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

RESEARCH PROJECT II

Romanian Natural Language Processing Tasks
Grammar Error Correction

Matei Vlad Cristian

Thesis advisor:

Șl. dr. ing. Dumitru-Clementin Cercel

BUCHAREST
2024

CONTENTS

1 INTRODUCTION

1.1 Context

We live in a world that's growing more complex every day. With a smart device connected to the internet at our fingertips, we can access information and connect with people from all corners of the globe. In this diverse and multicultural landscape, effective and accurate communication has become increasingly crucial. Together, through collaboration, we can truly make a difference.

Artificial intelligence has seen unprecedented advancements, becoming a hugely popular field, especially with the breakthroughs in natural language processing models like GPT-3 [1] and the enhanced GPT-4 [2]. These developments present numerous opportunities to improve our internet-mediated communication through the sophisticated analysis provided by AI models.

In this context, I believe that the Romanian language can benefit from technological advancements, with the potential to become more user-friendly and accessible to anyone interested. This progress would streamline both learning and communication processes, ultimately leading to a broader understanding and appreciation of Romanian culture.

One of the tasks that would greatly benefit the Romanian language is the development of an efficient grammar checker.

1.2 Problem

Currently, solutions for developing an efficient grammar checker for the Romanian language are limited, despite its potential usefulness in several areas of interest. These include aiding students in their journey of learning Romanian or assisting foreign students enrolled in the Romanian education system, checking emails in the corporate environment, serving as an auxiliary tool in journalism, or simply being used by writers, bloggers, and recommended by various social media platforms.

Another obstacle comes from the intricate nature of Romanian, highlighting the necessity for a grammar checker. Its complex grammar, full of exceptions and challenging rules, along with its diverse morphological landscape [3] influenced by Latin, Russian, and Germanic languages, contributes to a wide range of errors, compounded by various regional dialects.

[1] https://openai.com/index/gpt-3-apps/

[2] https://openai.com/index/gpt-4-research/

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

1.3 Objectives

This paper aims to expand the research boundaries and potential applications within the realm of natural language processing for the Romanian language. More precisely, the focus will be on addressing the primary task: rectifying grammatical errors.

The objective will be achieved through the publication of a novel parallel dataset, comprising pairs of correct and artificially generated incorrect sentences. This will enable the training of a new model from scratch, defined by a unique potential to achieve results comparable to state-of-the-art standards.

1.4 Structure

In the second chapter, relevant theoretical elements and similar approaches within the field will be presented in a logically ordered manner, providing the necessary information to understand the proposed solution.

The following chapter will introduce the proposed solution, along with the introduction of the new dataset, how it was constructed, and the creative ideas behind the error probability distribution.

In the fourth chapter, experiments will be conducted and the results will be presented through a performance analysis, including qualitative analysis and a comparison with the state-of-the-art in the Romanian language.

1.5 Research Overview: First Semester

Regarding my dissertation, the main focus will be advancing research on developing a set of NLP tools for the Romanian language. In the first semester, I introduced a completely new dataset for summarization and contributed to the creation of a new dataset with historical texts for NER. So far, I have conducted some experiments for summarization and outlined theoretically how the remaining experiments for both tasks should be conducted. This semester, I have introduced an additional task: grammatical error correction for the Romanian language. The final goal of the dissertation is to improve and complete the proposed experiments for these three NLP tasks for the Romanian language.

2 BACKGROUND

The upcoming chapter will present fundamental theoretical concepts essential for effectively implementing the proposed solution. Additionally, it will offer an overview of current methodologies within the research field.

2.1 Theoretical Concepts

One of the most successful approaches in grammatical error correction (GEC) is to treat the problem as a monolingual machine translation task, where you translate from possibly grammatically incorrect sentences to correct ones. This approach can be divided into two main categories: Statistical Machine Translation (SMT) [15] and Neural Machine Translation (NMT) [1].

2.1.1 Neural-Machine Translation

A Neural Machine Translation (NMT) system [1, 25], is a neural network that directly models the conditional probability $p(y|x)$, translating a source sentence, $x_1, \ldots, x_n$, into a target sentence, $y_1, \ldots, y_m$. Assuming translation occurs at the sentence level, this architecture can be perceived as a sequence-to-sequence model [23].

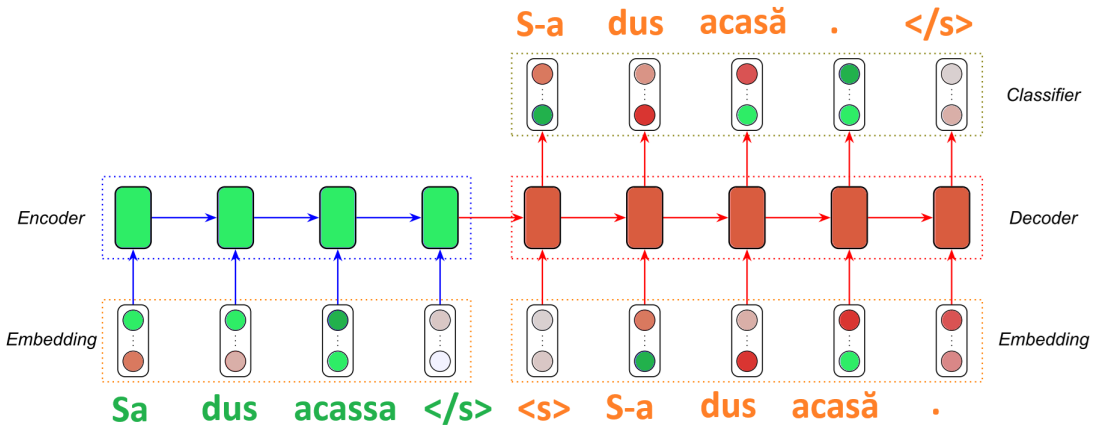

Figure 2.1 Neural-Machine Translation Seq2Seq Model [25]

