universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Automatic Error Correction of Bibliographic References in Academic Writing

verfasst von | submitted by

Elizaveta Ziulkova

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 587
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Joint-Masterstudium Multilingual Technologies
Betreut von | Supervisor:	Assoz. Prof. Mag. Dr. Dagmar Gromann BSc

Abstract

Existing approaches to Automatic Error Correction (AEC) have primarily focused on general grammar and spelling mistakes. However, academic domains face unique challenges, with strict requirements for grammar, sentence structure, and referencing. Due to the scarcity of suitable annotated data for referencing errors, this area remains underexplored. To address this gap, this thesis focuses on constructing a synthetic parallel referencing error corpus using a Large Language Model (LLM). Specifically, this study explores the effectiveness of two dataset creation strategies: synthetic data generation and synthetic error corruption. Additionally, this thesis investigates the development of an error type specification based on the textual Centre for Translation Studies (CTS) guidelines, which is used to guide the model in synthetic data generation and corruption pipelines. Experimental results show that a fine-tuned T5 model trained on a combination of synthetically generated and corrupted datasets achieves strong performance on real-world referencing error correction tasks, with an $F_{0.5}$ score of 72.85. However, the ability of a model to generalize to unseen error types remains limited, emphasizing the need for developing approaches for a more diverse and representative error type taxonomy. These findings contribute to the further development of robust academic error correction models and highlight future directions for research, such as exploring methods to expand error type coverage and improve the generalization of models to unseen errors.

Zusammenfassung

Bisherige Ansätze der automatischen Fehlerkorrektur (AEC) konzentrierten sich vorrangig auf allgemeine Grammatik- und Rechtschreibfehler. Akademische Fachbereiche stellen jedoch spezifische Anforderungen, insbesondere hinsichtlich Grammatik, Satzstruktur und Zitierweise. Aufgrund des Mangels an geeignet annotierten Datensätzen für Fehler im Bereich der Zitierweise ist dieses Themenfeld bislang unzureichend erforscht. Um diese Forschungslücke zu adressieren, widmet sich die vorliegende Arbeit der Generierung synthetischer Trainingsdaten zur Fehlerkorrektur bei Quellenangaben. Im Besonderen werden zwei Strategien zur Datensatzerstellung untersucht: die Erstellung synthetischer Daten sowie die gezielte Einbringung synthetischer Fehler. Die Ergebnisse verdeutlichen die jeweiligen Stärken und Schwächen beider Ansätze und zeigen, dass ein feinabgestimmtes T5-Modell mithilfe von synthetisch erzeugten und kombinierten (generierten und korrupierten) Datensätze in der Lage ist, Fehler in Quellenangaben in echten Beispielsätzen von Masterarbeiten effektiv zu korrigieren.

Contents

List of Tables

List of Figures

1. Introduction

Automatic Error Correction (AEC) is a task that involves taking an erroneous input sequence and producing a corrected output sequence. Despite decades of research (Sager, 1981; Kernighan et al., 1990; Alikaniotis and Raheja, 2019), this task remains challenging and presents significant opportunities for further research, especially in low-resource domains and languages (Náplava and Straka, 2019; Flachs et al., 2021). Existing approaches in AEC primarily focused on general grammar and spelling errors (Raheja et al., 2023; Faltings et al., 2021), while only fewer studies (Skitalinskaya et al., 2021) focused on errors occurring in specific domains, such as academic writing. Academic papers and theses, nonetheless, have a set of rules, with special requirements to grammar, sentence structure and referencing. This thesis addresses the challenge of automated revision in academic texts, primarily focusing on referencing errors based on guidelines from the Centre for Translation Studies (CTS). Referencing errors occur when publications are cited incorrectly. Due to the complexity of referencing rules, the full compliance often becomes difficult for thesis authors, which can easily lead to errors.

The biggest obstacle for the AEC task is the scarcity of aligned data needed for training the models. The models require a large amount of paired correct and incorrect sentences that are difficult to obtain. Despite the availability of corpora for general GEC tasks, such as the National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013), the 2013 Conference on Computational Natural Language Learning Shared Task (CoNLL-2013) (Kao et al., 2013), and the 2014 Conference on Computational Natural Language Learning Shared Task (CoNLL-2014) (Ng et al., 2014), these datasets are not applicable to task-specific challenges such as the correction of referencing errors. To cope with this issue, a GEC dataset was created based on the textual guidelines provided by CTS. Specifically, a table of error types was developed containing error categories and types derived from CTS guidelines. This specification table served as a source for synthetic generation and corruption approaches.

The creation of GEC datasets represents an important research direction, which comprises different strategies, such as manual corpus creation (Lee and Webster, 2012), data collection from online-collaborative platforms (Faltings et al., 2021; Grundkiewicz et al., 2019; Junczys-Dowmunt et al., 2018), back-translation (Lichtarge et al., 2019) or tagged corruption models (Stahlberg and Kumar, 2021). Additionally, LLMs open up enormous opportunities in GEC dataset creation

due to their strong generalization capabilities. However, this approach remains underexplored and has certain limitations. For example, LLMs sometimes produce too simplistic error patterns (Deng et al., 2025) or, on the contrary, tend to over-correct sentences (Li and Lan, 2025). Therefore, it is crucial to incorporate control mechanisms into the error generation pipeline.

This thesis contributed to the research on LLM performance as a GEC data source, incorporating concepts from Stahlberg and Kumar (2021), where correct sentences were corrupted with errors based on the error tag, and the controlled synthetic LLM pipeline approaches proposed by Li and Lan (2025). This thesis compares two different strategies for creating datasets with LLMs. In the first approach, both correct and erroneous sentences are generated by an LLM. In the second approach, clean sentences were collected from the real student master's theses. Then, the sentences containing referencing were filtered and further corrupted by an LLM.

This master's thesis aims to investigate the use of generic neural language models to correct the compliance of academic writing with very specific institutional guidelines. To this end, methods for generating sufficiently large and heterogeneous datasets are required as well as a real-world evaluations scenario. The two main approaches applied in this thesis are synthetic dataset creation as well as corruption of real-world examples. Finally, a real-world held-out test set of erroneous references from master's thesis completes the evaluation. Thus, the research question for this approach is: How effectively can generic neural language models fine-tuned on syntethically generated and corrupted datasets be applied to correct the compliance of real-world master's thesis examples with specific institutional guidelines?

One of the assumptions is that specific error correction patterns will be more successfully handled than others. Furthermore, the type of dataset used for fine-tuning is expected to impact the overall effectiveness measured by means of sucessfully corrected sequences. It can also be assumed that the length and complexity of sentences might impact the correction success.

While the general task of error correction has received substantial attention in research, the compliance with very specific guidelines for academic writing have not yet been investigated to the same degree. Such guidelines are at times very specific to the community or institute and not directly comparable across settings. Thus, it is interesting to evaluate the effectiveness of generic neural language models to check and potentially correct the compliance of bibliographic references in academic writing. As one major outcome of this thesis, the effectiveness of two main and state-of-the-art dataset creation strategies will be evaluated. It is interesting to investigate if synthetically created datasets can as effectively be used for fine-tuning as corrpued real-world examples. This result will be interesting to developers of GEC models as well as for academic institutions. Providing an automated and institution-specific approach for students to check the compliance

of their own writing with institutional guidelines would not only facilitate their writing process, but also relieve the burden on supervisors and revisors to point out failures to comply with institutional referencing guidelines. Furthermore, this thesis contributes to the overall practical and theoretical topic of customization of generic language models to very specific settings, providing the interesting use case of correcting non-compliant bibliographic references.

The code developed for this thesis, covering real-world sentence collection, synthetic dataset generation and corruption, and models training and evaluation, is released on GitHub[1].

[1] `https://github.com/trestoris/GEC_Referencing`

2. Deep learning theory

This chapter presents the preliminaries of Deep Learning necessary for this study. It discusses the fundamental concepts of the neural network and its core components, including the neuron, activation function, and loss function. Additionally, the chapter provides an overview of the Transformer architecture and advanced Transformer-based Sequence-to-Sequence models.

2.1. The Artificial Neuron Model

Deep learning, a field of machine learning, focuses on the learning of neural networks. A building block of a neural network is a neural unit, also known as a neuron. The structure of the basic neural unit is demonstrated in Figure 1.

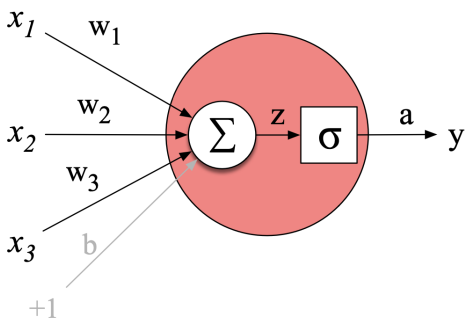

Figure 1.: A neural unit (Jurafsky and Martin, 2025, p. 134)

The concept of artificial neuron (McCulloch and Pitts, 1943) is inspired by biology, where the biological neuron receives signals as input, processes them, and transmits the output to the next neutrons. The artificial neuron receives multiple inputs (x_1, x_2, ..., x_n), each associated with its own weight (w_1, w_2, ..., w_n), in addition to a bias value b (Jurafsky and Martin, 2025, p. 134). The neuron computes the dot product of the inputs and weights and then adds the bias, as shown in Equation (1) (Jurafsky and Martin, 2025, p. 132).

$$z = \mathbf{w} \cdot \mathbf{x} + b \tag{1}$$

Additionally, the non-linear activation function f is applied to z, as shown in Equation (2).

$$y = a = f(z) \tag{2}$$

The activation function introduces the non-linearity into the neural networks by mapping the input value to a non-linear transformed value (Buduma et al., 2022, p. 12). There are several types of activation functions. One of them is the sigmoid function, as shown in Equation (3), which takes any real number and often return a value in the range of 0 to 1.

$$y = \sigma(z) = \frac{1}{1 + e^{-z}} \tag{3}$$

The characteristic of the sigmoid function is that its output asymptotically approaches 1 as z becomes very large, and asymptotically approaches 0 as z becomes very small, as shown in Figure 2.

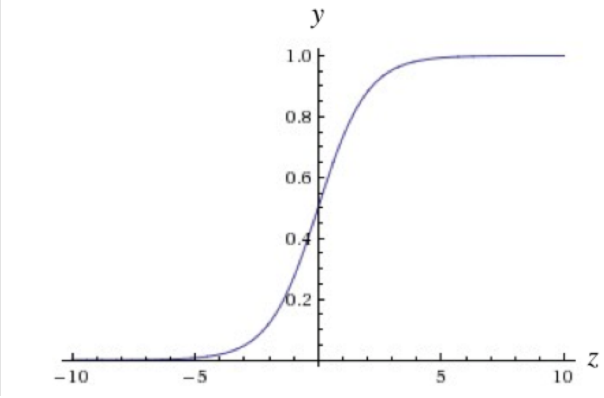

Figure 2.: Sigmoid activation function (Buduma et al., 2022, p. 13).

The tanh activation function maps real-valued input to value in the range $(-1, 1)$, as shown in Equation (4). The tanh activation function is often preferred to the sigmoid function because its output is zero-centered, as shown in Figure 3.

$$f(z) = \tanh(z) \tag{4}$$

The rectified linear unit (ReLU) function, as shown in Figure 4, returns the input for positive values; otherwise, it outputs 0, as shown in Equation (5).

$$y = \mathrm{ReLU}(z) = \max(z, 0) \tag{5}$$

Activation functions have different properties and advantages, and their selection depends on the specific network architecture and application requirements.

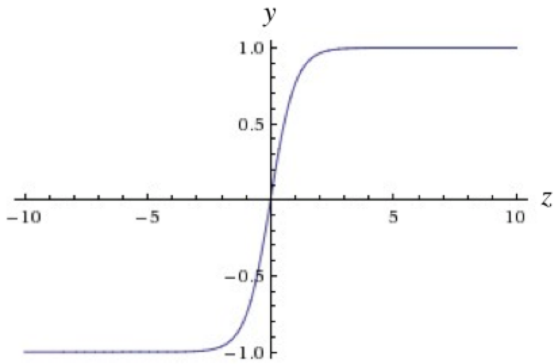

Figure 3.: Tanh activation function (Buduma et al., 2022, p. 13).

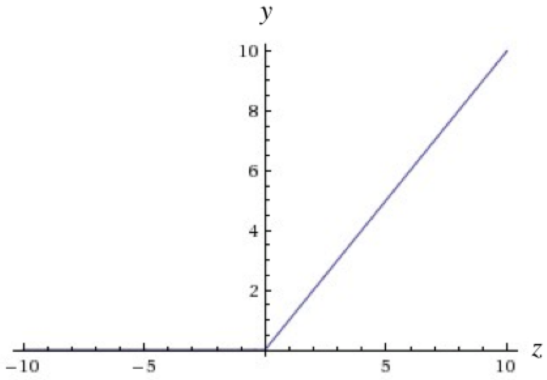

Figure 4.: ReLU activation function (Buduma et al., 2022, p. 13).

Additionally, the output layer can represent a probability distribution over the set of labels. For this purpose, a softmax layer is used. The softmax transforms the output of the layer by applying a normalization step so that the sum of the outputs $p_1, p_2, \ldots, p_n$ is equal to 1 (Buduma et al., 2022, p. 15). In classification tasks, the highest predicted class corresponds to the element in the softmax output vector with the highest value.

2.2. Feedforward Neural Networks

A neural network is composed of artificial neurons arranged in layers, typically including one input layer, one or multiple hidden layers h, and one output layer. After the neural unit receives an input, it processes it and passes the output to the subsequent layer. The neural network is called a feed-forward if the connections pass only from the lower layers to the higher layers, without connections between

neurons in the same layer (Buduma et al., 2022, p. 11). In a fully connected neural network, each neuron in one layer is linked to all neurons in the previous and the next layers, as shown in Figure 5.

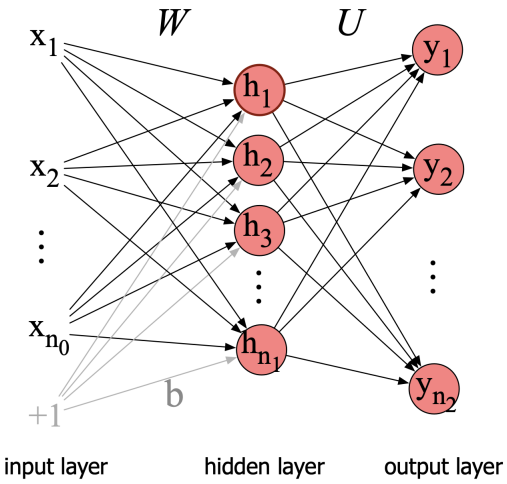

Figure 5.: A two-layer feed-forward neural network (Jurafsky and Martin, 2025, p. 139).

2.3. Neural Network Training

Neural networks can be trained using supervised, unsupervised, or self-supervised learning paradigms (Jurafsky and Martin, 2025, p. 140). Supervised learning, which will be described in details in this chapter, relates to settings where datasets with annotations, so-called gold labels, are used to train the model. Unspuervised refers to training scenarios where no such labels exist and the models are required to learn without a supervision signal. Finally, self-supervised is a training paradigm where the supervision signal is created directly from unannotated datasets. This can refer to masking sequences of a sentence or to predicting next sequences following an input sequence.

Supervised neural network training is the process of finding the optimal parameters W (weights) and b (biases) for each layer so that the predicted outputs of the model are as close as possible to the actual outputs (Jurafsky and Martin, 2025, p. 145). During the training the large number of examples with output are shown to a model, the model finds optimal weights and biases (parameters) by minimizing the loss function (Buduma et al., 2022, p. 17).

The loss function in the neural network measures the difference between the predicted output and the gold label. The lower the value of the loss function,

the closer the model prediction is to the ground truth. For neural networks, a common loss function is the cross-entropy loss function (Jurafsky and Martin, 2025, p. 145). During neural network training, the cross-entropy function (Jurafsky and Martin, 2025, p. 189) is minimized so that the correct answers are assigned higher probabilities and incorrect answers are assigned lower probabilities. This is achieved by minimizing the cross-entropy loss, which is equivalent to maximizing the log probability assigned to the correct outputs y given the inputs x.

The algorithm that minimize the loss function of a neural network is called gradient descent (Jurafsky and Martin, 2025, p. 90). At the beginning of training, the weights of the network are typically initialized randomly. To optimize the model, during training the loss function is calculated to measure how far the model prediction is from the desired output. In this step, the gradient is computed. The gradient indicates the direction of the steepest increase of the function. Therefore, to minimize the error, the weights are updated by moving in the direction opposite to the gradient, as shown in the Figure 6. This process is repeated by taking steps opposite to the gradient until the loss function reaches a local minimum. The

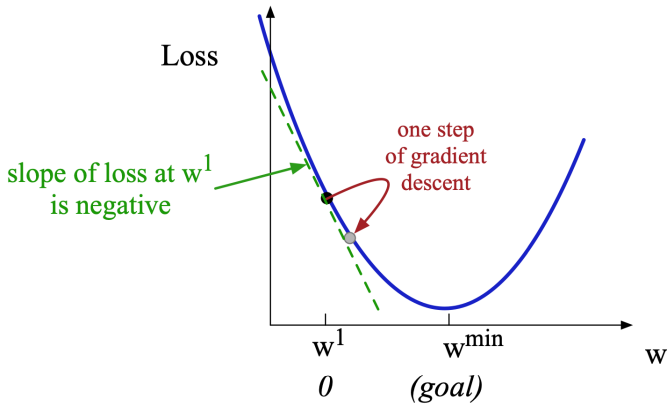

Figure 6.: A gradient descent step (Jurafsky and Martin, 2025, p. 90).

learning rate (Buduma et al., 2022, p. 21) is a hyperparameter that specifies the size of each step taken in direction opposite to the gradient at each update. In the case if the learning rate is excessively small, the training process can take longer, because it requires more steps to reach the minimum during gradient descent. However, if the learning rate is too big, there is a risk of constantly overshooting the minimum.

2.4. Transformer

The Transformer architecture (Vaswani et al., 2023), introduces a multi-head attention mechanism that captures contextual relationships in sequences. The Transformer includes two fundamental building blocks: an encoder and a decoder, as shown in Figure 7.

