UNIVERSITY POLITEHNICA OF BUCHAREST
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

RESEARCH PROJECT III

Romanian Natural Language Processing Tasks
Grammar Error Correction

Matei Vlad Cristian

Thesis advisor:

Șl. dr. ing. Dumitru-Clementin Cercel

BUCHAREST

2024-2025

CONTENTS

1 INTRODUCTION

1.1 Context

The internet, globalization and the emergence of global digital platforms have transformed our lives and transcended geographical and cultural borders [1]. Thanks to the internet, the Web and social media, it is easy to connect to anybody in the world, living in a different culture, in a different social context, with different social norms, with different habits and speaking a different language [1].

In the last years, the field of AI has made major progress in almost all its standard sub-areas, including vision, speech recognition and generation, and the specific domain of natural language processing. By using models such as ELMo, GPT, mT5, and BERT, the potential to improve internet-mediated communication through the analysis provided by AI models has been observed. With the emergence of popular models like GPT-3 [2] and GPT-4 [3], this has become even more evident.

As interest in second language acquisition grows, ensuring the precision and efficacy of written communication becomes increasingly essential. I believe the Romanian language can take advantage of technological progress, making it more intuitive and accessible to anyone interested. One method to make Romanian culture more approachable and enhance communication in the language is through the creation of an effective grammar correction tool.

1.2 Problem

The effort invested in developing a grammar correction tool has primarily advanced in the context of the English language. For Romanian, the efforts have been minimal, with only a few experiments and attempts. However, the need and interest for a specialized grammar corrector are significant. Such a tool could benefit foreign students enrolled in Romanian universities, bloggers, writers, editors of all kinds, and could certainly be used as a recommendation feature on social media platforms.

Another reason why the development of a specialized grammar corrector is necessary lies in the complexity and nature of the Romanian language. It features a rich and intricate grammar, full of exceptions and rules, with a diverse morphological structure inspired by Latin, Russian,

[1] https://digital-strategy.ec.europa.eu/en/news/diversity-aware-internet-when-technology-works-peo
[2] https://openai.com/index/gpt-3-apps/
[3] https://openai.com/index/gpt-4-research/

and Germanic languages [4] . As a result, grammatical errors are common, further encouraged by the differences among the specific regional dialects.

1.3 Objectives

This paper seeks to contribute to research in the field of natural language understanding for Romanian. Its value lies in experimenting with a new specialized grammar corrector designed for the resource-constrained Romanian language.

A new parallel dataset with a legal theme will be published, containing pairs of synthetically generated correct and incorrect sentences. This dataset will be used to train various models and will leverage additional techniques to evaluate its ability to provide rich and relevant information, aiming to achieve a new state-of-the-art for the Romanian language.

1.4 Structure

The following chapters will be structured as follows: Chapter Two will cover several theoretical elements to aid in understanding the main method. Chapter Three will introduce the proposed solution, along with the newly generated dataset and its analysis. Finally, Chapter Four will present the experiments, their results, and their interpretation.

1.5 Research Overview: Past Semesters

The primary focus of my dissertation will be to contribute to the development of a set of NLP tools specifically for the Romanian language.

During the first semester, I introduced an entirely new dataset for summarization and played a key role in creating a new dataset of historical texts for named entity recognition (NER). Additionally, I conducted experiments for summarization and established the direction for future work.

In the second semester, I developed a grammar corrector specialized for Romanian and built a new synthetic dataset. By training a T5 model, I achieved the best results for Romanian to date on the given dataset.

The experiments first involved creating an artificial parallel dataset with synthetically generated errors. This dataset, named MEID (Marcell Error Injection Dataset), was derived from MARCELL. The best results were 28.39 when the model was only pre-trained on the artificial dataset without beam search during the decoding step and 28.95 when beam search was

[4] https://en.wikipedia.org/wiki/Romanian_language

applied. As for full training, fine-tuning on RoNACC yielded 56.91 without beam search and 60.49 with beam search.

This semester, the goal is to further improve the grammar corrector's performance by expanding the experiments, refining specific aspects of the artificially generated dataset, and exploring additional training techniques, including the use of larger models.

2 BACKGROUND

The next chapter will cover key theoretical concepts crucial for the successful implementation of the proposed solution, along with an overview of existing methodologies in the research field.

2.1 Theoretical Concepts

2.1.1 Overview of Grammatical Error Correction

In this subchapter, I will show a history of grammatical error correction solutions proposed over time, as presented in [8].

Classifiers

Classifiers were among the earliest methods put up to address grammatical issues, possibly as a result of their attempt to address English problems in which second language speakers committed errors pertaining to articles or prepositions. One example of their application is in building a classifier that predicts one of {the, a/an, none} before each noun phrase in a sentence.

The classifier receives information representing the context of the analyzed word, and once the errors are flagged, it predicts the correction based on the confusion list, which can include articles, prepositions, noun forms, or verb forms.

The use of classifiers raises several concerns [8]. They have limitations in that they only correct specific errors, and the confusion lists are typically small. They also cannot correct fluency errors or recommend the use of more appropriate synonyms in a given context. Another issue is that they rely on a very specific local context, treating grammatical errors largely independently while assuming that the rest of the information in the context is correct.

Statistical Machine Translation

Unlike classifiers, Statistical Machine Translation (SMT) is designed to address all types of errors simultaneously without the need for specialized interventions.

Although it was designed for translation between languages, SMT [4] is also easily applicable

to GEC, where the task can be viewed as translating sentences from their incorrect forms to their correct versions.

Based on the noisy channel model [28], SMT can be mathematically formulated using Bayes' rules [8]:

$$\hat{C} = \arg\max_{C} P(C|E) = \arg\max_{C} \frac{P(E|C)P(C)}{P(E)} = \arg\max_{C} P(E|C)P(C)$$

, where the correct sentence C is passed through a noisy channel to produce an erroneous sentence E, and the goal is for the model to learn to reconstruct the correct sentence $\hat{C}$, starting from the erroneous sentence E.

Yuan and Felice [39] demonstrated the effectiveness of a POS-factored SMT system for error correction of a variety of error types in learner text in the CoNLL-2013 shared task. In 2014, SMT demonstrated state-of-the-art performance in general error correction, as evidenced by the top systems in the CoNLL-2014 shared task. This success solidified SMT as the dominant method in the field, motivating additional research and applications of SMT technology to GEC.

Neural Machine Translation

Given the rapid advancements in deep learning, Neural Machine Translation (NMT) has been developed and used heavily in Grammatical Error Correction (GEC). Unlike SMT, NMT uses one single neural network in the correction of grammatical errors without the use of feature engineering, thus offering an end-to-end solution. For this reason, soon after its development, NMT became the top methodology in the area [25].

NMT relies on an encoder-decoder architecture [9], where an input sequence $x = (x_1, x_2, ..., x_T)$ is passed through an encoder to produce a series of hidden state representations. These hidden states are then processed by a decoder to generate an output sequence $y = (y_1, y_2, ..., y_T)$. The decoder predicts each word y_t based on the sequence generated so far $y_1, y_2, ..., y_{t-1}$ [8]:

$$p(y|x) = \prod_{t=1}^{T} p(y_t \mid \{y_1, y_2, \ldots, y_{t-1}\}, x)$$

Various architectures have been proposed for constructing encoders and decoders, including RNNs, CNNs, and Transformers [33]. Similar to SMT, NMT also has some limitations. Although it can handle grammatical errors more complex in nature than SMT and does not require feature engineering, it needs an enormous amount of training data, which is costly and time-consuming to obtain. Additionally, due to the nature of deep neural networks, the predictions become non-interpretable, making it impossible to explain how the corrections are made.

Edit-Based Approaches

While previous approaches focused on generating a correct sentence from an incorrect one, this method offers a more cost-effective solution. The editing process primarily involves copying input tokens to the output with a high probability, with only the errors being modified.

This method guarantees faster inference speeds, though it also has limitations, particularly in its inability to perform well on fluency errors or very complex, context-dependent errors.

The generation of edits is viewed as a sequence tagging task [31]. For each token in the input sequence, the model predicts whether an edit operation should be applied to that token or not. This method requires creating a set of tags representing each individual operation. For operations like word insertion or word replacement, the model must analyze each word in the vocabulary, causing the number of tags to increase linearly with the number of unique words in the training data [8].

In addition to faster inference speed, one of the advantages of this method is the transparency and explainability of the correction process. However, the technique requires human engineering in defining the label set and their dimensions.

