Melnyk, Lidiia and Feld, Linda (2023) On Methods of Data Standardization of German Social Media Comments. Journal of Computer-Assisted Linguistic Research 7: 22-42. ISSN: 2530-9455.

On Methods of Data Standardization of German Social Media Comments

Lidiia Melnyk*, Linda Feld*
Friedrich Schiller University Jena, Germany
*Corresponding authors: lidiia.melnyk@uni-jena.de; linda.feld@uni-jena.de

Received: 22 June 2023 / Accepted: 1 September 2023 / Published: 18 December 2023

Abstract
This article is part of a larger project aiming at identifying discursive strategies in social media discourses revolving around the topic of gender diversity, for which roughly 350,000 comments were scraped from the comments sections below YouTube videos relating to the topic in question. This article focuses on different methods of standardizing social media data in order to enhance further processing. More specifically, the data are corrected in terms of casing, spelling, and punctuation. Different tools and models (LanguageTool, T5, seq2seq, GPT-2) were tested. The best outcome was achieved by the German GPT-2 model: It scored highest in all of the applied scores (ROUGE, GLEU, BLEU), making it the best model for the task of Grammatical Error Correction in German social media data.

Keywords: Grammatical Error Correction, LanguageTool, data augmentation, seq2seq, T5, GPT-2

1. Introduction

During the last two decades, large parts of discourses of all kinds have shifted from the non-virtual world to the virtual one. Opinions and arguments are being exchanged in a fast and often anonymous way on various online platforms, such as Twitter, Facebook, or YouTube. While those discourses (carried out online) represent a valuable resource for detecting trends of opinion, sentiment, and stance and can give insights into the linguistic realization of these, the often messy nature of textual social media data poses major challenges for computationally analyzing the data. It is the purpose of this study to address these challenges and find the most effective methods to standardize such data.

More specifically, the study is part of a larger project concerned with qualitatively and quantitatively analyzing YouTube comments regarding the topic of gender diversity – a topic that has received increasing attention and turned into a prominent and polarizing topic within

DOI: HTTPS://DOI.ORG/10.4995/JCLR.2023.19907
PUBLISHER: UNIVERSITAT POLITÈCNICA DE VALÈNCIA.

social, political, as well as scientific discourses, ultimately boiling down to the question of what defines a human being and their identity. YouTube – as a free and anonymous platform – is one of the various contexts in which ideas, opinions, and arguments relating to the topic of gender diversity are being exchanged. Ultimately, our project aims to give a holistic characterization of the discourse in terms of the content and the quality of the comments. Therefore, the first task in our project was to classify the comments according to their sentiment and stance (see Melnyk and Feld 2022). In order to approach the final task of classifying the data in terms of discursive strategies and analyzing the linguistic means employed in the comments, it is the aim of the study at hand to standardize the data (that were scraped from the comments sections below YouTube videos) and, thereby, provide error-free data that are required for the following tasks.

Therefore, we first applied an existing rule-based tool to correct the data, namely, LanguageTool (Section 3). The suggested corrections were validated by three annotators. Since the corrections suggested by the tool were not satisfactory, we then drew on language models frequently used to solve Natural Language Processing (NLP) tasks such as correcting spelling or grammar, namely a T5-based model, a German version of GPT-2, and a sequence-to-sequence (seq2seq) model with monotonic attention (Section 4). We used the validated corrections from the LanguageTool's output to train the models, and, due to the small amount of annotated data, we artificially augmented our data to achieve better performance of the models. The fine-tuned T5-based model demonstrated better performance than the custom-built seq2seq model when it comes to spelling and punctuation correction, but failed to correct casing errors. In total, the T5 model correctly identified 73% of errors, whereas the custom seq2seq model could not provide any coherent outcome. Both models were significantly outperformed by the German GPT-2 model, which was able to detect and correct 92% of errors that were also detected by our human annotators.

2. Data Sampling and Description

The comments were scraped according to a list of keywords linked to the videos and we used a JavaScript code in Google Apps Script in combination with the LangID tool in Python to filter out non-German comments. In total, about 383,000 unique comments from 450 videos posted between 2015 and the beginning of 2022 were gathered along with their metadata (link, creation date, author's name, number of likes and replies). Because of the vanishingly small number of comments before the middle of 2017, comments preceding September 2017 were excluded, resulting in a corpus of 350,000 comments.[1]

Social media data, and particularly the kind of data we are dealing with, i.e., short German texts or comments posted below YouTube videos, are not bound to any linguistic rules, which makes them highly unstandardized in terms of spelling and punctuation. Moreover, since the topic of the data is a highly controversial one, generating heated discussions and emotionally loaded arguments, large parts of the data reflect this strong emotional influence, resulting in even less standardized language use. Thus, due to the characteristics of social media discourses

[1] For a more detailed account of the corpus creation and description see Melnyk and Feld (2022). The corpus is available at https://www.kaggle.com/datasets/lidiiamelnyk/youtube-comments-on-gender-diversity.

(anonymous, quick, and emotionalized), the comments exhibit a high degree of irregularity, containing emojis, special characters, misspelled words, abbreviations, slang and dialectal variation, as well as a lack of or incorrect punctuation and grammar and even omissions. Consider the following example:[2]

(1) Viele Frauen wissen nicht einmal **das** sie dazu nicht in der Lage sind, werden dadurch **Depressiv** oder ähnliches. Abgesehen davon gibt es einige Krankheiten die die Fruchtbarkeit sowohl bei Männern als auch bei Frauen enorm verhindern, was es schwer macht **Kinder** zu zeugen. Ich find es auch so schlimm **das** wir heute im **21** Jahrhundert **immernoch teilweide** eine Denkweise haben vom 18 Jahrhundert.
(Many women do not even know that they are not able to do this, become depressed or something similar. Apart from that, there are some diseases that prevent fertility enormously in both men and women, which makes it difficult to have children. I also think it is so bad that today in the 21st century we still partly have a way of thinking from the 18th century.)

This comment exhibits a variety of errors: 1) missing comma before 'das' and the article 'das' is confused with the conjunction 'dass' (twice), 2) incorrect capitalization of the adjective 'depressiv', 3) missing comma before the relative pronoun 'die', 4) missing comma before the infinitive clause 'Kinder zu zeugen', 5) missing full stop after ordinals '21.' and '18.', 6) missing whitespace between 'immer' and 'noch', and 7) typo in 'teilweise'.

To be able to qualitatively work with the data, i.e., to computationally process the comments and annotate discursive strategies within them, standardized data are required. With this goal in mind, we set out to find the most reliable method for standardizing textual social media data.

3. Automated Spelling Correction with LanguageTool

3.1. LanguageTool API Overview

"Grammatical Error Correction (GEC) is the [NLP] task of correcting different kinds of errors in text such as spelling, punctuation, grammatical, and word choice errors." (Papers with code) Plenty of apps have been developed that build on different GEC approaches from rule-based models to transformers and large language models, promising fairly reliable, accurate, and fast correction of grammatical errors in the input data. However, most of these applications lack empirical assessment in terms of their accuracy of performance, as pointed out by Sahu et al. (2020). To fill this gap, these authors manually created a dataset of 500 sentences containing grammatical errors and annotations of the particular types of errors and tested the performance of five different AI-based apps: Grammarly, Ginger, LanguageTool, ProWritingAid and After the Deadline. The best results were achieved by Grammarly, outperforming the other apps by up to 15.6% in overall accuracy.