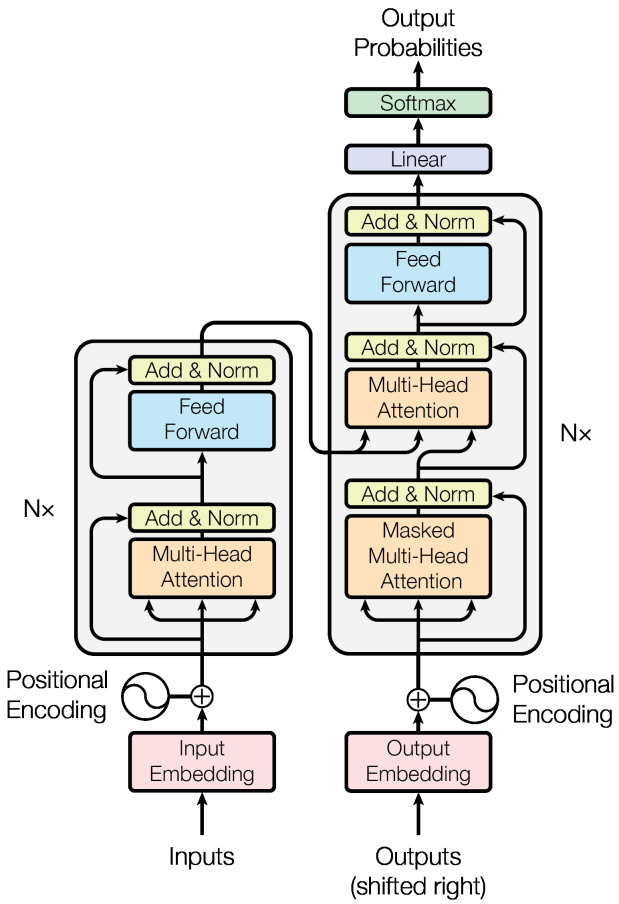

Figure 7.: Transformer architecture (Vaswani et al., 2023, p. 3).

In case of natural language-related tasks, the encoder receives the natural language input and transforms it into vector representations. The decoder receives these vectors as input and generates new tokens in natural language. During training, the prediction of the decoder is parallel across all tokens at once, however, during inference, the decoder generation process is autoregressive. Models may incorporate only the encoder, such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), only the decoder, such as Generative Pre-trained Transformer (GPT) (Brown et al., 2020), or both encoder-decoder

architecture, such as Text-to-Text Transfer Transformer (T5) Raffel et al. (2020).

2.4.1. Encoder

The encoder consists of a stack of identical layers, which is set to six in the original Transformer architecture (Vaswani et al., 2023). However, the number of layers is not fixed and can vary depending on the model architecture. For example, BERT-base incorporates repeated 12 layers. Each layer includes two core sub-layers: multi-head self-attention layer and a fully connected feed-forward layer (Tunstall et al., 2022, p. 60). Additionally, residual skip connections and layer normalization are applied after each sub-layer, which is referred to as 'Add & Norm' in Figure 7. The output embeddings of the encoder carry the contextual information of the word. The dimensionality of output embeddings of an encoder is the same as the dimensionality of input embeddings.

A residual connection (He et al., 2015) adds the input of a layer in addition to its output before passing the result forward. Therefore, the output of the layer always contains the input and the result of the transformation applied to the input. In case, if the transformation does not change the original input, the residual connection is equal to zero and it passes the input forward unchanged (Foster, 2023, p. 156). With the help of this, the neural network is learning on the layer difference instead of the direct output itself, which is called a residual mapping. This allows the paths for gradient propagation to be shorter, which helps to prevent gradients from becoming excessively small and therefore mitigates the vanishing gradient problem.

Layer Normalization (Ba et al., 2016) normalizes the input vector of each layer. The normalized vector is a vector where the mean equals to zero and the deviation equals to one (Jurafsky and Martin, 2025, p. 200). To achieve this, the mean (μ) and the standard deviation are calculated for the vector (σ^2), shown in Equations (6) and (7).

$$\mu = \frac{1}{d}\sum_{i=1}^{d} x_i \tag{6}$$

$$\sigma^2 = \frac{1}{d}\sum_{i=1}^{d}(x_i - \mu)^2 \tag{7}$$

Then the mean value is subtracted from each vector value and divided by the standard deviation, as shown in Equation 7.

$$\hat{x}_i = \frac{x_i - \mu}{\sigma} \tag{8}$$

Finally, the two learnable parameters are added to the formula, as shown in Equation 9.

$$\text{LayerNorm}(x_i) = \gamma \hat{x}_i + \beta \tag{9}$$

This makes the training process of the Transformer more stable, as it keeps the values of the hidden layers in a range that is suitable for gradient-based training.

2.4.2. Decoder

The decoder consists of a stack of repeated layers, and closely resembles the encoder architecture, including the self-attention and feed-forward layers with residual connections and layer normalization. However, the decoder has two differences from the encoder (Tunstall et al., 2022, p. 76). First, the self-attention mechanism in the decoder masks future tokens, forming the masked multi-head attention. With the help of this mechanism, each token on the current time step can attend only to previously generated tokens. This prevents the model from seeing future tokens and simply copying them as a target output. Secondly, the decoder has the encoder-decoder attention layer, which attends the output vectors of the encoder.

2.4.3. Attention

An attention mechanism allows neural networks to assign weight to each token in relation to all other tokens in the sentence (Tunstall et al., 2022, p. 61). This allows the model to capture how strongly each token is connected to the others in a sentence.

To achieve this, the attention mechanism uses three matrices: a query (Q), keys (K), and values (V). A query, Q, represents the token that is currently being processed by the model. Keys, K, represent all other tokens in the sentence, including the query token itself, to which the query can attend. The key encodes how relevant a token is to a given query. Values, V, are learned vectors that contain the information each token contributes. The attention mechanism uses these vectors to produce a weighted sum, based on how strongly the query matches each key.

The matrices Q, K, and V are obtained by multiplying the input embedding matrix X by three learned projection matrices W_Q, W_K, and W_V which are randomly initialized and updated during training, as shown in Equation 10. The embedded input sequence X has shape $N \times d$, where N is the input sequence length and d is the embedding dimension. Each row of X represents the embedding of one token in the input sequence.

$$Q = XW_Q, \qquad K = XW_K, \qquad V = XW_V \tag{10}$$

Then, the attention is computed using Equation 11.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\intercal}{\sqrt{d_k}}\right)V \tag{11}$$

2.4.4. Multi-Head Attention

The multi-head attention mechanism allows the model to simultaneously capture different types of connections between tokens. In multi-head attention, queries (Q), keys (K), and values (V) are first transformed separately for each head. The attention mechanism is then applied in parallel across all heads, the output of each is then combined and passed through a linear layer to generate the output, as shown in Equation 12.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(head_1, \ldots, head_h)W_O \tag{12}$$

where

$$head_i = \text{Attention}(QW_{Q_i}, KW_{K_i}, VW_{V_i}) \tag{13}$$

2.4.5. Positional embeddings

The Transformer architecture does not encode information about the position of tokens by default. To mitigate this limitation, the Transformer adds explicit positional embeddings to its input embeddings. Positional embeddings represent the information about the absolute positions of a token within a sentence. The resulting embeddings carry information about both the meaning of a word and its position within a sentence, as shown in Figure 8.

2.5. Sequence-to-sequence models architectures

As GEC is a sequence-to-sequence task, this section explores the most common encoder-decoder models, including Bidirectional and Auto-Regressive Transformers (BART) (Lewis et al., 2019), T5 (Text-to-Text Transfer Transformer) (Raffel et al., 2020), and Fine-tuned Language Net (Flan-T5) (Wei et al., 2022).

2.5.1. Bidirectional and Auto-Regressive Transformers (BART)

BART employs an encoder–decoder architecture, which incorporates a bidirectional encoder and an autoregressive decoder. The model was pre-trained on document corruption and reconstruction by optimizing the cross-entropy between the original

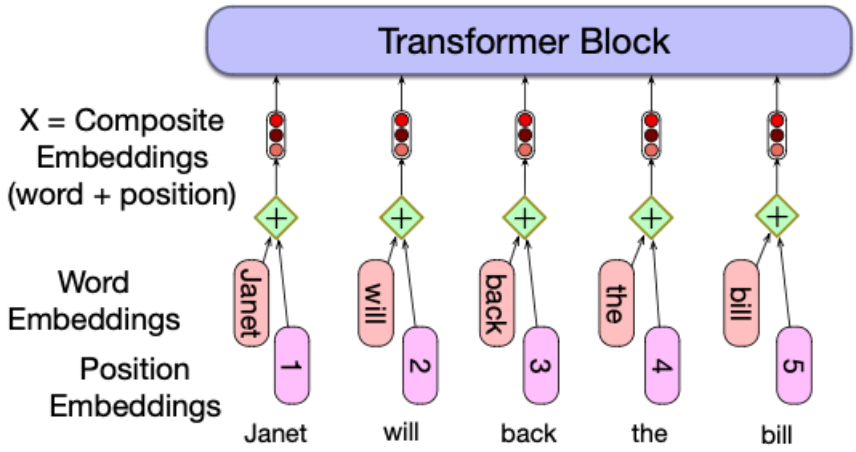

Figure 8.: Positional embeddings (Jurafsky and Martin, 2025, p. 198).

text and decoder output. The authors used different types of sentence corruption, such as token masking where tokens were masked and models needed to properly fill them; token deletion, when separate tokens were excluded from a sentence and the model needed both to identify the missing positions and insert the proper tokens; text infilling, where spans of tokens were masked and model needed to reconstruct them. Additionally, training included the sentence permutation, where sentences were shuffled and the model needed to receive the correct order, and document rotation, where the documents were rotated to start from a random place and the model needed to detect the token which was the original beginning. BART was fine-tuned to perform various tasks, such as question answering, summarization, and machine translation.

2.5.2. Text-to-Text Transfer Transformer (T5)

The T5 architecture was designed to address text-to-text tasks, in which the model accepts the text as input and produces the text as output. Machine translation, summarization, question answering are common text-to-text problems for T5, as shown in Figure 9.

The T5 architecture is based on the original encoder-decoder Transformer, inheriting the core building blocks such as an encoder-decoder structure, multi-head self-attention, feed-forward layers, residual connections, layer normalization, positional encodings, and softmax output. However, T5 introduces several modifications. Firstly, T5 applies layer normalization before each sub-layer and does not use additive bias terms in layer normalization. Secondly, T5 utilizes a different type of positional encoding. The original Transformer adds absolute positional vectors to

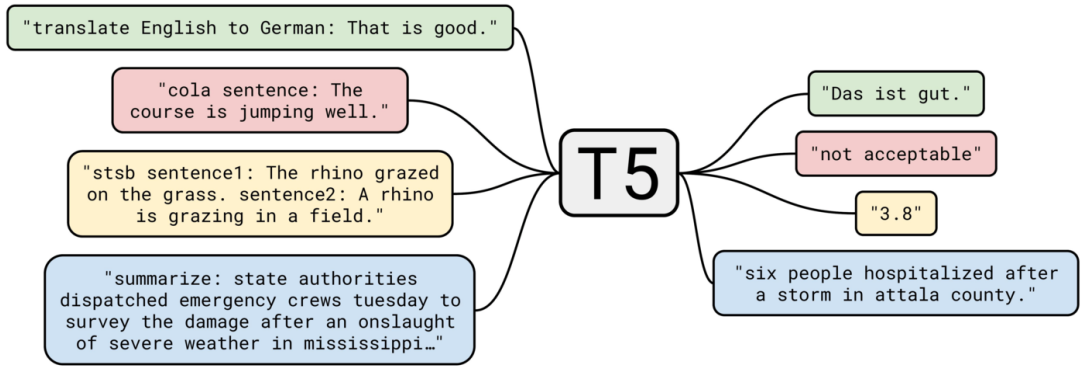

Figure 9.: Illustration of the Input and Output Structure of the T5 Model (Raffel et al., 2020, p. 3)

token embeddings as shown in Example (1).

(1) *Transformer Positional Encoding Example*
Sentence: The cat walks.
Token positions: 0, 1, 2
Positional vectors: [The + pos(0)], [cat + pos(1)], [walks + pos(2)]

However, T5 employs relative positional scalars, which capture the offset, that is, the pairwise distance between tokens for each query-key pair, as shown in Table 1. Moreover, unlike the original Transformer, these positional scalars are not added to token embeddings directly. Instead, T5 adds these scalars to the attention logits within each self-attention head.

Table 1.: Relative positional scalars for query token The (position 0)

Query (Q)	**Key (K)**	**Offset ($K - Q$)**	**Learned Scalar**
The (0)	The (0)	0	scalar[0]
The (0)	cat (1)	+1	scalar[+1]
The (0)	walks (2)	+2	scalar[+2]

The T5 model accepts a task-specific prefix at the beginning of the input sequence, which indicates the type of task to be performed by the model. This task-specific prefix can be formulated as a phrase, such as *translate English to German* or as a one-word command, such as *summarize*. During pre-training and fine-tuning, the model learns to align these task prefixes with their corresponding tasks.

T5 was pre-trained on a massive amount of data, primarily sourced from the Colossal Clean Crawled Corpus (C4), which was specifically constructed for T5. The C4 dataset contains over 750GB of English texts scraped from publicly available

web pages.

The model was pre-trained using a span corruption objective, in which it learned to predict the missing spans of tokens in a masked sentence, as illustrated in Example (2). This objective enables the model to adapt to various text generation tasks, such as translation, summarization, and question answering.

(2) *Example of span corruption pretraining used in T5*
Original sentence: The cat walks and loudly purrs.
Corrupted input: The `<extra_id_0>` and `<extra_id_1>`
Target output: `<extra_id_0>` cat walks `<extra_id_1>` loudly purrs

The T5 model is released in several sizes: small with 60M parameters, base with 220M parameters, large with 770M parameters, as well as two additional versions of a large model: T5 xl with 3B parameters and xxl with 11B parameters. Scaling up the number of parameters tends to improve the model performance; however, larger models require significantly more computational resources during training and inference.

2.5.3. FLAN-T5

Fine-tuned Language Net (FLAN) (Wei et al., 2022) is a framework in which language models, including GPT-3, are fine-tuned on more than 60 various NLP datasets annotated with natural language instructions. The process of fine-tuning models on datasets that include a set of instructions is called instruction-tuning. Instructions guide the model and are formulated in natural language, such as *Translate from Russian to German* or *Answer the question with yes or no* and are passed to the model along with the input and expected output during fine-tuning as shown in Example (3).

(3) *Example of instruction-tuning*
Instruction: Answer the following yes/no question.
Input: Is the cat a mammal
Output: yes

FLAN-T5 inherits the T5 architecture and extends it with the FLAN framework, which incorporates an additional instruction-tuning stage on tasks with natural-language instructions. The results of the research have demonstrated that the 137B-parameter FLAN fine-tuned model outperforms the larger 175B-parameter GPT-3 on the set of unseen tasks in a zero-shot setting.

2.5.4. mT5

mT5 is an adaptation of the T5 model (Xue et al., 2021), pre-trained using transfer learning on the multilingual dataset mC4, which was specifically created for mT5. This dataset contains texts scraped from web pages in more than 100 languages. The goal of transfer learning is to adapt a pre-trained model for a downstream task. For example, the T5 model was pre-trained primarily on an English-language dataset, therefore does not natively support other languages. To address these limitations, the authors released the mT5, the multilingual version of the T5 model, pre-trained on a corpus containing 101 languages. The multilingual data for mC4 were scraped from the entire Internet, resulting in an imbalanced distribution of languages within the dataset. To mitigate this issue, especially to support rare languages with critically low amounts of data in the corpus, the authors assigned sampling probabilities to each language and upsampled languages with lower probabilities to ensure more balanced training. The mT5 model was released in five sizes: Small with 300M parameters, Base with 580M parameters, Large with 1.2B parameters and XL with 3.7B parameters and XXL with 13B parameters. The model achieved state-of-the-art results on multiple benchmarks, demonstrating its strong multilingual capabilities.

3. GEC Dataset Creation approaches

Data scarcity remains one of the key challenges in addressing GEC tasks. GEC tasks require aligned sentence pairs, where one sentence contains errors and the other provides the corrected version. However, naturally aligned sentence pairs with error corrections are hard to obtain, making it more difficult to address the task. This chapter is dedicated to the main strategies for collecting GEC datasets, highlighting their strengths and drawbacks. Section 3.1 is focused on manual corpus creation strategies, where error corrections and corpus annotation were completed by humans. This approach is the most time- and resource-consuming; however, it enables the generation of high-quality corpora for various domains and settings. Section 3.2 is divided into corpus creation with scraping data from online-collaborative platforms with edit histories. The major advantage is the opportunity to collect waste amounts of parallel human-generated data. The drawback lies in the potentially low quality of the performed edits. Section 3.3 explores the back-and-forth data generation method, in which errors are artificially introduced into the text by translating it from the source language to a bridge language and then back again. This method effectively generates noisy GEC corpora, however, it does not cover all possible types of mistakes. Section 3.4 describes the GEC data generation pipeline with tagged corruption models. This method introduces control over the generation of erroneous outputs by specifying error types. Section 3.5 introduces the rule-based approach, in which grammatically correct sentences are corrupted with predefined error types based on specific linguistic rules. Lastly, Section 3.6 describes the strategies of generating parallel GEC corpus with LLMs.

3.1. Manual corpus creation

Manual corpus creation is one of the most challenging and time-consuming approaches for creation of GEC datasets. However, it has distinct advantages, especially for specialized fields such as academia, which require precise and accurate revisions. A key strength of manual corpus creation is the ability to maintain the standards of revision quality. Researchers can determine a procedure and select participants, which in most approaches are teachers (Lee and Webster, 2012) or professional editors (Du et al., 2022). However, in some cases, a self-revision can be implemented (Lee and Webster, 2012). The primary goal of the authors

was to specifically contribute to the self-revision process of argumentative writing. Participants were asked to write short opinion essays and then revise them based on teachers feedback that required them to provide examples both for and against the claim. This approach demonstrates that manual corpus creation allows for specification and determination of a revision procedure, resulting in more consistent and relevant results for solving the specific challenges in a particular domain.