Language Models as Discriminators and Generators

One common way in which language models were traditionally utilized in GEC was based on the assumption that sentences with low probabilities are more likely to contain grammatical errors compared to those with high probabilities. Consequently, GEC systems aimed to identify these low-probability sentences and transform them into grammatically correct, higher-probability sentences. To achieve this, language models played a crucial role as discriminators in guiding the correction process [5].

Another novel way to use language models for GEC is to use them as zero-shot or few-shot generators to generate grammatically corrected sentences from input sentences. For example, for a prompt such as "Correct the grammatical errors in the following sentence:" followed by a noisy input sentence, the language model will produce a corrected sentence given the input context. The viability of such a strategy has been mainly dependent on the improvements in Large Language Models (LLMs).

Regardless of the categorization of language models, one of the major advantages that come along with their usage is that they require only unannotated monolingual corpora, hence making them easier to be implemented on other languages. Although using language models in discriminative tasks does not yield state-of-the-art performance, their use in generative models has led to much improved performance [8].

2.1.2 Transformers

The Google researchers introduced a new deep learning architecture known as the Transformer [33], which revolutionized the AI field, particularly through the attention mechanism proposed in [18]. This technique processes text non-sequentially, enabling parallel understanding of context and relationships between words.

Model Architecture

The architecture of Transformers consists of a sequence of encoders and decoders units, similar to Seq2Seq models, with the distinction that contextualization in these modules is achieved through the attention mechanism.

The first type of module, the encoder, consists of several procedures. It begins by processing input embeddings, which include positional information. Next, a multi-head attention mechanism is applied, followed by a simple feedforward network. Residual connections [14] are also leveraged at each step, ensuring more stable gradients and contributing to normalization.

The second type of module, the decoder, also consists of two main components. While it closely mirrors the structure of the encoder, this module includes an additional step that applies attention over the outputs received from the encoder in the Transformer architecture.

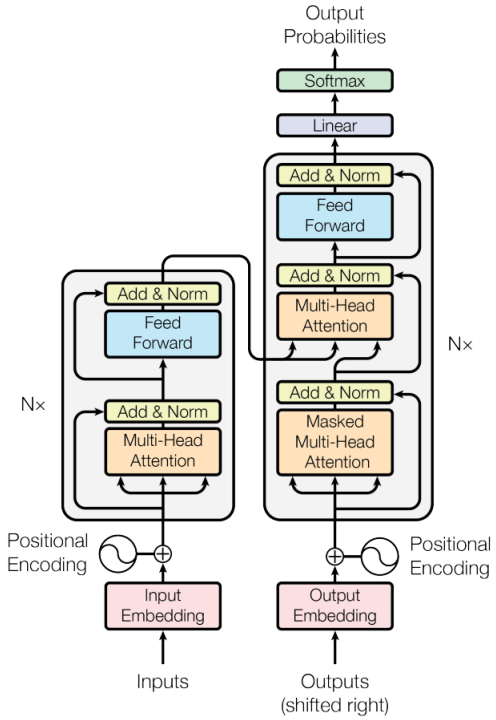

Figure 2.2 Transformers Architecture [33]

The attention mechanism can be described as a mapping system that responds to various queries with sets of answers (key-value pairs), much like a search engine. The process involves

applying scaled dot-product attention for each attention head, creating semantic similarities between words in the context. Each word calculates its correlation level with other words (keys) within the established context, and the resulting scores are based on value vectors. These vectors are updated using a weight matrix, which is initially initialized with the word embeddings themselves. The final step integrates the outputs from the multi-head attention mechanism into a single representation.

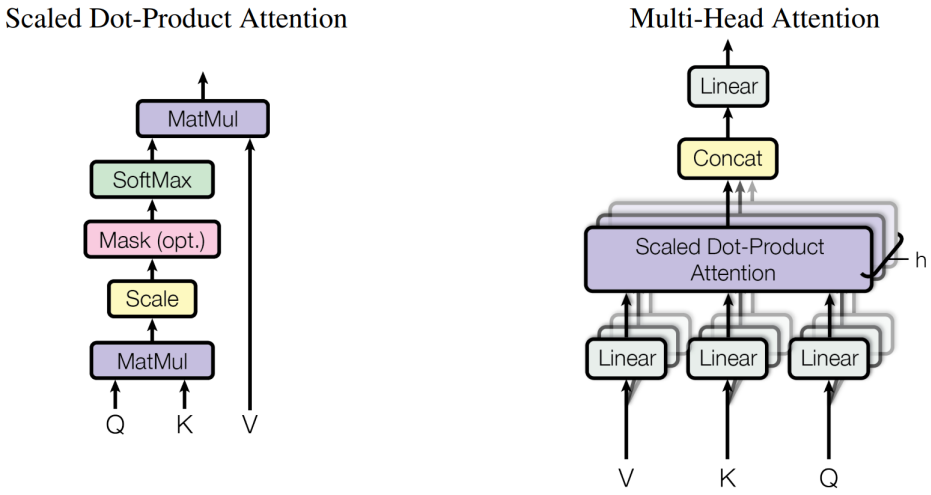

Figure 2.3 Attention Mechanism [33]

2.2 Existing Approaches

At a pivotal moment in the development of grammar checkers, the NLP community organized the CoNLL-2014 shared task conference [37], which aimed to correct errors in English essays. The field made further strides when [15] suggested that error correction could be approached as a translation task, transforming "incorrect" sentences into "correct" ones. Over the years, different models have been examined, beginning with SMT models introduced in [37] and advancing to NMT models [1, 30]. A major breakthrough occurred when these models began to utilize the attention mechanism [19].

As a result, research studies have presented the outcomes obtained when using sequence-to-sequence models [34, 38] for GEC. [11] showcases a phrase-based approach that incorporates neural features. Additionally, [16] explores efficiency improvements through the use of a deep recurrent neural network [2] and a transformer [33]. Finally, [26] proposes the use of a T5 model [23].

The challenge of adapting solutions for low-resource languages was a key topic at the BEA 2019 Shared Task conference [6]. The issue of limited data, a major hurdle for training transformers, was effectively addressed by [10, 13] in the "Low Resource Track" section, where [13] excelled, outperforming 14 submissions in the "Restricted Track" category. [10] also delivered encouraging results, especially through data augmentation techniques involving

controlled noise injection. Additionally, another method that employed a spellchecker to create confusion sets received recognition in [13].

One solution for low-resource languages is the artificial generation of errors. Several studies explore interesting ways to approach this issue, particularly in morphologically rich languages like Russian [27]. These studies highlight differences in the distribution of error types in Russian compared to English, as well as contrasts between mistakes made by native speakers and those made by foreign learners. The goal of synthetic errors is to closely emulate real-world mistakes. In the case of Ukrainian, [29] presents an alternative approach to adapting the commonly used ERRANT error taxonomy [7], proposing a different methodology. Another interesting approach to improving GEC model performance is using a back-translation [24] model to generate natural errors by transforming correct sentences into incorrect ones, thus creating new synthetic training data.

Regarding the Romanian language, relatively few studies have been conducted. [12] pretrains a vanilla transformer using an artificially generated dataset of approximately 300K examples. Eventually, by introducing a new dataset, RONACC, also referenced in this work—the model is finetuned, yielding good results for that type of architecture but ultimately surpassed by this paper. [21] proposes new evaluation methods and conducts additional experiments, further advancing research in the field for the Romanian language.

Additionally, [17] suggests a technique to enhance the performance of GEC models by using round-trip translation. This involves translating the text into a different language and then back again, which offers a stronger option for the final prediction. This approach has also been tested on Romanian, making it a focus of this study as well.

3 PROPOSED SOLUTION

3.1 Analysis of Datasets

In this section, I will present the two datasets utilized in the proposed method: RONACC and MARCELL, the latter being used to artificially create MEIDv2, a key contribution of this study.

3.1.1 RONACC

The Romanian National Audiovisual Council Corpus (RoNACC) is a dataset consisting of pairs of incorrect and corrected sentences gathered from various radio shows and Romanian TV broadcasts. It was introduced in [12] as the first dataset specifically proposed for Romanian Grammatical Error Correction.