In Seq2Seq models lacking the attention mechanism, there are two components: (a) an

encoder computing a representation s for each source sentence, and (b) a decoder generating one target word at a time, decomposing the conditional probability as:

$$\log p(y|x) = \sum_{j=1}^{m} \log p(y_j|y_{<j}, s) \tag{1}$$

By introducing the attention mechanism [1, 17], significantly improved results can be achieved, as the model selectively focuses on parts of the sentence throughout the translation process.

Neural machine translation aims at building a single neural network that efficiently handles intricate language patterns and long-distance word dependencies. This proficiency has positioned the arhitecture as the primary approach in machine translation [25], surpassing the limitations of SMT and offering more precise and fluent translations.

2.1.2 Transformers

Transformer [27], developed by Google researchers, signifies a breakthrough in AI, primarily due to its innovative use of attention, as proposed in [16]. This approach revolutionizes traditional sequential processing, allowing for parallel computation, which significantly improves performance.

Model Architecture

Transformers represent a class of neural networks known as Seq2Seq models, distinguished by their utilization of attention mechanisms. As a result, their architecture is structured around sequential processing, comprising encoding units and decoding units.

The encoder section comprises two primary elements. Initially, it processes input embeddings, which encode positional information. Subsequently, it utilizes a multi-head attention mechanism, followed by a simple feedforward network. Notably, each step incorporates a residual connection [12] and normalization to ensure stable gradient endurance, enhancing overall model stability.

Similarly, the decoder section consists of two main components. Alongside mirroring the encoder's structure, the decoder includes an additional subunit responsible for applying attention to the encoder's outputs. This enables effective information assimilation and facilitates the decoding process.

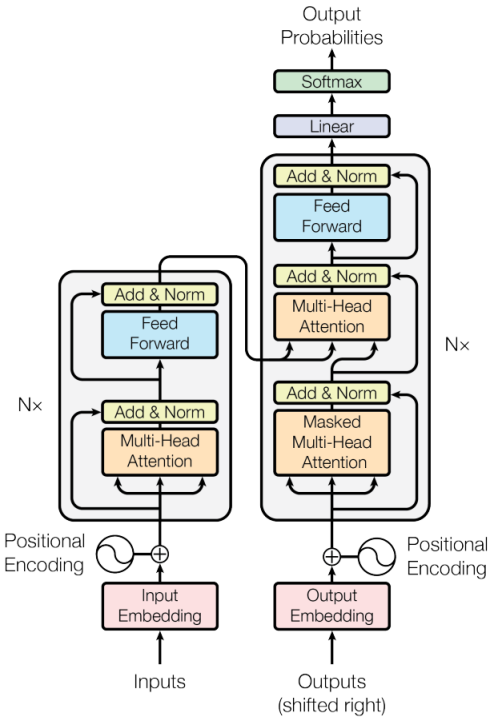

Figure 2.2 Transformers Architecture [27]

Regarding the attention mechanism, it functions as a mapping system, resembling a query interacting with a set of key-value pairs. Initially, it conducts scaled dot-product attention for each attention head, seeking semantic similarities across all text components. Each word query aims to establish correlations with context window words (keys), yielding similarity scores based on value vectors. These vectors are then updated by a weight matrix, initially initialized as the word embedding itself. Ultimately, the outputs from multiple attention heads are synthesized within the multi-head attention mechanism.

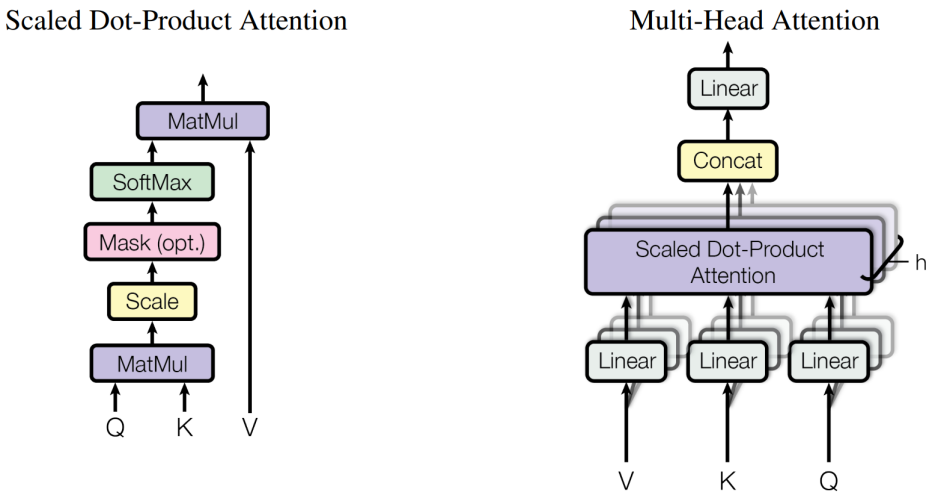

Figure 2.3 Attention Mechanism [27]

2.2 Existing Approaches

Error correction within the NLP community gained significant traction following the organization of the CoNLL-2014 shared task [30], which was dedicated to rectifying grammatical errors in English essays. The concept of 'translating' grammatically incorrect sentences into correct ones, thereby rectifying errors concurrently, was pioneered in [13]. During this event, statistical machine translation (SMT) models were also successfully introduced [30]. However, with the advent of neural machine translation (NMT) models [1, 25], especially those incorporating attention mechanisms [17], these models became the cornerstone architectures.

