HAW HAMBURG

MASTER THESIS
Thorben Schomacker

Simplification of German Narrative Documents with Longformer mBART

Faculty of Engineering and Computer Science
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE
WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Thorben Schomacker

Simplification of German Narrative Documents with Longformer mBART

Master thesis submitted for examination in Master´s degree
in the study course *Master of Science Informatik*
at the Department Computer Science
at the Faculty of Engineering and Computer Science
at Hamburg University of Applied Sciences

Supervisor: Marina Tropmann-Frick
Supervisor: Olaf Zukunft
Supervisor: Tillmann Dönicke
Submitted on: 15. Dezember 2022

Thorben Schomacker

Title of Thesis

Simplification of German Narrative Documents with Longformer mBART

Keywords

Language Generation, Transformer, Narrative Texts, German, Low-Resource, Few-Shot, Large Context

Abstract

Transformer-models have become the most prominent method for solving a multitude of natural language processing (NLP) tasks since their introduction in 2017. Natural Language Generation (NLG) is one of these problems. In this thesis we applied modern NLG-techniques to the problem of text simplification. Text simplification can be desribed as an intra-language translation task, where standard language is translated to simple language. Currently there are only a few German datasets available for Text Simplification. Even fewer with larger and aligned Documents, and not a single one with narrative texts. With this paper we firstly explore to which degree modern NLG-techniques can be applied to our newly proposed German Narrative Text Simplifications dataset. We used Longformer Attention and a pre-trained mBART model. Our findings indicate that currently avaible approach are not able to solve the task properly. We conclude on a few directions for future research to adress this problem.

Kurzzusammenfassung

Transformator-Modelle haben sich seit ihrer Einführung im Jahr 2017 zur Lösung einer Vielzahl von Aufgaben der natürlichen Sprachverarbeitung (NLP) durchgesetzt. Natural Language Generation (NLG) ist eines dieser Probleme. In dieser Arbeit haben wir moderne NLG-Techniken auf das Problem der Textvereinfachung angewendet. Textvereinfachung kann als eine innersprachliche Übersetzungsaufgabe beschrieben werden, bei der Standardsprache in einfache Sprache übersetzt wird. Derzeit gibt es nur wenige deutsche Datensätze zur Textvereinfachung. Noch weniger mit größeren und Dokumenten, die in beiden Versionen vorliegen. Und kein einziger mit narrativen Texten. In diesem Beitrag untersuchen wir zunächst, inwieweit sich moderne NLG-Techniken auf unseren neu eingeführten deutschen Datensatz für narrative Textvereinfachungen anwenden lassen. Wir

haben Longformer Attention und ein vortrainiertes mBART-Modell verwendet. Unsere Ergebnisse zeigen, dass die derzeit verfügbaren Ansätze nicht in der Lage sind, die Aufgabe richtig zu lösen. Wir schließen mit einigen Hinweisen für die zukünftige Forschung, um dieses Problem zu adressieren.

Contents

List of Figures

List of Tables

1 Introduction

With the rise of the internet, it has become convenient and often free to access an abundance of texts. However, not all people, who have access, can really read and understand the texts. Despite the fact that, they speak the language that the text is written in. Most often this problem originates in the too complex nature of the texts. Text Simplification can help to overcome this barrier. One of the first to outline the motivation for Text Simplification, were Chandrasekar et al. [1996]. In addition to this human audience, they defined five areas of natural language processing where Simple Language can help:

1. **Parsing**: Simpler sentences lead to faster parsing and less parse ambiguity.
2. **Machine Translation**: Reduced ambiguity.
3. **Information Retrieval**: Queries often result in large segments of texts, of which only a part has relevant information. Shorter and simpler sentences can help to extract shorter and more relevant segments.
4. **Summarization**: Simplification can improve the precision in sorting out irrelevant text and improve the overall summarization precision.
5. **Clarity of Text**: Simpler sentences and vocabulary helps to reduce ambiguities and redundant information.

Ruder [2019] names "Reasoning about large or multiple documents" as one of the four Biggest Open Problems in NLP. Simplification can, as stated above, make an important contribution, to increase the accessibility of longer texts for humans and machines.

Narrative forms are one of the primary ways humans create meaning [Felluga, 2011]. Narrative texts, then, make an important contribution to how we describe and shape our environment. Easy language also contributes to involving as many people as possible in

this process. The motivation to simplify narrative text is very appropriately worded by the Passanten Verlag[1]:

"Für alle, die Bücher lieben und denen es manchmal trotzdem schwer fällt zu lesen."
"For all those who love books and still sometimes find it hard to read."

Research Questions In this work, we investigated the following research questions:

1. What are the best methods to evaluate the quality of automatically generated text simplifications?
2. What is the most best method to automatically generate German document-level text simplifications?
3. Does Domain Adaptation Training improve the quality of automatically generated text simplifications?
4. Does Fine-Tuning a pre-trained Model improve the quality of automatically generated text simplifications?

Thesis Outline We tackle the research questions by using the following outline for this thesis:

- At the beginning of the thesis, we present natural language processing and the Transformer network in general terms in the Section 2.1, 2.2, 2.3 and 2.4. We describe the transformer architecture's inner workings in detail. Focussing on the Transformer Models and architectures used in this thesis.
- Followed by Section 2.5, in which we give an extensive overview on evaluation of text simplification. For this purpose, we created an overview table (Table 2.3) to provide the reader with a quick and visual guideline for the topic.
- Then, in Section 3 we describe our experimental setup, implementation details and evaluation methods we used. Both datasets, domain adaptation and fine-tuning data, are introduced and analyzed.
- The results of the experiments are shown and discussed in Section 4.

[1] `https://www.passanten-verlag.de/`

- The thesis is finalized with a conclusion and an outlook, which describes directions for future research in Section 5.

Source Code We made all code used for this thesis publicly and open-source available on Github:

Pre-Processing of the Domain Adapation based on Textgrid Texts:
github.com/tschomacker/textgrid-domain-adaptation-dataset

Pre-Processing of the Fine-Tuning Dataset based on Projekt Gutenberg, Gutenberg and PDF-Reading Samples Texts:
github.com/tschomacker/aligned-narrative-documents

Machine Learning Architecture and Implementation:
github.com/tschomacker/longmbart

Feel free to add an issue in the Github repository or contact us, for any questions regarding the thesis or the code.

2 Background and Previous Work

In this section, we describe the necessary background information for modern NLP models - specifically transformer-based models and going into the details on the nature of automatic text simplifications and previous work in this field. Firstly, we will give an overview of some essential concepts from natural language processing. We then present the standard transformer architecture [Vaswani et al., 2017]. We especially focus on the concept of attention, the core and success-bringing mechanism of the Transformer architecture, and how it can be applied on a document-level task. Furthermore, we conclude by outlining previous work on automatic German text simplification.

2.1 Natural Language Processing

Natural language processing (NLP) analyzes and exploits natural (human) language with computers. This discipline covers a wide range of tasks, such as Question-Answering or Translation. Applying neural networks in the NLP-field led to a new way of solving NLP tasks: neural language models. These models are achieving very promising results. Additionally, the usage of transfer learning to NLP has brought a major breakthrough in NLP: machines have exceeded human performance in several NLP tasks, e.g. DeBERTa [He et al., 2021] the SuperGLUE benchmark [Wang et al., 2019].

Training neural language models usually involves three steps: 1) Tokenize the input, that means separate and group the words into a more computationally efficient representation: a token. 2) These tokens are then transformed into a word embedding. This form of representations can be very information-rich and allows the use of mathematical operations. 3) Train a neural language model on the tokenized and embedded dataset [Sagen, 2021].

2.1.1 Tokenization

The transformation of human-readable text into a smaller sub-string of characters (token), is called tokenization [Grefenstette, 1999]. Tokens are the basis for most of the NLP techniques. There are multiple tokenization methods. To show some of the general challenges of tokenization, Sagen [2021] suggest a naive approach: Split each sentence based on space-separated words and list several limitations:

1. Only works for languages with spaces separated words.
2. Even for whitespace-separated languages, not all words follow this structure, e.g., concatenated or negated words.
3. Homonyms
4. Representing every possible word or even a fraction of them is costly
5. Theoretically character-level tokenization is more efficient than entire words since there are only 26 letters in the English alphabet. From a practical perspective, this low-level representation mostly fails to capture the full structure and relational interplay between the words [Sagen, 2021].

Ideally, tokenization should pose an optimal trade-off between sentence, word- and character-level representation, be language-independent and fast. In machine learning, finding the most effective tokens to split words into is covered by a tokenizer, that is trained on a language modeling task.

2.1.2 Embeddings

Every neural language model needs a vocabulary to work. Every token, that can be processed by the model, is stored as an *embedding*, usually a multidimensional vector that represents the token. Schomacker and Tropmann-Frick [2021] divided them into two categories: context-free and contextualized word embeddings. Context-free representations are traditionally used, and the most popular GloVe [Pennington et al., 2014] and Word2Vec [Mikolov et al., 2013] are based only on the characters the word consists of. Contextualized representations are based on the characters and additionally incorporate the adjacent tokens. Context adds valuable information, Figure 2.1 illustrates an example that shows the different representations of the word “bank” and how contexts

adds more detail to the representation. In the left sentence, it means a credit institution, and in the right sentence, a geographical phenomenon, although it has the same span of characters.

2.1.3 Encoder-Decoder Architectures

RNNs (recurrent neural networks) [Rumelhart et al., 1988] were previously the most used architecture family for processing sequential inputs such as text. Their key strength is that they can process and generalize across sequences of variable lengths. This strength relies on the fact, that RNNs can connect information that is distributed across the sequence via parameter-sharing. For example, where the model's task is to extract temporal information from a sequence. It has the following two sentences as input: "Today I am feeling sick" and "I am feeling sick today". In both cases, it should extract "today" independently of its position in the sequence [Schomacker and Tropmann-Frick, 2021].

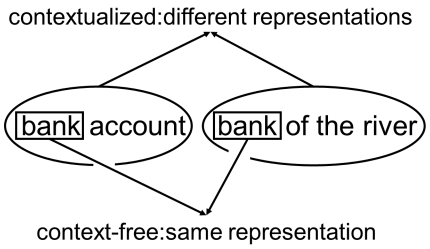


Figure 2.1: Comparing the representations of the same span of characters in a context-free and in a contextual model. (Source: [Schomacker and Tropmann-Frick, 2021])

The major limitation of RNNs is that their output length is determined by the input length. The encoder-decoder framework overcomes this limitation [Cho et al., 2014, Sutskever et al., 2014]. This framework consist of three main components:

Encoder: Extracts features by reading and converting the input into distributed representations, with one feature vector associated with each word position.

Context: Either a feature vector or a list of feature vectors based on the extracted features. If it is a list of feature vectors, it has the benefit that each feature vector can be accessed independently of its position in the input.

Decoder: Sequentially processes the context to generate the final output and solve the task

2.1.4 Simplification

The task of this paper is to create a machine learning model, that simplifies texts. This task can be seen as intralanguage translation, where input and output are written in the same language but in different versions of it. Therefore, many techniques and principles from interlanguage neural machine translation can also be applied to simplification. Intralanguage spans both simplification and summarization as tasks, both reduce the input text, and are thereby highly related. Summarizations reduce the content of input, not necessarily the language complexity. Simplification can cover both aspects. For German, there are official guidelines and constraints for Easy Language, [Freyhoff et al., 1998] and Simple Language is rather loosely regulated without any official constraints. Table 2.1 shows an overview of the relation between four different language versions.

Figure 2.2: European easy-to-read logo (Source: Inclusion Europe)

	Standard Language	Easy Language	Simple Language	Summary
Lingual reduction	-	+	+	-
Content reduction	-	+	-	+

Table 2.1: Comparison of Standard language, Easy Language, Simple Language and Summary. We extended and translated the table 1 from [Bredel and Maaß, 2016, p.527-530], which itself used the model from [Wagner, 2015]

In this paper, we further use the term *simple* language version to describe texts, with a reduced lingual complexity and possibly but not necessarily reduced content. Additionally, *simple* language, does not have any formal guidelines (e.g., [Freyhoff et al., 1998]) besides standard language orthography. So, we consider *simplification* as the task of intra-lingual translation between standard language and simple language. The differences of easy language and simple language are further described in Table 2.2.

2.2 Transfer Learning

Transfer learning as a collective term describes situations where what has been learned in one setting is exploited to improve generalization in another setting [Goodfellow et al., 2016, p.536-541]. This transfer learning situations can, according to [Pan and Yang, 2010], be labeled as "domain-" and "task-"related [Wilson and Cook, 2020]:

Easy Language	Simple Language
regulated by guidelines	less strictly regulated
useful especially for people with learning difficulties	useful also for other readers (elderly people, people with low knowledge of German, learners of a foreign language, etc.)
short main clauses, extensive renunciation of subordinate clauses	longer sentences; even subordinate clauses
Use of familiar words, explanation of difficult words	Use of even difficult terms
clear and large typeface	
a new paragraph after each punctuation mark	not necessarily a new paragraph after each punctuation mark
clear visual appearance of image and font	no strict regulation of the appearance of image and font

Table 2.2: More detailed comparison of Easy Language and Simple Language. We translated the table from [Bredel and Maaß, 2016, p.527-530], which itself used the model from [Magris and Ross, 2015]

Domain: Consists of a feature space (i.e., the features of the data) and a marginal probability distribution (i.e., distribution of the features in the dataset)

Task: Consists of a label space (i.e., the set of labels) and an objective predictive function (i.e., a predictive function learned from the training data)

So, a transfer learning aims to transfer knowledge from a source domain to a different target domain, transfer knowledge from a source task to a different target task or doing both. Furthermore, [Pan and Yang, 2010] introduced three terms to further describe transfer learning methods:

Inductive: The target and source tasks differ, the domains may or may not be different, and some labeled target data is required.

Transductive: The target and source tasks are the same while the domains differ, and both labeled source data and unlabeled target data is required.

Unsupervised: The target and source tasks differ, and there is no requirement of labeled data in either the source domain or the target domain.

Similarly, [Sagen, 2021] uses three common categories:

Regular transfer learning: trains on both the source and target task using all or a sufficient amount of training data.

Few-shot transfer learning: training on a few data samples for the target task.

Zero-shot transfer learning: no training on the target task.

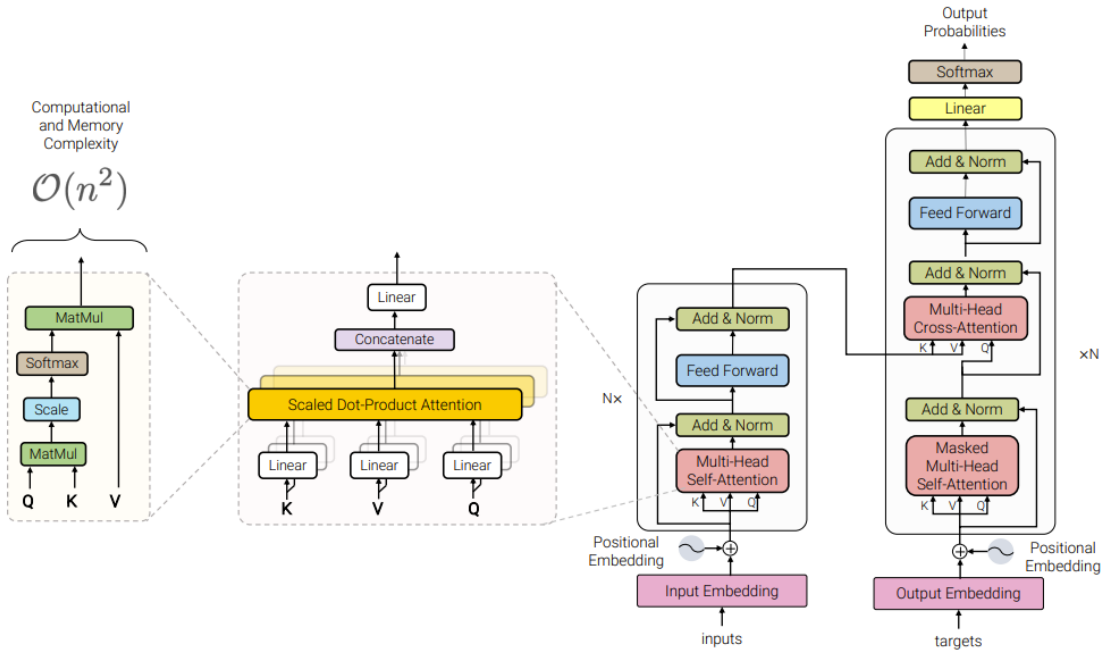


Figure 2.3: Illustration of the Vanilla Transformer Architecture [Vaswani et al., 2017] (Source: [Tay et al., 2022])

2.3 The Transformer Model

The Transformer model is an architecture that was introduced in [Vaswani et al., 2017]. It is distinct from previous approaches by entirely relying on attention to draw global dependencies between input and output instead of using recurrence. This makes possible to parallelize significantly more of the computation. This first Transformer model, often referred to as Vanilla Transformer, consists of an Encoder and Decoder Stack. The architecture is depicted in Figure 2.3.

2.3.1 An Overview to the Attention Mechanism

We humans perceive information in selective form, we attend more to aspects that seem more salient to us, than others. When seeing an image, it is also quite intuitive that

neighboring areas often highly correlate. For instance, it is easier to recognize a nose and ears if they are next to a pair of eyes.

Although the encoder-decoder framework performs efficiently on a variety of tasks, the framework's ability to understand long and complex inputs is limited, due to the fact that all the information is stored in a context. Attention is one way of overcoming this weakness and was first applied in machine translation [Bahdanau et al., 2016]. In an additional attention step, each annotation receives a weight that further determines its amount of influence on the output. This is illustrated in Figure 2.4, where each input x_i results in a hidden state h_i of the encoder, then an attention weight a_i determines the hidden state's weight when it is stored in the context. The context is afterwards used by the encoder. Without this mechanism, each hidden state would be added unweighted to the context. Which can, for instance, be problematic in situations where input has a large proportion of irrelevant and only a small proportion of relevant information because the relevant information would become incidental and the decoder's output less accurate.