Unfortunately, the tools exhibiting the best scores (Grammarly and ProWritingAid) do not

[2] Except for the examples (2) to (7) in Section 3.1, all of the listed examples are taken from our own corpus.

support German language texts yet. For that reason, LanguageTool[3] (ranking third in Sahu et al.'s (2020) study) was chosen for the correction of grammatical errors in our dataset. Similar to Grammarly, the top performer, LanguageTool follows a rule-based approach. It supports multiple languages, including German, and has a free-of-charge Python API. The rules it utilizes come from different sources such as language reference books and grammars, corpus analysis, community feedback, and linguistic research.

Among other resources, the tool is based on Hunspell, "a spell checker and morphological analyzer library and program" (McNamara et al. 2015).[4] Other resources that LanguageTool draws on for correcting German text are Jan Schreiber's list of German words,[5] POS-tagging data provided by Morphy, a freely available software package for morphological analyses in German,[6] and xxx-frami, a standard dictionary additionally containing words that do not belong to the core German vocabulary.[7] Moreover, LanguageTool leverages error collections containing errors found in different online resources and in e-mails.[8] In total, LanguageTool can detect 4,892 types of errors in German text, including 297 errors relating to punctuation and commas, as in (2), where there is a comma missing before 'sondern' ('but'), 1,554 errors relating to casing, as in (3), where the nominalized verb 'laufen' ('to walk') is not capitalized, 447 possible typos, as in (4), where an 's' was confused with an 'r', 451 easily confusable words, as in (5), where 'weist' ('to show') is confused with 'weißt' ('to know'), 653 errors regarding compounds, as in (6), where spaces were incorrectly placed within a compound consisting of an adjective and an infinitive clause with 'zu', and 527 grammatical errors, as in (7), where the grammatical gender of the noun 'Haus' (neuter) is not congruent with the grammatical gender of the article 'der' (masculine).[9]

(2) *Es ist nicht Sommer **sondern** Winter.
(It is not summer but winter.)

(3) *Das **laufen** fällt mir schwer.
(Walking is hard for him.)

(4) ***War** für eine riesige Überraschung!
(What a huge surprise!)

(5) *Das Auto **weißt** einige Kratzer auf.
(The car has several scratches.)

(6) *Er überprüfte die Rechnungen noch einmal, um ganz **sicher zu gehen**.

[3] Available at: https://languagetool.org/de/.

[4] See also http://hunspell.github.io/.

[5] Available at: https://sourceforge.net/projects/germandict/.

[6] Available at: https://morphy.wolfganglezius.de/.

[7] Available at: https://extensions.libreoffice.org/en/extensions/show/german-de-de-frami-dictionaries.

[8] The collection of German errors and incorrect words is based on Wikipedia (e.g., http://de.wikipedia.org/wiki/Rechtschreibfehler) and lists of incorrect words compiled by individual people (e.g., http://www.frank-roesler.de/dsdr.html or http://www.oberlehrer.org/naf.html). See https://dev.languagetool.org/error-collections for more details.

[9] https://community.languagetool.org/rule/list?lang=de. Examples (2) to (7) are taken from that list. Emphasis in bold was added.

(He checked the bills again to be sure.)

(7) ***Der** Haus wurde letztes Jahr gebaut.
(The house was built last year.)

The tool functions by splitting the given text into sentences and the sentences into words and assigning a part-of-speech (POS) tag to each word. After that, the POS-tagged text is matched against the rules provided by the respective XML file and Java code (containing rules that cannot be expressed as XML rules). Thus, LanguageTool does not correct the sentences by comparing them to correct ones but by mapping them against the rules defining the various errors a sentence can contain (LanguageTool).

3.2. Validation of LanguageTool's Output

LanguageTool's mode of operation creates the impression that the tool can reliably detect and correct any linguistic errors. However, as a rule-based approach it can, as outlined by Wang et al. (2021), generally suffer from multiple limitations, including an inability to account for the flexibility of naturally occurring language and the multitude of exceptions that characterize it, a lack of linguistic resources to generate an exhaustive set of rules for minority languages, as well as the amount of time and effort that is needed to develop such rules.

In order to examine how well LanguageTool performs on our data and to make the validation process more transparent and organized, the GEC was divided into three different categories, and the validation of each category was carried out step by step.[10] The first category to be corrected by the tool and validated by the annotators was casing. Then the annotators proceeded with the validation of the punctuation corrections and, finally, the spelling corrections.

Casing: Two categories of mistakes were checked: (1) wrong casing at the beginning of the sentence and (2) wrong casing of the words within a sentence. Of all the errors identified by LanguageTool, 20% were of the first type of error and 14% were of the second type. While only 6% of the corrections of the first type were identified as incorrect by the annotators, the percentage of corrections of the second type that were identified as incorrect is comparatively high, namely 20%.

Punctuation: LanguageTool identified and corrected 431 punctuation errors. Common reasons for the tool to detect a punctuation error are depicted in Figure 1.

[10] The annotators were two students (one proficient in German and the other a native speaker of German). Wherever there was a disagreement, a third annotator was consulted to make a decision.