Consequently, scholarly works emerged proposing sequence-to-sequence neural models for GEC [28, 31], while [8] integrated neural features into a phrase-based system. Moreover, [14] advocated for the utilization of a deep RNN [2] and a transformer model [27] to achieve superior outcomes. Furthermore, the introduction of a T5 model [20] was highlighted in [21].

Nevertheless, a notable challenge in employing transformers for GEC lies in their substantial demand for training data. Although the majority of GEC research has been centered on the English language, the BEA 2019 Shared Task on GEC [4] provided a framework for addressing low-resource datasets. As a result, [7, 11] achieved notable success in the "Low Resources Track," with [11] surpassing 14 submissions from the "Restricted Track" despite limited training data. The accomplishment of [7] is credited to their creation of artificial data using a noising function. Also, confusion sets extracted from a spellchecker, as suggested by [11], proved effective in addressing data scarcity. Nevertheless, despite these challenges, when compared to their outstanding performance on other tracks, the drop in results in this scenario amounted to just over 10 points [4].

Approaches to synthesizing errors synthetically can vary greatly. In languages with rich morphology and intricate rules, like Russian, [22] undertakes a comparison of error distributions in Russian versus English, along with a contrast of errors between native and non-native speakers. This is highly beneficial as artificially generated errors become more efficient when they closely resemble natural errors. In the case of Ukrainian, [24] introduces an alternative error taxonomy, diverging from the predominant use of ERRANT [5], and presents its own probability distribution of grammar errors.

Finally, in the context of Romanian language advancements, [9] crafts a specialized error distribution to effectively pre-train the model with a substantial 300k-sample dataset. Alongside, they introduce a novel dataset, RONACC, used for fine-tuning vanilla transformers. Moreover, [18] extends the research horizon by introducing alternative evaluation metrics and conducting varied experiments, enriching the ongoing discussion on linguistic analysis and system enhancement for the romanian language.

3 PROPOSED SOLUTION

3.1 Analysis of Datasets

In this section, I will present the two datasets used in our solution: RONACC and MARCELL. While RONACC is a parallel dataset containing naturally occurring grammatical errors, MARCELL is a simpler dataset that will be used in this research to obtain artificial generated errors.

3.1.1 RONACC

The authors [9] introduce, to the best of their knowledge, the first corpus for Romanian Grammatical Error Correction. The Romanian National Audiovisual Council Corpus (RONACC) is presented as a native corpus, encompassing corrected sentences spoken on Romanian TV and radio shows, alongside sentence pairs representing common written mistakes.

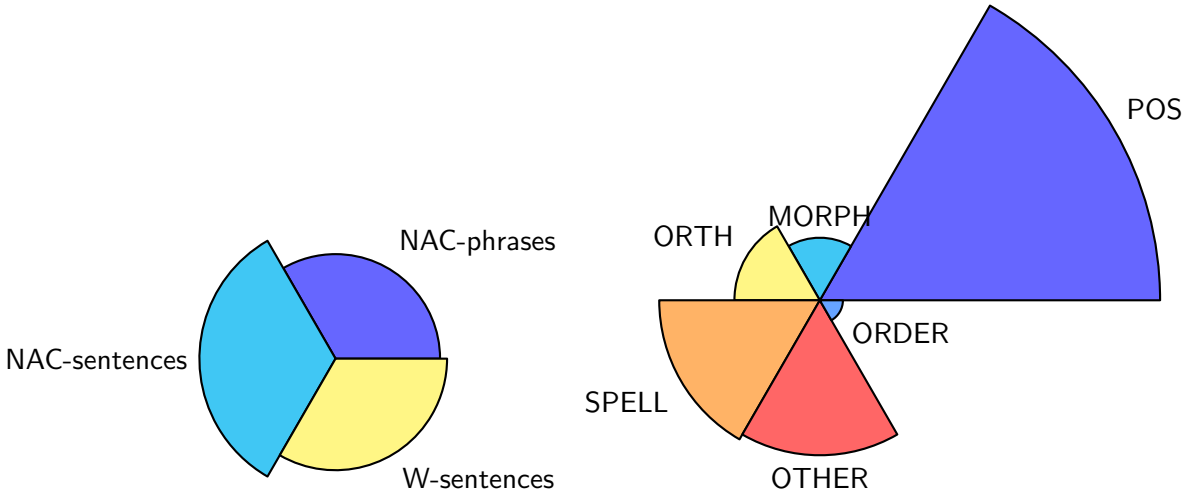

Figure 3.1 The composition of RONACC and the errors distribution

The corpus is categorized into three groups: NAC-phrases, NAC-sentences and W-sentences. NAC-phrases consist of phrases lacking verbs, NAC-sentences comprise well-formed sentences from spoken language, and W-phrases include well-formed sentences with written mistakes. Each phrase in the corpus is designed to contain sufficient information for correction without the need for additional context. The total number of GOLD labeled sentences is approximately 10k [1].