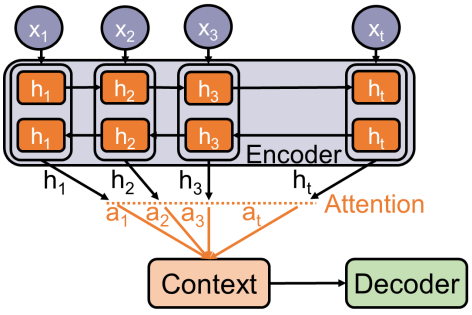


Figure 2.4: Visualization of the attention mechanism applied to an encoder-decoder model. (Source: [Schomacker and Tropmann-Frick, 2021])

2.3.2 Multi-Head Attention

One key factor for the success of the Transformer architecture [Vaswani et al., 2017] is its ability to tackle sequence to sequence tasks without any recurrence modules while increasing its performance and allow for parallel (GPU) training. This is done by using only self-attention layers to capture dependency between the input's tokens by applying attention to the tokenized input with itself. Vaswani et al. [2017]'s newly introduced *multi-head self-attention mechanism* firstly generates three different vector representations with randomly initialized weight created from the input: key K, value V, and query Q. Secondly, K and V are sent as input to the self-attention layer, which produces the output Q. The problem of learning to assign attention weights to each word can instead be viewed as: 1) given something to search for (query), 2) match the closest

keywords (key), and 3) return the most similar results based on your inquiry (value). This attention mechanism can be reformulated as:

$$Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.1}$$

This is also referred to as the scaled dot-product attention, which projects the expected keys from the source K onto the target output Q. All three matrices are created by multiplying the input sequence X with randomly initialized weights W to that, $Q = W^Q X, K = W_K X$, and $V = W^V X$ are learned during training by updating the weights W^Q, W^K, W^V.

In order to learn a more general and complete representation, [Vaswani et al., 2017] created multiple scaled-dot product attentions each using randomly initialized weight matrices, so that these attention weights could learn multiple word alignments of a sequence. This approach is called this *multi-head self-attention* and is visualized in the second dashed box from left from in Figure 2.3 and is formulated as:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \cdots, head_k)W^O \tag{2.2}$$

$$\text{head}_i = \text{Attention}(Q, K, V) \tag{2.3}$$

Where each attention head, $head_i$, is the scaled dot-product attention of the Q, K, V. And W^o is another randomly initialized weight matrix learned during training, h is the number of heads, which denotes the number of parallel attention layers computed and concatenated into a single multi-head attention layer. The Vanilla Transformer from [Vaswani et al., 2017] uses six stacked encoder- and six stacked decoder blocks, where each such block consists of a multi-head self-attention layer.

A downside of the multi-head self-attention is its quadratic run time and memory complexity caused by the matrix multiplications: Q, K, V are all matrices generated from a linear projection of the input text of length N and each token attends to every other token, the memory and computational complexity of these operations or, more specifically, the QK^T matrix multiplication is $\mathcal{O}(n^2)$ [Tay et al., 2022]. To mitigate this problem, several models have self-imposed a maximum sequence length of 512 or 1024 tokens, which a model can process at a time. For many tasks, this sequence length is sufficient.

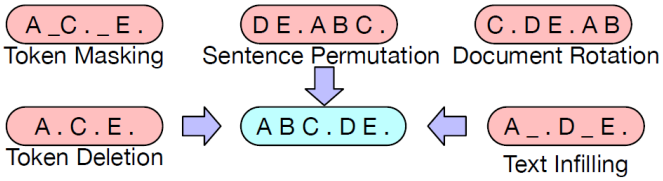


Figure 2.5: Noising strategies used in BART (Source: [Lewis et al., 2019])

However, not in document-based tasks, as in our scenario. Therefore, we elaborate methods to tackle this problem for large-context e.g., Document-level tasks in Section 2.4.4 and 2.4.5.

2.4 Transformer-based Language Representation Models

The general purpose of transformer-base models is to learn high-level language representations, so-called embeddings. It has become common practice to pre-train the models and incorporate some general language understanding and world knowledge in the embedding, and afterwards fine-tune the models on a downstream task. The baseline transformer architecture is very versatile and as evolved to the standard approach for machine learning models in NLP. Since it's appearance, many adaptations have been created, which are usually categorized (e.g., [Tunstall et al., 2022]) in into three branches:

Encoder-only: Such as BERT [Devlin et al., 2019], DistilBERT, RoBERTa, XLM, ALBERT, ELECTRA, DeBERTa

Decoder-only: Such as GPT, GPT-2, GPT-3, CTRL, GPT-Neo

Encoder-Decoder: Such as BART [Lewis et al., 2019], T5 [Raffel et al., 2020], M2M-100, BigBird

2.4.1 BART

BART is a denoising sequence-to-sequence model, introduced in [Lewis et al., 2019] , that uses denoising during pre-training. In denoising, the model firstly creates a corrupted version of an input and learns in a second step to map the corrupted one back to the original. It is a transformer-based [Vaswani et al., 2017] model, using an encoder and decoder stack. It incorporates newer findings from GPT [Radford et al., 2018] and change

ReLU activation functions to GeLUs [Hendrycks and Gimpel, 2020], which is a non-linear activation function and a modified expectation of adaptive dropout with initialize parameters from $\mathcal{N}(0, 0.02)$. Furthermore, BART is very similar to BERT, but differs in two aspects: (1) each layer of the decoder additionally performs cross-attention over the final hidden layer of the encoder (as in the Vanilla Transformer Encoder-Decoder model); and (2) BART does not use an additional feed-forward network before word prediction. BART uses five strategies for noising, which are illustrated in Figure 2.5:

Token Deletion: Random tokens in the input text are deleted. Contrary to token masking, the task is to decide at which positions inputs are missing.

Text Infilling: Random text spans in the input text with span lengths based on the Poisson distribution ($\lambda = 3$) is replaced with a single `[MASK]` token. That means that 0-length spans equals the insertion of `[MASK]` tokens. This procedure is inspired by SpanBERT [Joshi et al., 2020], but SpanBERT uses clamped geometric distribution for the span length, and strictly replaces each span with a sequence of `[MASK]` tokens of exactly the same length. In mBART text infilling, the model learns to predict how many tokens are missing from a span.

Sentence Permutation: All sentences from the input text are separated based on full stops, shuffled, and then reassembled in a random order.

Document Rotation: One token is randomly selected from the input text, and then the document in a way rotated that it begins with that token. With this task, the model learns to identify the start of the document.

2.4.2 mBART

[Liu et al., 2020] introduced mBART in 2020 as a sequence-to-sequence denoising auto-encoder pre-trained on large-scale monolingual corpora in many languages using the BART objective [Lewis et al., 2019]. They proposed the first method for pre-training a complete sequence-to-sequence model by denoising full texts in multiple languages. Previous approaches such as [Conneau and Lample, 2019, Edunov et al., 2019, Lewis et al., 2019, Raffel et al., 2020] only focused on the encoder, decoder, reconstructing parts of the text, or to only use an English corpus. This approach of pre-training a complete model allows offering a key feature: The model can be directly fine-tuned for supervised and unsupervised machine translation, with no task-specific modifications. The authors

tested mBART both for sentence-level and document-level machine translation. Figure 2.6 depicts the mBART architecture schematically.

The authors pre-trained mBART on 25 languages (CC25) extracted from the common crawl corpora of Wenzek et al. [2020], Conneau et al. [2020], which are based on internet texts, that are tagged with their language, filtered, and adjusted to improve the dataset quality. CC25 is multilingual, German has the fifth-largest proportion in the dataset after English, Russian, Vietnamese and Japanese.

The authors evaluated mBART on document-level machine translation tasks. Furthermore, they used document fragments during pre-training of up to 512 tokens, allowing the models to learn dependencies between sentences. They show that this pre-training significantly improves document-level translation. And interestingly, their mBART performs better on document-level with the document-level objective than with the sentence-level objective.

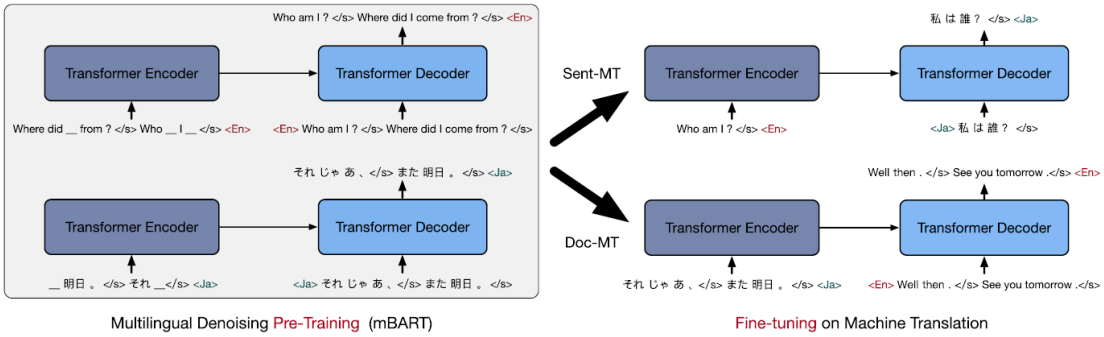


Figure 2.6: Illustration from [Liu et al., 2020] of the framework for the multilingual denoising pre-training (left) and fine-tuning on downstream MT tasks (right), where (1) sentence permutation and (2) word-span masking as the injected noise was used.

2.4.3 T5 & mT5

Today exists a broad landscape of transfer learning models for NLP. The Text-to-Text Transfer Transformer or *T5* [Raffel et al., 2020], published in 2020, aimed to explore what works best, and how far we can push the tools that already exist. T5 model transforms a wide variety of many-to-many and many-to-one NLP tasks to a uniformed text-to-text task. This allows T5 to be supervised multitask trained.

Massively Multilingual Pre-trained Text-to-Text Transformer or *mT5* [Xue et al., 2021] is the multilingual variant of T5. They used the same model architecture and training procedure as of T5. To be exact, they used the "T5.1.1" recipe, which improves T5 by using GeGLU [Shazeer, 2020] as the activation function, which is a combination of Gated Linear Units [Dauphin et al., 2017] and Gaussian Error Linear Units [Hendrycks and Gimpel, 2020] [1]. Additionally, pre-training on unlabeled data only with no dropout. mT5 uses the mC4 dataset, which includes 107 languages (101 languages and 6 language variants). German is the fourth-largest language in dataset (3.05%) after English (5.67%), Russian (3.71%), and Spanish (3.09%).

Pre-training multilingual models depends on a tradeoff: If low-resource languages (in this case: Languages, which make up only a small percentage) are sampled too often, the model may overfit; if high-resource languages are not trained on enough, the model will underfit. Xue et al. [2021] therefore, took the approach used by Devlin [2018], Conneau et al. [2020], Arivazhagan et al. [2019] and sampled examples according to the probability $p(L)\alpha|L|^{\alpha}$, where $p(L)$ is the probability of sampling text from a given language during pre-training and $|L|$ is the number of examples in the language to boost low-resource languages. The hyperparameter α (typically with $\alpha < 1$) allows controlling the degree of "boost" applied to the probability of training on low-resource languages. [Xue et al., 2021] set $\alpha = 0.3$.

2.4.4 Extending Transformer Models for Long-term Context

The maximum number of input tokens for most Transformer architectures is only limited by the underlying hardware, but most of them are only evaluated for a certain number of input tokens, e.g., BART used 1024 but can be extended to any number [JunhyunB, 2020]. With an increasing number of input tokens, optimized Transformer architecture tend to outperform. Whether to use optimized or simply extremely large-scaled models is a highly discussed topic [DickMan64, 2022].

[1] choosing the right activation function is in many cases trial and error. There is currently no scientific explanation why GeGLU works better. It is an ongoing debate. Possible reasons for the out performance could be that GeGLUE is smoother near zero and is in all ranges differentiable, thereby allowing gradients (although small) in the negative range[Kwag, 2022].

2.4.5 Efficient Transformers

Since it's debut in 2017 an almost unmanageable number of modifications of the Transformer have been published. [Tay et al., 2022] is a survey that focuses on *Efficient Transformers*, Vanilla Transformer adaptations that improve computation or memory efficiency. The authors selected significant models and classified them in a taxonomy, as depicted in Figure 2.7. In this section, we will describe a few of these efficient Transformers and additionally investigate the Infinity Former, a recently published model. We also considered [Lin et al., 2022], to see whether there are new "efficient Transformers", that did occur in [Tay et al., 2022].

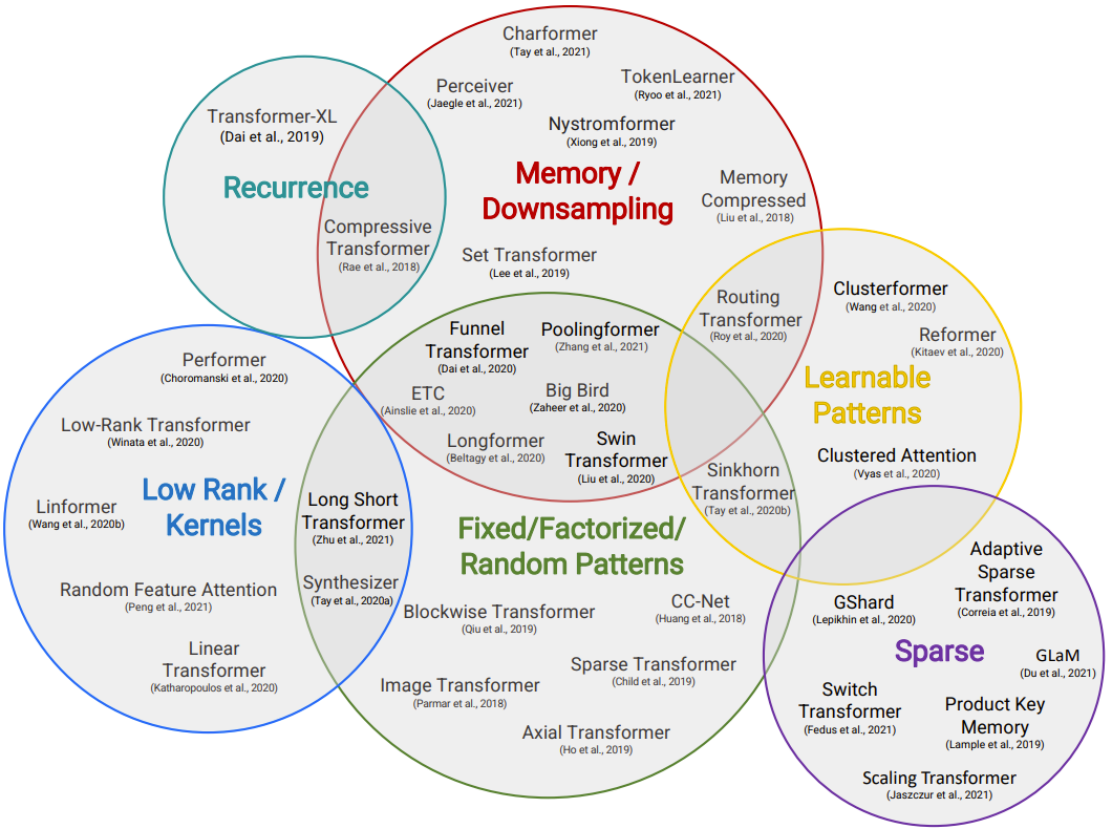


Figure 2.7: Taxonomy of efficient Transformers by [Tay et al., 2022]

Sparse Transformer [Child et al., 2019] introduced two novel attention patterns for training Sparse Transformers in a two-dimensional way:

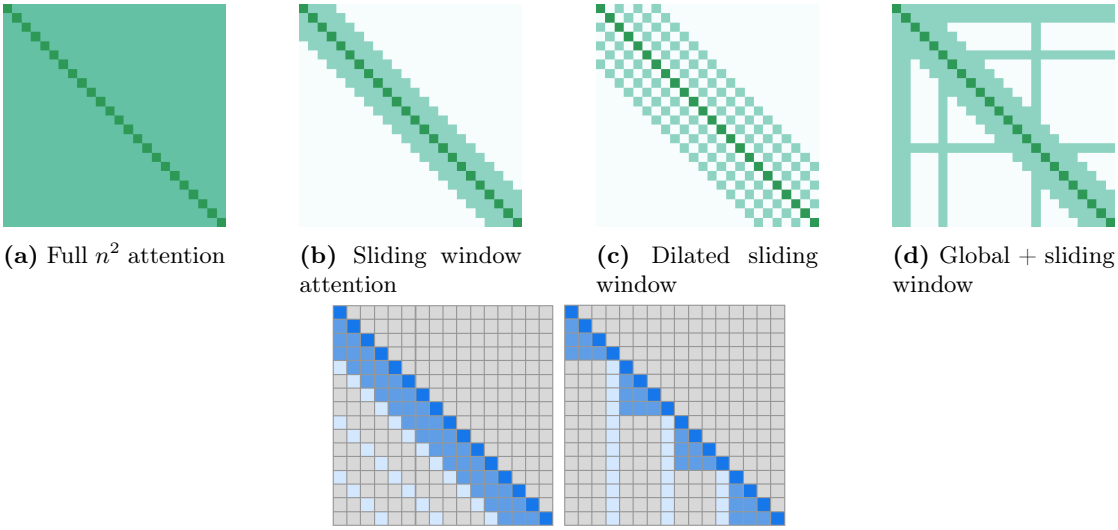
1. **Strided Attention**: In this type of attention, one head attends to the previous l locations, and the other head attend to every lth location, where l is the stride and chosen to be close to $\sqrt{n}$ (n is the total number of elements). This pattern is visualized in Figure 2.8(e).