Mita et al. (2024) focused on document-level revisions in the academic domain and introduced the Text Revision of ACL Papers (TETRA) corpus. This corpus consists of annotations made by professional editors on the abstract and introduction sections of ACL Anthology papers. The corpus used an XML-based format that contains annotations beyond the sentence level, in addition to word-level and phrase-level annotations. This allows for the revision of entire paragraphs, which was the main focus and contribution of the authors, as most approaches primarily address word-level corrections or changes within sentences. A key limitation of TETRA is its limiting scale: human-annotation, especially from experts, is an expensive and time-consuming process. The corpus contains only 61 annotated articles, which is insufficient for fine-tuning a GEC system. Instead of directly training the GEC system, the authors conducted a meta-evaluation, where the model had to determine the revised document in paired document comparisons. The accuracy achieved by BERT and GPT-4 demonstrated that language models can effectively distinguish between revised and unrevised versions of the same document.

An important advantage of manual corpus creation is that it allows researchers to introduce specific error categories for labeling. For example, ITERATER (Du et al., 2022) developed a classification for editing with a multilayer structure: meaning-changed edits on the higher layer; and clarity, fluency, style and coherence error types on the lower layer. The main motivation of this taxonomy is to make a step forward into the iterative editing process. The authors focused on the manual annotation, while the edits themselves were sourced from platforms with edit histories across different domains from arXiv, Wikipedia and News articles. However, despite the involvement of 11 qualified annotators and a team of linguists for annotation verification, the fully manually-labeled corpus, *ITERATER-HUMAN*, consisted only of 4,018 sentence pairs. To scale up the corpus, the authors implemented automatic annotation with a RoBERTa-based classification model (Liu et al., 2019). The model was fine-tuned on the human-based corpus and then applied to generate the *ITERATER-FULL* dataset. The authors compared *ITERATER-FULL-RAW* (dataset without annotations), *ITERATER-HUMAN* (dataset with human annotations), and *ITERATER-FULL* (dataset with automatic annotations) in the training of GEC models. The results showed that models trained on annotated datasets performed better than without annotations. Moreover, the

best results were achieved from a scaled annotated *ITERATER-FULL* dataset with automatic labeling, primarily due to the bigger size of the corpus. This demonstrates the importance of large-scaled data in training GEC models and points out the key constraints of human-crafted corpora.

Nevertheless, it remains possible to create a human-crafted corpus that is sufficient for training GEC models. One of the largest human-crafted GEC corpora is the National University of Singapore Corpus of Learner English (NUCLE) (Dahlmeier et al., 2013), which includes 1,400 essays, each 500 words long manually corrected and annotated with error tags by professional English instructors. The essays were written on various topics by undergraduate students learning English as a Second Language (ESL). The corpus contains over 50,000 sentences, categorized into 27 types of grammatical errors mainly based on the Centre for English Language Communication (CELC) error classification system with minor adjustments based on the feedback of the instructors. The large size of the corpus made it sufficient for the training of GEC models (Junczys-Dowmunt et al., 2018). Additionally, the authors calculated the error distribution statistics and described the most frequently occurring error tags in the corpus.

The creation of a large human-crafted corpus is time-consuming and resource-intensive, so it is crucial to estimate the feasible amount of annotated data that can be produced and determine if it will be sufficient to train the GEC model. However, even if the size of the human-annotated corpus is insufficient to train the model, it can still be valuable for developing GEC systems. It can be used as a base for scaling up the corpus with synthetic error or tag generation approaches (Du et al., 2022), or serve as a source for calculating error distribution, which can significantly improve synthetic data generation (Stahlberg and Kumar, 2021).

3.2. Data collection from online collaborative platforms

Another method of GEC dataset creation involves collecting data from online collaborative platforms that support text revisions, such as Wikipedia (Lichtarge et al., 2019). The primary advantage of this approach is a large amount of both human-written source sentences and their revisions. This extensive volume of revisions allows to build large corpora, such as WikiDocEdits with 12 million sentence-level edits (Faltings et al., 2021). Furthermore, the wide range of topics on Wikipedia provides valuable diversity for model training (Lichtarge et al., 2019). Additionally, extracting GEC data from online collaborative platforms can be highly beneficial for languages with limited data resources.

A notable benefit is the simplicity of the data extraction process, as both original

sentences and their revisions are presented in Wikipedia edit histories. The main steps of extraction revisions from Wikipedia were presented by Grundkiewicz and Junczys-Dowmunt (2014). Wikipedia edit histories are available in XML format and require several preprocessing steps: removing markup, splitting text into sentences, and applying filtering criteria, such as maximum sequence length or edit length constraints. Moreover, Wikipedia provides metadata that includes comments on edits, which can be a valuable resource for improving GEC datasets. The authors collected over 12 million pairs of sentences, covering spelling and grammar error corrections, stylistic improvements, and sentence paraphrases. However, Wikipedia edits include some revisions that are not valuable for the GEC task, such as changes to time references, updates to numerical information, as shown in Example (1):

(1) In [-May 2003-] +August 2004+ this percentage increased to [- 62-] +67+%

The biggest obstacle is that collaborative platforms allow anyone to perform revisions. As a result, some edits include intentionally wrong edits or even vandalistic changes. This brings an additional limitation to using online collaborative platforms for GEC corpus generation, since it requires additional quality control. For example, Lichtarge et al. (2019) flagged more than 1.7 million sentences as potentially harmful, which included vulgar terms, changes limited to dates or numeric values, and technical artifact sentences with markup tags.

Another significant limitation is the substantial amount of noise in Wikipedia-extracted corpora, as well as the large amount of errors which are not relevant for a target domain. For example, to adapt the Wikipedia-extracted corpus for ESL correction tasks, the authors (Grundkiewicz and Junczys-Dowmunt, 2014) kept in WikEd only mistakes similar to errors found in NUCLE. To achieve this, they applied the Longest Common Subsequence (LCS) algorithm to identify NUCLE error patterns and then converted them into regular expressions. By applying these regex patterns, they filtered the WikEd corpus, keeping only matching sentence pairs. The final corpus contained 3.9 million sentences, comprising just 32% of the original WikEd dataset. Alternatively, Wikipedia edits can act as an augmentation source for existing corpora, which can be especially beneficial for low-resource languages. For example, Boyd (2018) expanded a small gold GEC dataset by applying the ERRANT tagging system and further filtering corrections based on mistake types. This resulted in a 13.38% improvement in the model's $F_{0.5}$ score.

Wikipedia is not the only one possible source for gathering data for constructing GEC corpuses. For example, the ClaimRev corpus (Skitalinskaya et al., 2021) consists of 124 thousand sentence pairs sourced from the revision histories on the online debating platform Kialo[1]. Authors scraped aligned sentence pairs with Kialo editing tags, such as clarification (rephrasing sentences), typo/grammar correction

[1] `https://www.kialo-edu.com`

(linguistic edits), and adding/correcting links (modifying evidence sources). While a predefined error structure can be beneficial, it may also be restrictive, in case authors need a different error classification. It is important to consider that each platform has specific characteristics, such as a writing style, domain specifications, and other relevant settings that can influence corpus development. For example, Kialo has a moderation process for each revision, which can potentially improve the quality of revisions compared to non-moderated platforms. Moreover, it is possible to apply several revisions to the same sentence, and all changes are stored in the history. The authors noticed that the quality of edits tends to increase over the revision history, with earlier changes being less refined and later ones demonstrating the highest quality.

3.3. Back-and-forth translation

The strategies described in the previous section illustrate methods for creating a corpus through the error correction process, where incorrect sentences are transformed into their correct forms. However, this is not the only possible strategy for building a GEC corpus. Another strategy involves error corruption, where researchers collect clean sentences and generate ungrammatical versions by introducing artificial errors.

One method for introducing artificial errors is a round-trip or backtranslation (Somers, 2005), a strategy originally developed to enhance the quality of Seq2Seq models for Neural Machine Translation (NMT). However, it also shows effectiveness in error generation tasks (Xie et al., 2018; Kementchedjhieva and Søgaard, 2023), facilitating the creation of noisy parallel sentences for error correction. In this method, the correct source sentence is extracted from the text, translated into another language – which is also called a bridge language – and then translated back into the original language. This method introduces noise and errors into sentences based on the limitations of translation systems and the ambiguity of translations. The main benefit of backtranslation translation in GEC is the opportunity to generate a large amount of data, which is highly valuable for low-resource languages. This method requires only the source sentences and does not depend on instructions, predefined error tags, or specific categories. However, Lichtarge et al. (2019) notices that errors generated through round-trip translation are less random than natural human mistakes, which may limit the fine-tuned model's ability to correct human errors. In addition, this approach has several limitations. Firstly, round-trip translation sometimes produces identity translations, where output versions of a sentence are the same as the original one, resulting in a clean output without errors, which must be excluded from the corpus. Secondly, another limitation is the lack of typographical errors that researchers had to manually introduce into the corpus.

This involved adding extra characters, e.g. *spelling* changed to *spelilng*, deleting characters, e.g. *spelling* changed to *speling* or transposing characters, e.g. *spelling* changed to *spellign*. Finally, Lichtarge et al. (2019) observed that backtranslation can produce only a limited range of possible errors making the dataset less diverse compared to data sourced from Wikipedia.

3.4. Tagged corruption models

Another way to obtain datasets for GEC is synthetic error generation based on error tags. Stahlberg and Kumar (2021) proposed a method of training tagged corruption models to generate ungrammatical versions of clean sentences based on error tags. The generation of specific errors with the error type tags adds a lot of control on the corpus generation, which is not possible with previous approaches like crowd-sourced data parsing and backtranslation. Additionally it opens up a lot of opportunities for domain-specific cases with specific rules and error tags systems. The method involves collecting parallel corpora, tagging errors by type with the ERRANT annotation toolkit (Bryant et al., 2017), and training a Transformer Seq2Seq model to generate errors in clean sentences using the specified error tags. As shown in Example (2), these error tags, e.g. NOUN:INFL for incorrect noun inflection, guide the error generation process.

(2) NOUN:INFL There were a lot of sheep. → There were a lot of sheeps.

The key benefit of this approach is that it can produce large diverse datasets for GEC and allows taking control over the errors at the same time. Additionally, it can accurately reflect the error distribution of real-world data. To accomplish this, they transfer the ERRANT error distribution to their synthetic dataset, which allows them to replicate a realistic error type distribution. A limitation of the original method is that the tagged corruption models were initially trained on parallel corpora with error tags, where error tags were labeled through ERRANT. However, those powerful tools might not support all low-resource languages. This initial limitation can be mitigated through Few-Shot Instruction Tuning with Large Language Models, where the model is trained on several examples with error tags and corresponding correct and incorrect sentence versions to produce large number of corrupted sentences.

3.5. Rule-based method

The rule-based method involves generating parallel sentence corpora by applying rule-based functions to clean sentences to produce erroneous versions. For example,

Xu et al. (2019) utilized five main error types for their rule-based approach, each with corresponding subtypes, and additionally included a Word Tree to perform substitutions based on morphological relationships. As a result, they generated a corpus containing 3 billion tokens.

The rule-based approach also enables the generation of GEC corpora for low-resource languages and domains, which is more difficult due to the limited availability of open data and the lack of error annotation systems, such as ERRANT, designed specifically for English. For example, one of the challenges in addressing GEC tasks in Turkish is the existence of only one open corpus. Kara et al. (2023) proposed a synthetic generation pipeline for producing error sentences from the correct ones and presented the GECTurk corpus, comprising more than 130.000 parallel sentences from news articles generated through this pipeline. As a first step, the authors sourced a list of writing rules from the Turkish Language Association (TDK). They manually selected 20 rules, using as a main criterion that the errors could potentially be produced by first language speakers. For each rule, they develop the function (f) that represents a transformation rule introducing grammatical errors into sentences, as shown in Example (3).

(3) Traditionally, some pronouns are written adjacent. [hiçbir → hiç bir], [herhangi → her hangi]

The rule-based function checks whether a token is eligible for error transformation based on both morphological and syntactical analysis. It also includes probabilities calculated based on the analysis of the corpus and student essays. If both requirements are satisfied, the transformation substitutes the correct token with the error in the sentence. The result pair consists of correct and incorrect sentences, with annotations indicating the beginning and end indices, and the error type. The resulting corpus contains 138,000 aligned sentences with 25 error types, which were used by the authors to train GEC models.

3.6. Generating synthetic data with Large Language Models

Large Language Models (LLMs) open up new approaches for synthetic dataset creation. However, there is a lack of research on synthetic corpus creation with LLMs. LLMs have been described as an error explanation tool (Song et al., 2024) or an evaluation tool (Kobayashi et al., 2024) for GEC. The role of LLMs as a data generation source for GEC, along with its challenges and limitations, remains underexplored. Nonetheless, some studies have examined the performance of LLMs in addressing GEC tasks. For example, Katinskaia and Yangarber (2024) tested

GPT 3.5 to correct errors from the ERRANT dataset. Human evaluation reveals that 86% of English editing was acceptable, compared to 80% of cases in Russian. Furthermore, the authors observe that human corrections often differ from the model-generated suggestions, due to the existence of several options on how to correct the sentence. When the model's correction is accurate, it is accepted as correct, even if it differs from the human annotation. Additionally, the authors analyzed false corrections based on error tags and text categories. They noticed that the most challenging areas for GPT 3.5 were prepositions, hyphens, articles, and tenses. Regarding text categories, they observed that correcting government-related topics in Russian was the most challenging task for the model. This is likely due to the frequent use of non-standard sentence structures and lexical choices in legal terminology.

LLMs open up enormous opportunities in GEC dataset creation due to their strong generalization capabilities. However, this approach remains underexplored. It was noticed that LLMs can sometimes produce too simplistic error patterns (Deng et al., 2025). Moreover, Li and Lan (2025) noticed that regardless of the effectiveness of LLM in many tasks, in GEC cases it can produce unexpected output that can change the semantic meaning of sentences instead of actually correcting them. This highlights the importance of controlling the LLM, which can be achieved by shifting its role from the corrector to more controllable roles, such as augmentation. The authors propose a Type-aware method based on LLMs for Data Augmentation (TypeDA), a pipeline for generating ungrammatical sentences. TypeDA consists of two modules: mask modeling and error filling. The mask modeling step takes a clean sentence and masks text spans at positions where people make mistakes, based on the pre-trained Seq2Seq T5-base model. The T5-base was trained on parallel sentences to fill in the masks, as illustrated in Example (4).

(4) **source (X):** {I, **wants**, to, be, a, math, **teach**}
target (Y): {I, **want**, to, be, a, math, **teacher**}
masked output (M): {I, **[MASK]**, to, be, a, math, **[MASK]**}

The primary goal of masking is to give the LLM more precise control and guidance in performing error generation. Masking limits LLM corrections to predefined areas, helping to reduce unexpected outputs.

In the error filling step the researchers prompted GPT-4 (OpenAI et al., 2024) with an erroneous source sentence (X), the corresponding clean sentence (Y), masked version (M), error type (a) and one-shot example for the error type as illustrated in Example (5). The output ($\hat{X}$) is a sentence that contains grammatical errors based on the specified error type.

(5) **Source (X):** {She, **had**, been, working, hard}
Target (Y): {She, **has**, been, working, hard}
Masked Output (M): {She, **[MASK]**, been, working, hard}
Error Type (a): Verb tense error
One-shot Example: {He is going to school} → {He are going to school}
Output $\hat{X}$: {She, **have**, been, working, hard}

Additionally, in the error filling step the authors include the error distribution, as they have noticed that different types of errors have varying probabilities of occurrence in the real world. To account for this, the model samples error types based on their actual occurrence probabilities based on the given dataset.

After the output sentences are obtained, the authors perform automated checks where they a) check if the masking token *[MASK]* is not present in the output, b) extract the error type from the target-output paired sentences with ERRANT and compare it with the actual error type, and c) calculate the Levenshtein distance between the source (X) and output ($\hat{X}$) sentences. As a result, the authors have obtained an augmented dataset containing typed pairs of clean target sentences (Y) and erroneous sentences ($\hat{X}$) generated by the LLM.

The authors compared TypeDA with other popular methods for introducing artificial errors, including backtranslation, rule-based methods, and direct noise. The results demonstrated that TypeDA more accurately followed the pre-defined error categories and outperformed other methods, which, in some cases, like rule-based, were limited in the types of errors, and in others, like backtranslation, generated redundant and unnecessary errors.

4. GEC Model Training

This chapter presents the main approaches to train GEC models. GEC is a task that is primarily formulated as a sequence-to-sequence problem. Therefore, Transformer encoder-decoder architectures, such as T5, are commonly used to address this challenge. Within the Transformer architecture, the encoder processes the erroneous input sentence, while the decoder generates the corrected output. Prior studies have explored the performance of T5 in addressing GEC tasks in various domains (Katinskaia and Yangarber, 2023), languages (Rothe et al., 2021), and settings (Faltings et al., 2021), which have demonstrated the SOTA results on multiple benchmarks. Furthermore, several papers (Schick et al., 2022; Raheja et al., 2023) have analyzed the role of instructions in training GEC models. These studies have shown that task-specific instructions can significantly improve models performance on GEC tasks. This impact is particularly notable on model families that were initially pre-trained with instruction-tuning, such as FLAN-T5. Additionally, it was investigated the generalization abilities Raheja et al. (2023) of GEC models to perform edits on unseen tasks, instructions, and languages.

4.1. Sequence-to-Sequence models

Sequence-to-Sequence models, such as T5, have demonstrated strong performance in addressing AEC tasks (Katinskaia and Yangarber, 2023; Rothe et al., 2021; Faltings et al., 2021). For example, Katinskaia and Yangarber (2023) trained the T5 model to check the answers of Russian language learners for grammatical exercises. The authors performed the training of the T5 model in two stages. In the first stage, they pre-trained the model on a large synthetically corrupted dataset that was created by introducing errors into clean WMT News sentences (Bojar et al., 2017) with Aspell confusion sets (Grundkiewicz et al., 2019). The resulting dataset contained 10 million aligned sentence pairs, enabling the T5 model to learn and recognize the main grammatical correction patterns in the Russian language. In the second stage, the authors fine-tuned the pre-trained model on the manually-crafted RULEC corpus, containing only 50,000 aligned sentences. This dataset consists of sentences written by learners of the Russian language, each of which was manually corrected by two first language linguists. This step allows to adapt the model to high-quality, real-word data in a specific domain of Russian as a second language.