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

3.1.2 MARCELL-RO

As part of the MARCELL CEF Telecom project [26], which aims to collect and deeply annotate a large comparable corpus of legal documents, the MARCELL-RO-v2[2] corpus is presented as containing national legislative texts from the period 1881-2021.

This dataset mainly collects Romanian government decisions, orders, decrees, and laws. While this corpus spans back to laws promulgated in the 19th century, my aim is to select texts that adhere to contemporary grammatical standards.

Besides ensuring grammatical accuracy, the choice of this dataset was influenced by its exceptionally well-organized structure. The MARCELL-RO subcorpus is meticulously divided into sentences, each thoroughly analyzed, with every word annotated with pertinent information crucial for this research, such as part of speech, lemma, case, gender, number and form. The entire process was conducted using tools provided by RACAI[3]. These annotations will be particularly valuable, especially during the synthetic error generation phase.

3.2 Taxonomy of Grammatical Errors

ERRANT, the ERRor ANnotation Tool [5], is a tool designed to automatically extract edits from parallel datasets and classify them according to a framework based on over 50 rules, across various categories, thereby creating an error taxonomy. By facilitating an assessment of error types at various granular levels, this tool not only standardizes the way grammatical errors are perceived, easing the workload for researchers, but also enables obtaining a realistic and richly informative score at evaluation.

The functioning involves primarily aligning sentences based on a linguistics-driven algorithm, utilizing information such as lemma and part-of-speech [10]. Once the sentences are aligned, the next step is to identify errors and calculate the Damerau-Levenshtein edit distances between tokens of correct and incorrect sentences. To assign an error type to a correction, ERRANT uses a rule-based approach that considers information about the POS tags, lemmas, stems and dependency parses.

In this study, for the Romanian language, I utilized the ERRANT version adapted by [9], which relies on 14 Universal Dependencies (UD) POS tags instead of language-specific tags. The word dictionary is based on the vocabulary provided by the Hunspell spellchecker [4], while the POS tagging is performed using Romanian SpaCy [5].

A list of categorized error examples [9] from the RONACC dataset can be found below:

[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

Additionally, regarding the tags, they include the part of speech (e.g., *NOUN*), and an additional *FORM* tag is appended when various inflections occur. The MORPH category addresses errors related to word form, ORTH relates to capitalization and whitespace errors, SPELL indicates a spelling error, whereas ORDER refers to errors related to the reordering of tokens.

3.3 Synthetic Data Generation

3.3.1 Error Distribution Analysis

Once the synthetic error categories are established, it becomes easier to consider how the errors should look. Since the Romanian language lacks sufficient resources to train a generative-type model for GEC, the probability distribution of synthetic errors becomes crucial. Finding a good balance between these categories enables the model to learn more intelligent correlations just from the pretraining dataset.

Therefore, considering the complex morphology of the Romanian language, I have chosen to rely on a probability distribution similar to languages with slavic influence, such as Ukrainian and Russian.

In [22], the authors present a new dataset, RULEC, composed of approximately 12K sentences extracted from various essays, written by either native Russian speakers or non-native speakers. All sentences are manually annotated. Thus, in Table 2, the differences between the types

of speakers can be seen, which I find very interesting because it allows me to decide which segment of the population I want to better adapt my model to.

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 [22] (left) and types of errors with percentages for Ukrainian learners [24] (right).

The high number of spelling and punctuation errors among native speakers is notable, and I will also focus on this aspect in my research. Similarly, for the Ukrainian language, the UA-GEC dataset [24] exhibits a similar probability distribution. In their case, punctuation errors are predominant, and spelling errors also remain significant. However, I will not take into account this taxonomy of errors, which presents fluency as a distinct category.

3.3.2 Error Injection Rules

There are numerous methods to modify grammatically correct sentences. The survey [6] considers this automatic perturbation both as a general technique of **noise injection**, which can be achieved through **rule-based alterations**, and as a means of generating erroneous sentences using approaches like back-translation or round-trip translation.

To tackle this problem, I have chosen to focus exclusively on the method of **generating grammatical errors based on rules**, as highlighted in state-of-the-art research on the Romanian language [9]. This approach, recognized as one of the most intuitive methods for adding noise, involves applying rules that are determined arbitrarily, empirically, and based on observations of available data. These rules are built on a pre-defined probability distribution, with the objective of achieving a distribution similar to those discussed in Section 3.3.1.

Using the **Rule based** method typically covers most errors in the grammar and spelling categories. A particularly interesting approach is presented in the article [29], where the authors define several probability distributions used in error generation.

Accordingly, in this research, I began by defining the initial error probability distribution based on the length of a sentence, as illustrated in Table 3. Specifically, this distribution determines

the likelihood of the number of errors that can be generated.