2. **Fixed Attention**: Strided Attention works well, when the data naturally has a structure that aligns with the stride, e.g. images. Text and other forms of data without a periodic structure, [Child et al., 2019] found that the network can fail to properly route information with the strided pattern. For those cases, they use a fixed attention pattern, where specific cells summarize previous locations and propagate that information to all future cells. In Fixed Attention, one step t is defined as $t = l - c$ with l locations and an additional hyperparameter c. For example, if the stride is 128 and $c = 8$, then all future positions greater than 128 can attend to positions 120-128, all positions greater than 256 can attend to 248-256, and so forth. This pattern is visualized in Figure 2.8(f)

Longformer was presented in Beltagy et al. [2020] and trained with RoBERTa [Liu et al., 2019] checkpoint. This model's context was extended to decrease the computational cost. The window size for the diluted sliding window w was set to the maximum sequence length a RoBERTa model can attend to, in their case, 512 tokens. While the Longformer has a memory and time complexity of $O(n(w+k))$, in practice, it is only an improvement if the sequence length n is much greater than 512. Interestingly, the Longformer conversion can not only be applied to all RoBERTa based models, but the authors also stated that the general principle they can be applied to any transformer-based model.

Tay et al. [2022] describe Longformer as a variant of Sparse Transformer [Child et al., 2019], with "Dilated Sliding Windows" as a key distinction, which enables the Longformer to better cover long-ranges without sacrificing sparsity. Furthermore, Longformer achieves its efficiency improvement by using three self-attention window patterns instead of a dense-attention matrix. These patterns aim to capture longer dependency, they are illustrated in Figure 2.8 and described in the following:

1. **Sliding window attention**: A window-attention, with a fixed size, w surrounds each token. These windowed attention is stacked in multiple layer to create a large receptive field. In this field, the top layers have access to all input locations and are thereby capable to incorporate information across the entire input into the



(a) Full n^2 attention **(b)** Sliding window attention **(c)** Dilated sliding window **(d)** Global + sliding window

(e) Strided **(f)** Fixed

Figure 2.8: Comparison of full self-attention a), the Longformer patterns b), c) (Source: [Beltagy et al., 2020]) and the Sparse Transformer patterns d), e) (Source: [Child et al., 2019])

representations, similar to the convolution in CNNs [Wu et al., 2019]. Figure 2.8b illustrates that each token attends to $\frac{1}{2}w$ tokens on each.

According to Tay et al. [2022] windowed attention was first proposed in early local-based attention models (Image Transformer [Parmar et al., 2018], Compressed Attention [Liu et al., 2018] and/or Sparse Transformer [Child et al., 2019]).

2. **Dilated sliding window attention**: Dilating is a method that further increases the receptive field without increasing computation. It is inspired by dilated CNNs [Oord et al., 2016] where the window has gaps of size dilation d (see Figure 2.8c). Assuming a fixed d and w for all layers, the receptive field is $l \times d \times w$, which can reach tens of thousands of tokens even for small values of d. They found that settings with different dilation configurations per head improves performance. This allows some heads without dilation to focus on local context, while others with dilation focus on longer context.

3. **Global attention**: Windowed and dilated attention are not flexible enough to learn task-specific representations. So, Beltagy et al. [2020] added "global attention" on few pre-selected input locations. This attention operation is symmetric, meaning a token with a global attention attends to all tokens across the sequence,

and all tokens in the sequence attend to it. This symmetry is illustrated in Figure 2.8d. Despite that global attention is task specific, it is an easy way to add inductive bias to the model's attention. Furthermore, it is much simpler than existing task-specific approaches, which use rather complex architectures.

4. **Linear Projections for Global Attention**: Beltagy et al. [2020] modified the Transformer model [Vaswani et al., 2017] attention scores by using two sets of projections: 1) Q_s, K_s, V_s to compute attention scores of sliding window attention, and 2) Q_g, K_g, V_g to compute attention scores for the global attention. These additional projections provide flexibility, and they showed that this is critical for best performance on downstream tasks.

Reformer [Kitaev et al., 2020] reduces the memory and time complexity to $\mathcal{O}(n \log n)$ by using two concepts:

1. **locality sensitivity hashing**, a hashing function with the core idea that nearby vectors should obtain a similar hash while distant vectors should not. So, locality sensitivity means that attention is only computed amongst query and keys if they fall in the same hash bucket. Additionally, to maintain causal masking, Reformer assigns and maintains a position index for every query and key. So, it is therefore able to check if each query key comparison preserves the autoregressive property.
2. **reversible layers**: Normally, when training a deep learning model, backpropagation requires the activation function's output to be kept in memory. To save memory, the Reformer recomputes the input and output tensors only during the training. Then it uses the difference between these to approximate the gradients in the intermediate layers. With this technique, the model only needs to store the activations for the input and output layer once.

Linformer [Wang et al., 2020] is based on the idea of *low-rank self-attention*. In this form of attention, Low-Rank Projections on Length Dimensions are employed. Linformer uses additional projection layers to project the length dimension of keys N to a lower-dimensional representation k. So, this low-rank method ameliorates the memory complexity problem of self-attention because the $N \times d$ matrix is now decomposed to a smaller version, with $k \times d$ dimensions. The overall attention computation complexity becomes $\mathcal{O}(n)$, because k is a constant.

Transformer-XL [Dai et al., 2019] uses *segment-based recurrence*. In this approach, adjacent blocks are connected with a recurrent mechanism and allowing information flow between them. This approach reduces the inference time, but because the dense-attention calculations are not sparsified, the computational complexity remains $\mathcal{O}(n^2)$.

Poolingformer [Zhang et al., 2021] uses two-levels of attention as their core concept. The first level attention, a smaller sliding window pattern, is used to aggregate information from neighbor tokens. In the second level attention, the receptive fields are increased with a larger window size. Then pooling of the keys and values is used to reduce the computational cost to $\mathcal{O}(n)$.

∞-former [Martins et al., 2022] is not included in the efficient transformers survey [Tay et al., 2022] since it was released after the survey. It uses *unbounded long-term memory*. This relies on continuous-space attention, which attends over the long-term memory. This reduces the complexity to $\mathcal{O}(1)$ because it becomes independent of the context length. By this ∞-former made a trade-off between memory length with precision. To fine-grain this trade-off, ∞-former samples locations in the long-term memory, that more relevant by the previous step's attention. Consequently, ∞-former attributes a larger space in the long-term memory to the memories stored in those sample locations. This process is called "sticky memories" and improves the precision

Currently, Longformer is the only architecture, which can be applied to checkpoints of other pretrained models. Other models such Long Short Transformer [Zhu et al., 2021], Poolingformer [Zhang et al., 2021], and Cluster-Former [Wang et al., 2021] initialize (in other words copying the weights into a new model from scratch) their model based on pre-trained checkpoint but do not use the checkpoints directly (in other words continue the training of an existing model).

2.4.6 Grammatical Knowledge

Transformer and deep learning models, in general, showed some impressive performance in tasks that require extensive linguistic skills. One necessary aspect for such tasks is grammatical knowledge. [Linzen and Baroni, 2021] investigated whether these models are inducing human-like grammatical knowledge from their raw input data, consequently,

whether they can shed new light concerning which innate structure is necessary for language acquisition. [Linzen and Baroni, 2021] divides the discussion of Grammatical Knowledge in deep learning models in two different categories:

Nature Versus Nurture Both humans and machines do not start as tabulae rasae, when it comes to language learning, but their biases are quite different. Language models are not constrained to perform only syntactically defined, recursive operations, but **rather sequential left-to-right processing** (RNNs) and **content-addressable memory storage and retrieval** (gating, attention). If we see a neural language models perform and solve syntactic tasks in a way that is consistent with human syntactic competence, we could conclude that a human-like constrained system is not necessarily needed to acquire the relevant abilities. But the architectural bias should be ignored, and should not mislead to the conclusion, that statistical learning from data alone suffices for acquiring the relevant abilities. At the moment, Linzen and Baroni [2021] did not find which architectural features could be fundamental for learning syntax.

To address the bias-problem in neural language models, it could be fruitful to **inject linguistic constraints**. Linguistic Constraints are generally used for representing properties that an object must satisfy [Blache, 2000]. They can have a general grammatical nature such as "an output sentence has to have at least one subject, verb and object" or could be more specific such as "an output sentence has to fulfill the Leichte Sprache Guidelines" (see section 2.1.4). Practical implementations for this approach are still in their infancy.

Amount and Nature of Training Data A great imbalance quickly becomes apparent, when comparing the amount and nature of training data, that is used by human first language learner (children) and neural language models. Children use a few books, which are targeted for language learner, to learn a language, while nlm use entire libraries which contain an unspecified and broad selection of books. Furthermore, children additionally learn in an environment of social interaction or feedback, while nlm "sit in the dark" and only digest the textual data. The real question, according to Linzen and Baroni [2021], is asking is how much can be learned from huge amounts of written linguistic data alone.

2.5 Evaluation Measures for Text Simplification

To properly answer the research questions defined earlier, a suitable evaluation metric needs to chosen. Grabar and Saggion [2022] and other works conclude, that the evaluation of simplification remains understudied. It is a hard task to solve, which roots in the challenge of defining a standard simplification output. Grabar and Saggion [2022] names two reasons behind this root challenge: (1) it is not factual since it relies on transformations managed by "simplificators" (human or automatic nature) and (2) it is heavily based on own the knowledge and opinion of people and thereby not consensual. Furthermore, unlike for standard language, a *native simplified-language speaker does not exist* [Siddharthan, 2014].

Evaluation measures are intended to serve as an assessment of the quality of simplification research. Evaluation can be divided in human and automatic evaluation. In this work, we will only focus on the automatic approach. In theory, human evaluation should deliver more precise results, but automatic approaches are less expensive by far and aim to correlate with human evaluators. Since simplification can be considered as monolingual translation of documents from original to simplified languages, traditionally translation-related metrics are also applied to simplification. Grabar and Saggion [2022] concludes that metrics may be correlated with some of the three criteria (semantic, grammaticality and simplicity) but to this day none of them covers all the criteria.

In the following, we will give an extensive overview on the currently most used metrics for text simplification evaluation. To get a better overview, we selected the most-used metrics, listed them in Table 2.3, and clustered them by the method they used. These methods outline the subsection-strucutre of this section, where we further describe the metrics. We additionally investigated whether the metrics are structure- and grammar-aware, their point of reference (input or target), and if they employed any robustness mechanism.

2.5.1 Vocabulary

A simple way of measuring a text's level of readability, is to employ multiple language-level (e.g., defined by the Common European Framework of Reference for Languages [Europarat, 2020]) vocabularies and then check each word to see which vocabulary it is listed in. **OOV** (out of vocabulary) uses this idea. It describes the rate of words

Name	Output to Target	Output to Input	Grammar	Strucute	Robustness mechanism	Method	Similarity	Readability
OOV	X	O	-	-	-	vocabulary	-	X
BLEU	X	O	-	X	-	n-grams	X	-
ROUGE	X	O	-	X	-	n-grams	X	-
iBLEU	X	O	-	-	-	n-grams	X	-
FKBLEU	X	O	μ	-	-	n-grams, FKGL	X	X
FKGL	-	-	-	-	-	word, sentence, syllable quantity	-	X
SARI	O	X	μ	-	semantic annotation	n-grams	X	-
changed	X	-	-	-	-	edits	X	-
potential	X	O	-	-	-	edits	X	-
TERp	X	O	-	X	semantic-aware substitutions	edits	X	-
METEOR	X	O	-	X	-	alignment	X	-
METEOR++	X	O	X	-	-	alignment, vocabulary sub-set	X	-
SEMA	X	O	-	-	semantic alignment	alignment, relations	X	-
SAMSA	X	O	-	-	semantic annotation	alignment, relations	X	-
SWSS	X	O	-	-	semantic weights	alignment, relations, vocabulary sub-set	X	-
BERTscore	X	O	μ	μ	embeddings	embeddings	X	-
BLEURT	X	O	μ	μ	embeddings	embeddings	X	-
MoverScore	X	O	μ	μ	embeddings	embeddings	X	-
QuestEval	X	O	-	μ	-	factuality	X	-
SUP	-	-	-	X	-	entropy	-	-
BOW-Proba	-	-	-	-	-	entropy	-	-

Table 2.3: Where X means it was used this way in its publication paper, - it cannot be used this way, O means it could be theoretically used this way, and μ means that does not cover this aspect, but it is a side effect of its method.

which are not present in the reference vocabulary [Vu et al., 2014]. For simplification, a low number of out of vocabulary words indicates that a good readability of the text. This metric highly depends on the reference vocabulary. There is a wide variety of alternative readability scores to OOV. Vu et al. [2014] noticed that these metrics are correlated with syntactic simplicity, since simplifications often output longer sentences these metrics become less suitable [Wubben et al., 2012]. Grabar and Saggion [2022] even conclude that such measures are not correlated with simplicity. Metrics that only rely on vocabularies do not consider any structural information in the calculation.

2.5.2 N-Grams

N-grams are contiguous sequences of n items from a text sequence, these items can be e.g., phonemes, syllables, characters or words. In the following we present some metrics, that are based on finding certain n-grams in the generated output (or candidates) and in the target/reference text and evaluating the similarity. N-grams-based approaches have some form of tokenization as prerequisite and can be highly affected by the choice of the tokenization process.

BLEU or (bilingual evaluation understudy) [Papineni et al., 2002] consists of two components: n-gram precision and sentence brevity penalty. N-gram precision is an adaptation of precision, that takes word order (n-grams) into account. Just as with precision, the score increases with the degree to which the output is similar to the target. The modified precision score, p_n, for the entire test corpus is calculated as follows:

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{gram_n \in \{C\}} Count_{clip}(gram_n)}{\sum_{C' \in \{Candidates\}} \sum_{gram'_n \in \{C'\}} Count_{clip}(gram_n)} \tag{2.4}$$

$$Count_{clip} = min(Count, Max\_Ref\_Count) \tag{2.5}$$

where C is a candidate sentence of the Candidate Sentence Corpus $Candidates$, and $gram_n$ is an n-gram in C. $Count$ is the number of occurrences of the word in the candidate and Max_Ref_Count is the highest number of occurrences of the word in the references.

An evaluation metric should enforce the proper length of a candidate, since to to long candidates are already penalized by the modified n-gram precision measure, [Papineni et al., 2002] employed an additional sentence brevity penalty, to avoid the other negative extreme. They first summed the best match lengths for each candidate sentence in the corpus to obtain the test corpus' effective reference length r. And defined the brevity penalty, BP, to be a decaying exponential in $r = c$, where c is the total length of the candidate translation corpus:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{1-r/c} & \text{if } c \leq r \end{cases} \tag{2.6}$$

So, the complete BLEU is calculated as:

$$\text{BLEU} = \text{BP} \cdot exp\left(\sum_{n=1}^{n} w_n \log p_n\right) \tag{2.7}$$

where N is the length and w_n the positive weights.

Works such as [Martin et al., 2018] noticed that it correlates with grammaticality and semantics, but not with the degree of simplicity. Furthermore, BLEU is a corpus metric and is generally advised to apply it in a scenario where the corpus is large. In practice, BLEU was mostly used on texts containing more than 1,000 sentences [Marie, 2022].

ROUGE or Recall-Oriented Understudy for Gisting Evaluation [Lin, 2004], is the most commonly used evaluation metric for summarization. ROUGE compares a candidate to a set of reference summaries and computes the co-occurrences of n-grams between the candidate and each reference. It is similar to BLEU [Papineni et al., 2002] and there are two main flavors of ROUGE: ROUGE-N and ROUGE-L.

ROUGE-N calculates the recall of n-grams between the candidate and a set of reference texts :

$$\text{ROUGE-N} = \frac{\sum_{S \in \{References\}} \sum_{gram_n \in \{S\}} Count_{match}(gram_n)}{\sum_{S \in \{References\}} \sum_{gram_n \in \{S\}} Count(gram_n)} \tag{2.8}$$

where n stands for the length of the n-gram, $gram_n$, and $Count_{match}(gram_n)$ is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries.

ROUGE-L is based on the longest common subsequence (LCS), instead of n-grams. The LCS includes common words between the candidate and the reference in the same order, which do not need to be necessarily consecutive. A longer shared sequence should indicate more similarity between the two sequences. This approach has two advantages: 1) More robustness to meaning-invariant lexical permutations, because the words only have to be in the same order and not necessarily consecutive 2) It does not depend on the predefined n-gram length, since the LCS is automatically set up. The metric is an F-measure and is calculated as follows:

$$R_{lcs} = \frac{LCS(X,Y)}{m} \tag{2.9}$$

$$P_{lcs} = \frac{LCS(X,Y)}{n} \tag{2.10}$$

$$F_{lcs} = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{2.11}$$

Where m and n are the lengths of texts X and Y respectively, P_{lcs} is the precision measure, R_{lcs} is the recall measure and β specifies the weighting.

Gaskell et al. [2020] conclude that ROUGE only performs a surface-level comparison and even penalizes lexical and compositional diversity despite a high semantic similarity of the candidate, and it's reference. In regard to summarization, they further interpret ROUGE as a necessary but not fully satisfying condition; high scores do not necessarily mean the model is producing good outputs, but very low scores are a red flag. Since, a minimum text similarity should be achieved by the model, because certain n-grams overlap should be present when the input and output language is the same.

FKGL or Flesch-Kincaid Grade Level (FKGL) [Kincaid et al., 1975] is a readability metric which is commonly reported as a measure of simplicity. It's current main usage is as a tool for teachers, parents and others to judge the readability level of various texts. It is a score that is oriented to the U.S. grade level and, if the score is greater than 10, can also mean the number of years of education required to understand this text. A shortcoming of this metric is, that short sentences could get good scores even if they are ungrammatical, or do not preserve meaning because the metric relies on average sentence lengths and number of syllables per word, leading short sentences would get good scores even if they are ungrammatical, or do not preserve meaning [Wubben et al., 2012]. Therefore, FKGL scores should be interpreted with caution. The grade level is calculated with the following formula [Kincaid et al., 1975] :

$$0.39\left(\frac{\text{total words}}{\text{total sentences}}\right) + 11.8\left(\frac{\text{total syllables}}{\text{total words}}\right) - 15.59 \tag{2.12}$$

iBLEU [Sun and Zhou, 2012] is an extension of BLEU that focuses on measuring diversity and adequacy of the generated paraphrase.

FKBLEU [Xu et al., 2016] combines an existing metric for paraphrase generation iBLEU with the Flesch-Kincaid Index [Kincaid et al., 1975], a commonly used readability metric. That means that sentences with higher FKBLEU values are better simplifications with higher readability.