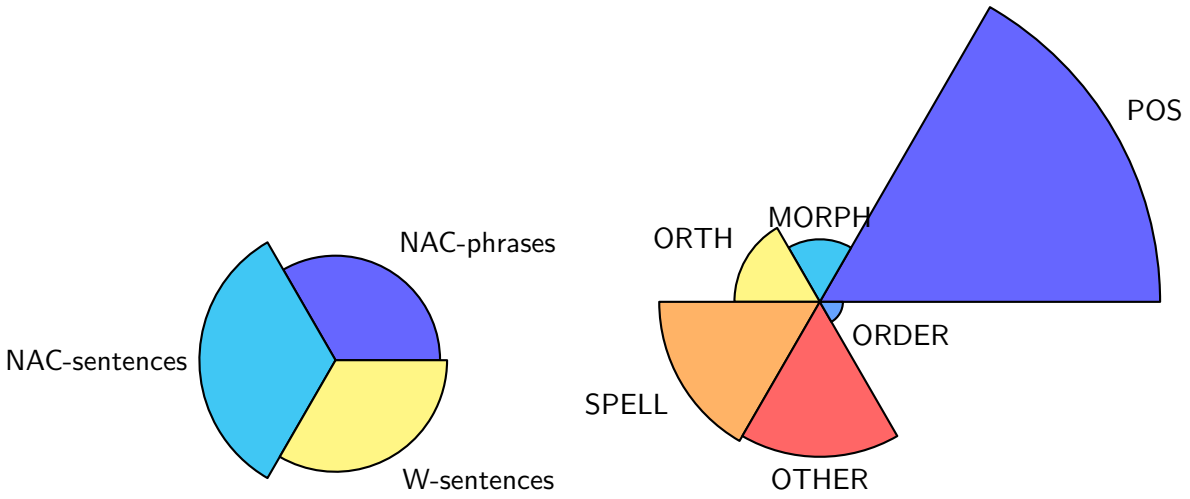

Figure 3.1 The composition of RoNACC and the errors distribution

The corpus can be classified into three categories: NAC-phrases, which contains sentences with missing verbs; NAC-sentences, focusing on the correct formation of sentences; and W-phrases, which includes writing errors. Each grammatical mistake can be corrected, as the provided context is sufficient. The dataset consists of approximately 10K sentence pairs [1].

[1] `https://nextcloud.readerbench.com/index.php/s/9pwymesT5sycxoM`

3.1.2 MARCELL-RO

As part of the MARCELL CEF Telecom project [32], which focuses on collecting and annotating a comprehensive corpus of legal documents, the MARCELL-RO-v2[2] corpus is introduced. This corpus includes Romanian national legislative texts from 1881 to 2021, mainly consisting of government decisions, orders, decrees, and laws.

While the corpus spans historical legal texts, the focus here is on selecting documents that do not contain grammatical errors. The dataset's exceptional organizational structure further influenced its selection. The MARCELL-RO subcorpus is precisely segmented into sentences, with each word carefully annotated to include relevant details, such as part of speech, lemma, case, gender, number, and form. These annotations, made possible by tools from RACAI[3], will be particularly useful in generating synthetic errors for the study.

3.2 Taxonomy of Grammatical Errors

To establish a well-defined categorization for evaluating and testing grammatical errors, I have chosen to use ERRANT, the ERRor ANnotation Tool [7]. This tool enables the automatic identification of errors in a parallel dataset by using a framework based on over 50 rules, creating a comprehensive error taxonomy. It provides a qualitative and informative evaluation, helping to optimize the final results.

For the Romanian language, [12] presents an adapted version of ERRANT, which relies on 14 universal dependency POS tags rather than language-specific ones. Additionally, the word dictionary is based on the vocabulary used by the Hunspell spellchecker [4], while part-of-speech tagging is carried out using Romanian SpaCy [5].

A list of categorized error examples [12] from the RONACC dataset can be found below:

The tags represent different grammatical features: POS (e.g., NOUN) and an additional FORM tag for variations in word forms. The MORPH category covers errors in word structure, ORTH identifies capitalization and spacing mistakes, SPELL marks spelling issues, and ORDER highlights mistakes in token arrangement.

[2] https://elrc-share.eu/repository/browse/marcell-romanian-legislative-subcorpus-v2/2da548428b9d11eb9c1a00155d026706ce94a6b59ffc4b0e9fb5cd9cebe6889e/

[3] https://relate.racai.ro/

[4] https://github.com/titoBouzout/Dictionaries

[5] https://spacy.io

Errant Type	Meaning	Description / Example
MORPH	Morphology	în cazul unei paciente [**internată** → internate] joi
ORTH	Orthography	Acum, în [**camera deputaților** → Camera Deputaților]
SPELL	Spelling	treceți la ceea ce [**vroiați** → voiați] să ziceți
ORDER	Word Order	Nu [**mai o** → o mai] da cotită
OTHER	Other	permis de [**port-armă** → portarmă]
NOUN	Noun	O rotiță care să-și aducă [**aportul** → contribuția].
NOUN:FORM	Noun form	să vedem cum a fost din [**punct** → punctul] de vedere al organizării
VERB	Verb	opoziția ar putea [**demara** → începe] procedura de suspendare
VERB:FORM	Verb form	Omul negru poate [**fii** → fi] folosit metaforic
ADJ	Adjective	cu personaje dubioase sau într-un context [**aiurea** → nepotrivit].
ADJ:FORM	Adjective form	E un pic [**ambiguu** → ambiguă] definirea termenului mită aici
ADV	Adverb	Și te costă și foarte, foarte [**ieftin** → puțin]
PRON	Pronoun	O să trebuiască să dați niște mesaje, [_ → niște] telefoane
PRON:FORM	Pronoun form	Pe [**aceeași** → aceeși] bandă circulai?
DET	Determiner	Șeful [_ → unui] aerodrom privat din Comana
DET:FORM	Determiner form	de liderii PSD, PNL și [**a** → ai] minorităților naționale
ADP	Adposition	80% [**din** → dintre] victimele traficanților
CCONJ	Coordinating conjunction	dânsa ca [**și** → _] persoană luptătoare
PUNCT	Punctuation	lovește mingea înapoi[_ → ,] dar au început să se apropie

Table 1: Example of ERRANT taxonomy from the RONACC dataset

3.3 Synthetic Data Generation

3.3.1 Error Distribution Analysis

To generate synthetic errors for Romanian, I rely on a probability distribution similar to those found in languages with Slavic influences, such as Ukrainian and Russian, due to the limited resources available for training a generative GEC model. The RULEC dataset [27], which includes around 12K manually annotated sentences written by both native and non-native Russian speakers, highlights various error types. The differences between speaker types help me decide which population segment to target for better model adaptation. Also, the high number of spelling and punctuation mistakes among native Russian speakers is particularly interesting, and this aspect will be a focus of my research.

I also note that a similar error distribution is found in the UA-GEC dataset [29] for Ukrainian, where punctuation and spelling errors are dominant. However, I choose not to incorporate the fluency category from this dataset in my analysis. Instead, I will focus on the most common errors, such as spelling and punctuation, in order to improve the accuracy of synthetic error generation for Romanian.

Foreign			Heritage		
Error	**%**	**Errors per 1,000**	**Error**	**%**	**Errors per 1,000**
Spell.	18.6	11.7	Spell.	42.4	15.7
Noun:case	14.0	8.8	Punc.	22.9	8.5
Lex. choice	13.3	8.3	Noun:case	7.8	2.9
Miss. word	8.9	5.6	Lex. choice	5.5	2.0
Punc.	7.6	4.8	Miss. word	4.7	1.7
Replace	6.3	3.9	Replace	2.8	1.0
Extra word	5.7	3.5	Extra word	2.4	0.9
Adj:case	3.9	2.4	Adj:case	2.1	0.8
Prep.	3.3	2.1	Word form	2.1	0.8
Word form	3.1	2.0	Noun:number	1.8	0.7
Noun:number	2.6	1.6	Verb agr.	1.6	0.6
Verb agr.	2.5	1.6	Prep.	1.5	0.6

Error type	**%**
Grammar (all)	14.4
Fluency (all)	23.6
Spelling	19.0
Punctuation	43.0

Table 2: Combined table of errors for Russian Foreign and Heritage learners [27] (left) and types of errors with percentages for Ukrainian learners [29] (right).

3.3.2 Error Injection Rules

In my second-semester research, I focused on generating grammatical errors through a rule-based method, as outlined in [12], which is widely recognized for its effectiveness in the Romanian language. This approach involves applying empirically determined rules based on a pre-defined probability distribution to generate errors. I reviewed various methods for error generation, including noise injection and back-translation, as mentioned in [8].

The rule-based technique is particularly useful for handling grammar and spelling errors, and I defined the initial error probability distribution in my research, considering sentence length, as shown in Figure 1, which also illustrates the adaptation of the new version. I chose to increase the probability of having more errors in sentences since the goal of this semester was to experiment with larger models capable of handling more complex incorrect sentences. This distribution controls the likelihood of generating specific errors, in line with findings in [36].