Length	Err.	Prob.	Length	Err.	Prob.
[1, 3)	0	0.50	[6, 9)	2	0.30
	1	0.50		3	0.45
[3, 6)	1	0.50		4	0.25
	2	0.50	[16, 20)	3	0.10
[9, 16)	3	0.15		4	0.30
	4	0.25		5	0.30
	5	0.30		6	0.15
	6	0.30		7	0.15
[20, 30)	4	0.10	[30, ∞)	5	0.10
	5	0.30		6	0.20
	6	0.30		7	0.20
	7	0.15		8	0.30
	8	0.15		9	0.20

Table 3: Error Injection Probability Distribution by Sentence Length

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.

After that, I defined two additional probability distributions [29] in Table 4: one for the type of error that will be generated, and the other for the number of misspells in a token, if the misspelling method is initially chosen.

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

Table 4: Token Error Types (left) and Misspells Probability Distribution (right)

The first three methods are quite self-explanatory and simple to comprehend, but they have low priority within the error generation process. The central point of interest is the modeling of Misspell and Substitution errors, as these methods truly generate the most complex grammatical mistakes. Additionally, Punctuation errors are addressed separately.

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.

Furthermore, there is an attempt to add several methods characterized by greater adaptability to the Romanian language, constructed in a creative manner.

The first method, called **Articulated Suffix Removal**, focuses on addressing specific errors related to Romanian grammar. In Romanian, definite articles are often formed by adding suffixes such as 'ul' or 'a' to the base form of a word, for example, turning 'baiat' (boy) into 'baiatul' (the boy) and 'fată' (girl) into 'fata' (the girl). By removing these suffixes, the model can better predict the correct word forms. Furthermore, a common grammatical error among native speakers involves miscounting the number of 'i' in the plural form of words already containing an 'i'. For instance, the word 'copil' (child) becomes 'copii' (children) in plural, and with the definite article added, it becomes 'copiii' with three 'i's. This method aims to improve the accuracy of grammatical predictions by focusing on these specific rules of Romanian grammar.

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

Table 5: Probability Distribution of Misspelling Error Types

The second method, called **Controlled Letter Replacement**, involves selecting a character and replacing it with one or more potential alternatives. Each character's list of alternatives is generated by examining adjacent keys on a standard Romanian keyboard, characters with similar shapes, and through careful analysis of common grammatical errors in Romanian, including both misspellings and mispronunciations. For example, the word 'celălalt' is frequently mispronounced as 'celălant', which prompts consideration of replacing the character 'l' with 'n'. Similarly, words such as 'culoarea ridichii' and 'președințe' are often misspelled

as 'culoarea ridichiei' and 'președinție', leading to the replacement of the letter 'i' with 'e' or 'ie'.

The probabilities associated with these methods, which form the misspelling rule, can be seen in Table 5.

Substitution Errors

This may be one of the most important methods, as it generates more complex errors that are linked to the deeper meaning of a sentence and can thus most accurately mimic the natural errors made by Romanian speakers.

The first substitution method utilizes **RoWordNet** [6], the Romanian version of the WordNet lexical database. This approach involves substituting a word token with a synonym from its synonym sets (synsets) provided by WordNet. The synsets are filtered based on their relevance to the context of the sentence, and a synonym is selected. This method serves a dual purpose: it enhances the model's understanding of the Romanian language by teaching it synonyms, and it aids in error generation.

Type	Prob.
RoWorNet	0.20
DexOnline	0.40
Spellchecker	0.40

Table 6: Probability Distribution of Misspelling Error Types

The second method uses **DEXOnline** [7], the web version of the explicative dictionary of the Romanian language. This approach involves substituting a word token with a conjugation or declination of the word. By utilizing web scraping, all potential conjugations or declensions of a word are extracted, with one of the available variants being selected to substitute the original word.

Additionally, a new approach has been introduced to save time and achieve comparable outcomes, while also enhancing creativity and generalization. If a word is found within a predefined list containing prepositions, conjunctions, possessive articles, demonstrative articles, or indefinite articles, the corresponding type of list is identified. Twenty-five percent of the time, a candidate is chosen from a created **confusion list**, if not, web scraping will be performed.

The third and final method is the **Spellchecker** method, designed to replace a word token with a similar word, which may or may not be a conjugation of the original word. This method utilizes the Phunspell spellchecker [8], a Python wrapper for the Hunspell spellchecker. It identifies words closely resembling the token by calculating their edit distance against all

[6] https://github.com/dumitrescustefan/RoWordNet
[7] https://dexonline.ro
[8] https://pypi.org/project/phunspell/

other words in its Romanian vocabulary. Once a list of suggestions is generated, one of them is selected to replace the token.

The probabilities associated with these methods, which form the substitution rule, can be seen in Table 6.

Punctuation Errors

To synthetically generate **punctuation errors**, the researchers in [3] developed an error probability matrix. This matrix allows each punctuation mark (including spaces between words) to be replaced with any other punctuation mark based on a randomly generated probability. This matrix is then applied to each sentence in the dataset. The resulting matrix demonstrates the probabilities of substituting one punctuation mark for another.

Within this research, the probabilities have been adjusted to better fit the Romanian language, as shown in Table 7.

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
-	0.80	0	0	0	0.20	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 7: Probability Distribution of Punctuation Error Types

3.3.3 The MEID (MARCELL Error Injection Dataset)

In this study, MEID [9] is introduced, a parallel dataset containing pairs of sentences with artificially generated errors based on the rules outlined in this chapter. As the name suggests, this dataset is built using legal documents that comprise the Romanian MARCELL dataset. One of the main reasons for choosing this dataset is its very low error rate in legislative documents. On the other hand, MARCELL is a very large dataset, allowing for flexibility in filtering and preprocessing.