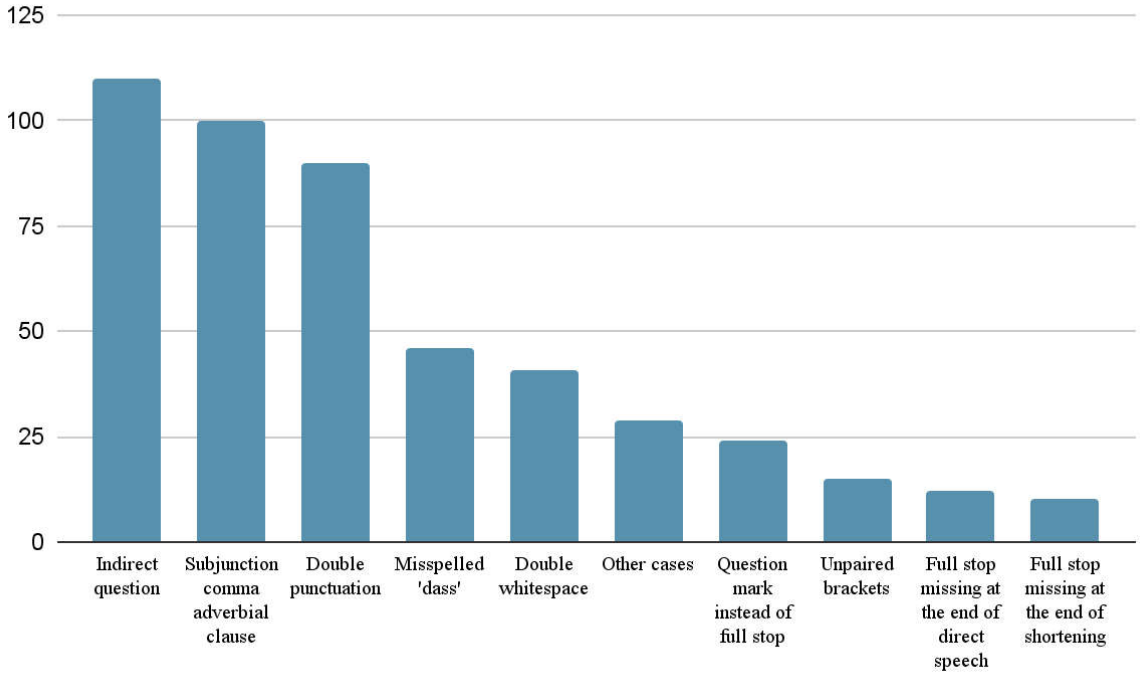

FIGURE 1. MOST COMMON PUNCTUATION CATEGORIES

10% of LanguagTool's punctuation corrections were marked as incorrect by the annotators. Among these, the following patterns occurred rather frequently: question marks at the end of indirectly reported questions, as in (8), detection of adverbial clauses of reason but the insertion of a comma at the wrong place, e.g., before 'weil' ('because'), as in (9), or detection of declarative content clauses because of misspelled article 'das' ('the') as 'dass' (conjunction 'that') and insertion of a comma, as in (10).

(8) Ich frage mich ob Trans eigentlich ein hervorgerufenes Phänomen der gesellschaftlichen Vorstellung von Geschlechtern ist**?**
(I wonder if Trans is actually a phenomenon caused by society's idea of gender)

(9) [...] das änder sich nicht **nur, weil**'s dich nicht betrifft.
([...] that doesn't change just because it doesn't affect you.)

(10) [...] und auch gerade Frauen die mal opfer von sexuellen Übergriffen geworden sind **wollen, dass** in der umkleide denke auch nicht unbedingt sehn.
(And especially women who have been victims of sexual assault also don't necessarily want to see that kind of thing in the changing room.)

Spelling: LanguageTool not only pays attention to the incorrect spelling of words but also to other phenomena such as missing, double, or misplaced whitespaces. In total, the tool identified and corrected 2,298 spelling errors in 2,411 possible cases. Figure 2 illustrates the six most frequent categories that were detected (in some cases as in the last category of the graphic below, LanguageTool did identify the error but could not match it to any of its categories).

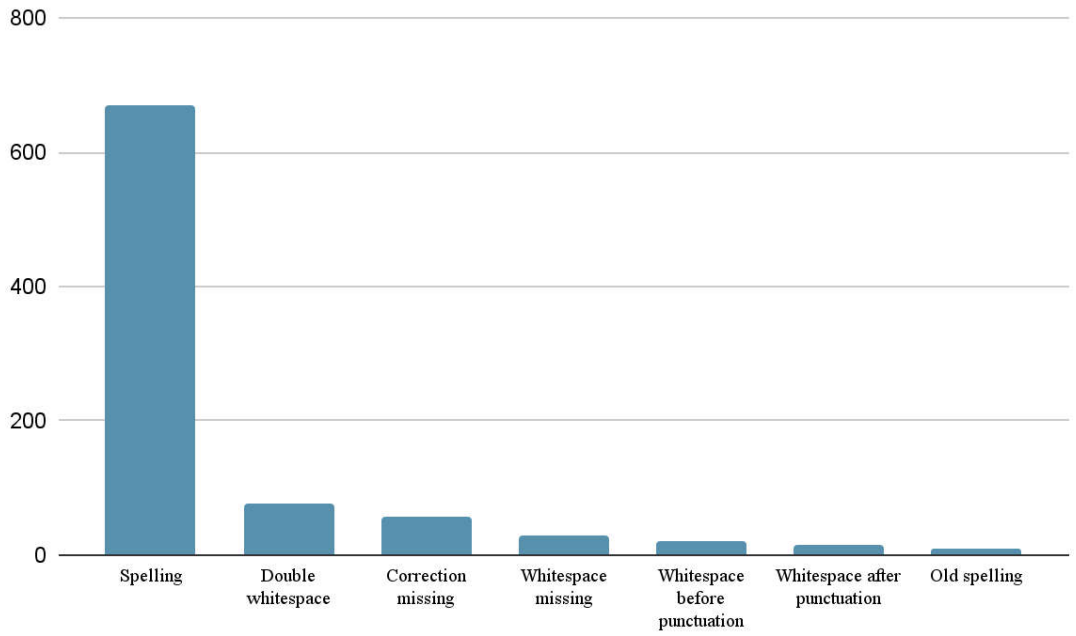

FIGURE 2. MOST COMMON SPELLING CATEGORIES

The most frequent category is the one pertaining to misspelled words in a narrow sense (80%), followed (with some distance) by double whitespaces (11.5%) and missing whitespaces after commas, full stops, and parentheses (3.7%). The annotators agreed with 86% of the spelling corrections suggested by LanguageTool. In the suggestions declined by the annotators, the tool, for instance, did not understand abbreviations, as in (11), where it corrected 'xD So' to 'PDS', or was incapable of properly correcting the misspelled word, as in (12), where the tool corrected 'Trnas' (misspelled 'Trans') to 'Trias':

(11) Das du die Brücke zwischen diesen Mädchen und der lgbtq Bewegung schlägst sagt wirklich viel über dich aus. Für was sind die noch so alles Schuld deiner Meinung nach ? **xD So** einen absoluten.... habe ich wirklich schon lange nicht mehr gelesen.
(That you make a connection between these girls and the LGBTQ movement says a lot about you. What else do you blame them for, in your opinion? xD I haven't read such an absolute... in a long time.)

(12) Das ist bloß kein wirkliches Argument. Frauen sind schon in den Frauenumkleidekabinen. Wenn jetzt aber eine Schüler, der **Trnas** ist, in die Mädels Umkleide kommt, dann ist das was anderes.
(That's not a real argument. Women are already in women's changing rooms. But if a student who is transgender enters the girls' changing room, that's a different matter.)

To ensure the quality of the annotation, inter-annotator agreement (IAA) was calculated using Cohen's Kappa score (Landis and Koch 1977). The best IAA was achieved for the casing task with a score of 0.94, followed by the spelling correction with 0.80 and the punctuation correction with 0.77. A closer look at the discrepancies sheds some light on the various reasons for the different annotation choices.

Casing: For instance, the automatic omission of (linked) names led to confusion in terms of capitalization of the (not necessarily semantically) first word of sentences as in:

(13) Team Gelb argumentiert sehr emotional, egozentrisch und sehr subjektiv. Die Gesellschaft hat sich nach ihnen zu richten. **[omitted name]** hat diese Thematik in seinen Netflixspecials gut aufbereitet.
(Team Yellow argues very emotionally, egocentrically, and subjectively. Society has to conform to their demands. [omitted name] has addressed this issue well in their Netflix specials.)

Here, the omitted POS is the sentence's subject. Consequently, capitalization of 'hat' (verb) is grammatically incorrect but correct in terms of 'hat' being the (new) first word of the sentence.

Punctuation: The most common disagreement between the annotators in terms of punctuation occurred in those cases where the tool inserted a comma between 'nur' or 'nicht' and 'weil', as in (14).

(14) [...] **Nur, weil** es nicht um ihre Geschichte ging! ('Nur weil' in the original)
([...] Just because it wasn't about her story!)