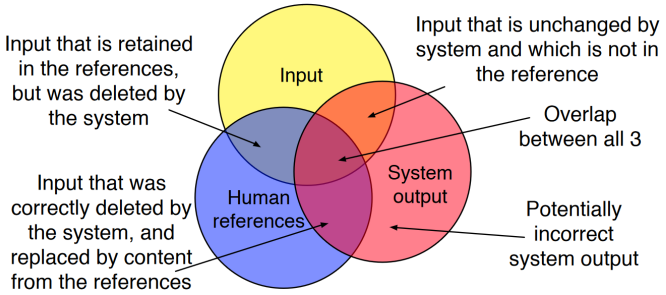


Figure 2.9: The different regions that are treated differently with SARI metric. (Source: [Xu et al., 2016])

SARI or **s**ystem output **a**gainst **r**eferences and against the **i**nput sentence [Xu et al., 2016] compares the generated output not only against the reference but, as the name indicates, against the source data. By individually treating certain aspects of the output, as shown in 2.9. Xu et al. [2016] showed in their work that both FKBLEU and SARI correlate stronger to human judges than BLEU.

2.5.3 Edits

Another way of comparing texts is by measuring the edit distance between two sequences. In other words, the distance between two texts is the minimum number of operations required to change one text into the other. The probably most famous member of this class is the Damerau–Levenshtein distance [Damerau, 1964], which works on a word-level and considers all insertions, deletions or substitutions of a single character, or transposition of two adjacent characters that are required to change one word into the other into the calculation. But there are a few more recent implementation of the edit-measuring approach.

TERp (Translation Edit Rate plus) Similar to the Damerau–Levenshtein distance, the Translation Edit Rate (TER) [Snover et al., 2006] counts all required operations to change one text into the other, but additionally normalizes the score by the average length of the references. The fewer transformations to fit the reference sentence, the better. Furthermore, TERp (Translation Edit Rate plus) [Snover et al., 2009] advance TER by using word alignment, stemming and synonymy detection to allow matches between semantically equal words. Moreover, it uses probabilistic phrasal substitutions to align phrases in the output and target.

changed [Horn et al., 2014] measures the percentage of test examples where the system suggested changes, regardless of their correctness or quality, with the objective to produce the highest number of changes.

potential [Paetzold and Specia, 2016] calculates how many of the generated output candidates are in the reference data, with the objective to produce the highest number of such candidates. In its publication paper, the authors used it to measure the quality of generated substitutions for complex words.

2.5.4 Alignment

Another approach for evaluation is to align parts (e.g., words or complete sentences) of the translation and measure their similarity. These aligned parts do not have to match on the character-level. For instance, the words 'run' and 'walk' can be aligned even though they are not a matching uni-gram. In machine translation, it is important, that all the original information is transferred into the translated output. This aspect is called *information retention*. Additionally, there is *semantic retention*, which involves not only the superficial factual information but also the semantics. Although for classical translation information both semantic and information transfer are very important, omission of information during text simplification is often allowed and sometimes even required (see Table 2.2).

METEOR [Denkowski and Lavie, 2011] evaluates translation candidates by aligning them to references and computing a sentence-level similarity scores. METEOR is a statistical model, that has to tuned language-specific, the authors in Denkowski and

Lavie [2011] showed that balanced tuned version of Meteor consistently outperforms BLEU.

METEOR++ [Guo et al., 2018] Extends METEOR with Copy Knowledge Extraction. The author discovered that copy knowledge in which the words always have a high possibility of co-occurrence in paraphrase pairs. Furthermore, they proposed a simple statistical method to extract copy knowledge based on the given parallel monolingual paraphrases.

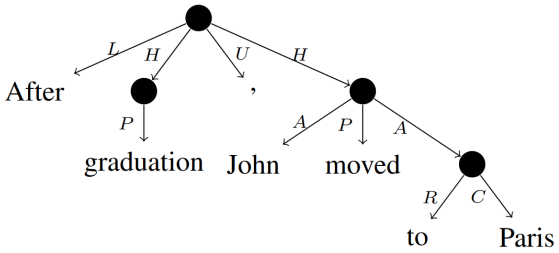


Figure 2.10: Tree approximation of UCCA annotations of an example sentence (Source: [Abend and Rappoport, 2013])

SAMSA or Simplification Automatic evaluation Measure through Semantic Annotation [Alva-Manchego et al., 2019] is the first structure-aware measure for text simplification in general, and the first to exploit semantic structures for this purpose. The authors developed SAMSA with the premise that a structurally correct simplification fulfills two conditions: (1) each sentence holds a single event (UCCA Scene) from the input (2) the main relation of each of the events and their participants are preserved to the output. SAMSA uses two core components: semantic annotation and word-to-word alignment.

In the word-to-word alignment, words in the input and one or zero words in the output are aligned. Thus permitting SAMSA not to penalize outputs that involve lexical substitutions.

They used UCCA as the semantic annotation scheme, [Abend and Rappoport, 2013] It aims to represent the text's main semantic phenomena and abstracting away from syntactic detail. UCCA uses a Scene as its basic notion of the foundational layer. It describes a movement, an action or a state which is persistent in time. Each Scene consists of one main relation, which can be either a Process or a State. Figure 2.10. The alignment can be done either manually or automatically, using the TUPA parser

(Transition-based UCCA parser; [Hershcovich et al., 2017]) for UCCA. TUPA officially support English, German and French [2].

SEMA or text Simplification Evaluation Measure through Semantic Alignment [Zhang et al., 2020b] is an optimization of SAMSA. It replaces the string alignment with semantic alignment, including three semantic alignment methods: full alignment (SEMA-base), partial alignment, and hyponymy alignment. SEMA makes it possible to evaluate semantic retention and Zhang et al. [2020b] showed that SEMA has high applicability in Chinese corpus, and where the first to apply a semantic retention metric on a Chinese corpus.

SWSS or Semantically Weighted Sentence Similarity [Xu et al., 2020] is similar to SAMSA and also uses core words from UCCA. Furthermore, they introduced three penalties concerning statistical differences of two UCCA representations to create a more accurate output: 1) The ratio between number of scenes of two representations. 2) The ratio between the counts of nodes of two representations; and 3) The ratio between counts of edges towards critical semantic roles of two representations, which are Process, State and Participant. This value is the sum of the number of scenes and the count of all arguments in the sentence. All three penalties follow the same hypothesis, that the number of these attributes should stay the same after the simplification if the content is completely retained. These penalties are applied to the Word Match Score and add more accuracy to the final score.

2.5.5 Embeddings

In the previous sections, we discussed metrics that worked with lexical tokens. The process of embedding transforms these tokens into representations, typically in the form of a vector (embeddings (see Section 2.1.2)). The following metrics are based on these representations or embeddings instead of solely tokens. The degree of knowledge that can be incorporated is an interesting and ongoing debate. They store some degree of structural and grammatical information, we discussed the limits of the grammatical perception in section 2.4.6.

[2] https://github.com/danielhers/tupa

BERTSCORE was introduced in [Zhang et al., 2020a] as a BERT-based [Devlin et al., 2019] evaluation metric, that is designed to be simple, task agnostic, and easy to use. It compares two texts based on the weighted cosine similarities of their embedded representations.

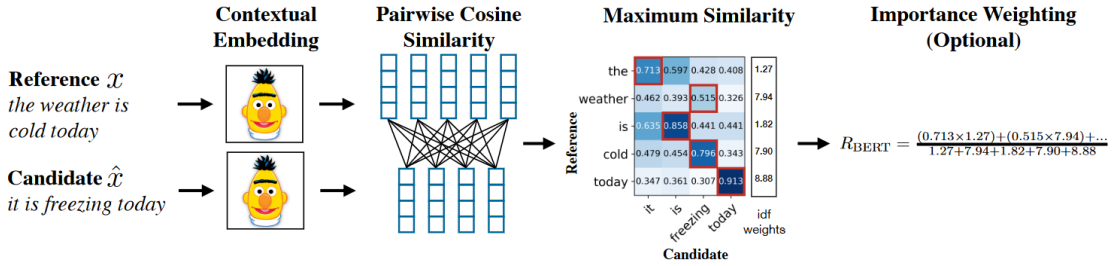


Figure 2.11: Schematic calculation of R_{BERT} (Source: [Zhang et al., 2020a])

To calculate the BERTSCORE a reference sentence $x = \langle x_1, \ldots, x_k \rangle$ and a candidate sentence $\hat{x} = \langle \hat{x}_1, \ldots, \hat{x}_l \rangle$ are represented as contextual embeddings (see section 2). Then, their matching is computed using cosine similarity and optionally weighted with inverse document frequency scores. Figure 2.11 illustrates this process. Using embeddings allows a soft approach of measuring similarity instead of exact-string or heuristic matching. The cosine similarity of a reference x_i and a candidate x_j is $\frac{x_i^\top \hat{x}_j}{||x_i|| ||\hat{x}_j||}$, [Zhang et al., 2020a] reduced the calculation to $x_i^\top \hat{x}_j$ by using pre-normalized vectors. Contextualized embeddings offer the great benefit that, despite the fact that this measure only uses isolated tokens, a non-negligible part of information from the rest of the sentence is included in the calculation.

Zhang et al. [2020a] conducted extensive experiments with various configuration choices for BERTSCORE and found that BERTSCORE achieves better correlation than common metrics. However, they did not find any configuration of BERTSCORE that clearly outperforms all others. In their setting, F_{BERT} was the most reliable, BERTSCORE and recommend it for machine translation evaluation in general.

In detail, Zhang et al. [2020a] used greedy matching to maximize the matching similarity score, so that each token is matched to the most similar token in the other sentence. Recall score matches each token in x to a token in $\hat{x}$, precision matches each token in $\hat{x}$ to a token in x and F1 combines recall and precision:

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} max_{\hat{x}_j \in \hat{x}_i} x^\top \hat{x}_j \tag{2.13}$$

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_i \in \hat{x}} max_{x_j \in x_i} x^\top \hat{x}_j \tag{2.14}$$

$$F_{BERT} = 2\frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}} \tag{2.15}$$

Furthermore, Zhang et al. [2020a] showed that incorporating importance weighting and baseline rescaling can improve the performance of BERTscore. The original source code [3] has been modified since the publication and supports numerous underlying pretrained models. Some support very long input sequences and various languages. A recent study [Alva-Manchego et al., 2021] compared different sentence simplification evaluation metrics (BLEU, iBLEUCH, SARI, BERTscore) and concluded in the fact that BERTscore should be used in the future.

Despite this recommendation, the underlying models are known to have flaws. [Hanna and Bojar, 2021] points out that BERT in particular has been shown to be, in certain scenarios: 1) insensitive to negation [Ettinger, 2020], 2) insensitive to word order [Pham et al., 2021], 3) inexact representations of numbers [Wallace et al., 2019], and 4) fragile to named entities [Balasubramanian et al., 2020].

Nonetheless, it is hard to say how these shortcomings might manifest in BERTscore or even in language or domain environments that are tested by the authors. [Hanna and Bojar, 2021] investigated the shortcomings of BERTscore:

- Penalizing translations that **incorrectly render function words** seems to be the most difficult for BERTscore. This includes sentences with tag questions. In one example, that was difficult for BERTscore, the reference is "You're crazy, aren't you?", and a secondary good translation is "You're crazy, right?", while incorrect sentences are "You're crazy, or?" and "You're crazy, are not you?".
- BERTscore fails to assign low scores when a bad candidate sentence has high lexical overlap with the reference in terms of content words.

[3] `https://github.com/Tiiiger/bert_score`

Despite this, [Hanna and Bojar, 2021] showed that BERTscore outperformed BLEU score across all the datasets and conditions they tested. And confirm the recommendation of [Alva-Manchego et al., 2021]. The main drawback of BERTscore in comparison to traditional metrics such as ROUGE is, that it is less transparent. Nonetheless, it correlates in a vast number of cases more with human-judgement than these traditional approaches. For example [Chan et al., 2021] found a correlation of low BERTscore of a project description and the degree of willingness of Kickstarter investors to invest in the described project.

One aspect that is inherently part of BERT is grammar. As described in Section 2.3 and 2.4 embeddings store context information and thereby the grammar and structure of a text to some degree. [Zaczynska et al., 2020] investigated the degree of syntactic grammar information, that is store in the German BERT. They explicitly tested for German-specific syntactic constructs, and observe that the model performs well. For this purpose, they created an agreement test. In this test, two sentences, a grammatical one and an ungrammatical one, are forwarded through a model. The sentences differed minimally from each other at only one locus of (un)grammaticality, i.e. one word. And monitored whether the model preferred the correct or the incorrect sentence.

BLEURT Sellam et al. [2020] pre-trained BERT to act as an effective evaluation metric, which is resistant to domain drift. The authors used unsupervised pre-training combined with fine-tuning on human evaluations. Its main contribution is the novel pre-training scheme using a synthetic dataset tailored to evaluation. Their dataset contains 6.5 million samples, each containing a sentence extracted from Wikipedia together with an artificially perturbed version of that sentence. Their intentions are that via the synthetic perturbations, the model will be familiar with many of the issues it might encounter between the hypothesis and target summaries when deployed as an evaluation metric. To this day, there are only English Versions of BLEURT available.

MoverScore [Zhao et al., 2019] was developed at the same time as BERTscore and could be described as a generalized version of BERTscore combined with Word Mover's Distance (WMD, [Kusner et al., 2015]). WMD defines the similarity of two documents as the minimum distance between their embedded representations. WMD is an instance of the widely-studied Earth Mover's Distance transportation problem, which has numerous efficient solvers.

BERTscore can be viewed as a hard-aligned case of WMD, meaning that each token 'travels' to the most semantically similar token in the other sequence, leading to a hard one-to-one mapping for each token in the sentence pairs. MoverScore, on the other hand, uses soft alignments and the mapping across the sentence pairs is determined by solving the following constrained optimization problem:

$$s.t.F1 = f_{x^n}, F^\top 1 = f_{y^n} \tag{2.16}$$

Source Text: In the Soviet years, the Bolsheviks demolished two of Rostov's principal landmarks St Alexander Nevsky cathedral (1908) and St George cathedral in Nakhichevan (1783-1807).			
Simplification: The Bolsheviks destroyed St. Alexander Nevsky cathedral and St. George cathedral in Nakhichevan during the Soviet years.			
Generated Question	**Answers**		**F1**
	Source	Simplif.	
When did the Bolsheviks demolish St George cathedral?	the Soviet years	Soviet years	0.8
What cathedral was demolished in 1908?	Rostov	Unanswerable	0.0

Table 2.4: Example of automatically generated and answered questions by QUESTEVAL given a source text and its simplification. (Source: [Scialom et al., 2021b]

2.5.6 Factuality

The previous metrics evaluated on a textual-level. An alternative approach is to measure which degree of factuality the input was transferred into the output. One representative of this class is **QuestEval** [Scialom et al., 2021a]. It is a question-answer based evaluation method. It can be used to access the factual consistency (i.e. precision) or the relevance (i.e. recall) of the evaluated output, in comparison to another document. QuestEval contributes to previous works of question-answer metrics by 1) unifying the precision and recall-based QA metrics to obtain a more robust metric 2) proposing a method to learn the saliency of the generated queries, to make integrating the notion of information selection possible. Furthermore, QuestEval does not require any reference, thus it is very well applicable in situation with very few or no references. Scialom et al. [2021b] showed that QuestEval shows good results on sentence simplification and the source code publicly available [4]. Figure 2.4 shows an example evaluation.

[4] https://github.com/ThomasScialom/QuestEval

2.5.7 Entropy

One goal of simplification is to make the text more accessible. For narrative texts, this could mean to increase the information density and delete side information. Furthermore, text generation models tend to become redundant with longer target outputs. Therefore, it is useful to measure the redundancy of a text. [Kontoyiannisy, 1997] suggest entropy as a characterization, or measurement, of redundancy.

SUP-Entropy [Kontoyiannisy, 1997] describes an entropy estimator that can be used to determine the entropy of a text by calculating the length of the shortest prefix starting at x_i, that does not appear starting anywhere in the previous i symbols x_0, x_1, x_{i1}, and denote this length by l_i. This prefix-length l_i can be thought of as the length of the next phrase, after the past up to time $(i-1)$ has been encoded. Since, as i grows, there is no restriction on how far into the past we can look for a long match. In other words, this metric measures the surprise value of a sub-string.

We further call it the **shortest unique prefix** or *SUP* metric. [Kontoyiannisy, 1997] used this metric on a bit-level, but it can as well be used on a word-level. Consider this example:

$$\text{Ich Du Du Ich Ich} \underbrace{\text{Du Ich Du}}_{l_5=3} \text{Ich Du Du Ich Ich Du Ich Ich Du}$$

then for $i = 5$ we get $l_5 = 3$. This is described by the following formula:

$$\hat{H}_N = \left[\frac{1}{N} \sum_{i=1}^{N} \frac{l_1}{log(i+1)} \right]^{-1} \tag{2.17}$$

where M is the largest index or the sequence length +1 and $N < M$. [Kontoyiannisy, 1997] did not elaborate on how M should be chosen.

BOW-Entropy Additionally, a **bag-of-word entropy** approach is possible:

$$I(w) = -\log_2(p(w)) \tag{2.18}$$

$$p(w) = \frac{count(w)}{n} \tag{2.19}$$

$$H(W) = \sum_{w \in W} p(w) \cdot -\log_2(p(w)) \tag{2.20}$$

$$= \sum_{e \in W} p(w) \cdot I(w) \tag{2.21}$$

where w is a word in the bag of words W, $I(w)$ is the entropy of w, $p(w)$ is the probability of w, $count(w)$ is the number of w in W, n the total length of W, and $H(W)$ the text level entropy. [Kontoyiannisy, 1997] showcased their metric on several different English texts, including the King James Bible, a concatenation of four novels by Jane Austen and two novels by James Joyce. Their results showed that their entropy metric captures statistical structure and descriptional complexity, but not the complexity that comes from the actual contextual and semantic meaning of the text. Since we have a similar domain, we hypothesize that this metric also captures this information in our scenario.