The model was tested and evaluated on the RULEC-GEC test set, which includes essays written by language learners, manually corrected, and annotated for GEC tasks. The researchers evaluated the obtained corrections using the M2 scorer (Dahlmeier and Ng, 2012), ERRANT's $F_{0.5}$ score, and also performed a manual evaluation. The results showed that the model achieved an $F_{0.5}$ score of 52.1 and an M2 score of 52.9, while the manual evaluation demonstrated that 62.4% of the model's corrections were accurate. The downside of these automatic metrics is that they compare the model's output with the gold corrections, ignoring the fact that various options can exist to correct the sentence. Furthermore, the authors revealed that the quality of the test set can be improved, since it sometimes contains unnecessary and ungrammatical corrections. The researchers decided to re-annotate the RULEC-GEC dataset, initially selected for evaluation, on a subset of 5,000 items. The authors performed their own corrections of the erroneous sentences, ensuring that the revised version maintained minimal distance to the gold correction. This step allows authors to ensure that the re-annotated corrections are of high quality, which contributes to the validity of the evaluation procedure. With this adjustment, the model $F_{0.5}$ score was raised to 55.4 on the re-annotated subset. Additionally, they observed that the model performed better in cases where the erroneous sentence required the token replacement compared to token insertion or deletion. The authors assume that this may be caused by the bias in synthetic data used for pre-training, as the dataset was primarily generated with the token replacement algorithm.

Sequence-to-Sequence models also demonstrated the capability to address multilingual GEC tasks, where a model trained on one language is capable of performing the correction in another. Rothe et al. (2021) trained the large-scale multilingual mT5 model (Xue et al., 2021) to perform error corrections in English, Czech, German, and Russian. The authors also tested how performance changes with scaling up the parameters in GEC tasks, including large models such as the XL model with 3B parameters and the XXL model with 11B parameters. Similar to Katinskaia and Yangarber (2023), they trained the mT5 model in two stages: GEC pre-training stage and fine-tuning. In the first stage, they trained the model on the large multilingual dataset mC4, where they synthetically introduced errors, primarily of types that can occur in any language, such as characters swap *error* $\rightarrow$ *erorr*, lowercasing *Germany* $\rightarrow$ *germany* or inserting random characters. This large dataset enables the Transformer model to acknowledge basic GEC patterns in a multilingual setting. In the second step, they fine-tuned the model on existing human-crafted datasets created for each language, such as RULEC-GEC for Russian (Rozovskaya, 2024). These datasets contain more specific error correction patterns that exist in each language. Additionally, apart from first pre-training and then fine-tuning, they have experimented with alternative training strategies.

For example, they combine both pre-training and fine-tuning datasets together or integrating these data with prefixes *pre-train* and *fine-tune*. However, the initial two-stage training approach, where the model was first trained on synthetic data and then on human-labeled dataset, achieved the best performance. The results showcased that the largest gT5 XXL model with 11 billion parameters demonstrated the best $F_{0.5}$ score on the test set and achieved state-of-the-art results in Czech, German, and Russian. The weakest performance was observed with the small mT5 model, highlighting the importance of the model size for the high generalization abilities for unseen tasks. Moreover, the authors performed the cleaning of the LANG-8 English corpus introducing the CLANG-8 dataset, which led to improved $F_{0.5}$ scores across the models, demonstrating that the cleaning of the training data can enhance the performance of the GEC models.

4.2. Sequence-to-Sequence models accepting instructions

This section presents approaches for fine-tuning the model with instructions. Initially, the original Seq2Seq T5 model does not process instructions, but it is possible to add a prefix tag to guide the model. Moreover, the T5 model has several modified architectures, such as FLAN-T5 and LM-adapted T5, which are specifically designed to process and follow instructions. This section demonstrates methods of instruction-based fine-tuning, revealing the benefits of this approach as well as possible limitations.

Faltings et al. (2021) fine-tuned the T5-base model to perform text revisions according to user commands. The authors trained the Iterative Editor model on the WikiDocEdits dataset, which comprises 12 million sentence-level edits sourced from Wikipedia. In addition to corrections, Wikipedia includes editor comments which often reflect the edit intention, such as *simplify the text* or *update the fact*. To transform these human-written comments to the edits, they utilized the Wikipedia edit intention classifier, which automatically assigned one of 13 labels such as *fluency*, *content* etc. During training, the T5 model was given both correct and erroneous sentences as input, along with related error tags produced by the classifier, grounding information, and context sentences, all concatenated together with a special <sep> separator tag. The training was conducted on 1 million paired sentences, with the validation and test sets comprising 10,000 sentences each. Notably, in contrast to existing approaches (Rothe et al., 2021; Katinskaia and Yangarber, 2023) that have trained large-scaled T5 models, the authors trained a T5-base which has only 220 million parameters. During the evaluation stage, the authors aimed to assess not only the accuracy of the edits but also the alignment of

the performed corrections with the given editing commands. To assess the quality of the edits, the authors used three evaluation metrics: BLEU score, word-level edit precision, and sentence-level accuracy.

The BLEU score (Papineni et al., 2002) is a metric commonly used in MT. It measures the similarity of the model translation to the gold reference based on the n-grams. Although originally designed for MT, the metric can also be applied to GEC tasks. The authors computed the BLEU score for each sentence using n-grams up to 4. However, this metric has several limitations. As the BLEU score is calculated based on the similarity between the reference and the output, it tends to be generally high for text editing tasks, since the editing operations are typically relatively small. Additionally, if the model fails to perform changes and simply copies the source sentence, the BLEU score also tends to be high regardless of the fact that the model did not perform the correction. To mitigate this issue, the authors computed calculations of word-level edit metrics (precision, recall, accuracy) and sentence-level accuracy.

For the word-level edit metrics, the authors compared the reference with the model output in terms of token-level operations such as deletion, insertion, and replacement of a token. For example, for the edit illustrated in the Example (1), the word-level edit would include two triplets, each containing information about the operation type, the token position, and the token itself: (replace, 4, "was "), (replace, 7, "funny").

(1) **Source (X):** {My, old, cat, Max, **were**, cool, and, **funnny**}
Target (Y): {My, old, cat, Max, **was**, cool, and, **funny**}

The word-level edit is calculated two times, first by comparing the original erroneous sentence with the reference corrections, and second by comparing the original example with the model corrected output. Then, these two edit tuples are compared based on the calculations of precision, which shows how many model edit operations were correct; recall, which shows how many actual edits the model matched; and F1, as a harmonic mean between these metrics.

The sentence-level accuracy is calculated based on the word-level edit calculations. If all existing word-level edits within a sentence match between the reference and the model output correction, the sentence-level accuracy is assigned a 1, otherwise a 0. This metric showcases the overall amount of exact matches, where the model corrections are fully aligned with the references. The evaluation metrics obtained for the Interactive Editor model were compared with the GPT-2 baseline The results revealed that the Interactive Editor outperformed the baseline across all evaluation metrics including BLEU, word-level edit precision, and sentence-level accuracy.

Raheja et al. (2023) presented COEDIT, a fine-tuned FLAN-T5 model that accepts user instructions in addition to input, such as *Paraphrase the sentence* or *Fix the grammar*. The authors aimed not only to address the grammar error correction task, but also text simplification, clarity improvement, paraphrasing, or style adjustment. To provide a model of the understanding of the edit intentions, the authors created a set of instructions for each type of task. For example, the simplification error tag includes various synonymic instructions, such as *Simplify this text*, *Make this easier to understand*, *Make this simpler* or just *Simplify*. This adjusts the model to various ways to determine the instruction prompt.

Additionally, the authors investigated two variants of instructions: general-purpose e.g. *Summarize this text* and task-specific e.g *Rewrite to be easier to understand*. The results demonstrated that task-specific instructions are more effective for fine-tuning, as they increase the density of the task distribution space, allowing the model to generalize better to unseen tasks. Additionally, it improves the model performance with composite instructions, which incorporate two or more intentions within a single prompt, e.g. *Make text readable and coherent*. COEDIT was fine-tuned on a parallel corpus of 82K instruction-based input-output pairs, primarily sourced from open corpora, including the ITERATER dataset (Du et al., 2022). The authors add the instructions to the parallel sentences with edits with the format *<instruction: source, target>*, developing a new open dataset called COEDIT. The architecture selected for fine-tuning was FLANT5, with different sizes of parameters, including 770B, 3B, and 11B. Notably, the authors aimed to demonstrate the performance of small models (less than 1B parameters) and medium models (1-10B parameters) in text editing tasks.

The author compared the COEDIT model with various baselines. First, they analyzed the performance of FLAN-T5 compared to T5. Acknowledging the fact that the T5 model cannot accept the instructions as part of a prompt, they changed the human-written instructions to task-specific prefixes, such as *gec*, *clarity.*, etc, comprising the following input *<task: source, target>*. The results demonstrate that instruction-tuned COEDIT surpasses T5-XL in all types of edits with a drastic improvement in the metrics. Secondly, the COEDIT was compared with decoder-only models GPT3 and LLaMA from 7B to 176B parameters with zero-shot learning. Furthermore, the authors compared COEDIT with decoder-only GPT-3 and instruction-tuned LLM including ChatGPT with few-shot learning on 4 examples. In both settings, COEDIT outperformed these models. Additionally, a human evaluation was conducted, in which the best-performing 3B COEDIT model was compared with the fine-tuned 175B GPT3-edit model. The annotators were asked to select which corrected output for the same input was more fluent, accurate, and meaningful. The results showed that in 64% of all cases the annotators preferred the COEDIT output. Additionally, the authors examined the model's generalization

abilities, by evaluating it on unseen tasks, such as sentence compression and politeness transfer. Although the model was not trained on these specific tasks, they are still closely related to the existing tasks in the task distribution. The results demonstrated that COEDIT-XL outperformed the other models in out-of-domain generalization for GEC tasks.

Schick et al. (2022) introduced Plan, Edit, Explain, Repeat (PEER), a model for iterative text editing that includes multiple stages such as planning, editing, and explaining edits, with the opportunity to iteratively repeat the process. Similar to Faltings et al. (2021), the authors trained a model on the Wikipedia edit histories. The authors applied several steps of filtering to improve the quality of obtained data, such as cleaning the sentences containing special symbols or the removal of too long corrections. The authors trained the model on 5 million sentence pairs containing corrections and corresponding editor comments. The PEER model represents a fine-tuned LM-Adapted T5 model (Raffel et al., 2020) with sizes ranging from 3B to 11B parameters. The notable benefit is that LM-Adapted T5 accepts natural language instructions, such as *Simplify this sentence* or *Add a reference*, which can be extracted from the Wikipedia editor comments.

To evaluate the model, the authors utilized several metrics. First, they used the Exact Match (EM) metric, which reveals the amount of model edits that are completely matched with the reference edits. Second, they used SARI, a metric that evaluates output based on three word-level edit operations: add, delete, and keep. It checks if the words added, deleted, or kept in the model output are also added, deleted, and left untouched in the reference. Finally, they have examined the GLEU (Napoles et al., 2015), a variation of a BLEU score adapted for GEC tasks, and the ROUGE score, a metric that measures the n-gram overlap in the reference correction and the model output.

The authors compared PEER with several baselines and SOTA models, such as GPT-3, on editing tasks from various datasets containing editing instructions, such as JFLEG, ITERATER, and FRUIT. Additionally, to evaluate PEER's editing capabilities in out-of-domain scenarios, the authors presented the Natural Edits dataset, a diverse collection of edits. In addition to Wikipedia articles, this dataset also includes news articles from Wikinews, and questions from StackExchange covering various topics, such as cooking, movies, and gardening. This introduces new domains at the test stage, allowing to evaluate the model's generalization capabilities. The results demonstrate that PEER performs strongly in various editing tasks, consistently outperforming the SOTA models, demonstrating its ability to follow instructions and perform the required edits.

5. Evaluation metrics for GEC

Text generation tasks, such as summarization and machine translation, use automatic metrics to access the quality of the model output. Similar to other text generation tasks, the GEC task employs various automatic evaluation metrics.

5.1. F-score

A common approach to accessing the GEC model is to compute the F-score – the harmonic mean of precision and recall to measure how close the model corrections are to the reference gold edits (Bryant et al., 2023, p. 670). To obtain this metric, the gold reference sentences are compared to the output of the model, where each difference is considered as an edit. True Positive (TP) is counted when the model edit matches the reference. False Positive (FP) occurs when the model proposes an incorrect or unnecessary edit. False Negative (FN) occurs when the model fails to apply a required correction present in the reference. A visual representation of the TP, FN and FP calculations is shown in Figure 10.

		TP		FN	FP		
Original	I	likes	to	drive	a	bicycle	.
Hypothesis	I	like	to	drive	the	bicycle	.
Reference	I	like	to	ride	a	bicycle	.

Figure 10.: A visual representation of TP, FN and FP calculations (Bryant et al., 2023, p. 671)

These scores are then used to compute precision and recall. Precision shows the proportion of how many edits proposed by the model that are also present in the gold edits, as shown in Equation (14). Recall shows how many of the errors that were present in the gold standard edits were identified by the model, as shown in Equation (15).

$$P = \frac{TP}{TP + FP} \tag{14}$$

$$R = \frac{TP}{TP + FN} \tag{15}$$

Finally, the F-score is computed, which is the harmonic mean of precision and recall, as shown in Equation (16). In contrast to the arithmetic mean, the harmonic mean strongly penalizes low values of precision or recall. The F-score provides a balanced measure of the ability of a model to perform correct edits and cover the edits presented in the gold reference.

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \tag{16}$$

However, in GEC research, it is common to report the $F_{0.5}$ score, which weighs precision twice stronger than recall, as shown in Equation (17).

$$F_{0.5} = \frac{(1 + 0.5^2)\, PR}{0.5^2 P + R} \tag{17}$$

This is motivated by the understanding that in GEC tasks it is generally more important for a model to avoid incorrect corrections than to find all possible errors.

5.2. Bilingual Evaluation Understudy (BLEU)

The Bilingual Evaluation Understudy (BLEU) score (Papineni et al., 2002), originally developed for MT, has widely been applied to evaluate GEC systems. The BLEU score compares the model-corrected output sentence (C) with the reference sentence (R) based on their n-gram overlap. However, this metric has several limitations for evaluating the model performance on GEC tasks. Firstly, in grammar correction tasks, the edit operations are typically relatively small. This can lead to situations in which unchanged source sentences without any corrections (S) may receive high BLEU scores, because they are similar to reference sentences (R) and have a high n-gram overlap. Therefore, even when the model completely fails to perform the correction and generates the unchanged sentence as output, the BLEU score can still be high.

5.3. Generalized Language Evaluation Understanding (GLEU)

The Generalized Language Evaluation Understanding (GLEU) (Napoles et al., 2015) score is a metric that was developed as an adaptation of BLEU for GEC tasks. This metric incorporates both the reference sentence and the source sentence

in the calculation. Considering the original sentence in the GLEU calculation helps to mitigate the tendency of BLEU to over-reward unedited or unchanged outputs. The core idea of GLEU is that if n-gram overlap is present in the model output and the reference, it is rewarded. However, if the model output has n-gram overlap with the original source sentence but not with the reference, it is penalized.

5.4. The System output Against References and against the Input sentence (SARI)

The System output Against References and against the Input sentence (SARI) (Xu et al., 2016) is a metric which evaluates the three editing operations which can be performed by the GEC model: keep, add, and delete a token. The metric compares all three sentences: source erroneous sentence (S), corrected model candidate sentence (C) and gold reference sentence (G). For each n-gram, the system rewards operations that are present in both model sentence (C) and reference sentence (G), but are absent from the input sentence (S) and penalizes the operations that appear in model output (C) but not in gold reference (G). The final score is obtained by averaging precision and recall across all three edit operations: keep, add, and delete.

6. Related work

The related work used for this study include approaches for generating error type systems (Kara et al., 2023). This approach served as inspiration for the development of the referencing error type table. Additionally, methods for developing GEC datasets were explored, including the use of tagging-corrupted models (Stahlberg and Kumar, 2021), in which errors are generated based on error tags, and an LLM-based approach (Li and Lan, 2025). Additionally, approaches for T5 model training were examined, including the method proposed by Katinskaia and Yangarber (2023), that employed two-stage fine-tuning using firstly a general-purpose dataset and secondly smaller but higher-quality smaller dataset. Moreover, Katinskaia and Yangarber (2023) proposed a method for GEC model training that incorporates instruction-tuning of FLAN-T5.

Stahlberg and Kumar (2021) addressed the lack of control in corpus generation by training tagged corruption models. This method allows to generate corrupted sentences based on the clean ones using specific error tags from the ERRANT toolkit. This approach adds precise control over the types and distributions of errors. As a result, a diverse and domain-specific GEC corpus was created that reflects real-world error patterns.

Addressing GEC in low-resource domains and languages is challenging due to data scarcity and the lack of error annotation tools such as ERRANT, that are available only for English. To address this issue, Kara et al. (2023) developed a rule-based synthetic generation pipeline for Turkish, using 20 transformation rules sourced from the Turkish Language Association, to produce corrupted versions of correct sentences. Based on this approach, the GECTurk corpus was created comprising 138,000 parallel sentences with 25 annotated error types. This corpus significantly expanded available resources for the Turkish GEC. Li and Lan (2025) observed that while large language models (LLMs) are effective for a range of tasks, including corpus creation for GEC. Nonetheless, the authors highlighted the importance of adding control over LLMs due to their tendency to either oversimplify or overcorrect sentences. To address this challenge, they proposed TypeDA, a type-aware data augmentation pipeline with mask modeling and error filling. This allows to guide LLMs in generating corrupted sentences based on specific error types. Experiments showed that TypeDA outperforms traditional augmentation approaches such as backtranslation and rule-based methods by producing more accurate and diverse errors.

There are different strategies for training efficient GEC models. For example, Katinskaia and Yangarber (2023) addressed the lack of annotated learner data by pre-training a T5 model on 10 million synthetically corrupted sentence pairs and then fine-tuning it on the high-quality, manually corrected RULEC corpus of 50,000 sentences. Evaluated on the RULEC-GEC test set using automatic and manual evaluation, the model initially achieved an $F_{0.5}$ score of 52.1, with 62.4% accuracy in manual checks. The authors found limitations in the test set and added several alternative correct options, which raised the $F_{0.5}$ score to 55.4.

Raheja et al. (2023) introduced COEDIT, a fine-tuned FLAN-T5 model designed to follow user instructions for tasks such as grammar correction, simplification, paraphrasing, and style adjustment. By creating a diverse set of instructions, the author fine-tuned the model on 82,000 instruction-based input-output pairs. The authors demonstrated that task-specific and composite prompts significantly improved performance and generalization to unseen tasks. Compared to baseline models, including T5, GPT-3, LLaMA, and ChatGPT, COEDIT consistently outperformed larger models particularly in human evaluations where annotators preferred COEDIT's outputs in 64% of cases.