Spelling: Unsurprisingly, the annotators' decisions only differed with respect to misspelled words, and within this category, the annotators disagreed in only 6.7% of the cases, many of which involved casing issues regarding specialized vocabulary, as in (15), or anglicisms, as in (16). Casing has already been checked as a separate category, but also identified as a new rule within the spelling category:

(15) Jeder kann sich als "**Trans**" ausgeben. ('trans' in the original)
(Anyone can claim to be "trans.")

(16) [...] finde ich es wichtig einen gewissen safe **Space** zu haben. ('space' in the original)
(I find it important to have a certain safe space.)

As the example shows, the line between spelling and casing correction is rather blurry, and possible casing errors – if not already corrected in the previous step – will be detected by the spelling correction as well. The reasoning of LanguageTool behind this is not fully transparent, and it is not possible to completely reconstruct the tool's process of categorizing errors into punctuation, casing, and spelling ones.

From the above, it can be concluded that while LanguageTool did detect and correct many errors correctly, it is not particularly suited for the task of correcting textual data collected from social media. It was not capable of properly dealing with acronyms (like 'XX-Chromosomen'), neologisms (like 'Genderfluide'), English words and phrases (like 'safe space') or anglicisms (like 'queer'), and it often replaced words it could apparently not map onto some rule or dictionary entry with more familiar words (like 'Klangfelder' replacing 'Klinefelter'). Therefore, LanguageTool needs further validation of the changes it suggests. We used the proofread LanguageTool corrected data for training and fine-tuning of our models in order to be able to skip the human annotation step in the future.

4. Testing of Deep Learning models

4.1. Deep Learning Models

4.1.1. T5 model

Transfer learning (cf. Torrey and Shavlik 2009) has become increasingly common in NLP. Models based on transfer learning are usually pre-trained on large amounts of unlabelled data using unsupervised learning, which "causes the model to develop general-purpose abilities and knowledge that can then be transferred down to downstream tasks" (Raffel et al. 2020, 2). While transfer learning was commonly used with recurrent neural networks (RNNs), more recent models using transfer learning are based on the *Transformer* architecture (Vaswani et al. 2017). Transformer-based models rely "entirely on self-attention to compute representations of [their] input and output without using sequence-aligned RNNs or convolution" (Vaswani et al. 2017, 2). Raffel et al. (2020) utilized this architecture and built an "encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and for which each task is converted into a text-to-text format" (HuggingFace), which they called T5 (Text-to-Text Transfer Transformer).[11] More specifically, in their model,

> an input sequence of tokens is mapped to a sequence of embeddings, which is then passed into the encoder. The encoder consists of a stack of "blocks", each of which comprises two subcomponents: a self-attention layer followed by a small feed-forward network. [...] The decoder is similar in structure to the encoder except that it includes a standard attention mechanism after each self-attention layer that attends to the output of the encoder. The self-attention mechanism in the decoder also uses a form of autoregressive or causal self-attention, which only allows the model to attend to past outputs. (Raffel et al. 2020, 4-5)

The authors trained their model on the Colossal Clean Crawled Corpus (C4),[12] which consists of text scraped from the web and made publicly available in the web archive Common Crawl. They trained four models differing in size and number of parameters: 'Base' (their baseline model with roughly 220 million parameters), 'Small' (fewer layers, 60 million parameters), 'Large' (more layers, 770 million parameters), and '3B and 11B' (more layers, 2.8 billion and 11 billion parameters, respectively). In experimenting with different NLP tasks, the largest model performed best and achieved state-of-the-art scores (Raffel et al. 2020).

T5 has a proven record of successfully accomplishing different NLP tasks, including spelling and punctuation prediction. Using mT5, a multilingual version of T5 developed by Xue et al. (2021) and pre-trained on a dataset covering more than 100 languages following a span-prediction objective, Rothe et al. (2021) built a language-agnostic GEC model that is capable of detecting errors independent of the input data's language. Therefore, they leveraged the multilingual data provided by the C4 corpus, artificially corrupted the sentences to generate synthetic training data, and fine-tuned their models using monolingual corpora in English,

[11] Available at: https://huggingface.co/docs/transformers/model_doc/t5.
[12] Available at: https://huggingface.co/datasets/c4.

Czech, German, and Russian. Their final, best model, the gT5, surpasses previous state-of-the-art results in GEC. Similarly, Švec et al. (2021) experimented with the T5 Base model and fine-tuned it to the task of restoring punctuation and casing in output produced by automatic speech recognition (ASR) systems in English, Czech, and Slovak. They showed that T5 (as well as BERT, another pre-trained, attention-based transformer) is an easily trainable model for detecting punctuation and restoring casing in a given language.

4.1.2. German DBMDZ GPT-2 Model

Error correction can be considered a generative task (i.e., generating new text), which is why we also decided to test a German GPT-2 (Generative Pretrained Transformer) model released by the Munich Digitization Center (MDZ) on HuggingFace.[13] This model is based on the autoregressive GPT-2 architecture introduced by OpenAI (Radford et al. 2019) and pre-trained for German language processing. While it is not an official German GPT-2 model, it is the only monolingual (German) GPT-2 model available (Bangura et al. 2023). The GPT-2 architecture has achieved state-of-the-art results in various NLP tasks, including language modeling, text generation, and machine translation.

"The model was trained on a 16GB and 2,350,234,427 tokens data set consisting of data from the Wikipedia dump, EU Bookshop corpus, Open Subtitles, CommonCrawl, ParaCrawl and News Crawl." (Bangura et al. 2023, 6) The utilization of such a large and varied training dataset contributes to the model's efficacy and proficiency in German language processing. The model exhibits exceptional language generation and analysis capabilities, making it a powerful tool for a wide range of applications.

4.1.3. Seq2seq Model with Monotonic Attention

Attention mechanisms (Bahdanau et al. 2015) were introduced to overcome the shortcoming of seq2seq models (Sutskever et al. 2014), where one context vector of a fixed length is given to the decoder that is supposed to capture all the information of the entire input sequence processed by the encoder, resulting in a loss of information when confronted with longer sequences (Cho et al. 2014). As a solution to that, in an attention model, "the encoder produces a sequence of hidden states (instead of a single fixed-length vector) which correspond to entries in the input sequence" (Raffel et al. 2017, 1). Implemented in a feed-forward neural network, the decoder pays attention to all intermediate states of the encoder while producing the output, which makes attention models extremely effective when dealing with tasks involving long sequences as inputs. At the same time, this architecture makes these models highly complex and non-linear. In contrast, the monotonic attention mechanism processes the input in a strict left-to-right manner, considering each input at a given output timestep and discarding any previous elements for subsequent output timesteps. This linearity reduces the complexity of the model and enables it to produce output sequences while processing the input sequence (Raffel et al. 2017). Figure 3 illustrates the two different attention mechanisms.

[13] Available at: https://huggingface.co/dbmdz/german-gpt2.

