state-of-the-art performance on various text generation tasks, we choose GRU to be the main neural network model to test our data supplementation methods due to insufficient training data.

Baseline: A bi-directional GRU. Only the seed corpus (SweLL-gold corpus) is used for the baseline.

We use a bidirectional GRU in both encoder and decoder to preserve the source context information more thoroughly. To minimize the information loss from long sequences, additive attention mechanism from Bahdanau et al. (2016) is applied. The decoder generates the following word based on the probability using the current output token, weighted encoder outputs, and previous hidden state of the decoder. To enhance the performance, packed padded sequences and masking are also applied to force our model to focus on actual word tokens other than padding tokens.

We further experiment on the Transformer-based T5 model(Raffel et al., 2020), due to its renowned achievement on various text generation tasks. T5 model consists of both encoder and decoder, pre-trained on multi-task mixture which outputs a result as a text-to-text format. In this experiment, we chose T5-base as the size of the pre-trained model. The library used for the experiment is described in the next section.

5 Experimental Setup

5.1 Data

Gold-standard Dataset The gold-standard dataset for the model training/evaluation are based on the SweLL-gold corpus (Volodina et al., 2019) provided by the project from SweLL-Infrastructure for L2 Swedish. It contains 502 essay texts written by adult learners of Swedish. Table 1 shows the number of sentences for both original and normalized sentences.

Type	Sentences	Tokens
Original	7,807	147,842 incl.punct.
Normalized	8,137	151,851 Incl.punct.

Table 1: Number of sentences/tokens in SweLL-gold corpus

SweLL correction annotation guidelines note the original sentences were normalized manually by correcting erroneous and deviant language, following the norms of standard Swedish.

In this study, the original version of the sentences is regarded as the source text, and the normalized sentences as the target text. We first pair the source and normalized sentences to the size of 8,137. Due to the size mismatch between original and normalized sentences, we compare the number of matched strings on each source sentences based on the target sentences and choose a source sentence that contains the most matched characters for each target sentences. The pairs consist of erroneous-clean text pairs and the identical pairs which do not contain any erroneous sentences. We delete the identical pairs and split the data into training, development, and test dataset. Both development data and test data are split from the SweLL-gold corpus in proportion of 70%, 10%, 20%, respectively. The sentence sizes for training/development/test dataset are listed in Table 2 below.

Training	Development	Test
4,451	636	1,272

Table 2: Baseline training, development, test set size

Datasets for Data Normalization For the data normalization, collecting corpus that contains erroneous sentences is essential. Considering this, we gathered more data from SpIn, Sw1203, and

TISUS, the three subcorpora of the SweLL-pilot corpus (Volodina et al., 2016b). The total number of sentences used for training is 9688, as shown in Table 3. First, the collected corpora will be adapted to Swegram (Näsman et al., 2017) for the lexical normalization. Before the training procedure, the sentence pairs that contain the length under 3 are deleted. Then, the original sentences are used as a source text and the normalized sentences as a target text. The normalized sentence pairs will be concatenated with original training data in the training process.

Corpus	Original	Final
Spln	4,247	3,455
Sw1203	3,136	3,089
TISUS	3,422	3,334
Before denoising	9,878	
After denoising	**9,688**	

Table 3: Number of sentences in SweLL-pilot collection before and after cleansing noises

Datasets for Data Augmentation For the synthetic data augmentation, we use the 8 sidor news article corpus from Språkbanken Text[3] due to its precise characteristic in terms of grammar. We randomly selected texts from the corpus with a condition that each sentence has more than three alphabetical words and stored separately in a proportion to the gold-training data to experiment on the effect of the synthetic data size on the model performance. The size of artificially generated sentences is shown in Table 4. These pseudo-sentence pairs are concatenated to the original training data and feed into the network in the training procedure.

Proportion	Size
Original	4,451
1:4	17,804
1:6	26,706
1:8	35,608
1:10	44,510
1:16	71,216

Table 4: Number of synthetic sentences used for training procedure according to the proportion to original data.