7. Methodology

This chapter describes the methodology of the study. Section 7.2 discusses the development of the error category system, which defines various types of referencing errors. This error category specification is used to guide LLMs in both data generation pipelines. Section 7.3 describes two strategies for GEC dataset generation employed in this study. The first strategy, presented in Subsection 7.3.1, involves generating both clean and erroneous sentence pairs using the LLM. The second strategy, described in Subsection 7.3.2, involves collecting real sentences containing references from theses and corrupting these sentences with errors using the LLM. Section 7.5 discusses the models selected for fine-tuning to perform reference error correction. Section 7.6 describes the experiments conducted in this research. Finally, Section 7.7 describes the evaluation procedure and presents the real-world dataset used as a test set to assess the model performance.

7.1. Dataset generation strategy

The dataset for training AEC models typically consists of aligned pairs of erroneous and corrected sentences. There are various approaches to collect parallel AEC corpora, which are described in Section 3. However, not all methods are suitable for Automatic Error Correction for Referencing. To effectively address AEC in academic referencing, the corpus must satisfy the following requirements: (a) sentences must be from the academic domain; (b) errors must occur only in the referencing part of the sentence; and (c) referencing rules must specifically follow the CTS guidelines.

Manual dataset creation requires a vast number of erroneous sentences that need to be corrected, and faces serious time and resource constraints. The main constraint for this method is the absence of an openly available large number of sentences containing referencing errors. The backtranslation approach is not appropriate for this task, as it is an uncontrollable method that does not allow determination of specific error types to corrupt sentences. Collecting parallel sentences with corrections from online-collaborative platforms is also unsuitable, as draft versions of theses and academic papers with corrections are not publicly accessible.

However, data scraping can be used to collect clean sentences, which can later

be corrupted with errors. For this purpose, approaches such as tagged corruption models (Bryant et al., 2017) or controlled generation of synthetic data with LLMs (Li and Lan, 2025) can be adopted. The core concept of tagged error corruption is to pass clean sentences to the model with an error tag to generate the erroneous sentence corresponding to the specified tag. In contrast to tagged corruption models, LLMs not only allow error corruption but also enable the generation of clean sentences for further corruption, making them highly beneficial for automated dataset creation. However, LLMs have several downsides and constraints. More specifically, one of the risks of LLMs in the task of error generation is overcorrection, which can be mitigated by masking and detailed input containing clear examples, and more precise prompt instruction control (Li and Lan, 2025).

An additional challenge in GEC dataset creation is that the model requires a large volume of training data, and collecting high-quality data is particularly difficult in low-resource domains. Recent research (Katinskaia and Yangarber, 2023; Li and Lan, 2025) has demonstrated that synthetic dataset generation can be an effective strategy to train GEC models. Moreover, Katinskaia and Yangarber (2023) noted the effectiveness of two-layer GEC model training using synthetic data. The initial stage involves a data generation approach that produces a large but non-diverse dataset, which serves as the first pre-training step. In the second stage, a more accurate, diverse, and high-quality dataset can be used for fine-tuning. Inspired by this two-step strategy, as well as by related works in LLM dataset creation (Deng et al., 2025; Li and Lan, 2025), this thesis explores the effectiveness of dataset creation approaches for addressing the Referencing Error Task:

- Controlled sentences parallel sentences generation (both correct and erroneous sentences)
- Controlled error corruption of real-word sentences with LLM

Controlled generation or corruption allows adding to the prompt for an LLM some constraints or directives which can enhance the model output, make it more precise, which is especially valuable for LLMs with a small number of parameters. To achieve this, an error tag system was developed to classify and systemise referencing errors based on the textual guidelines.

7.2. Error types and rules

In most artificial error generation approaches based on the error categories described in the theoretical chapter, such as tagged-based approaches (Section 3.4), LLM-based methods (Section 3.6), or rule-based methods (Section 3.5), the error tag system is usually predefined. Additionally, researchers have worked with English,

which already has a well-established error classification scheme and widely used error annotation tool ERRANT. In the case of a low-resource language or task, the error tag system must be constructed manually, as demonstrated by Kara et al. (2023). Based on the list of writing rules for the Turkish language, the authors determined the error tags themselves, resulting in 25 error tags derived from 20 manually selected rules. These error tags were used to guide a rule-based error generation model. The Center of Translation Studies of the University of Vienna has explicit guidelines for master theses, which can serve as a basis for developing the error tag system. The rules of referencing are described in Section 3.1 of the textual guidelines. The section contains paragraphs which can be interpreted as distinct rule categories. For example:

> "Beim Verweisen auf Werke, die von mehr als zwei AutorInnen oder HerausgeberInnen gemeinsam erstellt wurden, wird im Quellenverweis in der Regel nur der erste Name ausgeschrieben, gefolgt von (lat.) ‚et al.' (= und andere)".

This rule can be used as a source for an error category *Error in Using "et al." for Multiple Authors*. To facilitate future analysis and data generation, each category was assigned a specific error tag, such as ETAL in this case. After an error category is defined, specific error types are introduced that align with sentence examples. For example, Table 2 demonstrates several error types associated with the *ETAL* error category.

Table 2.: ETAL Error Types Examples

Error tag	**Example of Mistake**	**Corrected**
ETAL	Pym, Risku, and Shlesinger (2012) argue that translation theory is evolving.	Pym et al. (2012) argue that translation theory is evolving.
ETAL	Pym, Risku, Shlesinger, Baker (2012) explore the cultural aspects of translation.	Pym et al. (2012) explore the cultural aspects of translation.

Similarly, other paragraphs were viewed as sources for error categories, such as Incorrect Citation Order (ICO), Error in Using "et al." for Multiple Authors (ETAL), Incorrect Citation for Two Authors (TWO), and Incorrect Use of Page Number Abbreviations (PAGE). Other types were proposed after the analysis of real-word errors of the manually collected English part of the set, such as Wrong Choice of Type of Referencing (TYPE), Redundant entity (RED), and Incorrect Placement of Citations (PLC). The error type database consists of 8 error categories and 46 distinct error types and is provided in Appendix A. Each error type includes an error category tag, a unique identifier (error ID), a one-shot example consisting

of a clean and erroneous sentence, and a detailed rule explanation to guide the LLM more effectively.

The *ICO* rule includes the cases where two or more sources are cited in the wrong order, as shown in Table 3. According to the rule, when the authors have different surnames, the references should be listed in alphabetical order by surname rather than chronologically by year. However, if the author's surnames are the same, the works should be listed in chronological order.

Table 3.: ICO Error Types Examples

Error tag	**Example of Mistake**	**Corrected**
ICO	Research on translation strategies has been carried out by several scholars (Jones 2010; Smith 2015).	Research on translation strategies has been carried out by several scholars (Smith 2015; Jones 2010).
ICO	Translation strategies have been discussed in various studies (Baker 2008; Baker 2006).	Translation strategies have been discussed in various studies (Baker 2006; Baker 2008).

The *TYPE* error category applies to incorrect punctuation associated with the citation type, as illustrated in Table 4. Citation types include narrative citations, where the authors are part of the sentence, such as *Pym (2019) and Risku (1995) examined*, and parenthetical, where the authors are not directly involved in the sentence structure: *Research on translation strategies has been carried out by several scholars (Smith 2015; Jones 2010).* The error examples include cases in which these narrative and parenthetical citation types are applied incorrectly.

Table 4.: TYPE Error Types Examples

Error tag	**Example of Mistake**	**Corrected**
TYPE	(Pym 2019; Risku 1995) examined the impact of globalization on translation.	Pym (2019) and Risku (1995) examined the impact of globalization on translation.
TYPE	According to (Pym 2019), translation strategies are evolving rapidly.	According to Pym (2019), translation strategies are evolving rapidly.

The TWO tag includes errors that occur when the cited work has exactly two authors. In these cases, the referencing should include "and" or "&", but no other separators, as shown in Table 5.

The PAGE error type applies to references that include page numbers. Examples of PAGE-type errors include incorrect sequencing, such as using *Pym (2019: 18, 19, 20)* instead of *Pym (2019: 18ff.)* or *Pym (2019: 18, 19, 20, 21)* instead of *Pym (2019: 18–21)*.

The RED error type includes cases where part of a citation is unnecessarily repeated in the sentence. It is based primarily on an analysis of real-world citation

Table 5.: TWO Error Types Examples

Error tag	Example of Mistake	Corrected
TWO	The theory of translation was discussed by Pöchhacker, Risku (2005).	The theory of translation was discussed by Pöchhacker and Risku (2005).
TWO	Several studies have addressed this issue (Petrova, Ivanova 1995).	Several studies have addressed this issue (Petrova & Ivanova 1995).

errors, where authors often include surnames or year information redundantly in the citation, as illustrated in Table 6.

Table 6.: RED Error Types Examples

Error tag	Example of Mistake	Corrected
RED	As Pöchhacker (Pöchhacker 1995) suggests, equivalence plays a crucial role in translation.	As Pöchhacker (1995) suggests, equivalence plays a crucial role in translation.
RED	Pöchhacker in 1995 suggests that equivalence plays a crucial role in translation (Pöchhacker 1995).	Pöchhacker (1995) suggests that equivalence plays a crucial role in translation.

The PUN error type includes various punctuation errors, such as missing parentheses, incorrect parentheses types, omitted commas, or the use of a colon instead of a comma, among other similar mistakes, as shown in Table 7.

Table 7.: PUN Error Type Examples

Error tag	Example of Mistake	Corrected
PUN	Several scholars have written on translation theory (Catford 1965 Nida 1964 Newmark 1988).	Several scholars have written on translation theory (Catford 1965; Nida 1964; Newmark 1988)
PUN	According to Kaindl [2003], the theory of translation has evolved	According to Kaindl (2003), the theory of translation has evolved.

The PLC error type refers to cases where a citation is incorrectly placed in a sentence, as illustrated in Table 8. For example, in narrative citations, the reference should appear directly after the author's name instead of the end of the sentence.

7.3. Dataset generation procedure

This study examined the effectiveness of synthetic data in training Seq2Seq models to perform error corrections on real-world academic data. Specifically, this study explored two distinct approaches for generating synthetic GEC data. In the first

Table 8.: PLC Error Type Example

Error tag	Example of Mistake	Corrected
PLC	According to House, translation theories have been thoroughly examined and demonstrated the valuable results (2007).	According to House (2007), translation theories have been thoroughly examined and demonstrated the valuable results.

approach, the model generated both a clean sentence and its corrupted version. It receives as input an error tag, a rule description, and a one-shot example of a clean and corrupted sentence. In the second approach, real-world clean sentences collected from actual student work are corrupted using an LLM, based on the error tag, a description, and a one-shot example.

7.3.1. Clean and erroneous sentences generation with LLM approach

In this approach, both clean and erroneous sentence generation is performed using an LLM. To contribute to the investigation of smaller models, this study focuses specifically on the dataset generation with the GPT-4o-mini model (OpenAI et al., 2024), the current smallest and most cost-effective OpenAI model. This model provides API access, making it possible to execute a controlled data generation Python pipeline in environments such as VS Code.

Unlike in previous studies (Deng et al., 2025; Li and Lan, 2025), in this approach both clean and corrupted sentences are generated by an LLM. This indicates that the model required more precise guidance not only to introduce errors into clean sentences but also to generate the initial clean versions. For this purpose, one-shot examples were provided to guide the model in generating correct sentences. For sentence corruption, the model is guided by a detailed rule definition and a one-shot example of an erroneous sentence. Example (1) illustrates the data included in the prompt for generating both clean and erroneous sentences.

(1) **One-shot Example: Pym, Risku, Shlesinger, Baker 2012** explore the cultural aspects of translation. → **Pym et al. (2012)** explore the cultural aspects of translation.
Error Type: ETAL
Rule: When citing a work with three or more authors it should be used only first author surname with "et al."
Output $\hat{X}$: Smith, Brown, Thompson (2023) provide a comprehensive analysis of named entity recognition techniques applicable to legal texts. → **Smith et al. (2023)** provide a comprehensive analysis of named entity

recognition techniques applicable to legal texts.

Despite being instructed that sentence structure should vary from reference, the model consistently generated outputs closely resembling one-shot reference sentences. This limits dataset diversity compared to the second approach, which involves corrupting real-word clean sentences. After analyzing the initial subset of generated data, several downsides were noted:

1. The LLM fails to generate a diverse set of surnames in the sentences. Surnames such as Smith, Brown, Thompson, and Johnson are regularly repeated throughout the dataset. This can be a serious limiting factor, particularly for error types that rely on surname variation, for example for error category 'ICO' when authors must be listed in alphabetical order instead of chronological.

2. The LLM fails to generate a diverse set of numerical values (years, page numbers).

3. There is a lack of topical diversity in the dataset produced by the LLM. Sometimes, the model generated very similar sentences, primarily in the biological domain.

To address the first issue, controlled name prompting was used to increase diversity of the surnames in the generated data. Because the small model has limited ability to generate diverse surnames, they were randomly selected using the `names` Python library and included in the prompt. The `names` Python library allows random selection of first and last names, based on the 1990 US Census dataset, which contains more than 88,799 surnames. This large number of surnames allows to make the training data more diverse. Additionally, this generated data closely resemble real-word data, as the surnames were sourced from completed US census forms. For each sentence, five surnames were randomly selected from the library and directly included in the prompt using the instruction: *Use any of these surnames for sentence generation.*

To address the limitation of non-diverse numerical values, the Python random library was used to generate year values as a random number from 1800 to 2024 and page numbers in range from 1 to 800.

To address the limitation in topic diversity, 50 scientific problems related to translation and NLP were selected to introduce variation into the sentence generation process. With the random Python library, a random topic from this list was added to the prompt for each generation iteration: *Generate a clean academic sentence on the topic: topic-hint.*

To support a better model understanding of the corruption process, a rule was added for each error category. Moreover, for some rules, a 'details' column was included, in case additional comments need to be added for a specific error category to improve the quality of generated output. The input data schema passed to the prompt is illustrated in Figure 11.

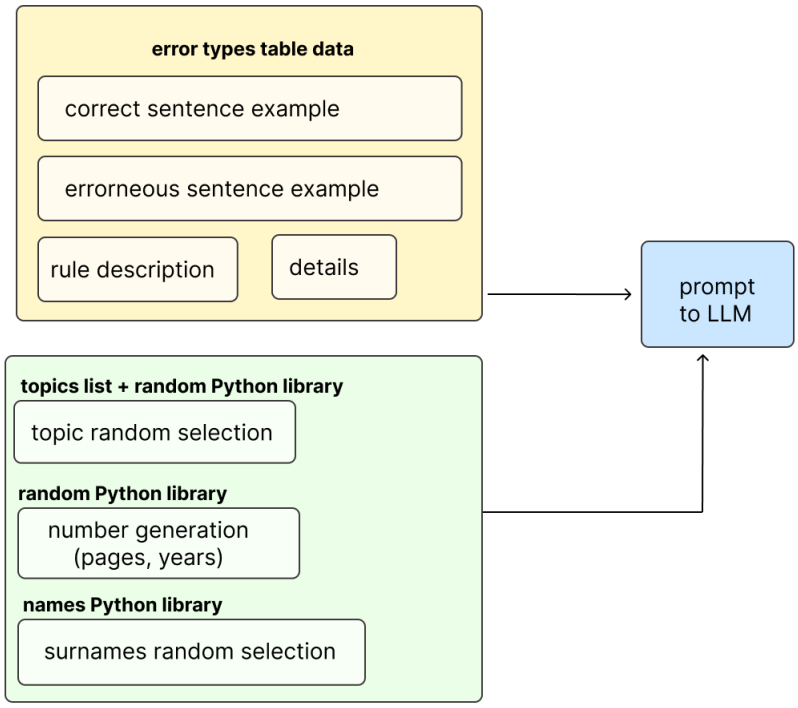

Figure 11.: Clean and erroneous sentences generation with LLM pipeline

As a result, the following prompt was used for the generation of each error category, as shown in Example (2).

(2) **One-shot Example: Pym, Risku, Shlesinger, Baker 2012** explore the cultural aspects of translation. → **Pym et al. (2012)** explore the cultural aspects of translation.
Error Type: ETAL
Topic (random selected): Named Entity Recognition
Rule: When citing a work with three or more authors, it should be used only the first author surname with "et al."
Surnames (random selected): Watkins, Kepner, Cassada, Chamberlain, Hardcastle
Year (random selected): 2023

Output $\hat{X}$: Chamberlain, Whittington, Church (2023) provide a comprehensive analysis of named entity recognition techniques applicable to legal texts. → **Chamberlain et al. (2023)** provide a comprehensive analysis of named entity recognition techniques applicable to legal texts.

For each of the 46 error types, 50 aligned pairs of erroneous and corrected sentences were generated. This results in two datasets consisting of 2,300 correction pairs for English and German. Additionally, using the same strategy, a synthetic test set was created, including 94 aligned pairs of erroneous and corrected sentences, which were used to evaluate the model.

7.3.2. Real-word sentences error corruption with LLM approach

This approach involves collecting real-world clean sentences from master's theses and their further error corruption. The first step involves collecting clean sentences, filtering out the reference sentences, and applying classification, as shown in Figure 12. As theses are available in PDF format, the initial step involves extracting all textual data from the PDF pages using libraries such as PDFPlumber. This library allows extracting PDF data and converting it into Python strings for further processing. Additionally, the library supports passing a page range as a parameter for textual extraction. As the focus is on content-relevant text, only pages from the introduction to the conclusion were selected, excluding the title page, abstract, acknowledgments, table of contents, and bibliography, as these sections are not relevant to the research.

For the research task, only sentences that contain reference information are relevant. To achieve this, a regular expression (Regex) was implemented to filter only the necessary sentences. The Regex pattern checks whether a sentence includes referencing content and extracts only those sentences into the database. The pattern was designed to collect any sentence that includes round parentheses containing a four-digit year beginning with 18, 19, or 20. This pattern effectively captures sentences containing references, as referencing sentences always include the publication year in round parentheses.

As a result, 2,491 German sentences and 513 English sentences were collected. Descriptive analysis of sentence length revealed that the mean length of collected reference-containing sentences is 36 words in English and 25 words in German. This is significantly higher than in the synthetically generated dataset, where the mean sentence length is 17 words for English and 16 words for German.