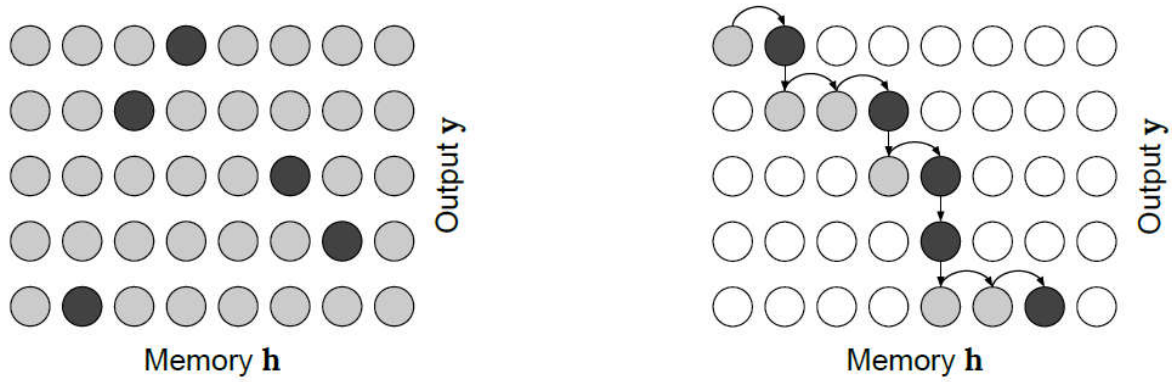

FIGURE 3: VISUALIZATION OF STOCHASTIC PROCESS UNDERLYING SOFTMAX-BASED ATTENTION DECODERS (LEFT) AND MONOTONIC STOCHASTIC DECODING PROCESS (RIGHT) (RAFFEL ET AL. 2017, 3)

4.2. Data Augmentation

The dataset we created is relatively small and, therefore, might turn out to be insufficient as a foundation for fine-tuning the models. To overcome this insufficiency, i.e., the scarcity of data annotated for (grammatical) errors, several methods have been proposed to artificially generate data for training GEC models (Madnani et al. 2012; Grundkiewicz and Junczys-Dowmunt 2014; Ge et al. 2018; Xie et al. 2018; Awasthi et al. 2019; Grundkiewicz et al. 2019; Lichtarge et al. 2019; Omelianchuk et al. 2020). For instance, synthetic data containing (grammatical) errors can be generated by taking error-free sentences and randomly substituting, inserting, or deleting words or characters, either "according to the frequency distribution observed in seed corpora" (Grundkiewicz and Junczys-Dowmunt 2014, 480) or based on confusion sets, which consist "of words that are commonly confused with each other" (Grundkiewicz et al. 2019, 254). Two other prominent approaches are extracting "source-target pairs from grammatical errors and their human-curated corrections gleaned from Wikipedia revision histories" (Lichtarge et al. 2019, 3291; cf. Grundkiewicz and Junczys-Dowmunt 2014) and introducing "noise into Wikipedia sentences via round-trip translation through bridge languages" (Lichtarge et al. 2019, 3291; cf. Madnani et al. 2012). Even though such data augmentation methods have proven to be highly effective and economically efficient, complete reliance on fully synthetic data might have such drawbacks as lower than original diversity of errors (Grundkiewicz and Junczys-Dowmunt 2014, 480) or the need for "language-specific hyperparameters and spelling dictionaries" (Rothe et al. 2021, 702). Despite these drawbacks and to save time, efforts, and resources (which would be needed to collect more data), we decided to artificially augment our existing dataset, too.

More specifically, we generated data by intentionally breaking some grammar and spelling rules and inserting mistakes in the input data, respectively. This approach might be described as hybrid, since we (1) took the real-life, messy, and not pre-processed training data and identified the most frequent errors with the help of a rule-based tool, (2) manually validated the corrections suggested by the tool, and then (3) added some of the repetitive grammar and spelling errors we noticed while validating the data. Thereby we tackle the first of the two drawbacks mentioned above, and having manually annotated high-quality seed data for training, our approach also overcomes the second drawback of requiring a specific spelling dictionary.

For each of the three categories tested by the tool and validated by the annotators (see Section 3.2), specific mistakes were chosen and selected to be inserted into the data. Casing: Based on the decision that the correct form of compound nouns containing the word 'trans' is hyphenated and starts with a capital letter, as in 'Trans-Mann' or 'Trans-Frau', every fifth instance of compound nouns containing the word 'trans' was lowercased and separated by a whitespace, as in 'trans Mensch'. This procedure resulted in 181 additional examples. Punctuation: 50% of the commas in the input data were deleted, resulting in 313 additional examples. Spelling: Every fifth instance of the conjunction 'dass' was replaced by its old spelling version 'daß', and 20% of the umlauts occurring in the data ('ä', 'ö', 'ü') were replaced by their vowel combinations ('ae', 'oe', 'ue'). This resulted in 481 additional examples. Following Rothe et al. (2021, 704), we also left 10% of the training data uncorrupted to teach the model that the input does not necessarily contain errors and can also be correct. In total, the data augmentation resulted in a significant increase in training data from 1,000 cases to 1,876 cases. These data were used to fine-tune the T5 and GPT-2 models and to train the seq2seq model.

4.3. Evaluation Metrics

To evaluate model performance, we implemented BLEU, GLEU and ROUGE scores. The Bilingual Evaluation Understudy (BLEU) score (Papineni et al. 2002) and the Generalized Language Evaluation Understanding (GLEU) score (Napoles et al. 2015) are both metrics used to evaluate the quality of NLP models. The BLEU score is a corpus measure that counts the number of n-gram matches between the input and generated output sequences (Casas et al. 2018). The score ranges from 0 to 1, with higher scores indicating better machine-generated translations (Papineni et al. 2002). Viewing GEC as a text-to-text rewriting task – and thus quite similar to machine translation – the output of GEC models can equally be evaluated by the BLEU score. The BLEU score can be calculated as follows:

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')} \qquad \text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} \omega_n \log p_n\right)$$

Napoles et al. (2015) criticized the 'off-the-shelf' application of such metrics as the BLEU score and showed that there is only little correlation between computational metrics like BLEU and human evaluations of GEC. Still inspired by BLEU, they developed the Generalized Language Evaluation Understanding (GLEU) score, a revised BLEU metric to better approximate human evaluations of GEC. Whereas the BLEU score weights accurately corrected text higher than the uncorrected changes left, the GLEU score "rewards correction while also correctly crediting unchanged source text" (Napoles et al. 2015, 590). It calculates the overlap of n-grams between the corrected sentence generated by the model and the human-corrected sentence, similar to the BLEU metric, but it "assigns more weight to n-grams that have been correctly changed from the source" (Napoles et al. 2015, 590). By rewarding correct edits, on the one hand, and penalizing

ungrammatical ones, on the other, and in using n-grams to capture both grammatical constraints and the fluency of the model's output, GLEU better models human evaluations of GEC than other evaluation metrics (Napoles et al. 2015). It can be computed as follows:

$$p'_n = \frac{\sum_{n\text{-}gram \in C} Count_{R\setminus S}(n\text{-}gram) - \lambda \left(Count_{S\setminus R}(n\text{-}gram)\right) + Count_R(n\text{-}gram)}{\sum_{n\text{-}gram' \in C'} Count_S(n\text{-}gram') + \sum_{n\text{-}gram \in R\setminus S} Count_{S\setminus R}(n\text{-}gram)}$$

$$\text{GLEU}(C, R, S) = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} \omega_n \log p'_n\right)$$

The same logic as that of these two scores can be applied to the ROUGE (Recall-Oriented Understudy for Gisting Evaluation) scores (Lin 2004). This set of metrics calculates the F-measure (the harmonic mean of precision and recall) between the generated text and the ground truth text, or in other words, the overlap between the generated text and the reference text, in terms of n-grams, where n can be 1, 2, or 3. ROUGE score 1 (ROUGE-1) measures the overlap of unigrams and ROUGE score 2 (ROUGE-2) extends the evaluation to bigrams to include important sequences and maintain the order of words. It is commonly used for evaluating automatic summarization and machine translation systems, but it can also be applied to machine translation and grammatical error collection tasks (Lin 2004). It is calculated as follows:

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count\,(gram_n)}$$

4.4. Results and Discussion

4.4.1. Overview

The results of the models – in terms of the different scores used to compare them – are summarized in Table 1.