[3]https://spraakbanken.gu.se/en/resources/attasidor

5.2 Training Details

Preprocessing We first tokenize sentences using spaCy tokenizer (Honnibal and Montani, 2017) for Swedish. The texts are tokenized and fed into the model through Torchtext 0.11.0 framework .

Training The recurrent seq2seq model used in this experiment was built on Pytorch 1.10.0(Collobert et al., 2011). The Transformer model used in this paper is built on simpleT5 library[4], constructed on top of PyTorch-lightning and Huggingface Transformers.

To experiment two data supplementation methods, we, first, distinguish the training data combined with SWEGRAM normalization (SpIn, Sw1203 and TISUS) and the training data with synthetic data generated from 8 Sidor corpus in a proportion listed in Table 4. Then, we feed each dataset into the GRU model to compare the performance according to the supplement methods and the size of synthetic data.

We, further, experiment to fine-tune T5, the pre-trained model using the dataset mentioned above. More details on the hyperparameters used in both models are listed in Appendix A.

5.3 Evaluation Metric

		correction by model	
		True	False
Errors on actual sentence	True	TP	FN
	False	FP	TN

Table 5: Error matrix on GEC task

In this paper, we list precision, recall and $F_{0.5}$ score as an evaluation scale. As shown in Table 5, precision is the proportion of sentences that contain grammatical errors among sentences corrected by the model (TP/(TP+FP)). Recall is the proportion of sentences that are actually corrected among the grammatically erroneous sentences (TP/(TP+FN)), related to 'finding the errors'. $F_{0.5}$ score reflects the change in precision to a greater extent than recall, which places more importance on 'properly correcting the errors'. We put more weight on recall and $F_{0.5}$ in this experiment to investigate the model's performance on both 'finding the errors' and 'correcting the errors'. We used nltk.translate.chrfscore module[5].

[4]https://pythonrepo.com/repo/Shivanandroy-simpleT5
[5]https://www.nltk.org/api/nltk.translate.chrf_score.html

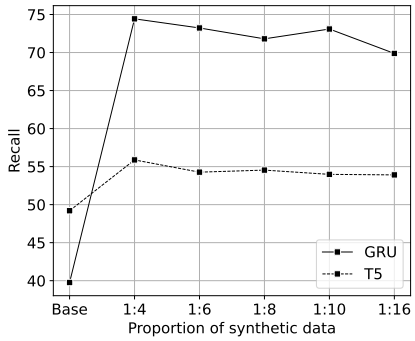

Figure 2: Recall score according to the synthetic data size

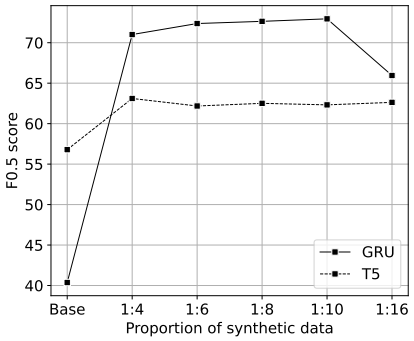

Figure 3: $F_{0.5}$ score according to the synthetic data size

6 Results

Data methods	P	R	$F_{0.5}$
Baseline	41.60	39.76	40.37
[Normalization]			
SWEGRAM	71.50	**68.69**	70.21
[Synthetic data]			
1:4	71.35	**74.43**	71.01
1:6	73.14	73.23	72.37
1:8	**73.99**	71.80	72.64
1:10	73.88	73.09	**72.95**
1:16	66.70	69.88	65.95

Table 6: Experimental results of our data supplementation methods on GRU. P refers to Precision, R refers to Recall, and $F_{0.5}$ refers to $F_{0.5}$ score.