Consequently, the dataset is filtered to include only documents dating from the year Romania joined the European Union in 2007. This reduces the number of documents used in this study from 163K to 104K, ensuring the dataset remains relevant. From these, sentences containing

[9] `https://huggingface.co/datasets/mateiaassAI/MEID`

fewer than 10 tokens are excluded. Ultimately, approximately 2 million sentence pairs are obtained, limiting the training set, although there is potential to obtain more data. The dataset generation process takes several days using only an i7-12800H processor and 32 GB RAM. The data is initially saved as JSONL files, which are then packed into multiple Parquet files and uploaded to Hugging Face, where the dataset is now hosted.

Errant Taxonomy Evaluation

Continuing, the error distribution in MEID is being presented based on the ERRANT taxonomy introduced in the previous section. It's important to note that these results are statistical and have been derived from evaluating a sample of approximately 30K examples, which is considered sufficient for generalization. The analysis utilizes ERRANT scripts to generate and compare m2 files. These files are produced by comparing altered sentences with their correct counterparts and by comparing correct sentences with themselves. The final step consists of comparing the m2 files to obtain the numbers presented in Table 8.

Category	Percentage
PUNCT	31.13%
SPELL	23.00%
OTHER	21.31%
ORTH	7.48%
WO	4.27%
NOUN:FORM	3.00%
ADP	2.82%
NOUN	2.70%
MORPH	0.67%
CCONJ	0.67%
VERB	0.61%
VERB:FORM	0.60%
ADJ	0.43%
DET:FORM	0.26%
PRON	0.23%
DET	0.19%
PRON:FORM	0.07%
ADV	0.07%
K	0.01%
ADJ:FORM	0%

Table 8: MEID distribution of errors

As anticipated, punctuation errors are the most prevalent, occupying the top spot. Following closely, SPELL errors are the second most frequent, bolstered by the contributions of the Hunspell spellchecker and other word-level modifications. Furthermore, there is a correct proportion of ORTH-type errors, although the low number of morphological errors generated is somewhat disappointing.

3.4 T5

Given the immense size of modern machine learning models, training them from scratch is nearly impossible due to the vast amount of data and computational power required. This challenge is effectively addressed by transfer learning [19], a technique that allows a model to leverage knowledge gained from solving one problem and apply it to a related task. In this approach, a model is pre-trained on a large, high-quality corpus to learn language structure, grammar, and semantics, and then fine-tuned for a specific downstream task.

The text-to-text transformer (T5) model [20] proposes a unified framework for studying transfer learning approaches in NLP, allowing for the analysis of different settings and the derivation of best practices. Among the methods it includes are machine translation, classification tasks, regression tasks, and other tasks such as summarization.

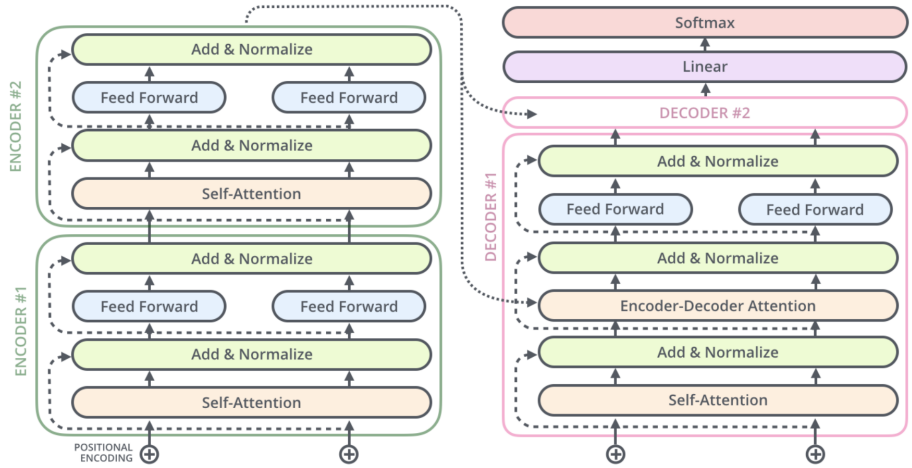

Figure 3.2 T5 Architecture [10]

The architecture of a T5 model does not differ from that of a vanilla encoder-decoder transformer. The official version used Common Crawl web-extracted text for training. For this paper, a variant trained on 80GB of Romanian text was used.

The **t5-base-romanian** [11] is a pre-trained version from scratch of the T5v1.1 base model (247 parameters), which has an encoder sequence length of 512 and a decoder sequence length of 256. Given that the model has been trained on multiple tasks, we will leverage the fact that GEC functions as a type of machine translation. We will provide it with sentence pairs along with a suggestive prompt. Each incorrect text will be formatted as follows: "Corectează: {*grammatically altered text*}", since I've observed slight improvements with this approach.

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

4 EVALUATION AND EXPERIMENTS

In this chapter, the conducted experiments will be presented, followed by an overview of the evaluation metrics and an interpretation of the obtained results.