Model	ROUGE	GLEU	BLEU
T5	0.5	0.25	0.72
GPT-2	0.9	0.63	0.9
Seq2seq	0.17	0.23	0.27

TABLE 1. EVALUATION METRICS FOR GEC MODELS

The worst scores were achieved by the seq2seq model, which can be attributed to the small amount of data, and it is likely that more training data will provide better results. While the fine-tuned T5 model performed better than the seq2seq model trained from scratch, it still exhibits a lack of ability to detect certain errors such as casing. The best results were achieved by the fine-tuned GPT-2 model, scoring highest in all the metrics used. This can be attributed to the comprehensive dataset on which the GPT-2 model was trained, which provided greater flexibility in performing generative tasks such as error-free text generation. More insights on the performance and training of the models are presented in the subchapters below.

4.4.2. T5 Model

We followed Rothe et al. (2021) and Švec et al. (2021) and utilized T5's encoder-decoder transformer architecture by fine-tuning the pre-trained base version of T5 (which is one of the several T5 models available in the Hugging Face's Transformers library). The T5 Base model[14] has 220 million parameters and is pre-trained on a large corpus of English text (C4) with a denoising objective, in which "the model is trained to predict missing or otherwise corrupted tokens in the input" (Raffel et al. 2021, 12). Being trained on multi-task data, the model is generally capable of dealing with any NLP task it is given, supporting English, French, Romanian, and German, the latter of which is the language of our data.

We set the batch size to 5 with an optimal learning rate of 2e-5, 100 training epochs, 0.01 weight decay, and 8 training and evaluation steps. During training, the model learns to correct errors in the input text by generating a new text containing the corrections. The training is done using the AdamW optimizer (Loshchilov and Hutter 2019), an improved version of Adam (Kingma and Ba 2015), with linear learning rate warmup and decay and gradient accumulation to simulate a larger batch size.

Step	Validation Loss	ROUGE-1	ROUGE-2
5	0.812300	45.460200	38.045000
10	0.722873	46.941400	41.051300
15	0.691845	46.118300	40.573400
20	0.680330	48.213100	43.470700
25	0.674840	50.326000	45.629300

TABLE 2. T5 MODEL FINE-TUNING

As can be seen in Table 2, during each step of the training process, the validation loss decreased and the ROUGE score increased. By the end of the final step, the ROUGE score reached slightly over 50%. This indicates that only about half of the generated corrections overlapped with the reference text. Similarly, the BLEU and GLEU scores remained relatively low (0.72 and 0.25, respectively). It is important to note that the dataset used for training was already pre-corrected

[14] Available at: https://huggingface.co/t5-base.

and did not require extensive changes, which may explain the relatively low ROUGE score. Apart from that, the low score can be attributed to the model's inability to recognize the casing patterns, as in Table 3, for example.

(Incorrect) Input	**Proofread output**	**T5 output**	***English translation***
"**sexual straftäter** im **knast**," mimimi. Die sind doch nicht ohne grund im Gefängnis. Dort sind sie doch nicht zum Spaß **wtf**	"**Sexual Straftäter** im **Knast**," mimimi. Die sind doch nicht ohne Grund im Gefängnis. Dort sind sie doch nicht zum Spaß **WTF**.	"**Sexual Straftäter** im **Knast**," mimimi. Die sind doch nicht ohne Grund im Gefängnis. Dort sind sie doch nicht zum Spaß **wtf.**	*"Sex offenders in jail," mimimi. They are in prison for a reason. They are not there for fun wtf.*
trans frauen auch. **als** nicht betroffene **person** sowas zu sagen ist wild	**Transfrauen** auch. **Als** nicht betroffene **Person** sowas zu sagen ist wil**d.**	**transfrauen** auch. **als** nicht betroffene **Person** sowas zu sagen ist wil**d.**	*Transwomen too. As a non-affected person to say something like that is wild.*

TABLE 3. T5 OUTPUT

As Table 3 shows, T5 is capable of identifying and correcting punctuation errors such as full stops at the end of sentences as well as spelling errors such as 'trans frauen', but fails when it comes to casing at the beginning of the sentence ('Transfrauen') and uppercasing abbreviations ('WTF').

4.4.3. German DBMDZ GPT-2 Model

In the context of limited training data available for fine-tuning the GPT-2 model, the alternative approach of prompting was employed. Prompting involves using annotated text as the desired outcome for GEC. This methodology allows for leveraging existing annotated data to guide the model's text generation process. In the text generation pipeline, specific settings were applied to control the output. The temperature parameter was set to 0.0, which ensures that the model's output remains deterministic and focused. Working with GPT models, we need to keep in mind the token limit for the input, which in the case of GPT-2 is 1,024 tokens. We introduced the recursive text splitter with 100 tokens overlap to care for the specific cases when the input comments are longer than the limitation.

(Incorrect) Input	**Proofread output**	**GPT-2 output**	***English translation***
asmr Das ändert nichts an der Wirklichkeit. Er wurde männlich geboren. Das lässt sich nicht rückwirkend ändern, indem man irgend ein Papier **umscreibt**.	**ASMR** Das ändert nichts an der Wirklichkeit. Er wurde männlich geboren. Das lässt sich nicht rückwirkend ändern, indem man irgend ein Papier **umschreibt**.	**ASMR** Das ändert nichts an der Wirklichkeit. Er wurde männlich geboren. Das lässt sich nicht rückwirkend ändern, indem man irgend ein Papier **umschreibt**.	*ASMR That doesn't change reality. He was born male. That can't be retroactively altered by rewriting some paper.*
Auf jeden Fall, ich **studier** Jura and das ist auch definitiv eine Folge, über die	Auf jeden Fall, ich **studiere** Jura and das ist auch definitiv eine Folge, über die viel	Auf jeden Fall, ich **studiere** Jura and das ist auch definitiv eine Folge, über die viel	*Definitely, I study law, and that's definitely a topic that is widely*

viel gesprochen wird, aber es ist einfach schwer mit einer (noch) nicht belegbaren Sache als Schwerpunkt zu argumentiere**n**	gesprochen wird, aber es ist einfach schwer mit einer (noch) nicht belegbaren Sache als Schwerpunkt zu argumentiere**n.**	gesprochen wird, aber es ist einfach schwer mit einer (noch) nicht belegbaren Sache als Schwerpunkt zu argumentiere**n.**	*discussed, but it's just difficult to argue with a (still) unproven matter as the main focus.*

TABLE 4. DBMDZ GPT-2 OUTPUT

As we can see from Table 4, the model's output is identical to the proofread output, identifying casing, spelling, and punctuation errors. The model scored high in all of the different evaluation metrics. It reached ROUGE scores of 0.91 (ROUGE-1) and 0.86 (ROUGE-2), already outperforming the previously tested T5 model, and BLEU and GLEU scores of 0.9 and 0.63, respectively.