Next, after determining the number of errors to be generated, the type of error is also established using a probability distribution, following the general hierarchy:

1. **Concatenation**: combine two consecutive tokens.
2. **Transposition**: the token exchange position with a consecutive token.
3. **Deletion**: delete the token.
4. **Misspelling**: introduce spelling errors into words.
5. **Substitution**: seven different types of substitutions are presented, including substitution between prepositions, articles, singular pronouns, plural pronouns, etc.
6. **Punctuation**: deleting/inter-changing symbols

After that, I updated two probability distributions [36] defined last semester, as shown in Figure 2: one for the type of error to be generated and another for the number of misspellings in a token if the misspelling method is selected.

Length	Err.	Prob.
[1, 3)	0	0.50
	1	0.50
[3, 6)	1	0.50
	2	0.50
[9, 16)	3	0.15
	4	0.25
	5	0.30
	6	0.30
[20, 30)	4	0.10
	5	0.30
	6	0.30
	7	0.15
	8	0.15

Length	Err.	Prob.
[6, 9)	2	0.30
	3	0.45
	4	0.25
[16, 20)	3	0.10
	4	0.30
	5	0.30
	6	0.15
	7	0.15
[30, ∞)	5	0.10
	6	0.20
	7	0.20
	8	0.30
	9	0.20

Length	Err.	Prob.
[1, 3)	0	0.50
	1	0.50
[3, 6)	1	0.50
	2	0.50
[9, 16)	3	0.15
	4	0.25
	5	0.30
	6	0.30
[20, 30)	4	0.10
	5	0.30
	6	0.30
	7	0.15
	8	0.15

Length	Err.	Prob.
[6, 9)	2	0.30
	3	0.45
	4	0.25
[16, 20)	3	0.10
	4	0.30
	5	0.30
	6	0.15
	7	0.15
[30, ∞)	5	0.10
	6	0.20
	7	0.20
	8	0.30
	9	0.20

Figure 1: Comparison of Error Injection Probability Distribution by Sentence Length (Old Version - Left, New Version - Right)

Type	Prob.
Concatenation	0.12
Transposition	0.08
Deletion	0.05
Misspell	0.45
Substitution	0.30

Tok. length	Err.	Prob.
[1, 3)	0	1.00
[3, 5)	1	1.00
[5, 10)	1	0.80
	2	0.20
[10, ∞)	1	0.75
	2	0.15
	3	0.10

Type	Prob.
Concatenation	0.10
Transposition	0.08
Deletion	0.05
Misspell	0.35
Substitution	0.40

Tok. length	Err.	Prob.
[1, 3)	0	1.00
[3, 5)	1	1.00
[5, 10)	1	0.30
	2	0.70
[10, ∞)	1	0.40
	2	0.30
	3	0.30

Figure 2: Comparison of Token Error Types and Misspells Probability Distribution (Old Version - Left, New Version - Right)

The first three methods are straightforward but have a lower priority in the error generation process. The main focus is on Misspell and Substitution errors, as they introduce the most complex grammatical mistakes. Additionally, Punctuation errors are handled separately. Finally, the probability of generating substitution errors has been increased to accommodate a new approach: substituting words with their synonyms from DexOnline. Additionally, it can be noted that the probability of generating multiple errors for a single word has also been increased.

Mispelling Errors

This specific category concerns errors made at the character level, where a number of characters are chosen from a word based on the probability described in Table 4 and modified using various techniques.

The main methods for generating mispelling errors are:

1. **Letter Transposition**: the swapping of positions between two characters.
2. **Letter Deletion**: the simple removal of a character.

3. **Letter Insertion**: the insertion of a random character.
4. **Letter Replacement**: the deletion of a character followed by an insertion.
5. **RoWordNet**: generating synonyms.
6. **DexOnline**: changing the words inflections.
7. **Confusion Lists**: for prepositions, conjunctions, articles.
8. **Spellchecker**: based on edit distances

To enhance adaptability to Romanian, new methods are introduced creatively. One approach, **Articulated Suffix Removal**, targets common grammatical mistakes by removing definite article suffixes (e.g., *baiatul* $\rightarrow$ *baiat*). Additionally, it addresses errors in plural forms, such as miscounting 'i' in words like *copiii* (children), refining grammatical accuracy.

The second method, **Controlled Letter Replacement**, involves replacing characters with potential alternatives. Alternatives are determined by examining adjacent keys on a Romanian keyboard, similar-looking characters, and common grammatical errors, such as misspellings and mispronunciations. For instance, the word *celălalt* may be mispronounced as *celălant*, leading to replacing *l* with *n*, or *culoarea ridichii* might be misspelled as *culoarea ridichiei*, prompting the replacement of *i* with *e* or *ie*.

Type	Prob.
Letter Transposition	0.25
Letter Deletion	0.25
Letter Insertion	0.15
Letter Replacement	0.10
Articulated Suffix Removal	0.05
Controlled Letter Replacement	0.20

Type	Prob.
Letter Transposition	0.30
Letter Deletion	0.25
Letter Insertion	0.20
Letter Replacement	0.10
Articulated Suffix Removal	0.10
Controlled Letter Replacement	0.20

Figure 3: Comparison of Probability Distribution of Misspelling Error Types (Old Version - Left, New Version - Right)

As seen in Figure 3, the probabilities are shown. In the new version of this research, I aimed to add more diversification. For example, some of the newly inserted letters can automatically be part of the transposition process, so this is the reason why the sum of probabilities does not equal 1. Additionally, I increased the chances for insertion, transposition, and articulated suffix removal, aiming to generate more complex errors.

Substitution Errors

The error generation process in this research focuses on generating more complex errors related to the deeper meaning of Romanian sentences. One important method involves substituting words using **RoWordNet** [6], which is the Romanian version of the WordNet lexical database.

Another method uses **DEXOnline (INFL)** [7], the online explicative dictionary of Romanian,

[6] https://github.com/dumitrescustefan/RoWordNet
[7] https://dexonline.ro

to substitute words with their conjugations or declensions. Web scraping extracts all possible variations, which are then selected and substituted in the text.

Type	Prob.
RoWorNet	0.20
DexOnline	0.40
Spellchecker	0.40

Type	Prob.
RoWorNet	0.20
Confusion List	0.10
DexOnline (INFL)	0.28
DexOnline (SYN)	0.12
Spellchecker	0.30

Figure 4: Comparison of Probability Distribution of Substitution Error Types (Old Version - Left, New Version - Right)

A new approach was introduced to speed up the process and enhance flexibility. If a word is part of a predefined list (such as prepositions, conjunctions, or articles), a candidate from a **confusion list** is chosen 25% of the time. Otherwise, web scraping is used to extract possible substitutions.

The third method, **Spellchecker**, utilizes the Phunspell spellchecker [8] to find and substitute words similar to the original word based on their edit distance.

Additionally, a method based on **DEXOnline (SYN)** has been introduced, which uses the DEXOnline dictionary to generate synonyms for words, adding new variants to the error generation process.

Punctuation Errors

In [3], a probability matrix is proposed for each punctuation mark, which allows the generation of synthetic errors. Each punctuation mark in the rows can be replaced with one from the columns based on the assigned probability.

Sign		,	;	:	-	.	?	!	...
	0.95	0.025	0	0	0	0.025	0	0	0
,	0.41	0.54	0	0	0	0.05	0	0	0
;	0.85	0.05	0.10	0	0	0	0	0	0
:	0.90	0.02	0	0.08	0	0	0	0	0
-	0.95	0	0	0	0.05	0	0	0	0
.	0.20	0	0.03	0	0	0.74	0	0	0.03
?	0.80	0	0	0	0	0.04	0.12	0.04	0
!	0.80	0	0	0	0	0.10	0.07	0.03	0
...	0.80	0	0	0	0	0	0	0	0.20

Table 3: Probability Distribution of Punctuation Error Types

The updated probability matrix introduces changes only for the "-" punctuation mark, and is shown in Table 3.

[8] https://pypi.org/project/phunspell/

3.3.3 The MEIDv2

In this paper, I will present MEIDv2, [9], the updated version of the dataset introduced in the previous report, MEID [10]. Ultimately, it results in a parallel dataset in Romanian, which contains correct sentences from the legislative section of the MARCELL corpus and those with synthetically generated grammatical errors, based on the new probability distributions.