2.6 Previous Work

The first (Rule-based) Automatic Text Simplification System Specia [2010] introduced statistical machine translation to the automatic text simplification task, using data from a small parallel corpus (roughly 4,500 parallel sentences) for Portuguese.

The first German – Simple German corpus The first German corpus, **GEO-GEOlino corpus**, on text simplification was introduced by Hancke et al. [2012] in 2012. Their corpus consists of unaligned articles from GEO (similar to National Geographic) and GEOlino (GEO's edition for children). They used this new data set to train statistical classifiers to predict the reading level of German texts. They took syntactic, lexical, and modeling features such as "Number of pronouns per sentence" or "suffix token ratio", which have shown good results for the same task on English data. Then they added a novel group of features: language-specific morphological complexity indicators. They examined a broad set of inflectional properties for German and used the derivational and inflectional morphology of nouns as features for readability classification of German for the first time. They showed that these novel morphological features are especially good

indicators for reading level, outperforming all other feature groups, when considered in isolation. Their corpus was later improved and enlarged later by Weiß and Meurers [2018]. They added unaligned transcripts of two German TV News shows: "Tagesschau" (targeting adults) and "Logo!" (targeting children). More recently, Aumiller and Gertz [2022] published a document-aligned dataset with a similar domain: lexicon articles for adults and for children.

The first *aligned* German – Simple German corpus Klaper et al. [2013] published the first sentence-aligned German simplification data set containing 270 articles from five different websites, mainly of organizations that support people with disabilities.

The first (Rule-based) Automatic Text Simplification System for German Suter et al. [2016] argues that the corpus from [Klaper et al., 2013] is not sufficiently enough large to train a statistical machine translation system that works reasonably well. They conducted the first (Rule-based) Automatic Text Simplification System for German and evaluated it on a short article on the arrival of the Swiss team at the Special Olympics in Korea. It consists of 135 words in six sentences and features many aspects of standard language.

The first data-driven Automatic Text Simplification System for German Säuberli et al. [2020] introduced the first parallel corpus for data-driven automatic text simplification for German. Their **APA** corpus contains 3,616 sentence pairs based on News Articles. They compared seven different Transformer encoder-decoder models and concluded their corpus was not large enough to sufficiently train a neural machine translation system that produces both adequate and fluent text simplifications. Later Spring et al. [2021] used the same neural machine translation models and further evaluated the levels of simplifications which were generated by the models.

Later Battisti et al. [2020] collected a larger corpus, where 378 texts contain document alignments. The corpus contains German, Austrian and Swiss PDFs and web-pages. Mostly from websites of governments, specialized institutions, and non-profit organizations (92 different domains). They applied sentence-alignment to their data and evaluated two freely available tools: Customized Alignment for Text Simplification (CATS) [Štajner et al., 2018] and MASSAlign [Paetzold et al., 2017]. For their data, CATS outperformed MASSAlign.

Rios et al. [2021] created a new corpus: 20m. Based on Swiss newspaper articles. They adapted mBART [Liu et al., 2020] with Longformer Attention [Beltagy et al., 2020] and applied it to the task of document-level text simplification, with their code available at [Rios, 2020].

Current baseline model Based on four corpora: the Web, APA, Wikipedia, and a corpus of their research group (capito) Ebling et al. [2022] created a gold standard for five sentence alignment methods (MASSAlign, CATS, LHA, SBERT, Vecalign; with CATS featuring three sub-methods). They found that LHA performed best on five out of the seven datasets (Web, Wikipedia, capito A1, capito A2, capito B1, APA A2, APA B1). Furthermore, they used the LHA alignments for the first sentence-based neural language model-based automatic simplification of German (baseline model).

Existing Tools Stodden and Kallmeyer [2022] published a web-tool to make the creation and modification of simplification corpora less difficult. However, they only offer very limited options of automatic sentence-alignment.

Furthermore, capito is currently the only company that offers the automatic translation to simple German.

3 Methodology

This section will describe the method used to train and evaluate the pre-training used for the extended language model and fine-tuning for the simplification task. We start by describing our methodology and the design choices made for evaluating. This is followed by a description of the training procedure, evaluation, and datasets used in relation to how it best would answer our research questions. Afterwards follows an in-depth description of the problems with training vast and memory-intensive models, hardware limitations, and methods that can be used to combat these problems. The chapter concludes with a description of the tooling, libraries, and hardware used to run the experiments.

3.1 Design of a Long-context Text Simplification Model

Our goal was to train a text generation model with a larger context [1], so that it can operate on document-level. Furthermore, this model should work on narrative texts. Since there is no publicly available dataset of aligned simple German and Standard German narrative documents, we constructed our own dataset. As described in Section 2.4.4 almost all efficient transformers employ a modification of their architecture in such a way that they must be retrained from scratch. *Longformer* is the only model we know to exist, which could extend the context on a pre-existing (and pre-trained) transformer. In a next step, we investigated pre-trained Encoder-Decoder Transformer models (see Section 2.4), that are pre-trained on German. We searched on huggingface.co[2], which is the largest platform for pre-trained models. First, we looked at all the text2text-generation models (8 551). Then we filtered all the models that include German as a language (225). Then we considered the models with more than 5 000 downloads (30). In the pre-last step we narrowed it down to all models, that are trained on a translation

[1] Most Transformer-based models (such as BERT) have a maximum input length of 510. So, we consider models with a maximum input length above 510 as large-context and models, that employ specific methods for improving the processing efficiency (see Section 2.4.5).

[2] accessed on 10/28/2022

task, that can be German to German. This leaves only facebook/mbart-large-50 and facebook/mbart-large-cc25, both introduced in [Liu et al., 2020] and described in Section 2.4.2. In the last step, we decided to take facebook/mbart-large-cc25 since it has learned fewer languages (25) in comparison to facebook/mbart-large-50 (50) and we reasoned, that the more relative pre-training time is spent on German the better. This process is visualized in Figure 3.1.

Creating simple German versions for long documents is a laborious task, so each sample is relatively expensive to produce. Since our training data set only contains 25 samples, which is an unusually low number for fine-tuning, the model only has few shots to learn the task. Similarly to [Rios et al., 2021] we will apply a text2text-generation pretrained Transformer model. They also used an adapted version of the mBART model [Liu et al., 2020] see section 2.4.2) for the task of document-level text simplification. Similarly, we will use `mbart-large-cc25`[3] with Longformer Attention (see section 2.4.5).

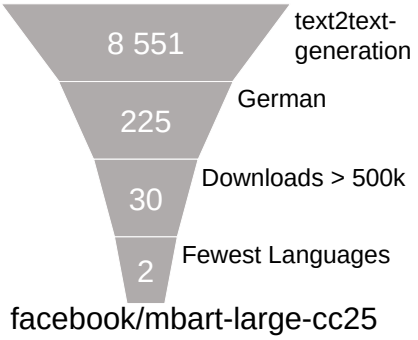


Figure 3.1: Our model selection process

mBART uses a specific input format consisting of the actual sentence and language-tag. Originally, each of the 25 languages have their own language tag. We additionally created two tags: `de_OR` and `de_SI`. `de_OR` indicates that the sentence is written in Standard Language and `de_SI` Simple Language. Both of them are derived from the original German tag `de_DE` and only modified during our fine-tuning process. In other words, we will train mBART to translate `de_OR` to `de_SI`.

[Rios et al., 2021] pointed out that Longformer is a GPU memory intense model. They reduced the original mBART vocabulary from 250k to 20k, keeping only those subwords and their embeddings that are most relevant for German by using a pre-defined word list. Since, we do not have access to such a word list, we left the original vocabulary untouched. On our server, we could train the model only with a batch size of one and a maximum input length of 1024. We still used, the same size of attention window as [Rios et al., 2021]. This results in the following configuration for fine-tuning and for domain adaptation:

[3] `https://huggingface.co/facebook/mbart-large-cc25`, accessed on 10/21/22

	standard mBART*	small mBART*	Fine-Tune mBART	Domain Adaptation mBART
max output length	1024	1024	1024	70
max input length	1024	1024	1024	70
Batch Size	1	4	1	12
Gradient Accumulation	60	15	-	60
attention dropout	0.1	0.1	0.1	0.1
dropout	0.3	0.3	0.3	0.3
attention mode	-	sliding chunks	sliding chunks	sliding chunks
attention window size	-	512	512	512
label smoothing	0.2	0.2	0.2	0.2
learning rate	3e-5	3e-5	3e-10	3e-10
Early Stopping Metric	rougeL	rougeL	BERTscore	-
patience	10	10	-	-
max epochs	-	-	1	1
min delta	0.0005	0.0005	0.0005	0.0005
lr scheduler	Reduce On Plateau	Reduce On Plateau	Reduce On Plateau	Reduce On Plateau
lr reduce patience	8	8	8	8
lr reduce factor	0.5	0.5	0.5	0.5
vocabulary size	250k	20k	250k	250k
beam size	6	6	4	4

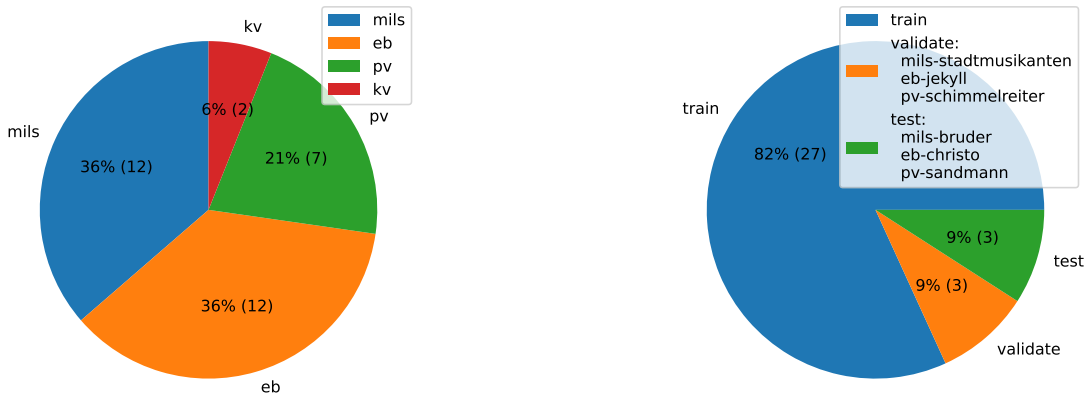
Table 3.1: Overview of our model configuration in comparison to the ones from [Rios et al., 2021] (marked with *)

3.2 Fine-tuning Corpus

We will fine-tune our model on a novel corpus: German Narrative Text Simplifications or short **GNATS** combines data from four different sources. One sample in GNATS is a German Narrative Text in a Simple Version, and it's original Standard Language Version. We had three sources for Standard Language Data:

gutenberg.org: is the oldest provider of free electronic books, founded by Michael Hart. Its collection is conducted by volunteers and is based on donations of public domain e-books (not currently protected by copyright in the United States). Typically, digitized versions of books that were published long ago, so that any related US copyright has expired.

projekt-gutenberg.org: is the largest full-text collection of German Literature. All books are copyright free, that means that the author, translator and illustrator died at least 70 years ago or the copyright owners agreed to the publication. Private usage is unrestricted, for a commercial use a license must be obtained.



(a) **(b)**

Figure 3.2: (a) Depicts the distribution of the data sub-sets and (b) the train-validate-test split in our corpus by number of documents

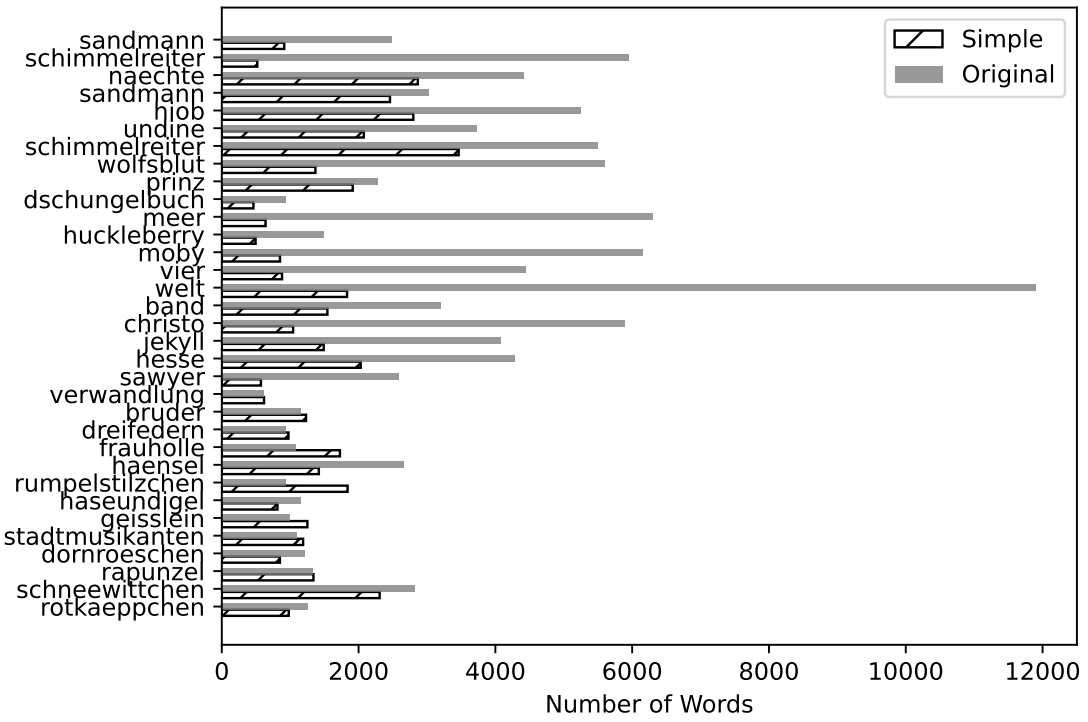


Figure 3.3: Comparison of the number of words of all documents in the corpus, in the Simple Language Version (Simple) and the Standard Language Version (Original)

textgridrep.org - The TextGrid Repository archives humanities research data for the long-term. It sets a value to provide an extensive, searchable, and reusable repos-

itory of texts and images, that is aligned with the principles of Open Access and FAIR.

We try to find a Standard Language Version of each document, preferring the gutenberg.org over the projekt-gutenberg.org version, and the projekt-gutenberg.org over the textgridrep.org version. Figure 3.2 and 3.3 depict the complete GNATS corpus. We selected `mils-stadtmusikanten`, `eb-jekyll` and `pv-schimmelreiter` for validation because their amount of words is close to the average and they represent a different sub-data set. For the same reasons, we selected `mils-bruder`, `eb-christo` and `pv-sandmann` for testing.

We look at six sources in total. The format of all reading samples from Arena Verlags Klassiker are not machine-readable. We had two sources of fairy tales in simple language: NRDS's Märchen in Leichter Sprache and kurzemaerchen. To not overrepresent fairy tales, we only used one of the two. We decided for Märchen in Leichter Sprache, since it is officially Easy Language certified, so the quality should be superior to kurzemaerchen. So we are left with four sub datasets. We listed all the documents, that are part of our dataset in Table 3.2 and all the documents and reason why we rejected them are listed in Table A.1 in the appendix. The complete GNATS-corpus consists of narrative texts, EB, PV and KV are all classics (or literary fiction) and only MILS covers fairy tales (a sub-genre of folklore [Wikipedia, 2022]). Unlike the others, MILS samples include the complete text and are not just excerpts in the form of a reading sample and differ in the literature genre.

We pre-processed all the selected samples. We manually align the simple language and the standard language version to match the extent of the simple language version. Most of the reading samples only covered the first section of the standard language version. So, it was important to truncate the longer version to ensure that both version are theoretically the same document in another language version. Algorithm 1 shows schematically our pre-processing. For more implementational details, please refer to the Github repository (Section 1)

Figure 3.2 show the distribution of the subsets in our corpus. Figure 3.3 compares the number of words of all documents in the corpus, in the Simple Language Version (Simple) and the Standard Language Version (Original).

Full Title	Source-ID	First Published	
Die Abenteuer von Tom Sawyer	eb-sawyer	English	1876
Moby Dick	eb-moby	English	1851
Der Graf von Monte Christo	eb-christo	French	1846
Die Abenteuer von Huckleberry Finn	eb-huckleberry	English	1885
Der seltsame Fall von Dr Jekyll und Mr Hyde	eb-hyde	English	1886
In 80 Tagen um die Welt	eb-welt	French	1873
Erzählungen von Hermann Hesse (Aus Kinderzeiten)	eb-hesse	German	1907
Sherlock Holmes. Das gesprenkelte Band	eb-band	English	1892
Sherlock Holmes. Das Zeichen der Vier	eb-vier	English	1890
20.000 Meilen unter dem Meer	eb-meer	French	1870
Die Verwandlung	eb-verwandlung	German	1912
Wolfsblut	pv-wolfsblut	English	1906
Der Schimmelreiter	pv-schimmelreiter	German	1888
Undine	pv-undine	French	1811
Hiob	pv-hiob	German	1930
Der Sandmann	pv-sandmann	German	1816
Weiße Nächte	pv-naechte	Russian	1848
Der glückliche Prinz	pv-prinz	English	1888
Der Sandmann	kv-sandmann	German	1816
Der Schimmelreiter	kv-schimmelreiter	German	1888
Kinder- und Hausmärchen - Brüder Grimm	All mils-documents	German	1858

Table 3.2: All documents in our corpus from einfachebuecher.de (eb) which are classified as "Klassiker" (Snapshot from 07/14/2022), and Passanten Verlag (pv) (Snapshot from 07/14/2022), Kindermann Verlag (kv) (Snapshot from 07/14/2022) and Märchen in Leichter Sprache (mils) (Snapshot from 07/14/2022)

3.3 Training Setup

We use [Rios et al., 2021]'s results as the basis of the parameters of our experiments. However, our data deviate strongly from them. Both their and our data are document-aligned and are available in German. However, our domain differs, as well as the length of the texts. In addition, we do not have a vocabulary, and we have significantly fewer data points. Therefore, we deliberately checked all parameters and changed them so that they should produce optimal results.