Data methods	P	R	$F_{0.5}$
Baseline	60.12	49.20	56.79
[Normalization]			
SWEGRAM	60.09	**49.39**	56.79
[Synthetic data]			
1:4	66.66	**55.88**	**63.11**
1:6	66.23	54.27	62.18
1:8	66.63	54.54	62.51
1:10	66.64	53.98	62.32
1:16	**67.27**	53.91	62.63

Table 7: Experimental results of our data supplementation methods on T5

We evaluate the performance of our data supplementation methods on the test dataset (1,272 sentences from SweLL-gold corpus). Table 6 and 7 summarize the results on both GRU and T5 model. As shown in Table 6, our synthetic data augmentation method achieved the best recall, nearly 34.67 points higher than the baseline on GRU model with a proportion of the synthetic data size to 1:10. The model using spelling normalization methods, also showed high performance although the amount of increased training data size was only around 10,000. Table 7 shows our additional experiment on T5, which is one of the Transformer-based networks. Using only seed datatset, T5 showed higher scores in every metrics. Due to the small-scale training, however, both recall and $F_{0.5}$ score stayed steady around 52 and 62 points, respectively.

7 Discussion

7.1 Analysis on Synthetic Data Size

As shown in Table 6, our synthetic data augmentation method can improve the GEC model up to nearly 35 points. In this section, We investigate the influence of synthetic data size on the model performance based on Figure 2 and 3 which show both recall and $F_{0.5}$ score throughout the synthetic data size. Recall score in Figure 2 represents how well both models performed on finding the erroneous sentences. Figure 3, on the other hand, represents whether the models properly corrected erroneous sentences.

As shown in Figure 2, The GEC model on GRU showed the best performance on finding the error sentences with a 1:4 synthetic data proportion. However, the model achieved the best result on correcting errors properly when the synthetic data proportion is 1:10. When the synthetic data size is increased to 1:16, the model's overall performance started to show a decline.

The GEC model built on T5, on the other hand,

Gold-standard	Om du vill sola eller träna , **du borde [→ bör du]** ta **solenkräm [→ solkräm]** .	
baseline	GRU	Om du vill åka eller , , , du kan ta . .
Normalization	GRU	Om du vill sola eller träna , du du ta solenkräm .
Synthetic data	GRU (1:4)	Om du vill sola eller träna , du borde ta solenkräm .
	GRU (1:10)	Om du vill sola eller träna , du borde ta solenkräm .
	T5 (1:16)	Om du vill sola eller träna , bör du ta solkräm .

Table 8: Examples of the error corrections. The phrase on the right of the arrow on the Gold standard indicate grammatical form. **Bold** indicates errors to be fixed by GEC model. Numbers in the parenthesis, e.g., (1:4) indicates the proportion of synthetic data.

performed better with only using the seed dataset than GRU, but showed that there was no marked increase or decrease on the performance of T5 model throughout the synthetic data size.

In summary, GRU model yielded a very strong performance in both finding and correcting sentence errors in synthetic data ratios between 1:4 and 1:10. T5, on the other hand, showed lower performance than GRU due to the data-hungry characteristics of Transformer, but it showed the highest precision in the proportion 1:16. This suggests the possibility of the development of the T5 model when a large amount of realistic pseudo-parallel datasets are augmented.

7.2 Case Analysis

In this section, we use a specific example to analyze the error corrections generated by GEC models with our data supplementation methods. Table 8 shows the corrections that the models generated on the test set. We randomly chose the models for the analysis. It demonstrates that the performance of the GEC model using only seed dataset produces unsatisfactory output. The model cannot detect errors in any case and unnecessarily changes most of the words. The model with seed corpus combined with normalized dataset shows better result, but it fails to choose the correct words, e.g. the pronoun *du* is chosen instead of auxiliary verb *kan*. The GEC models with seed corpus combined with synthetic data (proportion 1:4 and 1:8) starts to take shape in the sentence, but it does not correct the erroneous word correctly, e.g. (*solenkräm* → *solkräm*).

Both of the data supplementation methods we implemented does not consist of the sentence-level errors, such as word order related errors on Table 8 (*du borde* → *borde(bör) du*). The example above represents that the model based on GRU is not sufficiently trained to fix the sentence-level errors.