This version retains the core characteristics of the original dataset, keeping only articles from 2007 onward and selecting only sentences longer than 10 words. In the end, approximately 1.5 million pairs were obtained, processed over 2-3 days in parallel using an I7-12800H with 32GB RAM. The dataset was saved in JSONL files, which were then packed into multiple Parquet files and publicly uploaded to Hugging Face.

Errant Taxonomy Evaluation

The error distribution in MEID is calculated using the ERRANT taxonomy, based on 30K examples considered sufficient for generalization. The technique is similar to that used in previous research, utilizing scripts provided by the open-source community for ERRANT.

Category	Percentage	Category	Percentage
PUNCT	31.13%	VERB	0.61%
SPELL	23.00%	VERB:FORM	0.60%
OTHER	21.31%	ADJ	0.43%
ORTH	7.48%	DET:FORM	0.26%
WO	4.27%	PRON	0.23%
NOUN:FORM	3.00%	DET	0.19%
ADP	2.82%	PRON:FORM	0.07%
NOUN	2.70%	ADV	0.07%
MORPH	0.67%	K	0.01%
CCONJ	0.67%	ADJ:FORM	0.00%

Category	Percentage	Category	Percentage
PUNCT	23.56%	VERB	0.95%
SPELL	13.38%	VERB:FORM	0.59%
OTHER	32.73%	ADJ	0.82%
ORTH	6.56%	DET:FORM	0.36%
WO	4.68%	PRON	0.32%
NOUN:FORM	3.22%	DET	0.25%
ADP	3.68%	PRON:FORM	0.03%
NOUN	6.64%	ADV	0.12%
MORPH	0.61%	K	0.01%
CCONJ	0.84%	ADJ:FORM	0.64%

Figure 5: Comparison of MEID distribution of errors (Old Version - Left, New Version - Right)

As can be seen, by adjusting the probabilities for generating synthetic errors, significant differences emerge. The errors generated are more diverse, with the number of incorrect nouns being 2.5 times higher, twice as many incorrect adjectives, and a substantial increase in errors related to verbs, determiners, pronouns, adverbs, adjective forms, coordinating conjunctions, and other types.

I think it's a good sign that punctuation and spelling errors have decreased a bit compared to the likelihood of encountering other types of errors.

[9] https://huggingface.co/datasets/mateiaassAI/MEID3_v2

[10] https://huggingface.co/datasets/mateiaassAI/MEID

3.4 T5

Transfer learning [22] is an advanced paradigm in which the model can utilize knowledge acquired from a large-scale pre-training stage to transfer that knowledge to a related task. Rather than training from scratch, requiring huge amounts of data and computer power, a model is first trained on a general-purpose dataset that teaches the model general language patterns, grammar, and semantics. Therefore, on a smaller, task-specific dataset it can be fine-tuned, thus adapting to it faster and benefiting also from the application of previous knowledge. This boosts performance and speed of training under all circumstances and is especially beneficial in low-resource situations when labeled data is scarce.

The T5 (Text-to-Text Transfer Transformer) model [23] applies transfer learning by framing all NLP tasks as text-to-text transformations. This unified approach allows it to handle tasks such as translation, classification, and summarization efficiently, optimizing training and adaptation across different applications.

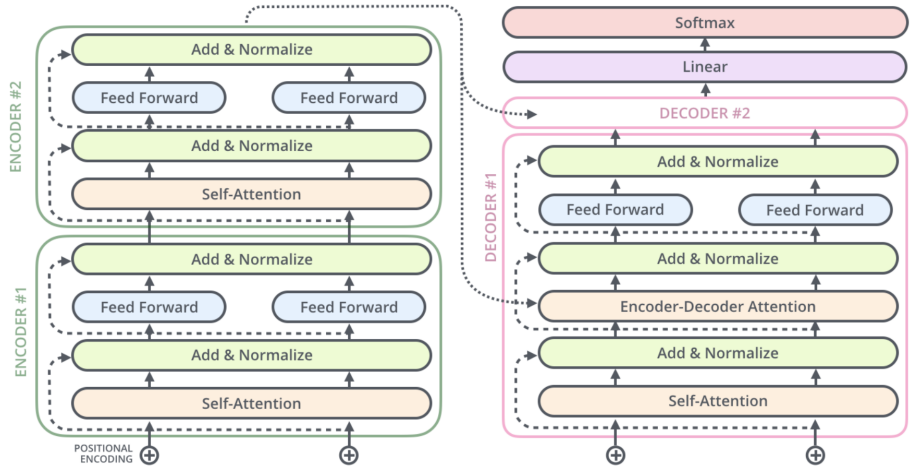

Figure 6: T5 Architecture [11]

The T5 model architecture follows the same structure as a standard encoder-decoder transformer. The official version was trained using text from Common Crawl, sourced from the web. For this paper, a variant trained on 80GB of Romanian text was utilized.

The t5-base-romanian model [12] is based on the T5v1.1 architecture (247 million parameters), with an encoder sequence length of 512 and a decoder sequence length of 256. Trained on a variety of tasks, it will be utilized for Grammar Error Correction (GEC), which can be seen as a type of machine translation. The model will process sentence pairs accompanied by a specific prompt to guide the correction. Each sentence with errors will be presented in the format: "Corectează: *grammatically altered text*", an approach that has shown slight improvements in previous tests.

[11] `https://jalammar.github.io/illustrated-transformer/`
[12] `https://huggingface.co/dumitrescustefan/t5-v1_1-base-romanian`

3.5 Data Augmentation Strategies

In this research, I focused on optimizing the final results. Thus, by using larger models, I aimed not to build the learning process solely on a fixed, rule-based probability distribution. I tried to introduce uncontrolled noise, and for this purpose, I used the two methods presented below.

3.5.1 Back-Translation

Back-Translation (BT) [24] is a noise generation method, successfully used in GEC [35]. The model learns to generate noise by taking a correct sentence as input and an incorrect one as reference, which can later augment the correct sentences during inference, as shown in the Figure 7:

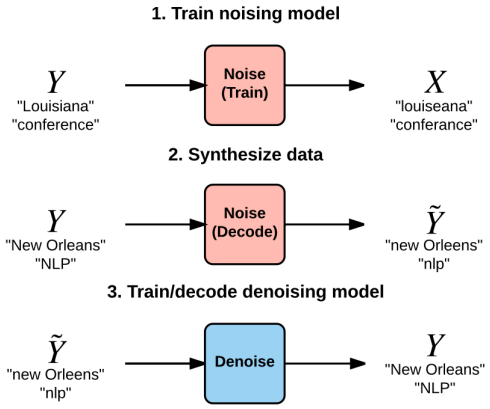

Figure 7: Back-Translation Method [35]

In the experiments, I used the training portion of RoNACC to train the model, ensuring that the noise generated closely resembles the natural errors encountered in real-world data.

3.5.2 Round-Trip Translation

A slightly more popular alternative to Back-Translation is Round-Trip Translation (RTT) [20]. This method involves translating the text using an imperfect translation model into a bridge language, which is then back-translated into the original language. This sequential process inevitably introduces grammatical errors, especially if error-prone methods are used.

I believe that the errors generated will be less severe compared to those produced using Rule-Based techniques and Back-Translation, either separately or tested together, and therefore, I will use the method presented in [17].

Successfully tested in GEC and even on RoNACC, with significant improvements, it involves

augmenting the input text both during training and inference in a structure of the form: "*input text* => *round-tripped text*."

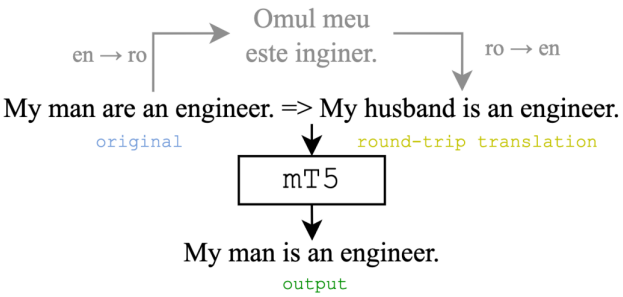

Figure 8: Round-Trip Translation Method [17]

With this technique, the model will be assisted in first finding an intermediate form that is less affected by errors, which will later be corrected to provide the final prediction.

I tested several languages as a bridge, such as Russian and Mongolian, but ultimately decided to use English. Since it is quite different from Romanian, it introduces fluency errors that are necessary to enrich the synthetic dataset. Additionally, by using the Google Translate API, the translation into English is not so flawed as to produce serious errors, allowing the model to more easily correct complex sentences.