The next step is to assign classification labels to the clean sentences. Sentences are classified according to the following criteria:

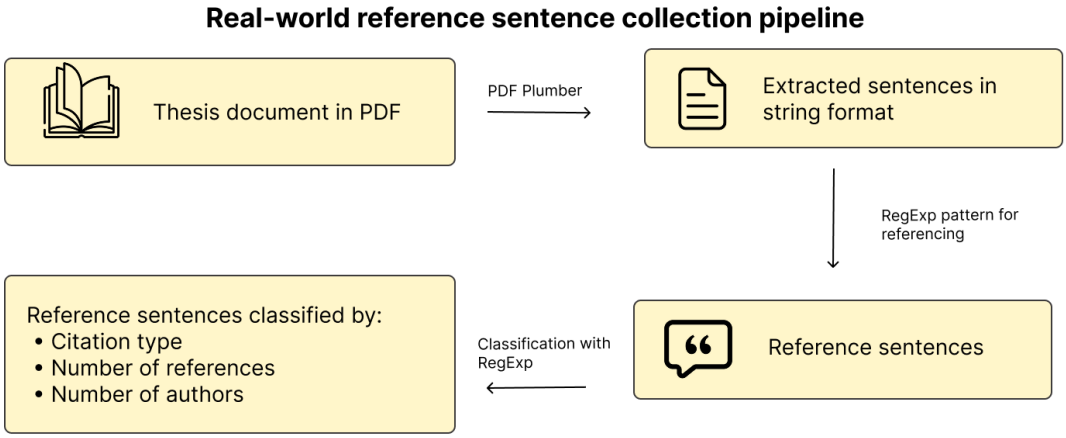

Figure 12.: Real-world reference sentence collection pipeline

- citation type: narrative or parenthetical
- number of sources – one, two, or multiple sources in a single sentence
- number of authors – one, two, or multiple authors in a single source
- Inclusion of page numbers – whether page numbers are specified in the citation

Each collected sentence was classified into these four categories using regular expressions.

The citation type can be narrative or parenthetical. In narrative citation, the author's name is a part of a sentence, such as *Pöchhacker (2004) argues that translation is a complex process.* In parenthetical citation, the reference with author name and year appears in parenthesis and is not integrated into the sentence, such as *Several studies have addressed this issue (Pym 2019; Risku 1995).*

Every sentence can be classified according to the the number of sources it cites: it may include a single source, such as *The study focused on these aspects (Baker, 2006).* or multiple sources, such as *Translation theories have been explored by Baker (2020), House (2007), and Venuti (2008).*

Additionally, a sentence can be classified based on the number of authors within a single reference: one author, such as *Pöchhacker (2002) argues that translation is an evolving field*, two authors, such as *In their study, Kaindl and Pöchhacker (2003) explore the relationship between language and culture*, or multiple authors, such as *Pym et al. (2012) discuss translation theory in depth.*

Finally, sentences can be classified according to the presence or absence of page numbers within their references.

Similarly, each error type from the error category table was assigned a classification label according to this system. In addition, classification categories for error types can include the tag 'ANY', indicating that any type of sentence is acceptable for this case. This classification is necessary to automatically align each sentence with the error types suitable for corruption, as shown in Algorithm 1.

Algorithm 1 Aligning and Corrupting Algorithm of Extracted Sentences Based on Error Types

Require: List of extracted sentences S, error type table T

for each error type t in T **do**
  for each extracted sentence s in S **do**
    if (sentence_type(t) = sentence_type(s) **or** sentence_type(t) = ANY)
  and
(num_authors(t) = num_authors(s) **or** num_authors(t) = ANY)
  and
(num_sources(t) = num_sources(s) **or** num_sources(t) = ANY)
  and
(with_page(t) = with_page(s) **or** with_page(t) = ANY) **then**
      perform_corruption(s, t)
    else
      return "Sentence not assigned a category"
    end if
  end for
end for

If the collected sentences match the error type by citation type, the number of sources, and the number of authors, they can be used for corruption. The input for the prompt structure used for error corruption with the LLM and the expected output are illustrated in Example (3).

(3) **Clean Source (X):** Finally, in 2016 neural networks were applied to machine translation **(Wu et al. 2016)**.
Error Type: ETAL
One-shot Example: The relationship between language and culture was widely discussed **(Pym et al. 2012).** → The relationship between language and culture was widely discussed **(Pym, Risku, Shlesinger, 2012)**.
Corrupted Output $\hat{X}$: Finally, in 2016 neural networks were applied to machine translation **(Wu, Zhang, Johnson 2016)**.

For the corruption process, only a subset of error types was selected. Specifically, this selection included errors in citation type choice (TYPE2, TYPE4), incorrect

citation of two authors (TWO2, TWO4), addition of redundant entities (RED1, RED2, RED3, RED4, RED5), and incorrect citation placement (PLC1). All these categories are detailed in the main error table presented in Appendix A. The selection primarily included error types that require greater manipulation of the sentence structure. Because real-world sentences tend to be structurally complex, including error types involving structural changes during corruption can help the model better adapt to varied sentence structures.

The sentences were collected separately for English and German. For each error type, 200 sentences were randomly selected from the collected dataset and corrupted, resulting in a corrupted dataset of 2,000 parallel sentences for each language. The corrupted dataset with a size of 2,000 items was concatenated with the synthetic dataset of 2,300 items, resulting in a dataset of 4,300 elements.

7.4. Real-word gold test dataset

The gold-standard test dataset consists of 104 examples of real students' referencing errors, which were selected and manually corrected by an actual professor and supervisor of master's theses Dagmar Gromann. The dataset includes the erroneous sentences and their corrected versions, the language, the error type, and a label indicating whether the error was present in the test set. The dataset comprises 104 sentences: 61 in English and 43 in German.

However, 14 examples were excluded from the final evaluation phase, as they included cases the model could not properly correct using sentence context. These cases primarily involved the MISS error type, where the model had to complete the sentence by adding a year or other missing reference elements. However, without access to the correct year as external knowledge, the model was unable to make the necessary correction. Additionally, the URL error category, which concerns the correction of web link referencing, was excluded, as none of the referencing types used for training addressed this case. After excluding these examples, the final test set comprised 90 sentences: 55 in English and 35 in German.

The dataset includes various types of referencing errors. As shown in Figure 13, the error distribution differs between English and German. For German, the most frequently occurring error types are TWO, RED, and ICO, whereas for English, the most common types are TYPE, PUN, and ICO.

Additionally, as mentioned in the Error Types creation section, real-world English examples were analyzed to define the error types. For example, error types RED3, RED4, and RED5 were developed based on the analysis of real-world referencing errors. However, the German test set was not included in the development of the Error Types specification table. Therefore, several error categories occur exclusively in the German gold test dataset. This enables testing the generalization capabilities

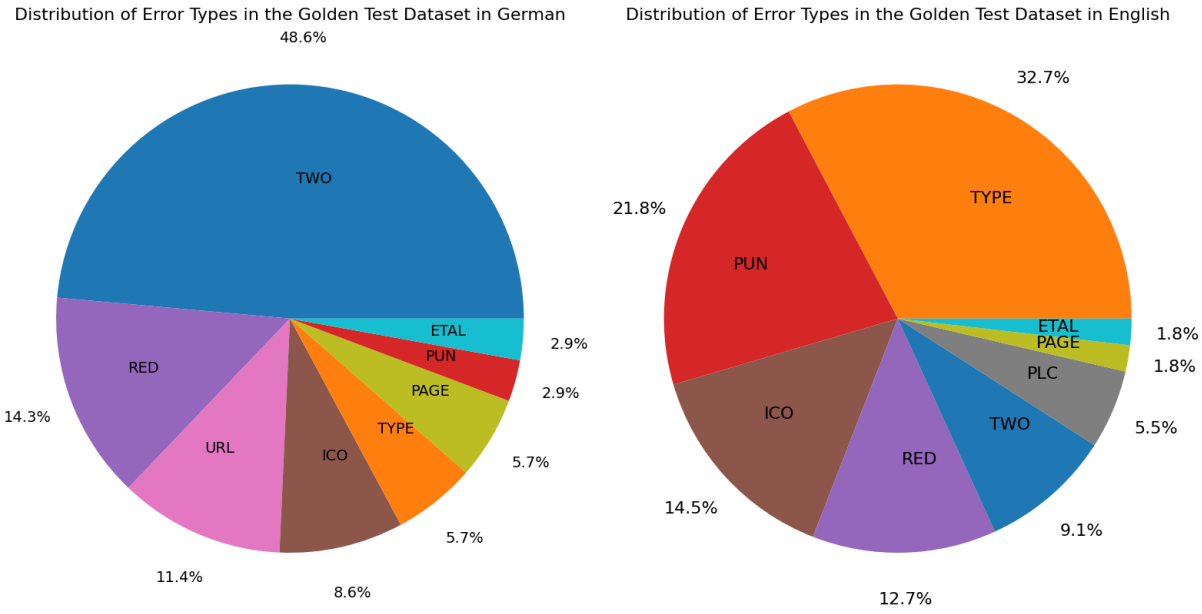

Figure 13.: Illustration of the distribution of error types on Gold Test set

of the Reference Error Correction model on unseen error categories.

7.5. Models

The models selected for AEC in referencing are the base T5 model, the multilingual variant mT5, and the instruction-tuned model FLAN-T5. T5 is a sequence-to-sequence model that is widely used in addressing GEC tasks, achieving SOTA performance. Although this model was not pre-trained on instruction-based tasks, it is capable of accepting a prefix tag or a full sentence as input. However, this model was not specifically pre-trained on datasets with instructions. Notably, in related work, the dataset sizes used for training T5 models was extremely large. This study aims to compare T5 and FLAN-T5 under conditions of limited training data, specifically with datasets of up to 10,000 examples. In contrast with T5, the FLAN-T5 has demonstrated the abilities of models to effectively perform edits even when fine-tuned on smaller amounts of data. The mT5 model was used to train models to perform corrections in German and to evaluate the cross-lingual editing capabilities, specifically to test whether models trained in English can correct German sentences and vice versa.

The model was trained using the Kaggle environment with access to two high-performance GPUs (2xGPU400). For FLAN-T5, T5 and mT5 the following hyperparameters were used: a learning rate of 2e-5, a per-device training batch size

of 4, and 5 training epochs. As FLAN-T5 can accept instruction in addition to the input, such as: *Fix the error in academic referencing in the following sentence, but keep the rest of the sentence unchanged except for the references.* The FLAN-T5, T5 and mT5 models were all used in their base versions with 200M parameters.

7.6. Experiments

This section describes the experiments conducted in the study. In the first experiment 7.6.1, T5 and FLAN-T5 were fine-tuned on a synthetically generated English dataset and evaluated on an unseen synthetic English test set. This experiment shows how models can correct referencing errors and enables analysis of the error categories that are most challenging for the models. However, since the test data was generated using the same strategy as the training data, this experiment does not demonstrate the model's ability to correct real academic sentences. In contrast, the second experiment 7.6.2 explores how the T5 model fine-tuned on synthetically generated data can perform corrections on manually collected real-world erroneous sentences. Additionally, this experiment examined the impact of model fine-tuning on a combination of synthetic and corrupted datasets compared to fine-tuning exclusively on a synthetic dataset. Finally, the third experiment 7.6.3 explores the editing capabilities of the mT5 model trained on German data to perform corrections. This experiment also investigates the editing capabilities of the model on unseen error categories.

7.6.1. Experiment 1: Synthetic test set evaluation

In this experiment, T5 and FLAN-T5 were evaluated for their overall capability to perform referencing error corrections. For this, both T5 and FLAN-T5 models were trained on synthetically generated sentence pairs, as described in Section 7.3.1. A 10 % validation split was applied to this dataset, resulting 2,070 sentences in the training set and 230 sentences in the validation set. The evaluation was conducted on the synthetic test set comprising 94 sentences with various error types. The test set was not used during model training and was separately created to assess the ability of models to perform error corrections.

For FLAN-T5, the following instruction prompt was used during both training and inference: *Fix the error in academic referencing in the following sentence, but keep the rest of the sentence unchanged except for the references.* For the T5 model, no instruction prefix was used during either the training or inference stages.

The goal of this experiment is to evaluate and compare the performance of the T5 model and the instruction-tuned FLAN-T5 model, and to assess the value of instruction for this type of task. Additionally, the experiment aims to analyze

which types of errors are more challenging for the models to correct in order to determine limitations and potential areas for improvement.

7.6.2. Experiment 2: English real-world evaluation

This experiment explores the ability of the T5 model trained on synthetic data to perform error correction on the real-world gold dataset. Specifically, this experiment compares models fine-tuned only on synthetically generated data and synthetic data mixed with synthetically corrupted real-word sentences approaches.

The REF-SYNT-EN model was fine-tuned on a synthetically generated dataset, with both source and erroneous sentences produced by an LLM, as described in Section 7.3.1. The dataset consisted of 2,300 items. A 10 % validation split was applied, resulting 2,070 sentences in the training set and 230 sentences in the validation set.

The REF-CORR-EN model was fine-tuned on a mixed dataset of 4,300 items. Applying a 10 % validation split resulted in 3,870 training sentences and 430 validation sentences. This dataset alongside synthetic data, also includes synthetically corrupted real-world sentences, as described in Section 7.3.2. This combination was designed to enhance the model ability to perform the correction in real-world error correction tasks. As real-world sentences are typically longer and often include multiple references compared to synthetically generated ones, the inclusion of synthetically corrupted sentences may improve the ability of the model to handle complex referencing structures.

The evaluation of both models in this experiment was conducted on a manually collected dataset comprising 55 paired sentences in English, detailed described in the Section 7.4. This experiment aims to explore how well models trained on synthetic data can perform real-world corrections, evaluated using the $F_{0.5}$ score, precision, and recall. Additionally, it investigates whether including synthetically corrupted sentences, in addition to synthetically generated sentences, can enhance the editing performance of the model.

7.6.3. Experiment 3: German real-world evaluation

This experiment investigates the ability of the mT5 model, trained on German synthetic data, to perform real-world corrections in German. Similar to the previous experiment, this experiment compared two models: REF-SYNT-DE, fine-tuned on synthetic data with both sentences synthetically generated, and REF-CORR-DE, fine-tuned on a combination of synthetic data and synthetically corrupted real-world sentences. However, a key difference is that the German test set was not analyzed for developing the Error Type table, based on error tags of which both synthetically generated and synthetically corrupted train datasets were produced.

Therefore, the model training data did not include sentences generated based on certain error types found exclusively in the gold test dataset. For example, the error type RED-NAME, in which the the author's given name is redundantly included in the reference, was not represented in the model's training dataset. This experiment explores the generalization abilities of the model to perform the corrections to unseen error types.

7.7. Evaluation

For all experiments, the selected evaluation metrics are precision, recall, and the $F_{0.5}$ score. The process of calculating these metrics is described in detail in Chapter 5. The first experiment was evaluated on a synthetically generated subset comprising 94 sentences. The second and third experiments were evaluated on real-world collected sentences with professional corrections: 55 English sentences and 35 German sentences, respectively. In addition to evaluation metrics, manual inspection was conducted to showcase examples of successful corrections and failures. As each erroneous sentence was labeled with an error category, the study additionally examined which error categories were most challenging for the models.

8. Results

This chapter presents the results of the experiments, specifically precision, recall, and $F_{0.5}$ score. Additionally, the chapter includes a manual analysis of model corrections to provide a more detailed insight into the most challenging cases. Section 8.1 presents the results of the first experiment, comparing T5 and FLAN-T5 models both trained and evaluated on synthetic datasets. Sections 8.2 and 8.3 present the results of the second and third experiments, evaluating T5 on English and mT5 on German, respectively, using real-world sentences. The results of these experiments compare the performance of models fine-tuned with different synthetic data generation strategies.

8.1. Synthetic test set evaluation

As described in Section 7.6.1, the first experiment evaluates the editing capabilities of the FLAN-T5 and T5-base models using a synthetic test subset. The comparison results for the FLAN-T5 and T5-base models in referencing error correction are presented in Table 9.

Table 9.: A comparative analysis of the T5 and FLAN-T5 models trained on synthetic data using a synthetic test set.

Model Name	**Precision**	**Recall**	$F_{0.5}$
FLAN-T5-REF-SYNT	93.33	91.50	92.96
T5-REF-SYNT	92.57	89.54	91.95

The results demonstrated that T5 without instructions achieved performance comparable to FLAN-T5 with explicit instructions. Both models consistently performed the required corrections while leaving the rest of the sentence unchanged. These findings indicate that in this experimental setup the use of uniform instructions for the entire dataset does not significantly influence model performance.

Notably, because both models share a similar architecture, they encountered difficulties with similar categories of errors. Specifically, the most challenging error type for both FLAN-T5 and T5 was the ICO tag, which involves incorrect citation order. Models occasionally performed incorrect reorderings, as demonstrated in Example (1).

(1) *T5 model error correction ICO failure*
Wrong sentence: "The integration of digital tools in language learning has been extensively explored by various scholars **(Gould 2017; Armijo 2018; Becraft 2020)**."
Reference: "The integration of digital tools in language learning has been extensively explored by various scholars **(Armijo 2018; Becraft 2020; Gould 2017)**."
Prediction: "The integration of digital tools in language learning has been extensively explored by various scholars **(Armijo 2018; Gould 2017; Becraft 2020)**."

Additionally, sometimes, the models faced unexpected surname hallucinations. When attempting to perform the desired correction, it replaced the existing author surname with a new one, as shown in Example (2).

(2) *T5 model error correction failure*
Wrong sentence: "Krohn in 1963 emphasizes the importance of named entity recognition in the analysis of legal texts **(Krohn, 1963)**."
Reference: "**Krohn (1963)** emphasizes the importance of named entity recognition in the analysis of legal texts".
Prediction: "**Krumholz (1963)** emphasizes the importance of named entity recognition in the analysis of legal texts".

Specifically, the model frequently has faced hallucinations in cases of surname reordering, as shown in Example (3).

(3) *T5 model error correction failure*
Wrong sentence: "The development of academic writing skills in English for Academic Purposes (EAP) is essential for non-native speakers to succeed in higher education (**Floyd 1839**; Kemerer 1936; Coronado 2009)."
Reference: "The development of academic writing skills in English for Academic Purposes (EAP) is essential for non-native speakers to succeed in higher education (Coronado 2009; **Floyd 1839**; Kemerer 1936)."
Prediction: "The development of academic writing skills in English for Academic Purposes (EAP) is essential for non-native speakers to succeed in higher education (Corondo 2009; **Flayd 1839**; Kemerer 1936)."