Surprisingly, T5 model with synthetic data (proportion 1:16) did not achieve the high $F_0.5$ score, but it successfully corrected the errors related to word order, which manifests its flexibility on correcting various error types. The result proves a possibility that T5 model can be effective to fix more diverse errors beyond the lexical-level errors than GRU.

8 Conclusion & Future Work

In this paper, we provide a neural baseline on the SweLL-gold corpus for Swedish GEC. We also improved the baseline by using two data supplementation methods - the sentence normalization and synthetic data augmentation using pre-defined rules. Synthetic data augmentation method, in particular, proved to be beneficial to the GEC task, which will be further explored in the future work applying recent studies, such as an utilization of annotated error tags (Wan et al., 2020; Stahlberg and Kumar, 2021). The normalization method using SWEGRAM, also proved its strong possibility due to its diverse linguistic annotation tools. Future work includes implementing more realistic error generator that covers from lexical, morphological level to synthetic level, such as noising the word order that L2 Swedish learners frequently make by using the POS tag annotation through SWEGRAM. In addition, the domain of the data used for augmentation should be taken into consideration. Edunov et al. (2018) noted that synthetic data can be highly effective if the domain suits for the real-world data. Finding the corpus that is not skewed to one domain should be counted in the future experiment. Lastly, Transformer model using copying mechanism that achieved the state-of-the-art performance on the GEC task (Zhao et al., 2019) should also be explored in the future work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

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. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. 2019. A neural grammatical error correction system built on better pre-training and sequential transfer learning. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, Florence, Italy. Association for Computational Linguistics.

R. Collobert, K. Kavukcuoglu, and C. Farabet. 2011. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Marcin Junczys-Dowmunt, Kenneth Heafield, Hieu Hoang, Roman Grundkiewicz, and Anthony Aue. 2018. Marian: Cost-effective high-quality neural machine translation in C++. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 129–135, Melbourne, Australia. Association for Computational Linguistics.

Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, and Kentaro Inui. 2020. Encoder-decoder models can benefit from pre-trained masked language models in grammatical error correction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4248–4254, Online. Association for Computational Linguistics.

Jared Lichtarge, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. Corpora generation for grammatical error correction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3291–3301, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Jesper Näsman, Beáta Megyesi, and Anne Palmér. 2017. SWEGRAM – a web-based tool for automatic annotation and analysis of Swedish texts. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 132–141, Gothenburg, Sweden. Association for Computational Linguistics.

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

Shared Task, pages 1–14, Baltimore, Maryland. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Elena Volodina, Lena Granstedt, Arild Matsson, Beáta Megyesi, Ildikó Pilán, Julia Prentice [Grosse], Dan Rosén, Lisa Rudebeck, Carl-Johan Schenström, Gunlög Sundberg, and Mats Wirén. 2019. The swell language learner corpus: From design to annotation. *The Northern European Journal of Language Technology*, 6:67–104.

Elena Volodina, Ildikó Pilán, Ingegerd Enström, Lorena Llozhi, Peter Lundkvist, Gunlög Sundberg, and Monica Sandell. 2016a. Swell on the rise: Swedish learner language corpus for european reference level studies. *CoRR*, abs/1604.06583.

Elena Volodina, Ildikó Pilán, and David Alfter. 2016b. *Classification of Swedish learner essays by CEFR levels*.

Zhaohong Wan, Xiaojun Wan, and Wenguang Wang. 2020. Improving grammatical error correction with data augmentation by editing latent representation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2202–2212, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

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

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, San Diego, California. Association for Computational Linguistics.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 156–165, Minneapolis, Minnesota. Association for Computational Linguistics.

A Appendix A

GRU	
Architecture	Bi-directional GRU (1-layer)
Learning rate	0.001
Batch size	32
Optimizer	Adam ($b_1 = 0.9, b_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$)
Max epochs	10
Loss	cross-entropy
Dropout	0.3
Embedding dimension	256
Hidden dimension	512
T5	
T5 type	T5-base
Max length of encoder	50
Max length of decoder	50
Batch size	8
Max epochs	5

Table 9: Training details for each model