4.1 Experimental Settings

As mentioned in the previous chapter, the newly formed dataset MEID will be used for pretraining the GEC task. The architecture is based on a T5 model in its base version, tailored for the Romanian language. Following the pretraining phase with MEID, the model will be fine-tuned on the main dataset, RONACC, which will also be used for evaluation. Next, I will present the experiment details

4.1.1 Data Preprocessing

Although I have published MEID, a parallel dataset comprising approximately 2 million sentence pairs, I found that meaningful results can be achieved with only a subset of these pairs. Furthermore, I considered several important filtering criteria. I believe that every sentence written in Romanian should end with a punctuation mark. I observed that MARCELL, and consequently MEID, contain a significant percentage of sentences that do not end with a punctuation mark. After filtering the sentences to include only those with at least 10 words and ending with one of the following punctuation marks: period, semicolon, ellipsis, exclamation mark, or question mark, the dataset used for pretraining was limited to 1 million sentence pairs.

4.1.2 Training Details

The model was trained using the free resources provided by Kaggle [1]. Therefore, training was conducted utilizing the capabilities of the P100 GPU, which has 16GB of RAM.

Regarding the hyperparameters that needed to be set, among those general and valid for both pretraining and fine-tuning, I specify: *input length*, *decoder generation length*, and *decoder length*. Following an analysis of the MEID dataset, I noticed that approximately 90% of the sentences are shorter than 500 words. Therefore, for the *input length*, I considered truncating the examples with the same value the T5 model was pre-trained with, which is 512. As for

[1] `https://www.kaggle.com/`

the *decoder length*, I remained consistent with 256, as it was pre-trained similarly. However, concerning the *decoder generation length*, I used a constant value of 64, allowing the model to generate fluent text, even if it means exceeding the initial length.

For adjusting the other hyperparameters, I used the Weights&Biases platform. [2].

PreTraining Details

Regarding the pretraining process, checkpoints were used, and the training was split into 2 parts, with a checkpoint triggered when reaching 500K examples. This allowed me to adhere to the maximum 12-hour limit for a single session.

It should be noted that for the first 500K examples from MEID, a learning rate of 3.5e-4 was used, while for the last 500K examples, the learning rate was adjusted to 1.75e-4. Additionally, the batch size was set to 4, and the weight decay was 0.01. The model was pretrained for only one complete epoch. The training dataset consisted of 1 million sentences, with approximately 10K sentences reserved for testing.

FineTuning Details

For the fine-tuning phase, I used the same official data split as presented in the state-of-the-art RONACC model [9]: 70% for training, and 15% each for validation and testing. However, in my training process, I concatenated the training and validation sets, retaining only the test set for evaluation.

It should be noted that the training was conducted over 10 epochs, with a batch size of 16 and a learning rate of 3e-5.

4.2 Evaluation Metrics

To evaluate the corrections made by the model, a confusion matrix will be created with the help of ERRANT to capture the similarities and differences between the edits required for correcting the sentences. Using this confusion matrix, the precision and recall of the model's predictions will be calculated.

Given that precision and recall alone do not fully capture our model's performance, and the F1 score's balanced approach may not be ideal, I focus on minimizing false positives by prioritizing precision over recall. Therefore, the primary evaluation metric will be the F0.5 score, which emphasizes precision more heavily.

[2] `https://wandb.ai/site`

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

In this chapter, the outcomes of the experiments and evaluations will be presented in detail. The results are divided into three main subsections: the performance evaluation using the ERRANT taxonomy, the qualitative analysis, and the state-of-the-art comparison.

4.3.1 Performance Evaluation

For my architecture, which consists of a T5-base model pretrained on the MEID dataset and fine-tuned on the RONACC dataset, the final testing was conducted using beam search decoding with a beam width of 8. The performance evaluation results are presented in Table 9 as a confusion matrix based on the error categories defined by the ERRANT taxonomy.

POS	TP	FP	FN	Pr	Re	F0.5
ADJ	2	9	16	18.18	11.11	16.13
ADJ:FORM	11	7	17	61.11	39.29	55.00
ADP	110	41	85	72.85	56.41	68.84
ADV	30	19	21	61.22	58.82	60.73
CCONJ	7	12	8	36.84	46.67	38.46
DET	23	18	24	56.10	48.94	54.50
DET:FORM	36	6	14	85.71	72.00	82.57
MORPH	15	5	12	75.00	55.56	70.09
NOUN	16	48	52	25.00	23.53	24.69
NOUN:FORM	111	43	70	72.08	61.33	69.64
ORTH	67	21	97	76.14	40.85	64.92
OTHER	176	137	235	56.23	40.84	52.29
PRON	0	68	7	0	0	0
PRON:FORM	4	0	4	100	50.00	83.33
PUNCT	151	120	150	55.72	50.17	54.51
SPELL	366	131	194	73.64	65.36	71.82
VERB	3	13	39	13.33	4.88	9.99
VERB:FORM	85	12	59	87.63	59.03	79.89
WO	8	3	8	72.73	50.00	66.67
Total	1220	713	1132	**63.11**	**51.87**	**60.49**

Table 9: T5-base Performance Results

The results show mixed outcomes, with notable performance in critical areas. Spelling errors are well-detected, with an F0.5 score of 71.82, indicating high certainty. Punctuation corrections are more challenging, reflected in an F-score of 54.51. Reliable error corrections are also observed in verb form, pronoun form, noun form, word order, and determiner form,

highlighting the model's ability to understand and correct various inflections of different parts of speech. This is one of the benefits of using confusion sets and the DexOnline method in generating artificial errors.