Gradient Accumulation In many cases, it is necessary that model updates are based on a certain batch size. For example, with the Cifar-100 dataset, a batch size of 1 makes little sense because too few different labels are covered in this batch. However, the batch size depends on the GPU memory. In order to use larger batch sizes despite non-existing GPU capacities, Gradient Accumulation can be used, for example [Rotenberg, 2020].

Gradient accumulation can be described as creating a large batch and splitting it into smaller mini batches. These mini batches are run sequentially on the model, but without updating the model. In the final step, the gradients of the mini batches are accumulated to simulate the gradient of a large batch, then the model is updated with this accumulated gradient. We set the Gradient Accumulation parameter `accumulate_grad_batches` in our Trainer to 1, that means we practically turned Gradient Accumulation off. Gradient Accumulation is particularly useful in scenarios, with a large amount of data and relatively small GPU memory. Since our dataset is rather small, Gradient Accumulation would probably have no positive effect on the performance and could even worsen the performance.

Attention Mode Attention mode is a parameter that is part of the Longformer attention. There are two options possible: sliding chunks and sliding chunks without overlap.

Algorithm 1: GNATS Pre-Processing

ebDocs ← All reading samples (Leseproben) pdf texts from einfachebuecher.de that are listed as Klassiker (Classics)
kvDocs ← All reading samples (Leseproben) pdf texts from Kindermann Verlag that are listed as Weltliteratur für Kinder (World Literary Classics for Kids)
pvDocs ← All reading samples (Leseproben) pdf texts from Passantenverlag
milsDocs ← All HTML texts from NRDS's Märchen in Leichter Sprache
for $\underline{subdataset \in \{ebDocs, kvDocs, pvDocs\}}$ **do**
    Discard all fairy tales
    Discard all plays
    Only consider books with a Standard Language Version available at our sources
    Parse the Reading Sample with PyPDF2
    Remove page numbers from Reading Sample
    Truncate Standard Language Version to the Length of the reading sample
    Align truncated Standard Language Version and Reading Sample
    Define a title
end
for $\underline{subdataset \in \{milsDocs\}}$ **do**
    Remove the intro: "Es war einmal: So fangen Märchen an. Ein Märchen ist eine sehr alte Geschichte. Dieses Märchen heißt: <Headline>. Das Märchen geht so:"
    Remove the outro: "Das war das Märchen von <Headline>."
    Extract the title from the url
end
for $\underline{subdataset \in \{ebDocs, kvDocs, pvDocs, milsDocs\}}$ **do**
    emove syllable separator: "Rot·käppchen" is changed to "Rotkäppchen"
    Remove all escape characters, e.g., line breaks
end

We followed [Rios et al., 2021] and used sliding chunks. In theory, sliding chunks have a higher accuracy, but the accuracy of sliding chunks without overlap should be very close [Hailong Li, 2001]. So, the choice of this parameter should not have a very high impact on the models' performance. These chunks are visualized in Figure 2.8 b), c), d).

Optimizer We used the Adam Optimizer and tune the learning rate with PyTorch Lightning Learning Rate Finder, and we used [Rios, 2020] as a guideline for the boundaries of the search area. The minimal learning rate is $3e-20$, the maximal is $3e-1$, the number of trainings is 1000 (following the Discussion in PyTorch Lightning Issue 4846), and the mode is exponential (default value). The rate, that delivered the best results in each case, was subsequently used in fine-tuning and is listed in the table for each scenario. Figure 3.4 shows exemplary how the tested learning rates performed for the one-shot Fine-Tuning without Domain Adaptation Scenario.

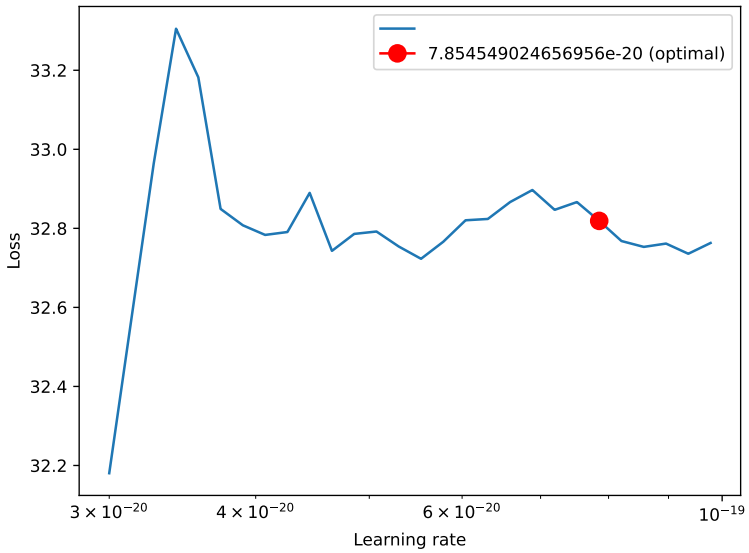


Figure 3.4: Results of the different learning rates on the loss. The red dot, is the chosen learning rate.

We note that the learning rate finder, we used, was introduced in 2020, is rather sparsely documented, and has not been changed very much since then. So results from alternative tuning libraries could differ. But a recent article showcased the usefulness of the learning rate optimizer in comparison to hand-picked learning rates [Kwaśniak, 2021].

Early Stopping Especially with little training data, overfitting can occur during the course of the training. Overfitting is indicated by the fact that the performance on the training data improves while it deteriorates on the validation data, in other words: The ability of the model to generalize deteriorates and the training data is "learned by heart". Ideally, the training process should run until the first occurrence of overfitting and then stop. Since it is not possible to determine this point in advance, we measure a validation parameter (`monitor`, Early Stopping Metric) and monitor how it changes. To be resilient to volatility, we do not stop the training exactly at a negative change, but wait for a certain number of epoch (`patience`). This process is called Early Stopping. Additionally, you could set a `min delta`, to determine when changes are relevant. We set `min delta`=0, so all changes are considered [Goodfellow et al., 2016, p.246-252].

3.3.1 Tools, Frameworks and Experimental Environment

The Server for our experiments, was gratefully provided to us by the Data Science Lab of the HAW Hamburg. Our setup is listed in Table 3.3:

Server Setup	**Framework**
CPU: Intel Core i9-10980XE CPU @ 3.00GHz × 36	Python 3.8
GPU: NVIDIA GeForce RTX 3090 (24576 MB)	Transformers [4]
RAM: 125,5 GiB	PyTorch 1.12.1
CUDA Toolkit 11	PyTorch Lightning 1.2
Operating System: Ubuntu 20.04.4 LTS	Spacy 3.4

Table 3.3: Tools, Frameworks and Hardware in our experimental Environment

3.4 Evaluation Measures

Evaluation plays a big role in this project. It serves three different purposes: 1) defining the early-stopping point while training; 2) compare the end results to references, and 3) determine the impact of the simplification on reflective passages. For each purpose, we will employ a different metric.

Furthermore, we described in section 2.5 multiple methods or approaches for simplification evaluation. We did not consider a vocabulary-based approach because they need a pre-defined vocabulary and do not offer the robustness we need for our purpose. We

employed two n-grams based approaches: **BLEU** and **ROUGE**. They are the most commonly used metrics for text generation and facilitate the comparability of our results. We do not consider any edits nor word alignment based approach. Simply because we already have three metrics that calculate the similarity between output and reference. Our main metric is **BERTscore** (see section 2.5.5). It determines the early stopping point in training, and we consider it the main indicator of text similarity, because it correlates the most with human judges and is resilient to paraphrasing (see Section 2.5.5). We deliberately chose not to measure the factuality of our models, since deleting information is allowed in the text simplification process and even become a necessity in longer texts. When we started the first trials of our experiments, one problem was that some sentences, within the generated text, were repeated very frequently. To make this effect visible during the experiments, we additionally measure the **entropy** within the generated texts using the measure described in section 2.5.7.

3.4.1 BERTscore

BERTscore is currently the recommend way of comparing Text Simplification candidates and references. As discussed in section 2.5 it is a soft metric, and it's F_{BERT} works relatively stable. We examined the overview and evaluation of the available pretrained models for bertscore: BERTScore Default Layer Performance on WMT16 [5] filtered it by Max Length $>$ 1022, multilingual (especially German) support, ordered them by their rank and checked their compatibility with the transformers version required for longmbart, e.g., `Microsoft/mdeberta-v3-base`, which fits our criteria and was ranked at 65th position needs a more modern transformer version [6]. So we used: `google/mt5-xl` and `google/mt5-large` are too large for our resources, so we used `google/mt5-base` (see Section 2.4.3). Following the recommendations in Section 2.5.5, we used F_{BERT} to determine the early stopping point.

3.4.2 Entropy

We used, as described in Section 2.5.7, the Bag-Of-Word Entropy and the Shortest Unique Prefix metric to measure the entropy of the output. In both cases, we used a German

[5] `https://docs.google.com/spreadsheets/d/1RKOVpselB98Nnh_EOC4A2BYn8_201tmPODpNWu4w7xI`, accessed 09/26/2022

[6] `https://github.com/Tiiiger/bert_score/issues/128`

Spacy pipeline [7] to tokenize the generated output. We considered all tokens including punctuation marks and lowercased them. So, the sentences "Ich" and "ich" should have the same entropy score and "Ich!" should have a higher score than the two other. For **the shortest unique prefix**-metric, we defined M as $M = Truncate(N * 0.5)$. This metric considers the text structure in the calculation, so offer this addition to the **bag-of-word** metric, which only considers the words and is not structure-aware.

3.5 Domain Adaptation

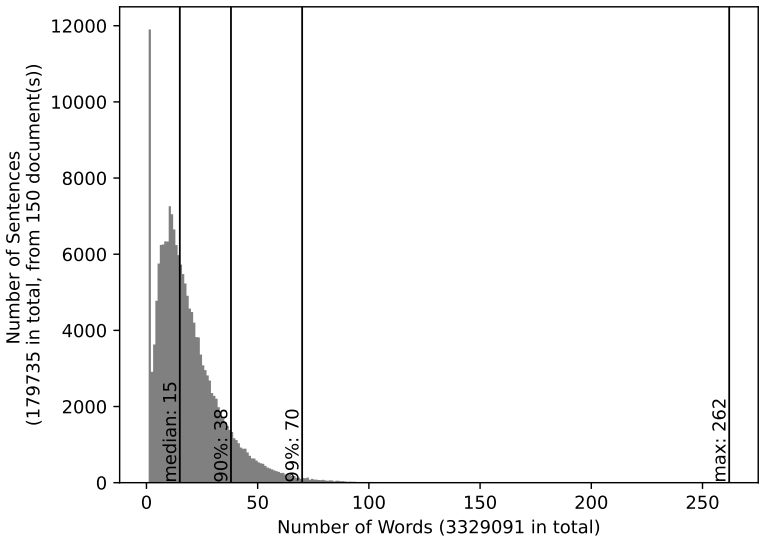


Figure 3.5: Number of words per sentence in the Textgrid Domain Adaptation dataset

Domain adaptation methods are often used to tackle the low-resource issue and to help models quickly adapt to target domain tasks. Despite their practicality, no studies have been done on Domain Adaptation for text simplification and very few studies have used domain adaptation methods on the related abstractive summarization task in a low-resource scenario. [Yu et al., 2021] is a recent work that addresses the abstractive summarization research gap. They added a second phase of pre-training on a BART model [Lewis et al., 2019] (see section 2.4.1) and systematically investigate in this area under three settings: 1) source domain pre-training (**SDPT**) based on a labeled source domain data; 2) domain-adaptive pre-training (**DAPT**) based on a substantial amount of unlabeled domain-related data; and 3) task-adaptive pre-training (**TAPT**) based on

[7] `https://spacy.io/models/de#de_core_news_sm`

unlabeled small-scale task-related data. The second phase of pre-training could cause the catastrophic forgetting in the pre-trained model. Their experimental results showed that both SDPT and TAPT can generally improve the overall performance, while the effectiveness of DAPT is correlated to the similarity between the pre-training data and the target domain task data. Furthermore, SDPT even outperforms DAPT and TAPT in terms of the averaged ROUGE-1 score, and adding RecAdam into the second phase of pre-training can generally further boost the adaptation performance for SDPT and TAPT.

After we created the longmbart-model we started the domain adaptation process. We downloaded all documents from TextGrid in the category "prose" and randomly sampled 60 documents. In a next step, we used the German (`decore_news_sm`) spacy pipeline to split the documents into sentences, shuffled them and masked 15% of the words in the sentences. We used these masked and unmasked sentence-pairs for a single epoch training of the model. Both sides of the pair are tagged as 'plain' German with the `de_DE` tag. We aimed to enrich the vocabulary with previously unseen words and adapt the existing embeddings to the narrative text domain and the historical environment of the texts.

We set the learning rate to $3e-10$, which is the half of the later-used learning rate for fine-tuning. The attention window is already set to 512, despite this, our max input and output length is set to 70. This number is the 99th percentile, that means only 1% of the sentences is right-side truncated (see Figure 3.5). The maximum batch size we could use with our setup was 8. See whether more domain adaptation data increases the results we created multiple buckets: one with 50 and one with 100 documents. After the training, we evaluated this model with the same test data set that we use later for fine-tuning and compared it to the not domain adapted model. The test results are listed in Table 4.1.

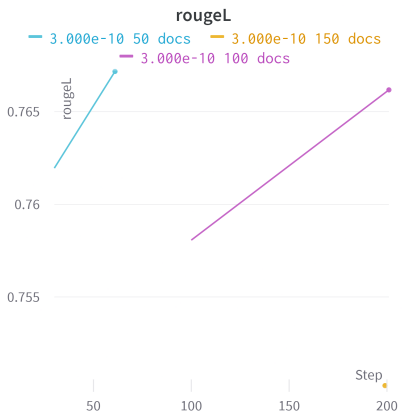


Figure 3.6: RougeL during 1 Epoch of Domain Adaptation

4 Results

In the following, we present and briefly discuss the results in section 4.1 and display some output of our best-performing model in section 4.2

4.1 Comparison

We ran several experiments to answer our research questions. As Table 4.1 indicates, Fine-Tuning and Domain Adaptation effected the model's performance negatively. Furthermore, BertScore seems to be not working properly. In Section 5 we analyze reasons and possible solutions for this behavior.

We analyze the model's performance via two kinds of metrics: similarity-based (BERTscore, BLEU and ROUGE) and entropy-based (SUP and BOW). Table 4.1 shows that the model without fine-tuning and domain adaptation performs the best both in terms of entropy and similarity. A single epoch of fine-tuning seems not to affect the models' performance, but fine-tuning it for 11 epochs worsens it drastically. Similarly, domain adaption without and with 1 epoch of fine-tuning drops below all non-domain-adapted models. Both domain adaptation set-ups (50 and 100 documents) perform the same, so the number of domain adaptation documents seems to have no effect on the performance. Interestingly, with more fine-tuning (11 epochs) the SUP entropy is improved, while the BERTscore-similarity further drops.

The model without domain adaptation and without fine-tuning performed the best and the more we trained the model, the more frequently individual text elements are repeated—first individual clauses, then words, and in the end only characters. These are results that no longer represent meaningful texts, let alone a high-quality text simplification.

Domain Adaptation Docs	F-BERT	RougeL	BLEU	SUP	BOW	Fine-Tune Epochs ♠	learning rate
-	**0.682**	**0.127**	**1.43**	**1.000**	**6.685**	0	-
-	0.682	0.127	1.43	1.000	6.685	1	7.8e-20
-	0.318	0	0	340.000	0.003	11 (100;10)	8.1e-07
50	0.301	0	0	123.666	0.038	1	3e-10 ♣
50	0.301	0	0	123.666	0.038	0	-
100	0.301	0	0	123.666	0.038	0	-
100	0.301	0	0	123.666	0.038	1	3e-10 ♣
100	0.298	0	0	49.666	0.0441	11 (100;10)	3e-10 ♣

Table 4.1: All models are tested on the GNATS test set and Beam Size = 6.
♠ Best epochs with max epochs and patience, if used, in brackets. We used early stopping as described earlier.
♣ The lr auto was unable to find an optimal learning rate; so we used a predefined value.

4.2 Generated Output

In this section, the generated output of our best performing model is written out. We added double-spacing, line numbers and the identifier as the headline. Besides this, we did not change the text in any way. Each test-text has its paragraph and analyzed in section 4.3, where the sources of the Standard Language and Simple Language Version are listed.

Des Teufels rußiger Bruder (mils-bruder)

abgedankter Soldat hatte nichts zu leben und wußte sich nicht mehr zu helfen. Da ging er hinaus in den Wald, und als er ein Weilchen gegangen war, begegnete ihm ein kleines Männchen, das war aber der Teufel. Das Männchen sagte zu ihm ›was fehlt dir? du siehst ja so trübselig aus.‹ Da sprach der Soldat ›ich habe Hunger, aber kein Geld.‹ Der Teufel sagte ›willst du dich bei mir vermieten und mein Knecht sein, so sollst du für dein Lebtag genug haben; sieben Jahre sollst du mir dienen, hernach bist du wieder frei. Aber eins sag ich dir, du darfst dich nicht waschen, nicht kämmen, nicht schnippen, keine Nägel und Haare abschneiden und kein Wasser aus den Augen wischen.‹ Der Soldat sprach ›frisch

dran, wenns nicht anders sein kann,‹ und ging mit dem Männchen fort, das führte ihn geradewegs in die Hölle hinein. Dann sagte es ihm, was er zu tun hätte: er müßte das Feuer schüren unter den Kesseln, wo die Höllenbraten drinsäßen, das Haus rein halten, den Kehrdreck hinter die Türe tragen und überall auf Ordnung sehen: aber guckte er ein einziges Mal in die Kessel hinein, so würde es ihm schlimm ergehen. Der Soldat sprach ›es ist gut, ich wills schon besorgen.‹ Da ging nun der alte Teufel wieder hinaus auf seine Wanderung, und der Soldat trat seinen Dienst an, legte Feuer zu, kehrte und trug den Kehrdreck hinter die Türe, alles, wie es befohlen war. Wie der alte Teufel wiederkam, sah er nach, ob alles geschehen war, zeigte sich zufrieden und ging zum zweitenmal fort. Der Soldat schaute sich nun einmal recht um, da standen die Kessel rings herum in der Hölle, und war ein gewaltiges Feuer darunter, und es kochte und brutzelte darin. Er hätte für sein Leben gerne hineingeschaut, wenn es ihm der Teufel nicht so streng verboten hätte: endlich konnte er sich nicht mehr anhalten, hob vom ersten Kessel ein klein bißchen den Deckel auf und guckte hinein. Da sah er seinen ehemaligen Unteroffizier darin sitzen: ›aha, Vogel,‹ sprach er, ›treff ich dich hier? du hast mich gehabt, jetzt hab ich dich,‹ ließ geschwind den Deckel fallen, schürte das Feuer und legte noch frisch zu.