8.2. Real-word test set evaluation on English

In this experiment, two models were evaluated on a manually constructed real-world dataset of student errors in English, which consisted of 55 sentences. The

T5-REF-SYNT model was trained solely on the synthetic dataset, while T5-REF-CORRUPT was trained on both synthetic and real-world corrupted data. The evaluation metrics are presented in the Table 10.

Table 10.: Evaluation results on real-world English test set

Model Name	**Precision**	**Recall**	$F_{0.5}$
T5-REF-SYNT	53.03	33.33	47.43
T5-REF-CORRUPT	75.86	62.86	72.85

The results show that the T5-REF-CORRUPT achieved up to 20 $F_{0.5}$ points more than T5-REF-SYNT. This demonstrates that augmenting synthetic data with real-world corrupted sentences allows the model to significantly improve the ability to perform accurate referencing corrections on real-word settings. It is interesting to see that the different fine-tuned models seem to struggle most with recall, indicating that the detection of erroneous sequences remains to be challenging. In general, there still is room for improvement for the overall performance, as with approximately 70 $F_{0.5}$ points a substantial proportion of examples are not correctly handled.

The T5-REF-SYNT-EN model, trained only on synthetic data, consistently struggles to fix errors in long sentences, especially when multiple corrections are required, as shown in the Example (4). This failure may be connected to a limitation in the synthetic dataset, where each generated sentence contained only one reference and, consequently, only required a single correction.

(4) *Analysis of the corrections of T5-REF-SYNT-EN*
Wrong Sentence: "In recent years, information retrieval (IR) approaches have been continuously researched and improved, through the development of toolkits such as **Pyserini Lin et al. (2021)**, adapted through strategies such as inverted file structures **Zhang et al. (2023a)** or data augmentation **Lin et al. (2023)**" (Anonymized Student Submission).
Reference: "In recent years, information retrieval (IR) approaches have been continuously researched and improved, through the development of toolkits such as Pyserini **(Lin et al. 2021)**, adapted through strategies such as inverted file structures **(Zhang et al. 2023a)** or data augmentation **(Lin et al. 2023)**".
Prediction: "In recent years, information retrieval (IR) approaches have been continuously researched and improved, through the development of toolkits such as **Pyserini Lin et al. (2021)**, adapted through strategies such as inverted file structures **Zhang et al. (2023a)** or data augmentation **Lin et al. (2023)**".

Additionally, the T5-REF-SYNT-EN model often struggled to correct citations that contained page numbers. Specifically, the model, when attempting to correct certain errors, performed the intended correction, but occasionally failed to omit page numbers or misplaced parentheses, as shown in Example (5).

(5) *Analysis of the corrections of REF-SYNT-EN*
Wrong Sentence: "As **Olohan** states, the rather close relation between the two can be perceived and described in a multitude of ways **(2016: 7)**" (Anonymized Student Submission).
Reference: "As **Olohan (2016: 7)** states, the rather close relation between the two can be perceived and described in a multitude of ways" (Expert Correction of Anonymized Student Submission).
Prediction : "As **Olohan (2016)** states, the rather close relation between the two can be perceived and described in a multitude of ways".

Wrong Sentence: "**In 2006, (Snover et al. 2006: 225)** introduced a new, more intuitive automatic measure for evaluating MT output that is based on the amount of editing a human would have to perform in order to fix the output so that it exactly matches a human reference translation" (Anonymized Student Submission).
Reference: "**Snover et al. (2006: 225)** introduced a new, more intuitive automatic measure for evaluating MT output that is based on the amount of editing a human would have to perform in order to fix the output so that it exactly matches a human reference translation".
Prediction : "**Snover et al. (2006): 225)** introduced a new, more intuitive automatic measure for evaluating MT output that is based on the amount of editing a human would have to perform in order to fix the output so that it exactly matches a human reference translation".

This limitation may be connected to the fact that the synthetically generated instances for the RED and TYPE error categories did not include page numbers. This has prevented the model from learning to properly performing corrections for instances with pages.

An additional challenge encountered by the T5-REF-SYNT-EN model involved correcting sentences that began with introductory phrases, such as *For exmample* or *However*. Although the model correctly addressed specific referencing errors, such as removing redundant author and year information, it often incorrectly removed these introductory phrases.

Additionally, the T5-REF-SYNT-EN model often struggled to implement the intended corrections, frequently leaving the sentences unchanged. Furthermore, the

model failed to correct previously unseen typographical errors, such as *Vieira et all. (2021)*, where the source sentence contained an extra *l*.

The T5-REF-CORRUPT-EN model demonstrated substantially improved performance, compared to the T5-REF-SYNT-EN. The model achieved nearly twice the scores of precision, recall, and $F_{0.5}$ score. The model performed more accurate corrections. Compared to the T5-REF-SYNT-EN model, T5-REF-CORRUPT-EN handled longer sentences more effectively. However, in longer sequences with four or more references, the model occasionally applied only partial corrections instead of correcting all errors, as shown in the Example (6).

(6) *Analysis of the corrections of REF-CORRUPT-EN*
Wrong Sentence: "However, the common ground of these approaches and methods is that even though they dominate the research field by setting new standards and benchmarks, they are usually only tested on English datasets, such as Natural Questions **Kwiatkowski et al. (2019)**, BEIR **Thakur et al. (2021)**, LoTTE **Santhanam et al. (2021)**, GooAQ **Khashabi et al. (2021)** and most prominently MS MARCO **Bajaj et al. (2016)**." (Anonymized Student Submission).
Reference: "However, the common ground of these approaches and methods is that even though they dominate the research field by setting new standards and benchmarks, they are usually only tested on English datasets, such as Natural Questions **(Kwiatkowski et al. 2019)**, BEIR **(Thakur et al. 2021)**, LoTTE **(Santhanam et al. 2021)**, GooAQ **(Khashabi et al. 2021)** and most prominently MS MARCO (Bajaj et al. 2016)."
Prediction : "However, the common ground of these approaches and methods is that even though they dominate the research field by setting new standards and benchmarks, they are usually only tested on English datasets, such as Natural Questions **(Kwiatkowski et al. 2019)**, BEIR **Thakur et al. (2021)**, LoTTE **Santhanam et al. (2021)**, GooAQ **Khashabi et al. (2021)** and most prominently MS MARCO **(Bajaj et al. 2016)**."

Additionally, the model was more effective in handling cases that posed difficulties for the previous model, particularly when T5-REF-SYNT-EN mistakenly omitted page numbers during corrections. As shown in Example (7), the T5-REF-CORRUPT-EN successfully handled these cases, correcting the errors and preserving critical reference details such as page numbers during corrections.

(7) *Analysis of the corrections of T5-REF-CORRUPT-EN*
Wrong Sentence: "As **Olohan** states, the rather close relation between the two can be perceived and described in a multitude of ways **(2016: 7)**"

(Anonymized Student Submission).
Reference: "As **Olohan (2016: 7)** states, the rather close relation between the two can be perceived and described in a multitude of ways."
Prediction : "As **As Olohan (2016: 7)** states, the rather close relation between the two can be perceived and described in a multitude of ways."

Wrong Sentence: "**In 2006, (Snover et al. 2006: 225)** introduced a new, more intuitive automatic measure for evaluating MT output that is based on the amount of editing a human would have to perform in order to fix the output so that it exactly matches a human reference translation." (Anonymized Student Submission).
Reference: "**Snover et al. (2006: 225)** introduced a new, more intuitive automatic measure for evaluating MT output that is based on the amount of editing a human would have to perform in order to fix the output so that it exactly matches a human reference translation."
Prediction : "**Snover et al. (2006: 225)** introduced a new, more intuitive automatic measure for evaluating MT output that is based on the amount of editing a human would have to perform in order to fix the output so that it exactly matches a human reference translation."

Additionally, the model encounters less frequent issues with the redundant removal of introductory phrases, such as *However*, as shown in Example (8).

(8) *Analysis of the corrections of REF-CORRUPT-EN*
Wrong Sentence: "For example, **Zheng and Ye (2009)** achieved high accuracy when applying SVM to classify sentiment in travel reviews **(Zheng and Ye 2009)**." (Anonymized Student Submission).
Reference: "For example, **Zheng and Ye (2009)** achieved high accuracy when applying SVM to classify sentiment in travel reviews."
Prediction: "For example, **Zheng and Ye (2009)** achieved high accuracy when applying SVM to classify sentiment in travel reviews."

Wrong Sentence: "However, **(Lichtarge et al., 2019)** notices that errors generated errors generated through backtranslation or round-trip translation are cleaner and less random than natural human mistakes." (Anonymized Student Submission).
Reference: "However, **Lichtarge et al. (2019)** notices that errors generated errors generated through backtranslation or round-trip translation are cleaner and less random than natural human mistakes."

Prediction: "However, **Lichtarge et al. (2019)** notices that errors generated errors generated through backtranslation or round-trip translation are cleaner and less random than natural human mistakes."

The model consistently produced accurate corrections. However, correcting the author order remained challenging. In some instances, it reordered authors alphabetically as required, but in others, reordered author names wrongly.

8.3. German synthetic and real-world test set

In this experiment, the mT5 model was fine-tuned on synthetic data MT5-REF-SYNT-DE and synthetic data with corrupted data MT5-CORRUPT-SYNT-DE. The results on the German real-world test set consisting of 35 sentences are shown in Table 11.

Table 11.: Evaluation results on real-world German test set

Model Name	**Precision**	**Recall**	$F_{0.5}$
mT5-REF-SYNT	26.09	25.00	25.86
mT5-REF-CORRUPT	51.35	36.54	47.50

The results showed that mT5-REF-CORRUPT considerably outperformed mT5-REF-SYNT by 20 $F_{0.5}$ points. This finding supports the claim that longer and more realistic examples in training data are crucial for this task, even across languages.

However, both mT5-REF-SYNT and mT5-REF-CORRUPT achieved notably lower metrics than the previous models trained on English datasets. One reason for this is that the German dataset contained many instances of error types that were new to the model. One such error is the redundant inclusion of the author's name or the title of the academic work, as shown in Example (9).

(9) *mT5-REF-CORRUPT error correction failure*
Wrong Sentence: "**Markus Ramlow (2009)** befasste sich in seiner Arbeit ‚Die maschinelle Simulierbarkeit des Humanübersetzens' mit verschiedenen Aspekten der maschinellen Übersetzung." (Anonymized Student Submission).
Reference: "**Ramlow (2009)** befasste sich mit verschiedenen Aspekten der maschinellen Übersetzung."
Prediction: "**Markus Ramlow (2009)** befasste sich in seiner Arbeit ‚Die maschinelle Simulierbarkeit des Humanübersetzens' mit verschiedenen Aspekten der maschinellen Übersetzung."

Additionally, the other new error type was the mentioning author in the text, as shown in Example (10).

(10) *mT5-REF-CORRUPT error correction failure*
Wrong Sentence: "Die Rechtssprache ist eng mit der Rechtsordnung eines Staates verbunden, die wiederum die Struktur der Gesellschaft widerspiegelt, so betont **Griebel (2013: 152)**" (Anonymized Student Submission).
Reference: "Die Rechtssprache ist eng mit der Rechtsordnung eines Staates verbunden, die wiederum die Struktur der Gesellschaft widerspiegelt **(Griebel 2013: 152)**."
Prediction: "Die Rechtssprache ist eng mit der Rechtsordnung eines Staates verbunden, die wiederum die Struktur der Gesellschaft widerspiegelt."

This highlights the importance of including a diverse set of error categories and types, as the model has limited generalization capabilities for unseen error types. Additionally, the case of incorrect citation order was still very complicated for the model, when the model needed to order author mentions alphabetically instead of chronologically. Similar to the results demonstrated in the previous experiment, the mT5-REF-CORRUPT model, whose training set included not only synthetically generated sentences but also synthetically corrupted real-world sentences, achieved higher performance. The mT5-REF-CORRUPT could better perform correction in more complicated sentence and referencing structures, as shown in the Example (11).

(11) *mT5-REF-CORRUPT and and mT5-REF-SYNT error correction failure*
Wrong Sentence: "**Liu & Zhang (2015)** sowie **Stein (2009)** bieten eine ausführliche Übersicht über die unterschiedlichen Ansätze und Systeme der maschinellen Übersetzung" (Anonymized Student Submission).
Reference: "**Liu und Zhang (2015)** sowie **Stein (2009)** bieten eine ausführliche Übersicht über die unterschiedlichen Ansätze und Systeme der maschinellen Übersetzung."
mT5-REF-SYNT Prediction: "**Liu und Zhang (2015)** bieten eine ausführliche Übersicht über die unterschiedlichen Ansätze und Systeme der maschinellen Übersetzung" Model Correction of Anonymized Student Submission).
mT5-REF-CORRUPT Prediction: "**Liu und Zhang (2015)** sowie **Stein (2009)** bieten eine ausführliche Übersicht über die unterschiedlichen Ansätze und Systeme der maschinellen Übersetzung."

Wrong Sentence: "Sowohl **Burchard & Harris (2017)** als auch **Läubli & Orrego-Carmona (2017)** klagen nämlich über den Mangel an Kooperation zwischen ForscherInnen, EntwicklerInnen und BenutzerInnen, was zu Frustration und Skepsis auf beiden Seiten führe" (Anonymized Student Submission).
Reference: "Sowohl **Burchard und Harris (2017)** als auch **Läubli und Orrego-Carmona (2017)** klagen nämlich über den Mangel an Kooperation zwischen ForscherInnen, EntwicklerInnen und BenutzerInnen, was zu Frustration und Skepsis auf beiden Seiten führe." (Expert Correction of Anonymized Student Submission).
mT5-REF-SYNT Prediction: "**Burchard und Harris (2017)** klagen nämlich über den Mangel an Kooperation zwischen ForscherInnen, EntwicklerInnen und BenutzerInnen, was zu Frustration und Skepsis führe."
mT5-REF-CORRUPT Prediction: "Sowohl **Burchard und Harris (2017)** als auch Läubli und **Orrego-Carmona (2017)** klagen nämlich über den Mangel an Kooperation zwischen ForscherInnen, EntwicklerInnen und BenutzerInnen, was zu Frustration und Skepsis auf beiden Seiten führe."

9. Discussion

The experimental results demonstrate that sequence-to-sequence models trained on synthetically generated data can effectively perform referencing error corrections. The synthetic data generation approach opens up enormous opportunities to address GEC tasks in low-resource domains or settings. However, as noted in previous research, LLMs tend to perform under-correction or, in opposite, overcorrection of the sentences which highlighted the importance of addition of control of LLM, for example with the error tags. In contrast to previous LLM research on synthetic GEC corpus creation (Deng et al., 2025; Li and Lan, 2025), this study investigated both the synthetic corruption of collected sentences and the generation of new clean and erroneous sentences. The research revealed that the sentence generation pipeline requires more control than error corruption. Apart from introducing the error tag, one-shot example, and rule description, the error generation pipeline additionally requires the random generation of numerical instances, such as page numbers and years, surnames and topics, to provide enough variety to the dataset. However, even with this approach, the generated sentences lacked the length and structural complexity in comparison to real-world academic sentences. Nevertheless, the model trained on this synthetically generated dataset was able to correct errors, demonstrating the potential of this approach for developing effective error correction systems. One of the major limitations of this approach is the syntactical simplicity of generated errors and the overall resemblance among the generated sentences. This limitation was addressed in the synthetically corrupted dataset, where sentences collected from master's theses were augmented using an LLM controlled by error tags. This approach facilitates the creation of a dataset that more closely mirrors the complexity and characteristics of real-world academic settings.

The first experiment demonstrated that both FLAN-T5-REF-SYNT-EN and T5-REF-SYNT-EN were capable of correcting referencing errors and showed strong performance when evaluated on a synthetically generated test set. However, real-world sentences have a more complex structure than synthetically generated sentences. Specifically, as demonstrated by the descriptive analysis, real-world sentences are significantly longer than synthetic sentences. Consequently, the second experiment revealed that T5-REF-SYNT-EN achieved substantially lower performance in the evaluation on the gold-standard real-world test dataset. However, the incorporation of synthetically corrupted sentences in the training data enhanced editing capabilities of T5-REF-CORRUPT-EN. This was especially not-

able on longer and more structurally diverse sentences or sentences that included complex references with multiple authors or contained page information. In the German setting, the combination of both types of training data also considerably improved the editing performance of the model. However, this approach requires the development of a sentence classification system. Since not every error type can be introduced in every sentence, both the sentence and the error tag must be assigned to appropriate categories. Only in case if these categories match, the correction can be applied. However, this introduces additional complexity to the approach, potentially increasing the implementation effort required for scalable and systematic error corruption.

The best performance was achieved by the T5-REF-CORRUPT-EN model, which was trained on a combination of two types of datasets: synthetically generated and synthetically corrupted data. The model achieved an $F_{0.5}$ score of 72.85, which is considered a strong result in the GEC research. For example, the T5 model trained to check the answers of Russian language learners on grammatical exercises achieved an $F_{0.5}$ score of 52.1 (Katinskaia and Yangarber, 2023), while the T5-base models trained by Rothe et al. (2021) achieved scores of 54.10 for English and 71.88 for Czech.

However, the addition of real-world synthetically corrupted sentences improved only the model's ability to perform error correction on more complicated structures, but did not mitigate the issue of unseen error types. The nature of referencing errors can be highly diverse. Although the referencing error table includes 46 error types, some error types were still underrepresented in the training dataset. The third experiment on real-world German examples has demonstrated that models have low generalization abilities for unseen error types. If the model has never seen an error type, such as redundantly omitting the given name of the author in the referencing, it will not perform the desired correction. This highlights the importance of generating a diverse error table specification as a crucial component of efficient GEC model training.

While several experiments have been conducted to evaluate the effectiveness of dataset creation methods and GEC potentials of models in real-world settings, the proposed experiments are still limited in size and linguistic coverage. They represent a first step towards testing the feasibility of models in custom settings, such as institutional guidelines. However, this experiment used small datasets and was limited in architectural choice, relying solely on variations of the T5 model. In addition, the study explored and compared only two data generation strategies. Employing larger datasets, exploring diverse data generation techniques, and expanding the range of model architectures could provide valuable insights in future research.