The version of Google Translate used was 2.0.0, and during augmentation, 10 times more sentences were generated by using creative decoding techniques.

4 EVALUATION AND EXPERIMENTS

This chapter presents the experiments conducted, followed by an overview of the evaluation metrics and an interpretation of the results obtained.

4.1 Experimental Settings

To address the task, I will use the second version of the MEID dataset. The model architecture will be T5, which will be used in both its base and large forms. Experiments conducted in the second semester will be repeated, and all intermediate results will be presented, from pretraining to finetuning, along with the effect of each optimization method used.

4.1.1 Training Details

The models were trained using the resources provided by the Faculty of Automatic Control and Computers at the Polytechnic University of Bucharest. Specifically, I used two NVIDIA H100 Tensor Core GPUs. The training time varied from 12 to 50 hours, depending on the number of epochs, model size, and the data subset used for pretraining.

Therefore, for the input length, I decided to truncate the examples to 256, down from 512. As for the decoder length, I kept it consistent at 256, as it had been pre-trained in a similar manner.

The number of examples used from MEIDv2 was 300K, in order to better compare the results with [12], avoid any tendency for overfitting, and also achieve shorter training times.

4.2 Evaluation Metrics

To assess the corrections made by the model, I will use ERRANT to generate a confusion matrix, which will help highlight both the similarities and differences between the corrections needed for the sentences. This matrix will be used to compute the model's precision and recall values.

Since precision and recall alone don't fully reflect the model's performance, and the F1 score may not be the most suitable in this case, I will focus more on minimizing false positives by prioritizing precision over recall. Therefore, the primary evaluation metric will be the F0.5 score, which gives more weight to precision.

The F0.5 score is calculated using the following formula:

$$F_{0.5} = 1.25 \cdot \frac{\text{precision} \cdot \text{recall}}{0.25 \cdot \text{precision} + \text{recall}}$$

4.3 Results

4.3.1 Performance Evaluation

In my experiments, I pre-trained the T5 model on the MEID dataset and then tested various approaches, including applying new data augmentation methods, different decoding techniques, or continuing with fine-tuning on the RoNACC dataset. Table 4 presents the differences between the final results obtained using the updated MEIDv2 dataset and the new T5Large architecture.

POS	**Old Results (T5Base)**			**New Best (T5Base)**			**New Best (T5Large))**		
	Pr	**Re**	**F0.5**	**Pr**	**Re**	**F0.5**	**Pr**	**Re**	**F0.5**
ADJ	18.18	11.11	16.13	54.55	25	**44.12**	44.44	22.22	37.04
ADJ:FORM	61.11	39.29	55.00	78.95	53.57	**72.12**	73.68	53.85	68.63
ADP	72.85	56.41	68.84	69.31	50.72	64.58	74.68	63.78	**72.22**
ADV	61.22	58.82	60.73	68.09	66.67	**67.80**	67.50	57.45	65.22
CCONJ	36.84	46.67	38.46	57.14	47.06	**54.79**	36	45	37.50
DET	56.10	48.94	54.50	65.62	60	64.42	69.57	69.57	**69.57**
DET:FORM	85.71	72.00	**82.57**	76.92	69.77	75.38	82.98	78	81.93
MORPH	75.00	55.56	70.09	73.68	66.67	72.16	95.45	72.41	**89.74**
NOUN	25.00	23.53	24.69	52.33	50.56	**51.96**	40	32.73	38.30
NOUN:FORM	72.08	61.33	69.64	71.93	48.24	65.50	81.25	72.22	**79.27**
ORTH	76.14	40.85	64.92	76.26	57.30	71.52	81.37	53.90	**73.84**
OTHER	56.23	40.84	52.29	70.17	65.38	69.16	91	84.92	**89.72**
PRON	0	0	0	8.86	46.67	**10.57**	4.05	42.86	4.95
PRON:FORM	100	50.00	**83.33**	4	22.22	34.48	83.33	62.50	78.12
PUNCT	55.72	50.17	54.51	59.63	49.62	57.32	66.02	56.15	**63.77**
SPELL	73.64	65.36	71.82	70.25	61.30	68.25	82.64	78.49	**81.78**
VERB	13.33	4.88	9.99	56.41	39.29	**51.89**	27.27	8.82	19.23
VERB:FORM	87.63	59.03	79.89	78.95	54.74	72.53	92.92	73.43	**88.24**
WO	72.73	50.00	66.67	100	35.71	73.53	85.71	75	**83.33**
Total	**63.11**	**51.87**	**60.49**	**67.86**	**60.21**	**66.18**	**82.46**	**75.33**	**80.93**

Table 4: T5 Performance Results Comparison (Old vs New)

As can be seen, in its new version with T5Base, I achieved an improvement of approximately 6%. The performance gain is attributed to both the new synthetic data from MEIDv2 and the fact that the current model uses both Back-Translation and Round-Trip Translation. Ultimately, using T5Large, the improvement over last semester is 20%. In the case of the last example, the best F0.5 score was obtained without any data augmentation. Therefore, it is

clear that significant differences can be observed in terms of identifying and correcting certain types of errors.

It is very clear how the new version of the MEID dataset influences the final results. Errors such as DET:FORM and PRON:FORM were much easier to correct before, but now ADJ, NOUN, PRON, and VERB are more easily identified and corrected. This could be the effect of the new probability of replacing words with their synonyms, and the addition of the new method DexOnline - SYN.

Furthermore, it is evident that, in general, the results obtained with T5Large are superior to those obtained with T5Base. However, there are cases where the new version of T5Base outperforms T5Large, which is somewhat unusual. A possible explanation for these results could be the influence of the augmentation methods, which are not reflected in the final results of T5Large. Specifically, through Back-Translation and Round-Trip Translation, errors of the VERB type are much easier to identify, with a score of 51.89% (compared to 9.99% and 19.23%), PRON with a score of 10.57% (from 0% and 4.95%), NOUN with 51.96% (from 24.69% and 38.30%), and similarly for ADJ, ADJ:FORM, ADV, and CCONJ.

To better visualize the effect of the augmentation methods, I will present the results obtained with the new version of T5Base, both with and without the augmentation techniques.

POS	**T5Base - No Data Augmentation**			**T5Base - Data Augmentation**		
	Pr	**Re**	**F0.5**	**Pr**	**Re**	**F0.5**
ADJ	33.33	16.67	27.78	54.55	25	**44.12**
ADJ:FORM	65	46.43	60.19	78.95	53.57	**72.12**
ADP	74.52	60.00	**71.08**	69.31	50.72	64.58
ADV	67.35	64.71	66.80	68.09	66.67	**67.80**
CCONJ	38.10	53.33	40.40	57.14	47.06	**54.79**
DET	61.36	57.45	60.54	65.62	60	**64.42**
DET:FORM	86.05	74	**83.33**	76.92	69.77	75.38
MORPH	100	74.07	**93.46**	73.68	66.67	72.16
NOUN	30.88	30.88	30.88	52.33	50.56	**51.96**
NOUN:FORM	76.58	66.85	**74.42**	71.93	48.24	65.50
ORTH	83.72	43.90	70.87	76.26	57.30	**71.52**
OTHER	64.24	45.01	59.18	70.17	65.38	**69.16**
PRON	1.25	14.29	1.53	8.86	46.67	**10.57**
PRON:FORM	83.33	62.50	**78.12**	4	22.22	34.48
PUNCT	65.88	55.81	**63.59**	59.63	49.62	57.32
SPELL	78.37	68.57	**76.19**	70.25	61.30	68.25
VERB	23.08	7.32	16.13	56.41	39.29	**51.89**
VERB:FORM	86.61	67.36	**81.93**	78.95	54.74	72.53
WO	77.78	43.75	67.31	100	35.71	**73.53**
Total	**68.54**	**56.59**	**65.76**	**67.86**	**60.21**	**66.18**

Table 5: T5 Performance Results Comparison (with/-out Data Augmentation)

As observed, the effect of the augmentation methods primarily lies in providing synonym variations to the base text. Some of their suggestions - replacing words with synonyms, in the

case of Round-Trip Translation, - or the noise introduced by Back-Translation are appropriate choices, leading to significant improvements in scores for ADJ, ADJ:FORM, CCONJ, NOUN, PRON, VERB, and WO. However, when the model relied on the error probability distribution from MEIDv2, which emphasizes inflection errors, spelling, and morphology, the results for DET:FORM, MORPH, NOUN:FORM, PRON:FORM, SPELL, PUNCT, and VERB:FORM were much better.