Der Graf von Monte Christo (EB-christo)

28. Februar 1815 gab die Hafenwache von Notre-Dame das Signal vom Heransegeln des Dreimasters 'Pharaon', der von Smyrna, Triest und Neapel kam. Ein Küstenpilot steuerte alsogleich aus dem Hafen und erreichte das Fahrzeug zwischen dem Kap Morgion und der Insel Rion. Auch hatte sich, wie sonst immer, die Plattform des Kastells Saint-Jean mit Neugierigen gefüllt; denn in Marseille ist die Landung eines Schiffes stets von großer Wichtigkeit, zumal wenn es einem Reeder dieser Stadt gehört.

Der Sandmann (PV-sandmann)

Ofel an LotharGewiß seid Ihr alle voll Unruhe, daß ich so lange - lange nicht geschrieben. Mutter zürnt wohl, und Clara mag glauben, ich lebe hier in Saus und Braus und vergesse mein holdes Engelsbild, so tief mir in Herz und Sinn eingeprägt, ganz und gar. - Dem ist aber nicht so; täglich und stündlich gedenke ich Eurer aller und in süßen Träumen geht meines holden Clärchens freundliche Gestalt vorüber und lächelt mich mit ihren hellen Augen so anmutig an, wie sie wohl pflegte, wenn ich zu Euch hineintrat. - Ach wie vermochte ich denn Euch zu schreiben, in der zerrissenen Stimmung des Geistes, die mir bisher alle Gedanken verstörte! - Etwas Entsetzliches ist in mein Leben getreten! - Dunkle Ahnungen eines gräßlichen mir drohenden Geschicks breiten sich wie schwarze Wolkenschatten über mich aus, undurchdringlich jedem freundlichen Sonnenstrahl. - Nun soll ich Dir sagen, was mir widerfuhr. Ich muß es, das sehe ich ein, aber nur es denkend, lacht es wie toll aus mir heraus. - Ach mein herzlieber Lothar! wie fange ich es denn an, Dich nur einigermaßen empfinden zu lassen, daß das, was mir vor einigen Tagen geschah, denn wirklich mein Leben so feindlich zerstören konnte! Wärst Du nur hier, so könntest Du selbst schauen; aber jetzt hältst Du mich gewiß für einen aberwitzigen Geisterseher. - Kurz und gut, das Entsetzliche, was mir geschah, dessen tödlichen Eindruck zu vermeiden ich mich vergebens bemühe, besteht in nichts anderm, als daß vor einigen Tagen, nämlich am 30. Oktober mittags um 12 Uhr, ein Wetterglashändler in meine Stube trat und mir seine Ware anbot. Ich kaufte nichts und drohte, ihn die Treppe herabzuwerfen, worauf er aber von selbst fortging.Du ahnest, daß nur ganz eigne, tief in mein Leben eingreifende Beziehungen diesem Vorfall Bedeutung geben können, ja, daß wohl die Person jenes unglückseligen Krämers gar feindlich auf mich wirken muß. So ist es in der Tat. Mit aller Kraft fasse ich mich zusammen, um ruhig und geduldig Dir aus meiner frühern Jugendzeit so viel zu erzählen, daß Deinem regen Sinn alles klar und deutlich in leuchtenden Bildern aufgehen wird. Indem ich anfangen will, höre ich Dich lachen und Clara sagen: 'Das sind

ja rechte Kindereien!' - Lacht, ich bitte Euch, lacht mich recht herzlich aus! - ich bitt Euch sehr! - Aber Gott im Himmel! die Haare sträuben sich mir und es ist, als flehe ich Euch an, mich auszulachen, in wahnsinniger Verzweiflung, wie Franz Moor den Daniel. So ist es in der Tat. Mit aller Kraft fasse ich Euch aus meiner frühern Jugendzeit so viel zu erzählen, daß Deinem regen Sinn alles klar und deutlich in leuchtenden Bildern aufgehen wird. Indem ich anfangen will, höre ich Dich lachen und Clara sagen: 'Das sind ja rechte Kindereien!' - Lacht, ich bitte Euch, lacht mich recht herzlich aus! - es ist, als flehe ich Euch an, mich auszulachen, in wahnsinniger Verzweiflung, wie Franz Moor den Daniel. So ist es in der Tat. So ist es in der Tat. Mit aller Kraft fasse ich Euch aus meiner frühern Jugendzeit so viel zu erzählen, daß Deinem unglückseligen Krämers gar feindlich auf mich wirken muß, ja, daß wohl die Person jenes unglückseligen Krämers gar feindlich auf mich wirken muß, ja, daß wohl die Person jenes unglückseligen Krämers gar feindlich auf mich wirken muß. So ist in der Tat. So ist es in der Tat. - Nun fort zur Sache!Außer dem Mit aller dem Mittagsessen, das alter Sitte gemäß schon um sieben Uhr aufgetragen wurde, das alter Sitte gemäß schon um sieben Uhr aufgetragen wurde. Er mochte mit seinem Dienst. Er mochte mit seinem Dienst viel beschäftigt sein. Nach dem Abendessen. Er mochte mit seinem Dienst viel beschäftigt sein. Nach dem Abendessen, das alter Sitte gemäß, das alter Sitte gemäß, das alter Sitte gemäß, das alter Sitte gemäß von uns um sieben Uhr aufgetragen. Nach dem Abendessen, daß er aber von selbst fortging, daß er aber von selbst fortging.

4.3 Discussion

We compared all three generated output sequences to the Standard Language Version and the Simple Language Version. The following subsections give a detailed discussion of each of the texts. In summary, we found, that the model:

1. copies the input text to a very high degree without any modifications

2. in cases where the model discarded parts of the inputs, it did not recognize the importance of the sequence, such as spelled-out antecedents for pronouns.

3. truncates rather randomly and without any semantic reason.

Der Sandmann (PV-sandmann)

We manually compared the Output of Der Sandmann (4.2) with the Standard Language Version and the Simple Language Version of it. We used the Gutenberg.org Version [1] and the Passanten Verlag Reading Sample [2]. Beginning with "Nathanael an Lothar" until "Clara an Nathanael", which matches the extent of the reading sample. From line 31 to 64, the complete text is equivalent to the Standard Language Input. After "Franz Moor den Daniel." the model inserted the passage "So ist es in der Tat." ("That is indeed how it is.") and repeats it two times. This passage already occurred in line 59. Then the passage "Mit aller Kraft fasse ich Euch aus meiner frühern Jugendzeit so viel zu erzählen, daß Deinem" ("With all my strength, I will tell you so much from my early youth, that your") follows, which also previously occurred in line 59. Followed by three times "unglückseligen Krämers gar feindlich auf mich wirken muß," ("I can't help but think that the unfortunate grocer must have a hostile effect on me,) in line 65, again a repetition from line 52. Then, again: "So ist es in der Tat.". Repetition in general manifest in bad results in the entropy metric.

Beginning in line 69, the model discarded information from the input text. "Außer dem Mit" ("Besides the Mit") is followed by "aller dem Mittagsessen, das alter Sitte gemäß schon um sieben Uhr aufgetragen wurde," ("of all the lunch, which was served according to old custom already at seven o'clock,") in the simplification and "tagsessen sahen wir, ich und mein Geschwister, tagüber den Vater wenig. Er mochte mit seinem Dienst viel beschäftigt sein. Nach dem Abendessen, das alter Sitte gemäß schon um sieben Uhr aufgetragen wurde," ("Apart from lunch, we, me and my siblings, saw little of our father during the day. He might have been busy with his duties. After supper, which was served according to the old custom at seven o'clock, was served,") in the original. Interestingly, with this reduction, the model output deviates from the truth. In the standard language version, supper is served at seven o'clock and in the simplification,

[1] https://www.gutenberg.org/ebooks/6341 which is congruent with the MONACO-version https://gitlab.gwdg.de/mona/korpus-public/-/blob/master/Hoffmann__Der_Sandmann/Hoffmann__Der_Sandmann.txt

[2] https://www.passanten-verlag.de/Leseproben/Sandmann_Leseprobe.pdf

lunch is served at seven o'clock. The reference simple language version, from Passanten Verlag, completely discarded the facts about dinner and supper. Boiling this passage down to a brief introduction of the father and mentioning, that he was busy with his work and that he told fascinating stories to his kids.

This fact about the time of supper and dinner, is unimportant for the core story line, but still shows weaknesses of the model. Another aspect of the reduction in the model output is the fact, that the model does not mention the father. This is the first introduction of this character in the story. So the model discarded an important character from this text passage. Then, "Er mochte mit seinem Dienst.", "Nach dem Abendessen" and "das alter Sitte gemäß" are repeated a few times. So, the model did not fully discard the father, "Er mochte mit seinem Dienst." ("he liked with his work") still refers to the father by the pronoun "Er" ("he"), despite the fact that the character was never introduced or referred to via his antecedent (the noun that the pronoun corresponds to). For a reader, that has only access to the model's output, it is impossible to understand who "Er" ("he") is. A clean or complete removal of a character would show some simplification capability, even if it was an important character. In this case, it was an incomplete removal of an arguably important character.

In lines 72-74 the model constructs another new sentence: "Nach dem Abendessen, das alter Sitte gemäß, das alter Sitte gemäß, das alter Sitte gemäß, das alter Sitte gemäß von uns um sieben Uhr aufgetragen. " ("After supper, the old custom, the old custom, the old custom, the old custom was served by us at seven o'clock."). Which is another new fact by the model. This time it does not contradict the original, but stills adds the information, that supper was served by "uns" ("we"), which would be interpreted here as the narrator and his siblings.

Most of the repeated sentences do not contain information that is important to follow the story. In this respect, there is actually no need to transfer them into the simplification, let alone repeat them. Especially sentences like "So ist es in der Tat" ("So it is indeed"), are only a linguistic emphasis and arguably add linguistic complexity without additional content.

The only sentence containing arguably important information that was repeated was "unglückseligen Krämers gar feindlich auf mich wirken muß,". For this sentence concludes, the narrator's first account of the meeting with the barometer seller, Copolla. It also describes the narrator's fear of Coppolla, which has so far prevented the narrator

from writing: "Ach wie vermochte ich denn Euch zu schreiben, in der zerrissenen Stimmung des Geistes, die mir bisher alle Gedanken verstörte! - Etwas Entsetzliches ist in mein Leben getreten!" (line 38) ("Oh, how could I write to you, in the torn mood of the mind? you, in the torn mood of the spirit, which so far has disturbed all my disturbed all my thoughts! - Something terrible has come into my life! entered my life!"), this "Entsetzliche" ("horrible") is the barometer seller Copolla or the grocer.

If we assume that repeated sentences are perceived as important by the model [3], the model correctly recognized an importance only in this case. The model does only partially demonstrate a good perception of named entities. While Copolla, the barometer seller, is emphasized, the first mention of the father was deleted in the simplification process, as we have discussed above.

Der Graf von Monte Christo (EB-christo) & Des Teufels rußiger Bruder (MILS-bruder)

For MILS-bruder), we used the Projekt-Gutenberg.org Standard Language Version [4] and the Märchen in Leichter Sprache (MILS) Simple Language Version from NDR [5]. And for EB-christo, we used the Projekt-Gutenberg.org Version [6] and the Märchen in Leichter Sprache (MILS) from einfachebuecher.de [7].

In both cases, the model output is a truncated version of the input text. This output texts are short and stayed far below the maximum output length. For MILS-bruder), there are 791 words (69%) and for EB-christo) there are 5,524 words (99%) of the input text, which are not represented at all in the output text.

[3] Profound hypothesis on the causes of repetition are sparse. We base our conjecture on the results of Xu et al. [2022]. Assuming a correlation between initial probability and repetition rate. If a text fragment occurs more often in the course of the document, it is more likely to be repeated. Therefore, we would say that "unglückseligen Krämers" has a high initial probability for the model. In this respect, it is information that should be repeated more frequently in the text and can therefore be considered as important.

[4] https://www.projekt-gutenberg.org/grimm/maerchen/chap143.html

[5] https://www.ndr.de/fernsehen/barrierefreie_angebote/leichte_sprache/Des-Teufels-russiger-Bruder,bruderleichtesprache100.html

[6] https://www.projekt-gutenberg.org/dumasalt/montechr/montechr.html from the beginnging until "Bei Dantes überstürzten sich Gedanken und Vermutungen."

[7] https://einfachebuecher.de/WebRoot/Store21/Shops/95de2368-3ee3-4c50-b83e-c53e52d597ae/MediaGallery/Leseproben/Klassiker/Der_Graf_von_Monte_Christo_Leseprobe.pdf

5 Conclusion & Outlook

In this thesis, we aimed to investigate methods to efficiently and practically extend the context of transformer-based models for generating automatic text simplifications on a document-level. Furthermore, we investigated the usage of fine-tuning and domain adaptation. To do that, we evaluated current metrics, and applied a selection in our experiments. We finalize this paper by concluding our results 5.1 and outline ideas for future research 5.2.

5.1 Conclusion

We analyze the model's performance via two kinds of metrics: similarity- and entropy-oriented. When looking at Table 4.1 entropy (SUP, BOW) drastically worsens in the course of domain adaptation and fine-tuning, so the more gets more repetitive. Equally, the model's output deviates more and more from the target (F-BERT, RougeL, BLUE). Furthermore, we did a qualititve analysis on output from the model in Section 4. Which also show repetition and some phenomenons, that can explain the similarity performance. We did not manage to definitively conclude on our research questions or reasons why the performance decreased, but we can conclude that:

1. Fine-Tuning does not necessarily improve the task-specific performance of a pre-trained text generation model.
2. Domain Adaptation does not necessarily improve the domain-specific performance of a pre-trained text generation model.
3. BERTscore is highly dependent on the underlying model. We strongly recommend running at least one n-gram based metric parallel to double-check the results. In our case, BERTscore did not reliably correlate with the other similarity metrics. So, it's result alone can not serve as a trustworthy indicator of the model's performance.

4. Entropy metrics provide additional guidance on the quality of generated text simplification. They can show very well to what degree the presented text generation model is affected by the repetition problem. Furthermore, they are very inexpensive to use.

The model without Domain Adaptation and without Fine-tuning performed the best. As described in Section 4.3, the outputs are mostly copied from the original text. The more we trained the model, the more frequently individual text elements were repeated. First, individual partial sentences, then words, and in the end only characters. These are results that no longer represent meaningful texts. Let alone a high-quality text simplification.

Catastrophic Forgetting The model in our experiments was previously trained on inter-language translation (from one language to another) and we further fine-tuned it on intra-language translation (from one version of a language to another version of the same language). Similarly, Domain Adaptation can also be seen as an intra-language task, that differs from the original mBart task (see Section 2.4.2. The model's general text generation capability dropped after fine-tuning and domain adaptation. One phenomenon, that is described in scientific literature and matches this characterization, is *catastrophic forgetting*. We argue that this phenomenon is the main reason for this negative trend. Catastrophic forgetting can occur in all scenarios, where machine learning models are trained on a sequence of tasks and the accuracy on earlier tasks drops significantly. The catastrophic forgetting problem manifests in many sub-domains of machine learning tasks. Ramasesh et al. [2021] demonstrated that forgetting is concentrated at the higher models layers. In their setup, these layers changed significantly and erased earlier task subspaces through sequential training of multiple tasks. All the mitigation methods they investigated stabilize higher layer representations, but varied on whether they enforce more feature reuse, or store tasks in orthogonal subspaces. They used an image dataset, but their general finding, that catastrophic forgetting mostly happens in the higher layers and that it should be mitigated there, can probably be applied very well to our work and NLP-problems in general.

There are several other possible reasons for this behavior and opportunities to improve the models' performance. In the following subsection we give an outlook, on possible ways of adjustment:

5.2 Outlook

Our work contributed to the field of automatic German Text Simplifications. This field is understudied, and future works, that want to build on top of our findings and other previous works, could research the following areas:

Noising Strategies We only used one *Token Masking* strategy of the five noising strategies from BART (see section 2.4.1). We used it, because it is the most common strategy. Theoretically, the domain adaptation outcome could be improved by using all five or systematically investigate which works best for domain adapting mBart.

Multi- vs. Monolingual pre-trained Text Generation Models On the one hand, pre-training a model for a specific language, has in theory no capacity dilution (i.e., the complete model capacity is being used for the language of interest). On the other hand, there might be some benefit of using the additional pre-training data from multiple (related) languages. Doddapaneni et al. [2021] conclude that it is not clear whether Multi- vs. Monolingual pre-trained Text Generation Models are always better than monolingual models or vice versa. So it would be beneficial, if a case study compared a German vs. Multilingual pre-trained Text Generation Model for this specific task. In this work, we only used a Multilingual model, so a similar German model could outperform our results.

Controllability & Learning Strategies [Erdem et al., 2022, 1165-1168] names a few resources, where adding metadata such as named entities or part-of-speech to the input can be used as an advanced learning strategy to improve results and offer more controllability over the output. We did not add any metadata and showed in Section 4.3, that our model is not properly able to recognize named entities, so inserting corresponding metadata could further improve the model's performance.

Catastrophic Forgetting Yu et al. [2021] investigated catastrophic forgetting in automatic text simplification. They speculated that their second phase of pre-training resulted in some form of catastrophic forgetting for the pre-trained model, which could have hurt the adaptation performance. They recommend to use, RecAdam [Chen et al., 2020], a specific optimizer to mitigate the problem. So, it could be very fruitful to use

this optimizer in the future. We did not use it because we lacked the time, but it is applicable to our existing implementation.