10. Conclusion

This thesis has demonstrated that sequence-to-sequence models trained on synthetic data can effectively perform real-world error correction in referencing. In particular, the study investigated the role of different synthetic data generation methods for model training. Although models trained solely on synthetic data perform well on similarly structured test sets, their performance drops when they faced with real-world sentences. However, the incorporation of synthetically corrupted sentences into the training data improved the model's ability to edit real-word sentences that usually have more complex referencing structures. The model fine-tuned on a combination of synthetic and corrupted datasets achieved the best results across the English and German settings, demonstrating that this can be an effective strategy for fine-tuning GEC models.

The addition of real-world corrupted sentences to the training stage was especially valuable for performing corrections of longer real sentences that include multiple authors or page information. Nonetheless, this approach did not fully address the challenge of unseen error types, as the generalization capabilities of the models remained limited. The findings demonstrate that the diversity of error categories and types is a critical factor for robust model performance.

This work contributes to the field of GEC by providing new insights into the development of synthetic data for training of GEC models specifically for referencing error correction. The findings also highlight current limitations in the generalization capability of Seq2seq models for referencing error correction. Error types that were not included in the error type specification were particularly challenging for the models to correct. This points to a direction for future research, such as expanding error type coverage and improve generalization of models to unseen errors.

Bibliography

Dimitris Alikaniotis and Vipul Raheja (2019). 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. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). Layer normalization. *CoRR*, abs/1607.06450.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi (2017). Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Adriane Boyd (2018). Using Wikipedia edits in low resource grammatical error correction. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 79–84, Brussels, Belgium. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, and et al. (2020). Language models are few-shot learners. *CoRR*, abs/2005.14165.

Christopher Bryant, Mariano Felice, and Ted Briscoe (2017). Automatic annotation and evaluation of error types for grammatical error correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe (2023). Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, pages 643–701.

Nikhil Buduma, Nithin Buduma, and Joe Papa (2022). *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*. O'Reilly Media.

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

Daniel Dahlmeier, Hwee Tou Ng, and Siew Mei Wu (2013). Building a large annotated corpus of learner English: The NUS corpus of learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 22–31, Atlanta, Georgia. Association for Computational Linguistics.

Jiayi Deng, Chen Chen, Chunyan Hou, and Xiaojie Yuan (2025). InstructGEC: Enhancing unsupervised grammatical error correction with instruction tuning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 110–122, Abu Dhabi, UAE. Association for Computational Linguistics.

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

Wanyu Du, Vipul Raheja, Dhruv Kumar, Zae Myung Kim, Melissa Lopez, and Dongyeop Kang (2022). Understanding iterative revision from human-written text. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3573–3590, Dublin, Ireland. Association for Computational Linguistics.

Felix Faltings, Michel Galley, Gerold Hintz, Chris Brockett, Chris Quirk, Jianfeng Gao, and Bill Dolan (2021). Text editing by command. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5259–5274, Online. Association for Computational Linguistics.

Simon Flachs, Felix Stahlberg, and Shankar Kumar (2021). Data strategies for low-resource grammatical error correction. In *Proceedings of the 16th Workshop*

on Innovative Use of NLP for Building Educational Applications, pages 117–122, Online. Association for Computational Linguistics.

David Foster (2023). *Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*, 2nd edition. O'Reilly Media.

Roman Grundkiewicz and Marcin Junczys-Dowmunt (2014). The wiked error corpus: A corpus of corrective wikipedia edits and its application to grammatical error correction. In *Advances in Natural Language Processing*, pages 478–490, Cham. Springer International Publishing.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield (2019). Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield (2018). 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. Association for Computational Linguistics.

Daniel Jurafsky and James H. Martin (2025). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 3rd edition. Online manuscript released January 12, 2025.

Ting-Hui Kao, Yu-Wei Chang, Hsun-Wen Chiu, Tzu-Hsi Yen, Joanne Boisson, Jian-Cheng Wu, and Jason S. Chang (2013). CoNLL-2013 shared task: Grammatical error correction NTHU system description. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 20–25, Sofia, Bulgaria. Association for Computational Linguistics.

Atakan Kara, Farrin Marouf Sofian, Andrew Bond, and Gözde Şahin (2023). GECTurk: Grammatical error correction and detection dataset for Turkish. In *Findings of the Association for Computational Linguistics: IJCNLP-AACL 2023 (Findings)*, pages 278–290, Nusa Dua, Bali. Association for Computational Linguistics.

Anisia Katinskaia and Roman Yangarber (2023). Grammatical error correction for sentence-level assessment in language learning. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 488–502, Toronto, Canada. Association for Computational Linguistics.

Anisia Katinskaia and Roman Yangarber (2024). GPT-3.5 for grammatical error correction. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 7831–7843, Torino, Italia. ELRA and ICCL.

Yova Kementchedjhieva and Anders Søgaard (2023). Grammatical error correction through round-trip machine translation. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2208–2215, Dubrovnik, Croatia. Association for Computational Linguistics.

Mark D. Kernighan, Kenneth W. Church, and William A. Gale (1990). A spelling correction program based on a noisy channel model. In *COLING 1990 Volume 2: Papers presented to the 13th International Conference on Computational Linguistics*.

Masamune Kobayashi, Masato Mita, and Mamoru Komachi (2024). Large language models are state-of-the-art evaluator for grammatical error correction. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 68–77, Mexico City, Mexico. Association for Computational Linguistics.

John Lee and Jonathan Webster (2012). A corpus of textual revisions in second language writing. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 248–252, Jeju Island, Korea. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer (2019). Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Xinyuan Li and Yunshi Lan (2025). Large language models are good annotators for type-aware data augmentation in grammatical error correction. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 199–213, Abu Dhabi, UAE. 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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov (2019). Roberta: A robustly optimized bert pretraining approach. *CoRR*, abs/1907.11692.

Warren S. McCulloch and Walter Pitts (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.

Masato Mita, Keisuke Sakaguchi, Masato Hagiwara, Tomoya Mizumoto, Jun Suzuki, and Kentaro Inui (2024). Towards automated document revision: Grammatical error correction, fluency edits, and beyond. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 251–265, Mexico City, Mexico. Association for Computational Linguistics.

Jakub Náplava and Milan Straka (2019). Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

Courtney Napoles, Keisuke Sakaguchi, Matt Post, and Joel Tetreault (2015). Ground truth for grammatical error correction metrics. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 588–593, Beijing, China. Association for Computational Linguistics.

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.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks,

Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski

Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph (2024). Gpt-4 technical report. *CoRR*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu (2002). 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. 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. *Journal of Machine Learning Research*, 21(140):1–67.

Vipul Raheja, Dhruv Kumar, Ryan Koo, and Dongyeop Kang (2023). CoEdIT: Text editing by task-specific instruction tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5274–5291, Singapore. Association for Computational Linguistics.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn (2021). A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Alla Rozovskaya (2024). Universal Dependencies for learner Russian. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 17112–17119, Torino, Italia. ELRA and ICCL.

Naomi Sager (1981). *Natural Language Information Processing: A Computer Grammar of English and Its Applications*. Addison-Wesley Longman Publishing Co., Inc., USA.

Timo Schick, Jane Dwivedi-Yu, Zhengbao Jiang, Fabio Petroni, Patrick Lewis, Gautier Izacard, Qingfei You, Christoforos Nalmpantis, Edouard Grave, and Sebastian Riedel (2022). Peer: A collaborative language model.

Gabriella Skitalinskaya, Jonas Klaff, and Henning Wachsmuth (2021). Learning from revisions: Quality assessment of claims in argumentation at scale. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1718–1729, Online. Association for Computational Linguistics.

Harold Somers (2005). Round-trip translation: What is it good for? In *Proceedings of the Australasian Language Technology Workshop 2005*, pages 127–133, Sydney, Australia.

Yixiao Song, Kalpesh Krishna, Rajesh Bhatt, Kevin Gimpel, and Mohit Iyyer (2024). GEE! grammar error explanation with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 754–781, Mexico City, Mexico. Association for Computational Linguistics.

Felix Stahlberg and Shankar Kumar (2021). 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. Association for Computational Linguistics.

Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022). *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*. O'Reilly Media, Sebastopol, CA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin (2023). Attention is all you need.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le (2022). Finetuned language models are zero-shot learners.

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

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch (2016). Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel (2021). mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

A. Appendix A

Table 12.: Examples of Incorrect Citation Order Errors and Their Corrections

Error Category	**Error tag**	**Error ID**	**Example of Mistake**	**Corrected**
Incorrect Citation Order	ICO	ICO1	Research on translation strategies has been carried out by several scholars (Smith 2015; Jones 2010).	Research on translation strategies has been carried out by several scholars (Jones 2010; Smith 2015).
Incorrect Citation Order	ICO	ICO2	Several scholars have written on translation theory (Nida 1964; Zack 1988; Catford 2003).	Several scholars have written on translation theory (Catford 2003; Nida 1964; Zack 1988).
Incorrect Citation Order	ICO	ICO3	Translation strategies have been discussed in various studies (Baker 2008; Baker 2006).	Translation strategies have been discussed in various studies (Baker 2006; Baker 2008).
Incorrect Citation Order	ICO	ICO4	The concept of equivalence has been explored in several works (Nida 2005b; Nida 2005a).	The concept of equivalence has been explored in several works (Nida 2005a; Nida 2005b).
Incorrect Citation Order	ICO	ICO5	Translation theories have been explored by House (2005), Baker (2007), Venuti (2006) and Venuti (1998).	Translation theories have been explored by Baker (2007), House (2005), Venuti (1998) and Venuti (2006).

A. Appendix A

Table 13.: Examples of Punctuation Errors and Their Corrections

Error Category	**Error tag**	**Error ID**	**Example of Mistake**	**Corrected**
Punctuation Error	PUN	PUN1	Translation theories have been discussed in works by Bassnett (2002), Nida (1964) and Venuti (1995).	Translation theories have been discussed in works by Bassnett (2002); Nida (1964); and Venuti (1995).
Punctuation Error	PUN	PUN2	Several scholars have written on translation theory (Catford 1965 Nida 1964 Newmark 1988).	Several scholars have written on translation theory (Catford 1965; Nida 1964; Newmark 1988).
Punctuation Error	PUN	PUN3	Multiple studies (Baker 1994, Risku 2005, Pöchhacker 1998) emphasizes a change in required competencies of translators.	Multiple studies (Baker 1994; Risku 2005; Pöchhacker 1998) emphasizes a change in required competencies of translators.
Punctuation Error	PUN	PUN4	The theory was first presented by Baker 1994.	The theory was first presented by Baker (1994).
Punctuation Error	PUN	PUN5	Pöchhacker, 2004, argues that translation is a complex process.	Pöchhacker (2004) argues that translation is a complex process.
Incorrect Punctuation	PUN	PUN6	The work by Risku 2005 is significant.	The work by Risku (2005) is significant.
Incorrect Punctuation	PUN	PUN7	This topic has been widely researched Baker 2006.	This topic has been widely researched (Baker 2006).
Incorrect Punctuation	PUN	PUN8	Studies by Kaindl, 1999 and Pöchhacker, 2005 have examined this.	Studies by Kaindl (1999) and Pöchhacker (2005) have examined this.
Incorrect Punctuation	PUN	PUN9	The study focused on these aspects. (Baker 2006)	The study focused on these aspects (Baker 2006).
Incorrect Punctuation	PUN	PUN10	This topic has been widely researched (Baker (2006)).	This topic has been widely researched (Baker 2006).
Incorrect Punctuation	PUN	PUN11	The concept was discussed by Nida (1964); Baker (1994); and Venuti (2000).	The concept was discussed by Nida (1964), Baker (1994), and Venuti (2000).
Incorrect Punctuation	PUN	PUN12	Several studies have addressed this issue (Pym 2019, Risku 1995).	Several studies have addressed this issue (Pym 2019; Risku 1995).
Incorrect Punctuation	PUN	PUN13	According to Kaindl [2003], the theory of translation has evolved.	According to Kaindl (2003), the theory of translation has evolved.
Incorrect Punctuation	PUN	PUN14	Translation theory has been debated (Kaindl, 2000).	Translation theory has been debated (Kaindl 2000).

Table 14.: Examples of Incorrect Citation Type Errors and Their Corrections

Error Category	**Error tag**	**Error ID**	**Example of Mistake**	**Corrected**
Incorrect Citation Type	TYPE	TYPE1	For example, (Pym 2019; Risku 1995) examined the impact of globalization on translation.	For example, Pym (2019) and Risku (1995) examined the impact of globalization on translation.
Incorrect Citation Type	TYPE	TYPE2	This topic has been widely researched Baker (2006).	This topic has been widely researched (Baker 2006).
Incorrect Citation Type	TYPE	TYPE3	Several studies Baker (2006), Risku (1995) examined the impact of globalization on translation.	Several studies (Baker 2006; Risku 1995) examined the impact of globalization on translation.
Incorrect Citation Type	TYPE	TYPE4	(Risku 1995) analyzed the cognitive aspects of translation processes.	Risku (1995) analyzed the cognitive aspects of translation processes.
Incorrect Citation Type	TYPE	TYPE5	(Pym 2019, Risku 1995) examined the impact of globalization on translation.	Pym (2019) and Risku (1995) examined the impact of globalization on translation.

Table 15.: Examples of Errors in Using *et al.* for Multiple Authors and Their Corrections

Error Category	**Error tag**	**Error ID**	**Example of Mistake**	**Corrected**
Error in Using *et al.* for Multiple Authors	ETAL	ETAL1	Pym, Risku, Shlesinger (2012) argue that translation theory is evolving.	Pym et al. (2012) argue that translation theory is evolving.
Error in Using *et al.* for Multiple Authors	ETAL	ETAL2	Pym, Risku et al. (2012) provide an analysis of translation strategies.	Pym et al. (2012) provide an analysis of translation strategies.
Error in Using *et al.* for Multiple Authors	ETAL	ETAL3	Pym, et al. (2012) discuss translation theory in depth.	Pym et al. (2012) discuss translation theory in depth.
Error in Using *et al.* for Multiple Authors	ETAL	ETAL4	The relationship between language and culture was widely discussed (Pym et al., 2012).	The relationship between language and culture was widely discussed (Pym et al. 2012).
Error in Using *et al.* for Multiple Authors	ETAL	ETAL5	The relationship between language and culture was widely discussed (Pym, Risku, Shlesinger, Baker, 2012).	The relationship between language and culture was widely discussed (Pym et al. 2012).
Error in Using *et al.* for Multiple Authors	ETAL	ETAL6	The relationship between language and culture was widely discussed (Pym, et al. 2012).	The relationship between language and culture was widely discussed (Pym et al. 2012).

Table 16.: Examples of Incorrect Citation for Two Authors and Their Corrections

Error Category	**Error tag**	**Error ID**	**Example of Mistake**	**Corrected**
Incorrect Citation for Two Authors	TWO	TWO1	The theory of translation was discussed by Pöchhacker, Risku (2005).	The theory of translation was discussed by Pöchhacker and Risku (2005).
Incorrect Citation for Two Authors	TWO	TWO2	The theory of translation was discussed by Pöchhacker & Risku (2005).	The theory of translation was discussed by Pöchhacker and Risku (2005).
Incorrect Citation for Two Authors	TWO	TWO3	The study has addressed this issue (Petrova, Ivanova 1995).	The study has addressed this issue (Petrova & Ivanova 1995).
Incorrect Citation for Two Authors	TWO	TWO4	The study has addressed this issue (Petrova and Ivanova 1995).	The study has addressed this issue (Petrova & Ivanova 1995).

Table 17.: Examples of Redundant Entity Errors and Their Corrections

Error Category	Error tag	Error ID	Example of Mistake	Corrected
Redundant entity	RED	RED1	As Pöchhacker suggests, equivalence plays a crucial role in translation (1995).	As Pöchhacker (1995) suggests, equivalence plays a crucial role in translation.
Redundant entity	RED	RED2	As Pöchhacker (Pöchhacker 1995) suggests, equivalence plays a crucial role in translation.	As Pöchhacker (1995) suggests, equivalence plays a crucial role in translation.
Redundant entity	RED	RED3	As Pöchhacker (1995) suggests, equivalence plays a crucial role in translation (Pöchhacker 1995).	As Pöchhacker (1995) suggests, equivalence plays a crucial role in translation.
Redundant entity	RED	RED4	Pöchhacker in 1995 suggests that equivalence plays a crucial role in translation (Pöchhacker 1995).	Pöchhacker (1995) suggests that equivalence plays a crucial role in translation.
Redundant entity	RED	RED5	In 1956, Risku developed a new translation tool.	Risku (1956) developed a new translation tool.

Table 18.: Examples of Incorrect Placement of Citations and Their Corrections

Error Category	Error tag	Error ID	Example of Mistake	Corrected
Incorrect Placement of Citations	PLC	PLC1	According to House, translation theories have been thoroughly examined and demonstrated the valuable results (2007).	According to House (2007), translation theories have been thoroughly examined and demonstrated the valuable results.
Incorrect Placement of Citations	PLC	PLC2	Pöchhacker argues that translation is an evolving field (2006: 249).	Pöchhacker (2006: 249) argues that translation is an evolving field.

Table 19.: Examples of Incorrect Use of Page Number Abbreviations and Their Corrections

Error Category	Error tag	Error ID	Example of Mistake	Corrected
Incorrect Use of Page Number Abbreviations	PAGE	PAGE1	Pym (2019: 18, 19, 20, 21) discusses the evolution of translation theory.	Pym (2019: 18–21) discusses the evolution of translation theory.
Incorrect Use of Page Number Abbreviations	PAGE	PAGE2	Pym (2019 18) discusses the concept of equivalence.	Pym (2019: 18) discusses the concept of equivalence.
Incorrect Use of Page Number Abbreviations	PAGE	PAGE3	Pym (2019: 18, 19) discusses the translation process.	Pym (2019: 18f.) discusses the translation process.
Incorrect Use of Page Number Abbreviations	PAGE	PAGE4	Pym (2019: 18, 19, 20) discusses the translation process.	Pym (2019: 18ff.) discusses the translation process.
Incorrect Use of Page Number Abbreviations	PAGE	PAGE5	Pym (2019: 18f – 21f) examines the shift in translation theory.	Pym (2019: 18–21) examines the shift in translation theory.
Incorrect Use of Page Number Abbreviations	PAGE	PAGE6	Pym (2019: 19 : 25) examines the shift in translation theory.	Pym (2019: 19–25) examines the shift in translation theory.