The impressive capabilities of the GPT-2 model raise the question of the necessity of intermediary steps for preparing and preprocessing training data. It suggests that the model may have the potential to identify errors without requiring explicit examples of the desired output. However, due to the specific nature of the data being worked with, which comprises unstandardized YouTube comments containing various forms of expression (e.g., trans-Frauen or Transfrauen), fine-tuning the GPT-2 model with the available training data becomes essential. This fine-tuning process enables the model to learn the preferred spelling conventions and account for neologisms that were not included during the open-source model's initial training.

4.4.4. Seq2seq Model with Monotonic Attention

We used German GloVe (Pennington et al. 2014) to create a pre-trained word embedding matrix, which was then used as an input to the neural network. This embedding matrix mapped each word in the input text to a high-dimensional vector representation, allowing the network to better understand the meaning and context of the words. We then composed a seq2seq model, in which both the encoder and decoder consist of LSTM layers (Hochreiter and Schmidhuber 1997) and the decoder additionally includes a dense layer for the output prediction. We trained the model on 100 epochs with 16 steps for each epoch. The learning rate was set to 0.005 and the optimizer of choice was Adam (Kingma and Ba 2015), which has a proven record of being effectively used with many models, including seq2seq models. We applied a custom loss function using Sparse Categorical Cross Entropy to ensure that the padded values of the heavily padded dataset are not being calculated.

The model displayed good training results with decreasing loss stopping at 0.07 and increasing accuracy, reaching 93% accuracy by the end of training. However, it displayed unsatisfactory performance on the validation data. We predicted the embedding indices and through that constructed the corrected sentences only to find out that even though we accounted for punctuation in training the model, it failed to replicate the correct punctuation in its predictions. Also, it did not manage to keep German sentence structure and sometimes repeated the predictions multiple times, making the output almost unreadable, as in (17).

(17) kontra Dank der bei hat das das Transmänner Lia ich gleich nicht und bei bei bei sei so wenn wenn Frau wenn wenn wenn wenn Frau Frau groß Ich Frau als als als

als als als machen es machen sich eigentlich sich neuen sich durch neuen sich sich sich sich neuen neuen mit sich sich sich sich neuen sich Mutter Mutter Mutter bescheuerte Übergangsphase ihm ihm jeweilige jeweilige ihn Geschlecht 2 andere Ende andere sie sie andere auf auf auf auf auch auf auf auf auf hinaus Laut Grundgesetz Rechte Schule Rechte Rechte Rechte + religiösen uns religiösen religiösen benachteiligt genauso bevorzugt eine eine und und und und und Gleiches eine eine eine eine und und und und und und und…

(counter Thanks to the (unintelligible) the Trans men Lia I not and at at at if if woman if if if if woman woman big I woman as as as as as as make it make actually make themselves new themselves through new themselves themselves themselves new new with themselves themselves themselves themselves new themselves mother mother mother stupid transitional phase him him respective respective him gender 2 other end other they they other on on on on also on on on on out according to the Basic Law rights school rights rights rights + religious us religious religious disadvantaged equally favored one one and and and and and the same one one one and and and and and and and…)

Compared to the other two models, the ROUGE score for this model was the lowest with around 0.17, translating to little to no overlap between the input and the generated output. Similarly, the model reached a BLEU score of 0.27 and a GLEU score of 0.23. We assume that the unsatisfactory performance is due to an insufficient amount of training data, which is both not enough for the model to learn and also does not allow for the increased complexity of the model architecture. It is evident that for GEC tasks, fine-tuning should be preferred over training from scratch in case of little training data.

5. Conclusions

In this article, we set out to find the most suitable method to standardize German social media data, or more specifically, to correct spelling, punctuation, and casing errors in YouTube comments relating to the topic of gender diversity. The first tool that we tested for the task of GEC was LanguageTool. However, the validation of the corrections by our annotators revealed that this tool is not particularly suitable for the task, since approximately 10% of the corrections were marked as incorrect by the annotators. Therefore, we decided to fine-tune a T5 and a German GPT-2 model and train a seq2seq model with monotonic attention using the data-validated output by LanguageTool.

While the results of LanguageTool were not satisfactory themselves, the data validated and corrected by the annotators served as high-quality training data. Additionally, we synthetically generated more training data by purposefully breaking some rules and inserting errors into the data. The resulting training data were then utilized for fine-tuning (T5 and GPT-2) and training (seq2seq) the models, allowing for adjustments to the specific use case. This approach acknowledged that not all aspects needed correction or modification, and it accommodated the users' creativity in their written expressions.

The fine-tuned German GPT-2 model exhibited superior performance compared to the fine-tuned T5 model and the seq2seq model trained from scratch. Thus, it can be concluded that this

model is the most suitable one for detecting and correcting errors in German social media data. Especially if the amount of available training data is limited, fine-tuning a generative language model is to be preferred over training a model, such as the seq2seq one, from scratch for the GEC task.

References

Awasthi, Abhijeet, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. "Parallel Iterative Edit Models for Local Sequence Transduction." In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, Hong Kong, China, November 03-07. Association for Computational Linguistics. 4260–4270. doi:10.18653/v1/D19-1435.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. "Neural Machine Translation by Jointly Learning to Align and Translate." Paper presented at ICLR 2015, San Diego, California, USA, May 07-09. https://arxiv.org/pdf/1409.0473.pdf.

Bangura, M., K. Barabashova, A. Karnysheva, S. Semczuk, and Y. Wang. 2023. "Automatic Generation of German Drama Texts Using Fine Tuned GPT-2 Models." https://arxiv.org/pdf/2301.03119.pdf

Casas, Noe, José A. R. Fonollosa, and Marta R. Costa-jussà. 2018. "A differentiable BLEU loss. Analysis and first results." Paper presented at ICLR 2018, Vancouver, Canada, April 30-May 03. 1–12. https://openreview.net/pdf?id=HkG7hzyvf.

Cho, Kyunghyun, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches." In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Doha, Qatar, October 25. Association for Computational Linguistics. 103–111. doi:10.3115/v1/W14-4012.

Ge, Tao, Furu Wei, and Ming Zhou. 2018. "Fluency Boost Learning and Inference for Neural Grammatical Error Correction." In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, Melbourne, Australia, July 15-20. Association for Computational Linguistics. 1055–1065. doi:10.18653/v1/P18-1097.

Grundkiewicz, Roman, and Marcin Junczys-Dowmunt. 2014. "The WikEd Error Corpus: A Corpus of Corrective Wikipedia Edits and Its Application to Grammatical Error Correction." In *NLP 2014: Advances in Natural Language Processing, 9th International Conference on NLP, PolTAL 2014*, Warsaw, Poland, September 17–19. Springer. 478–490. doi:10.1007/978-3-319-10888-9_47.