Among the two approaches, I believe that the version utilizing the augmentation methods is more suitable, as it provides a more balanced framework for error correction.

4.3.2 Qualitative Analysis

For the qualitative analysis, I selected four examples from the dataset that I found particularly relevant. These examples are also featured in the state-of-the-art study [12] and in last semester's research report. The detailed analysis is presented in Table 6.

Type	Example 1	Example 2	Example 3	Example 4
Original	**Oameni** nu **ii judeca** pe **barbatii ca** nu sunt **tati** buni	**Terminand** cu aventurile sale, Ben **sia** scos omnitrix-ul **si** a crescut de la un **baiat**mic **intrun** adolescent pe care te **poti** baza.	Mohamed care a sosit la Londra luna **trecuta** pentru **opeartie v-a fii** externat mai **tarziu** din spital.	**in ultimile** trei zile
Gold	**Oamenii** nu **îi judecă** pe **bărbați că** nu sunt **tați** buni	**Terminând** cu aventurile sale, Ben **și-a** scos omnitrix-ul **și** a crescut de la un **băiat** mic **într-un** adolescent pe care te **poți** baza.	Mohamed**,** care a sosit la Londra luna **trecută** pentru **operație, va fi** externat mai **târziu** din spital.	**în ultimele** trei zile.
T5Base Old-Version	**Oamenii** nu ii judeca pe **barbati** ca nu sunt **tați** buni	**Terminând** cu aventurile sale, Ben **și-a** scos omnitrix-ul **și** a crescut de la un baiat mic **într-un** adolescent pe care te **poți** baza.	Mohamed**,** care a sosit la Londra luna **trecută** pentru **operațiune, va fi** externat mai **târziu** din spital.	**în ultimele** trei zile
T5Base	**Oamenii nu-i judecă pe bărbați că** nu sunt **tați** buni	**Terminând** cu aventurile sale, Ben **și-a** scos omnitrix-ul și a crescut de la un **baiat** mic **la un** adolescent pe care te **poți** baza.	Mohamed care a sosit la Londra luna **trecută** pentru **operație va fi** externat mai tarziu din spital.	**în ultimele** trei zile
T5Large without DA	**Oamenii** nu ii judeca pe barbatii ca nu sunt tati buni	Terminand cu aventurile sale, Ben **si-a** scos omnitrix-ul si a crescut de la un baiat mic **într-un** adolescent pe care te poti baza.	Mohamed care a sosit la Londra luna trecuta pentru **operație va fi** externat mai tarziu din spital.	**în ultimele** trei zile
T5Large with BackT	**Oamenii** nu ii **judecă** pe **bărbați că** nu sunt **tați** buni**.**	**Terminand** cu aventurile sale, Ben **si-a** scos omnitrix-ul si a crescut de la un baiat mic **la un** adolescent pe care te poti baza.	Mohamed**,** care a sosit la Londra luna trecuta pentru **operație**, **va fi** externat mai tarziu din spital.	**în** ultimile trei zile

Table 6: Qualitative Analysis

From the analysis, it is evident that all models continue to struggle with the correct use of diacritics. However, overall performance is satisfactory, with predictions closely matching

the ground truth. The T5Base model is the only one that benefits from Back-Translation and Round-Trip Translation techniques, and their influence is noticeable in terms of fluency. This is particularly evident in Example 1, where the model creatively uses the grammatical construction "nu-i."

In Example 2, the newer models intuitively predict the phrase "la un" instead of "într-un." Although this is grammatically incorrect, it suggests a deeper semantic analysis of the sentence, given that even the GOLD version lacks explicit clarity.

For Example 3, all newer models successfully identify the correct form of the noun "operație" and grammatically correct most of the errors, including missing punctuation marks.

The final example is straightforward, leaving little room for further interpretation.

The T5Large model without data augmentation was expected to deliver the best results based on evaluation metrics. However, T5Base, which incorporates both optimization techniques, demonstrates a more balanced approach to this grammatical error correction case. This is reflected in the qualitative analysis, where it produces the most interesting results.

4.3.3 SOTA Comparison

This section presents a comparison between the existing results on RoNACC up until now. The left subtable shows the SOTA results [12, 17] at the top, followed by the results from my previous research. The middle subtable contains the results obtained with T5Base, while the right subtable displays the results obtained with T5Large.

Model	Pr	Re	**F0.5**
RoGEC with Beam Search			
Artificial data	17.33	17.27	17.32
Fine-tuning	56.05	46.19	53.76
Yova et al. [17]			
MT5Large	71.70	58.30	68.60
MT5XLarge	72.30	56.90	68.60
Old T5Base			
Artificial data	30.14	23.09	28.39
Fine-tuning	55.23	64.79	56.91
Old T5Base with Beam Search			
Artificial data	30.82	23.29	28.95
Fine-tuning	63.11	51.87	60.49

Model	Pr	Re	**F0.5**
T5Base			
Artificial data	28.45	34.06	29.42
Fine-tuning	68.54	56.59	65.76
T5Base + RTT			
Artificial data	-	-	-
Fine-tuning	67.14	59.71	65.51
T5Base + BT			
Artificial data	52.58	45.11	50.89
Fine-tuning	68.31	57.74	65.90
T5Base + BT + RTT			
Artificial data	57.36	42.60	53.64
Fine-tuning	67.86	60.21	66.18

Pr	Re	**F0.5**
T5Large		
38.30	53.29	40.58
82.46	75.33	80.93
T5Largee + RTT		
-	-	-
-	-	-
T5Large + BT		
74.38	67.26	72.84
80.88	78.60	80.41
T5Large + BT + RTT		
70.20	62.51	68.51
78.70	74.44	77.81

Table 7: General Comparison of the Final Results

Regarding the state-of-the-art, RoGEC [12] introduced the RoNACC dataset and conducted experiments using a vanilla transformer, achieving modest results of 17.32%, and 53.76% after fine-tuning. [17] focused on Romanian GEC, testing various MT5 models and obtaining the best score of 68.60% with MT5XLarge and MT5Large when applying RTT. In experiments conducted last semester, I demonstrated the impact of artificial datasets on final results,

showing that a T5Base model outperformed a vanilla transformer by approximately 7% when finetuned.

An important observation is the performance of the new T5Base model on the artificial dataset compared to the previous version. Although the new artificial dataset contains only 300K examples, compared to 1 million in the earlier version, the model achieves a better result of approximately 29.42%, compared to 28.39%. This suggests that MEIDv2 is more suitable to be used and the 300K examples are sufficient for a T5 model to learn the probability distribution based on a simple, fixed, rule-based technique without uncontrolled noise.

Without additional optimizations, results improve significantly with larger architectures. The T5Large model achieves an F0.5 score of 80.93%, which is approximately 12% higher than the state-of-the-art MT5XLarge, 37% higher than RoGEC, and 20% higher than the results from previous research.

[17] evaluates the impact of applying the Round-Trip Translation method on models. The conclusion is that while RTT aids in grammatical error correction, it also introduces unnecessary changes, which can risk lowering overall performance. This effect becomes more evident as the model size increases. Similar observations are made in this study, where RTT does not have a positive overall contribution for the T5Large model, even though it helps balance the error distribution.

Following the experiments, it was observed that the most effective data augmentation method was Back-Translation. Round-Trip Translation proved more beneficial when applied alongside BT. This suggests that when noise becomes uncontrolled or more natural, the alternatives introduced by RTT are easier for the model to rationalize.

In conclusion, the data augmentation methods proved to be highly effective. The same T5Base model, trained only on artificial data, nearly matches the performance of the older T5Base fine-tuned on the RoNACC dataset when combined with MEIDv2, Back-Translation, and Round-Trip Translation —achieving 53.64% compared to 56.91%. Moreover, simply applying BT during the pretraining phase boosted the results from 29.42% to 50.89%, an improvement of 20.5 percentage points. In combinatie cu BT, si RTT a imbunatati rezultatele cu 3% in cadrul acestui pas.

For the T5Large model, the results are even more impressive. Using only artificial data and BT, the model achieved a score of 72.84%, surpassing the state-of-the-art and outperforming any T5Base model fine-tuned on RoNACC. The improvement brought by the BT method is around 32.30%. In this near-saturated context, applying Round-Trip Translation does not lead to further improvements, confirming findings from [17]. Ultimately, for the T5Large model, the best fine-tuned results are achieved without additional augmentation. This is logical for a capable model like T5Large, as the noise generated during BT is based on the same data, and the model demonstrates its ability to decode information directly from the source.