However, the lower score obtained in the case of parts of speech without [FORM] signifies the need for injecting more synonyms, which could lead to a better understanding of the correct words and their meanings in their context.

4.3.2 Qualitative Analysis

Regarding the qualitative analysis, I selected four examples from the dataset that I found relevant and which are also presented in the state-of-the-art [9]. The analysis is available in Table 10.

Original	Gold	Predicted
Oameni nu **ii judeca** pe **barbatii ca** nu sunt **tati** buni	**Oamenii** nu **îi judecă** pe **bărbați că** nu sunt **tați** buni	**Oamenii** nu ii judeca pe **barbati** ca nu sunt **tați** 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.	**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.	**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 **trecuta** pentru **opeartie v-a fii** externat mai **tarziu** din spital.	Mohamed**,** care a sosit la Londra luna **trecută** pentru **operație, va fi** externat mai **târziu** din spital.	Mohamed**,** care a sosit la Londra luna **trecută** pentru **operațiune, va fi** externat mai **târziu** din spital.
in ultimile trei zile	**în ultimele** trei zile	**în ultimele** trei zile

Table 10: Qualitative Analysis

From this analysis, it is observed that there could be improvements in detecting diacritics. The model often fails to correct missing or incorrect diacritical marks, which are crucial for accurately representing Romanian language nuances. However, overall, the model performs well in identifying spelling and orthographic errors.

However, the substitution of the incorrect word "opeartie" with "operatiune" instead of "operatie" indicates the model's struggle with accurately replacing nonsensical nouns. This suggests that while the model can handle basic spelling corrections, it may not fully understand the contextual meaning of certain words, leading to less accurate corrections in more complex sentences. This highlights the need for further refinement, especially in handling semantic context and word usage.

4.3.3 SOTA Comparison

This section presents a comparison between the results obtained by our T5-GEC model and those achieved by the Neural Grammatical Error Correction for Romanian (RoGEC) model, as detailed in [9], which represents the current state-of-the-art in this domain. The comparative results are summarized in Table 11.

The T5 model demonstrates significant advancements over the RoGEC model, particularly when utilizing beam search decoding. This improvement is attributable to the T5 model's larger and more intricate architecture, which enables it to capture more nuanced patterns in the data and the MEID dataset on which our model was pre-trained. The quality of the dataset plays a crucial role, as does the difference in pre-training scale: RoGEC was pre-trained on only 300K examples, while our model was pre-trained on 1 million examples in this research.

In terms of fine-tuned results, the T5 model with beam search achieves a substantial F0.5 score of 60.49, surpassing RoGEC's 53.76, highlighting the effectiveness of our approach in reducing grammatical errors.

Moreover, the T5 model demonstrates robust performance even without beam search, achieving results that still surpass those of RoGEC with beam search. This consistency underscores the model's inherent capability to generalize well across different data distributions and error types.

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
T5			
Artificial data	30.14	23.09	28.39
Fine-tuning	55.23	64.79	56.91
T5 with Beam Search			
Artificial data	30.82	23.29	28.95
Fine-tuning	63.11	51.87	60.49

Table 11: Comparison with State-of-the-Art

Despite these achievements, the analysis also reveals areas for improvement. Certain grammatical categories, such as those without specific forms, do not perform as well, indicating a need for further refinement. This gap highlights the importance of continuous development, particularly in enhancing the model's ability to understand and correct more complex syntactic and semantic errors.

5 CONCLUSIONS

In conclusion, this research can be considered a success as it has achieved the objectives set forth in the introduction. The Romanian language has very few resources dedicated to the task of grammatical error correction, making the publication of a new parallel dataset, MEID, composed of 2 million sentence pairs, a significant improvement.

Furthermore, the methodology used to generate errors was creative, drawing inspiration from recent papers and the state-of-the-art in the GEC task for the Romanian language, aimed at enhancing the quality of artificial data. An additional advantage of the published dataset is its size, as it stands as the largest parallel dataset for GEC in the Romanian language to my knowledge.

Moreover, employing a modern T5 architecture, even in its base form and with minimal hyperparameter tuning, yielded quite promising results, surpassing by a significant margin the most performant existing solution to date.

Finally, this work enhances the capabilities of the Romanian language in digital environments by developing an efficient grammar checker. It supports communication, aids learners, supports professionals, and enriches digital content creation, positioning Romanian to leverage the latest technological advancements in modern communication.

5.0.1 Future Directions

Regarding future directions, I will certainly attempt to generate a parallel dataset containing artificially generated grammatical errors in a more balanced manner, with a probability distribution aimed at addressing the weaknesses of the current MEID dataset.

Additionally, I will explore training with other architectures and better parameter tuning to enhance model performance and accuracy.

Furthermore, I plan to investigate and implement creative methods for error generation that can improve generalization, including the consideration of fluency factors when correcting text.

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] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

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

[10] Mariano Felice, Christopher Bryant, and Ted Briscoe. Automatic extraction of learner errors in ESL sentences using linguistically enhanced alignments. In Yuji Matsumoto and Rashmi Prasad, editors, *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 825–835, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[11] 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.

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

[13] 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.

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

[15] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, page 48–54, USA, 2003. Association for Computational Linguistics.

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

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

[18] 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.

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

[20] 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.

[21] 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.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

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

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

[29] 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.

[30] 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.

[31] 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.