Grundkiewicz, Roman, Marcin Junczys-Dowmunt, and Kenneth Heafield. 2019. "Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data." In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, Florence, Italy, August 02. Association for Computational Linguistics. 252–263. doi:10.18653/v1/W19-4427.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. "Long Short-Term Memory." *Neural Computation* 9(8): 1735–1780. doi:10.1162/neco.1997.9.8.1735.

HuggingFace. "T5." Accessed June 20, 2023. https://huggingface.co/docs/transformers/model_doc/t5.

Kingma, Diederik P., and Jimmy Lei Ba. 2015. "Adam: A method for stochastic optimization." Paper presented at the 3rd International Conference for Learning Representations, San Diego, California, May 7-9. http://arxiv.org/pdf/1412.6980.pdf.

Landis, J. Richard, and Gary G. Koch. 1977. "The Measurement of Observer Agreement for Categorical Data." *Biometrics* 33(1): 159–174. doi:10.2307/2529310.

LanguageTool. "Development Overview." Accessed June 20, 2023. https://dev.languagetool.org/development-overview.

Lichtarge, Jared, Chris Alberti, Shankar Kumar, Noam Shazeer, Niki Parmar, and Simon Tong. 2019. "Corpora Generation for Grammatical Error Correction." In *Proceedings of NAACL-HLT 2019*, Minneapolis, Minnesota, June 02-07. Association for Computational Linguistics. 3291–3301. doi:10.18653/v1/N19-1333.

Lin, Chin-Yew. 2004. "ROUGE: A Package for Automatic Evaluation of Summaries." In *Text Summarization Branches Out. Proceedings of the ACL-04 Workshop*, Barcelona, Spain, July 25-26. Association for Computational Linguistics. 74–81. https://aclanthology.org/W04-1013.pdf.

Madnani, Nitin, Joel Tetreault, and Martin Chodorow. 2012. "Exploring Grammatical Error Correction with Not-So-Crummy Machine Translation." In *NAACL HLT '12: Proceedings of the Seventh Workshop on the Innovative Use of NLP for Building Educational Applications Using NLP*, Montréal, Canada, June 03-08. Association for Computational Linguistics. 44–53. doi:10.5555/2390384.2390389.

McNamara, Caolan, Németh László, n.a. Pander, and Paweł Hajdan Jr. 2015. "Hunspell." SourceForge. Last modified July 07. https://sourceforge.net/projects/hunspell/

Melnyk, Lidiia, and Linda Feld. 2022. "Sentiment Analysis and Stance Detection on German Youtube Comments on Gender Diversity." Journal of Computer-Assisted Linguistic Research 6: 59–86. doi:10.4995/jclr.2022.18224.

Napoles, Courtney, Keisuke Sakaguchi, Matt Post, and Joel Tetreault. 2015. "Ground Truth for Grammatical Error Correction Metrics." In P*roceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers)*, Beijing, China, July 26-31. Association for Computational Linguistics. 588–593. doi:10.3115/v1/P15-2097.

Omelianchuk, Kostiantyn, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. "GECToR – Grammatical Error Correction: Tag, Not Rewrite." In *Proceedings of the 15th*

Workshop on Innovative Use of NLP for Building Educational Applications, Seattle, WA, USA/Online, July 10. Association for Computational Linguistics. 163–170. doi:10.18653/v1/2020.bea-1.16.

Papers with code. "Grammatical Error Correction." Accessed June 20, 2023. https://paperswithcode.com/task/grammatical-error-correction.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. "BLEU: a Method for Automatic Evaluation of Machine Translation." In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, Pennsylvania, USA, July 07-12. Association for Computational Linguistics. 311–318. doi:10.3115/1073083.1073135.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. 2014. "GloVe: Global Vectors for Word Representation." In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, October 25-29. Association for Computational Linguistics. 1532–1543. doi:10.3115/v1/D14-1162.

Radford, Alec, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. "Language Models are Unsupervised Multitask Learners." https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

Raffel, Colin, Minh-Thang Luong, Peter J. Liu, Ron J. Weiss, and Douglas Eck. 2017. "Online and Linear-Time Attention by Enforcing Monotoni Alignments." In *ICML'17: Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, August 06-11. Association for Computing Machinery. 2837–2846. doi:10.5555/3305890.3305974.

Raffel, Colin, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." *Journal of Machine Learning Research* 21(1:140): 1–67. doi:10.5555/3455716.3455856.

Rothe, Sascha, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. "A Simple Recipe for Multilingual Grammatical Error Correction." In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Short Papers)*, Online, August 01-06. Association for Computational Linguistics. 702–707. doi:10.18653/v1/2021.acl-short.89.

Sahu, Subham, Yogesh Kumar Vishwakarma, Jeevanlal Kori, and Jitendra Singh Thakur. 2020. "Evaluating Performance of Different Grammar Checking Tools." *International Journal of Advanced Trends in Computer Science and Engineering* 9(2): 2227–2233. doi:10.30534/ijatcse/2020/201922020.

Schmaltz, Allen, Yoon Kim, Alexander M. Rush, Stuart M. Shieber. 2016. "Sentence-Level Grammatical Error Identification as Sequence-to-Sequence Correction." In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, San Diego, California, June 16. Association for Computational Linguistics. 242–251. doi:10.18653/v1/W16-0528.

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. 2014. "Sequence to Sequence Learning with Neural Networks." In *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems*, Montreal, Canada, December 08-13. Association for Computing Machinery. 3104–3112. doi:10.5555/2969033.2969173.

Švec, Jan, Jan Lehečka, Luboš Šmídl, and Pavel Ircing. 2021. "Transformer-Based Automatic Punctuation Prediction and Word Casing Reconstruction of the ASR Output." In *Text, Speech, and Dialogue: 24th International Conference, TSD 2021, Proceedings*, Olomous, Czech Republic, September 06-09. Springer. 86–94. doi:10.1007/978-3-030-83527-9_7.

Torrey, Lisa, and Jude Shavlik. 2009. "Transfer Learning." In *Handbook of Research on Machine Learning Applications*, edited by E. Soria, J. Martin, R. Magdalena, M. Martinez, and A. Serrano, 242–264. Hershey, PA: IGI Global.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." In *Advances in Neural Information Processing Systems 30: NIPS 2017*, Long Beach, CA, USA, December 04-09. Association for Computing Machinery. 5998–6008. doi:10.48550/arXiv.1706.03762.

Wang, Yu, Yuelin Wang, Kai Dang, Jie Liu, and Zhuo Liu. 2021. "A Comprehensive Survey of Grammatical Error Correction." *ACM Transition on Intelligent Systems and Technology* 12(5:65): 1–51. doi:10.1145/3474840.

Xie, Ziang, Guillaume Genthial, Stanley Xie, Andrew Y. Ng, and Dan Jurafsky. 2018. "Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction." In *Proceedings of NAACL-HLT 2018*, New Orleans, Louisiana, June 01-06. Association for Computational Linguistics. 619–628. doi:10.18653/v1/N18-1057.

Xue, Linting, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Bara, and Colin Raffel. "mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer." In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, June 06-11. Association for Computational Linguistics. 483–498. doi:10.18653/v1/2021.naacl-main.41.