5 CONCLUSIONS

This research can be considered a success, as it achieved the objectives outlined in the introduction. A new version of the dataset, MEIDv2, was introduced. The need to adjust the probability distribution for synthetic error generation was demonstrated, and the results were analyzed with respect to the final error distribution.

Additionally, new state-of-the-art results were achieved for RoNACC, the standard dataset for evaluating GEC models in Romanian. The effectiveness of the artificial dataset MEIDv2 was proven, helping the T5Large model achieve an F0.5 score of approximately 81%. The study also demonstrated the effectiveness of each data augmentation method, interpreted the most suitable contexts for their application, and identified their limitations in supporting large, highly capable architectures.

Ultimately, essential and significant techniques were provided for the development of GEC in Romanian, offering alternative solutions for a language with limited resources and few explorations.

The work presented a valid model capable of correcting grammatical errors in Romanian, positioning it as a serious candidate for practical use in everyday life.

5.0.1 Future Directions

For future work, I aim to improve the experiments conducted on the Romanian summarization task.

Additionally, regarding GEC, I plan to explore the effects of data augmentation using the predictive power of LLMs and evaluate their performance on the RoNACC dataset.

BIBLIOGRAPHY

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

[2] Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. Deep architectures for neural machine translation, 2017.

[3] Maksym Bondarenko, Artem Yushko, Andrii Shportko, and Andrii Fedorych. Comparative study of models trained on synthetic data for Ukrainian grammatical error correction. In Mariana Romanyshyn, editor, *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 103–113, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[4] Chris Brockett, William B. Dolan, and Michael Gamon. Correcting ESL errors using phrasal SMT techniques. In Nicoletta Calzolari, Claire Cardie, and Pierre Isabelle, editors, *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Sydney, Australia, July 2006. Association for Computational Linguistics.

[5] Christopher Bryant and Ted Briscoe. Language model based grammatical error correction without annotated training data. In Joel Tetreault, Jill Burstein, Ekaterina Kochmar, Claudia Leacock, and Helen Yannakoudakis, editors, *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 247–253, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[6] Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. The BEA-2019 shared task on grammatical error correction. In Helen Yannakoudakis, Ekaterina Kochmar, Claudia Leacock, Nitin Madnani, Ildikó Pilán, and Torsten Zesch, editors, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–75, Florence, Italy, August 2019. Association for Computational Linguistics.

[7] Christopher Bryant, Mariano Felice, and Ted Briscoe. Automatic annotation and evaluation of error types for grammatical error correction. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 793–805, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[8] Christopher Bryant, Zheng Yuan, Muhammad Reza Qorib, Hannan Cao, Hwee Tou Ng, and Ted Briscoe. Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, page 1–59, July 2023.

[9] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation, 2014.

[10] Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. A neural grammatical error correction system built on better pre-training and sequential transfer learning. In Helen Yannakoudakis, Ekaterina Kochmar, Claudia Leacock, Nitin Madnani, Ildikó Pilán, and Torsten Zesch, editors, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 213–227, Florence, Italy, August 2019. Association for Computational Linguistics.

[11] Shamil Chollampatt and Hwee Tou Ng. Connecting the dots: Towards human-level grammatical error correction. In Joel Tetreault, Jill Burstein, Claudia Leacock, and Helen Yannakoudakis, editors, *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 327–333, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[12] Teodor-Mihai Cotet, Stefan Ruseti, and Mihai Dascalu. Neural grammatical error correction for romanian. pages 625–631, 11 2020.

[13] Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. Neural grammatical error correction systems with unsupervised pre-training on synthetic data. In Helen Yannakoudakis, Ekaterina Kochmar, Claudia Leacock, Nitin Madnani, Ildikó Pilán, and Torsten Zesch, editors, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 252–263, Florence, Italy, August 2019. Association for Computational Linguistics.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pages 770–778, 06 2016.

[15] Marcin Junczys-Dowmunt and Roman Grundkiewicz. The AMU system in the CoNLL-2014 shared task: Grammatical error correction by data-intensive and feature-rich statistical machine translation. In Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant, editors, *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task*, pages 25–33, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[16] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, and Kenneth Heafield. Approaching neural grammatical error correction as a low-resource machine translation task, 2018.

[17] Yova Kementchedjhieva and Anders Søgaard. Grammatical error correction through round-trip machine translation. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2208–2215, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[18] Minh-Thang Luong, Hieu Pham, and Christopher Manning. Effective approaches to attention-based neural machine translation. 08 2015.

[19] Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation, 2015.

[20] Nitin Madnani, Joel Tetreault, and Martin Chodorow. Exploring grammatical error correction with not-so-crummy machine translation. In Joel Tetreault, Jill Burstein, and Claudia Leacock, editors, *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 44–53, Montréal, Canada, June 2012. Association for Computational Linguistics.

[21] Ioan Florin Cătălin Nitu and Traian Eugen Rebedea. Intelligent linguistic system for the grammar of the romanian language. *International Journal of User-System Interaction*, 13(4):183–198, 2020.

[22] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[23] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[24] Marek Rei, Mariano Felice, Zheng Yuan, and Ted Briscoe. Artificial error generation with machine translation and syntactic patterns. In Joel Tetreault, Jill Burstein, Claudia Leacock, and Helen Yannakoudakis, editors, *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 287–292, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[25] Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. A simple recipe for multilingual grammatical error correction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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, August 2021. Association for Computational Linguistics.

[26] Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. A simple recipe for multilingual grammatical error correction. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *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, August 2021. Association for Computational Linguistics.

[27] Alla Rozovskaya and Dan Roth. Grammar error correction in morphologically rich languages: The case of Russian. *Transactions of the Association for Computational Linguistics*, 7:1–17, 2019.

[28] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[29] Oleksiy Syvokon, Olena Nahorna, Pavlo Kuchmiichuk, and Nastasiia Osidach. UA-GEC: Grammatical error correction and fluency corpus for the Ukrainian language. In Mariana Romanyshyn, editor, *Proceedings of the Second Ukrainian Natural Language Processing Workshop (UNLP)*, pages 96–102, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics.

[30] Zhixing Tan, Shuo Wang, Zonghan Yang, Gang Chen, Xuancheng Huang, Maosong Sun, and Yang Liu. Neural machine translation: A review of methods, resources, and tools, 2020.

[31] Maksym Tarnavskyi, Artem Chernodub, and Kostiantyn Omelianchuk. Ensembling and knowledge distilling of large sequence taggers for grammatical error correction. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3842–3852, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[32] Tamás Váradi, Svetla Koeva, Martin Yamalov, Marko Tadić, Bálint Sass, Bartłomiej Nitoń, Maciej Ogrodniczuk, Piotr Pezik, Verginica Barbu Mititelu, Radu Ion, Elena Irimia, Maria Mitrofan, Vasile Păiș, Dan Tufiș, Radovan Garabík, Simon Krek, Andraz Repar, Matjaž Rihtar, and Janez Brank. The MARCELL legislative corpus. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 3761–3768, Marseille, France, May 2020. European Language Resources Association.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 06 2017.

[34] Ziang Xie, Anand Avati, Naveen Arivazhagan, Dan Jurafsky, and Andrew Y. Ng. Neural language correction with character-based attention, 2016.

[35] Ziang Xie, Guillaume Genthial, Stanley Xie, Andrew Ng, and Dan Jurafsky. Noising and denoising natural language: Diverse backtranslation for grammar correction. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *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, June 2018. Association for Computational Linguistics.

[36] Shuyao Xu, Jiehao Zhang, Jin Chen, and Long Qin. Erroneous data generation for grammatical error correction. In Helen Yannakoudakis, Ekaterina Kochmar, Claudia Leacock, Nitin Madnani, Ildikó Pilán, and Torsten Zesch, editors, *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 149–158, Florence, Italy, August 2019. Association for Computational Linguistics.

[37] Zheng Yuan and Ted Briscoe. Grammatical error correction using neural machine translation. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California, June 2016. Association for Computational Linguistics.

[38] Zheng Yuan and Ted Briscoe. Grammatical error correction using neural machine translation. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 380–386, San Diego, California, June 2016. Association for Computational Linguistics.

[39] Zheng Yuan and Mariano Felice. Constrained grammatical error correction using statistical machine translation. In Hwee Tou Ng, Joel Tetreault, Siew Mei Wu, Yuanbin Wu, and Christian Hadiwinoto, editors, *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task*, pages 52–61, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.