Unify Designations Designations of people are interchangeably used in standard language. Unifying them, e.g. via the "Gleiche Wörter" (equal words) section in the corresponding Hurraki (A German lexicon for Easy Language) article, could help. A good example is the father in Der Sandmann, he is mostly addressed as "Vater" (father) but also as "Papa" (dad) by his children and as "Herr" (Sir) as in "Herr des Hauses" (Master) by his house staff. All these words mean the same and are referring to the same person. So, it could by unifying same to "Vater".

Repetition Problem Su and Collier [2022] suggest a categorization of current approaches for natural language generation with language models into two classes: 1) **maximization-based** methods, such as greedy search and beam search; and 2) **stochastic** methods, such as top-k sampling [Fan et al., 2018]. They further describe, that maximization-based approaches tend to produce text that contains undesirable repetitions and stochastic methods tend to produce text that is semantically inconsistent with the given prefix.

We used Beam Search in our approach and experienced a significant increase of repetition during the training. Xu et al. [2022] divided approaches for **mitigating repetition** into 1) training-based [Welleck et al., 2020, Lin et al., 2021, Xu et al., 2022] and decoding-based [See et al., 2017, Fan et al., 2018, Holtzman et al., 2020] approaches. Recently, two new decoding approaches: Nucleus [Holtzman et al., 2020] and Contrastive Search [Su and Collier, 2022] have shown promising results in terms of reducing repetition and improving the overall quality of generated text.

Future work could apply these newer decoding methods to the task of document-level text simplification. Although, there is an increasing number of mitigating techniques, the causes of the repetition problem are still under-investigated. Fu et al. [2021] analyzed the problem by assuming the language models can be approximated to first-order Markov models. In contrast, Holtzman et al. [2020] indicate that the repetition probability has complex relationships with a long-range context and conclude that language models may not be simplified as first-order Markov models. Recently, Xu et al. [2022] investigated

quantitatively why maximization-based decoding models prefer consecutive sentence-level repetitions. They found the tendency to repeat previous sentences and the self-reinforcement effect are causes for repetition.

A Appendix

A.1 Additional Resources

	Full Title	Reason for rejection
kv	Faust	Is a play
kv	Macbeth	Is a play
kv	Der Widerspenstigen Zähmung	Is a play
kv	Leonce und Lena	Is a play
kv	Der Kaufmann von Venedig	Is a play
kv	Der zerbrochene Krug	Is a play
kv	Hamlet	Is a play
kv	Die Räuber	Is a play
kv	Viel Lärm um Nichts	Is a play
kv	Das Käthchen von Heilbronn	Is a play
kv	Ein Sommernachtstraum	Is a play
kv	Wilhelm Tell	Is a play
kv	Nathan der Weise	Is a play
kv	Götz von Berlichingen	Is a play
kv	Romeo und Julia	Is a play
kv	Der Sturm	Is a play
kv	Kleider machen Leute	reading sample too short
eb	Im Westen nichts Neues	No Original found
eb	Romeo und Julia	Is a play
eb	Sherlock Holmes. Der Mann mit der Narbe	No Original found
eb	Robinson Crusoe	Unable to obtain any text from the simple version; seems protected. Whether the Python library nor PDF24 OCR could anything extract but nonsense. (Original)
eb	Tristan und Isolde	Is a play
eb	Dracula	No Original found
eb	Eine Weihnachtsgeschichte	No Original found
eb	Das Phantom der Oper	No Original found
eb	Frankenstein	Same problem as Robinson Crusoe (Original)
eb	Erzählungen von Heinrich Böll	No Original found

eb	Momo	No Original found
eb	Sherlock Holmes. Ein Skandal in Böhmen	Original available but the reading sample is too short
eb	Der kleine Prinz	No Original found
eb	Heimatlos	No Original found
eb	Till Eulenspiegel	Simple Version is too far away from the Original. Simple Version seems more about the person Till Eulenspiegel than a specific Original document.
eb	Märchen von Hans Christian Andersen	contains fairy tales
eb	7 Kilo in 3 Tagene	No Original found
eb	Die Magie des Spiels	No Original found
eb	Feiertage, Feste und Bräuche	No Original found
eb	Deutschstunde	No Original found
eb	Die Abenteuer von Baron Münchhausen	Original available, but the reading sample is too short
eb	Grimms Märchen	contains fairy tales
pv	Mit Kobolden tanzen	contains fairy tales
pv	Paradies Federn	No Original found
pv	Hölderlin leuchtet	No Original found
pv	Der Augsburger Kreidekreis	No Original found
pv	Die Frau in der Tür	No Original found
pv	Die wilden Schwäne / Der Tannenbaum	contains fairy tales
pv	Russische Märchen	contains fairy tales
pv	Moby Dick	already in eb corpus
pv	Er kam zu spät	No Original found
pv	Bärenzart	No Original found

Table A.1: kv - All documents that are NOT in our corpus from Kindermann Verlag (Snapshot from 07/20/2022) and the reason we did not add them to our corpus
eb - All documents that are NOT in our corpus from einfachebuehcher.de which are classified as "Klassiker" (Snapshot from 07/14/2022) and the reason we did not add them to our corpus.
pv - All documents that are NOT in our corpus from Passanten Verlag (Snapshot from 07/20/2022) and the reason we did not add them to our corpus.

Erklärung zur selbstständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original

Bibliography

Omri Abend and Ari Rappoport. Universal Conceptual Cognitive Annotation (UCCA). In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 228–238, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://aclanthology.org/P13-1023.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. EASSE: Easier Automatic Sentence Simplification Evaluation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations, pages 49–54, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-3009. URL https://aclanthology.org/D19-3009.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. The (Un)Suitability of Automatic Evaluation Metrics for Text Simplification. Computational Linguistics, 47 (4):861–889, December 2021. ISSN 0891-2017. doi: 10.1162/coli_a_00418. URL https://doi.org/10.1162/coli_a_00418.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges, July 2019. URL http://arxiv.org/abs/1907.05019. arXiv:1907.05019 [cs].

Dennis Aumiller and Michael Gertz. Klexikon: A German Dataset for Joint Summarization and Simplification. In Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 2693–2701, June 2022.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473 [cs, stat], May 2016. URL http://arxiv.org/abs/1409.0473. arXiv: 1409.0473.

Sriram Balasubramanian, Naman Jain, Gaurav Jindal, Abhijeet Awasthi, and Sunita Sarawagi. What's in a Name? Are BERT Named Entity Representations just as Good for any other Name? In Proceedings of the 5th Workshop on Representation Learning for NLP, pages 205–214, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.repl4nlp-1.24. URL https://aclanthology.org/2020.repl4nlp-1.24.

Alessia Battisti, Dominik Pfütze, Andreas Säuberli, Marek Kostrzewa, and Sarah Ebling. A Corpus for Automatic Readability Assessment and Text Simplification of German. In Proceedings of the 12th Language Resources and Evaluation Conference, pages 3302–3311, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.404.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The Long-Document Transformer, December 2020. URL http://arxiv.org/abs/2004.05150. arXiv: 2004.05150.

Philippe Blache. Constraints, Linguistic Theories, and Natural Language Processing. volume 1835, pages 221–232, June 2000. ISBN 978-3-540-67605-8. doi: 10.1007/3-540-45154-4_21.

Ursula Bredel and Christiane Maaß. Leichte Sprache theoretische Grundlagen, Orientierung für die Praxis. Sprache im Blick. Dudenverlag, 2016. ISBN 978-3-411-75616-2.

C.S. Richard Chan, Charuta Pethe, and Steven Skiena. Natural language processing versus rule-based text analysis: Comparing BERT score and readability indices to predict crowdfunding outcomes. Journal of Business Venturing Insights, 16:e00276, November 2021. ISSN 23526734. doi: 10.1016/j.jbvi.2021.e00276. URL https://linkinghub.elsevier.com/retrieve/pii/S2352673421000548.

R. Chandrasekar, Christine Doran, and B. Srinivas. Motivations and Methods for Text Simplification. In COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics, pages 1041–1044. COLING, 1996. URL https://aclanthology.org/C96-2183.

Sanyuan Chen, Yutai Hou, Yiming Cui, Wanxiang Che, Ting Liu, and Xiangzhan Yu. Recall and Learn: Fine-tuning Deep Pretrained Language Models with Less Forgetting. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 7870–7881, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.634. URL https://aclanthology.org/2020.emnlp-main.634.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating Long Sequences with Sparse Transformers, April 2019. URL http://arxiv.org/abs/1904.10509. arXiv:1904.10509 [cs, stat].

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. arXiv:1409.1259 [cs, stat], October 2014. URL http://arxiv.org/abs/1409.1259. arXiv: 1409.1259.

Alexis Conneau and Guillaume Lample. Cross-lingual Language Model Pretraining. In Advances in Neural Information Processing Systems, volume 32, pages 7027–7037. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c04c19c2c2474dbf5f7ac4372c5b9af1-Abstract.html.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised Cross-lingual Representation Learning at Scale. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 8440–8451, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.747. URL https://aclanthology.org/2020.acl-main.747.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models beyond a Fixed-Length Context. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285.

Fred J. Damerau. A technique for computer detection and correction of spelling errors. Communications of the ACM, 7(3):171–176, March 1964. ISSN 0001-0782. doi: 10.1145/363958.363994. URL https://doi.org/10.1145/363958.363994.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML'17, pages 933–941, Sydney, NSW, Australia, August 2017. JMLR.org.

Michael Denkowski and Alon Lavie. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In Proceedings of the Sixth Workshop on Statistical Machine Translation, pages 85–91, Edinburgh, Scotland, July 2011. Association for Computational Linguistics. URL https://aclanthology.org/W11-2107.

Jacob Devlin. BERT Multilingual, 2018. URL https://github.com/google-research/bert/blob/master/multilingual.md. Library Catalog: github.com.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

DickMan64. [D] Why are transformers still being used?, June 2022. URL www.reddit.com/r/MachineLearning/comments/vo2br1/d_why_are_transformers_still_being_used/.

Sumanth Doddapaneni, Gowtham Ramesh, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. A Primer on Pretrained Multilingual Language Models, December 2021. URL http://arxiv.org/abs/2107.00676. arXiv:2107.00676 [cs].

Sarah Ebling, Alessia Battisti, Marek Kostrzewa, Dominik Pfütze, Annette Rios, Andreas Säuberli, and Nicolas Spring. Automatic Text Simplification for German. Frontiers in Communication, 7:706718, February 2022. ISSN 2297-900X. doi: 10.3389/fcomm.2022.706718. URL https://www.zora.uzh.ch/id/eprint/218829/. Publisher: Frontiers Research Foundation.

Sergey Edunov, Alexei Baevski, and Michael Auli. Pre-trained language model representations for language generation. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human

Language Technologies, Volume 1 (Long and Short Papers), pages 4052–4059, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1409. URL https://aclanthology.org/N19-1409.

Erkut Erdem, Menekse Kuyu, Semih Yagcioglu, Anette Frank, Letitia Parcalabescu, Barbara Plank, Andrii Babii, Oleksii Turuta, Aykut Erdem, Iacer Calixto, Elena Lloret, Elena-Simona Apostol, Ciprian-Octavian Truică, Branislava Šandrih, Sanda Martinčić-Ipšić, Gábor Berend, Albert Gatt, and Grăzina Korvel. Neural Natural Language Generation: A Survey on Multilinguality, Multimodality, Controllability and Learning. Journal of Artificial Intelligence Research, 73:1131–1207, April 2022. ISSN 1076-9757. doi: 10.1613/jair.1.12918. URL https://www.jair.org/index.php/jair/article/view/12918.

Allyson Ettinger. What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models. Transactions of the Association for Computational Linguistics, 8:34–48, 2020. doi: 10.1162/tacl_a_00298. URL https://aclanthology.org/2020.tacl-1.3. Place: Cambridge, MA Publisher: MIT Press.

Europarat, editor. Common European framework of reference for languages: learning, teaching, assessment ; companion volume. Council of Europe Publishing, Strasbourg, 2020. ISBN 978-92-871-8621-8.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical Neural Story Generation. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 889–898, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1082. URL https://aclanthology.org/P18-1082.

Dino Felluga. General Introduction to Narratology, January 2011. URL https://www.cla.purdue.edu/academic/english/theory/narratology/modules/introduction.html.

Geert Freyhoff, Gerhard Heß, Linda Kerr, Elizabeth Menzel, Bror Tronbacke, and Kathy Van Der Veken. Europäische Richtlinien für die Erstellung von leicht lesbaren Informationen für Menschen mit geistiger Behinderung für Autoren, Herausgeber, Informationsdienste, Übersetzer und andere interessierte Personen. Technical report, Europäische Vereinigung der ILSMH, June 1998. URL https://web-4-all.de/wp-content/uploads/2012/12/EURichtlinie_sag_es_einfach.pdf.

Zihao Fu, Wai Lam, Anthony Man-Cho So, and Bei Shi. A Theoretical Analysis of the Repetition Problem in Text Generation. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 12848–12856, Online, May 2021. AAAI. doi: 10.1609/aaai.v35i14.17520. URL https://ojs.aaai.org/index.php/AAAI/article/view/17520. Number: 14.

Alexander Gaskell, Dr Pedro Baiz, Lucia Specia, Hugo Barbaroux, and Dr Eric Topham. On the Summarization and Evaluation of Long Documents. page 98, September 2020.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts London, England, 2016. ISBN 978-0-262-03561-3. OCLC: 955778308.

Natalia Grabar and Horacio Saggion. Evaluation of Automatic Text Simplification: Where are we now, where should we go from here. In Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale, pages 453–463, Avignon, France, June 2022. ATALA. URL https://aclanthology.org/2022.jeptalnrecital-taln.47.

Gregory Grefenstette. Tokenization. Text, Speech and Language Technology, page 117, 1999. ISSN 1386-291X. URL https://www.academia.edu/375430/Tokenization.

Yinuo Guo, Chong Ruan, and Junfeng Hu. Meteor++: Incorporating Copy Knowledge into Machine Translation Evaluation. In Proceedings of the Third Conference on Machine Translation: Shared Task Papers, pages 740–745, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6454. URL https://aclanthology.org/W18-6454.

Hailong Li. Question about the implemented sparse attention · Issue #157 · allenai/longformer, April 2001. URL https://github.com/allenai/longformer/issues/157.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. Readability Classification for German using Lexical, Syntactic, and Morphological Features. In Proceedings of COLING 2012, pages 1063–1080, Mumbai, India, December 2012. The COLING 2012 Organizing Committee. URL https://aclanthology.org/C12-1065.

Michael Hanna and Ondřej Bojar. A Fine-Grained Analysis of BERTScore. In Proceedings of the Sixth Conference on Machine Translation, pages 507–517, Online, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.wmt-1.59.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with Disentangled Attention. In Proceedings of the International Conference on Learning Representations 2021, pages 1–21, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

Dan Hendrycks and Kevin Gimpel. Gaussian Error Linear Units (GELUs), July 2020. URL http://arxiv.org/abs/1606.08415. arXiv:1606.08415 [cs] version: 4.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. A Transition-Based Directed Acyclic Graph Parser for UCCA. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1127–1138, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1104. URL https://aclanthology.org/P17-1104.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. In International Conference on Learning Representations, pages 1–16, March 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Colby Horn, Cathryn Manduca, and David Kauchak. Learning a Lexical Simplifier Using Wikipedia. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 458–463, Baltimore, Maryland, June 2014. Association for Computational Linguistics. doi: 10.3115/v1/P14-2075. URL https://aclanthology.org/P14-2075.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving Pre-training by Representing and Predicting Spans. Transactions of the Association for Computational Linguistics, 8:64–77, 2020. doi: 10.1162/tacl_a_00300. URL https://aclanthology.org/2020.tacl-1.5. Place: Cambridge, MA Publisher: MIT Press.

JunhyunB. Does BART support more than 1024 tokens in inference of summarization task? · Issue #1685 · facebookresearch/fairseq, February 2020. URL https://github.com/facebookresearch/fairseq/issues/1685.

J. P. Kincaid, Jr. Fishburne, Rogers Robert P., Chissom Richard L., and Brad S. Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel:. Reports - Research ED108134, Defense Technical Information Center, February 1975.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The Efficient Transformer, February 2020. URL http://arxiv.org/abs/2001.04451. arXiv:2001.04451 [cs, stat].

David Klaper, S. Ebling, and Martin Volk. Building a German/Simple German Parallel Corpus for Automatic Text Simplification. In Klaper, David; Ebling, S; Volk, Martin (2013). Building a German/Simple German Parallel Corpus for Automatic Text Simplification. In: The Second Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR 2013), Sofia, Bulgaria, 8 August 2013., pages 11–19, Sofia, Bulgaria, August 2013. University of Zurich. doi: 10.5167/uzh-78610. URL https://www.zora.uzh.ch/id/eprint/78610/.

I Kontoyiannisy. The Complexity and Entropy of Literary Styles. Technical report, October 1997.

Matt J. Kusner, Yu Sun, Nicholas I. Kolkin, and Kilian Q. Weinberger. From word embeddings to document distances. In Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37, ICML'15, pages 957–966, Lille, France, July 2015. JMLR.org.

Chul Hyun Kwag. relu, gelu , swish, mish activation function comparison, May 2022. URL https://chadrick-kwag.net/relu-gelu-swish-mish-activation-function-comparison/.

Mateusz Kwaśniak. How to decide on learning rate, June 2021. URL https://towardsdatascience.com/how-to-decide-on-learning-rate-6b6996510c98.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, October 2019. URL https://arxiv.org/abs/1910.13461v1.

Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/W04-1013.

Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. AI Open, 3:111–132, January 2022. ISSN 2666-6510. doi: 10.1016/j.aiopen.2022.10.001. URL https://www.sciencedirect.com/science/article/pii/S2666651022000146.

Xiang Lin, Simeng Han, and Shafiq Joty. Straight to the Gradient: Learning to Use Novel Tokens for Neural Text Generation. In Proceedings of the 38th International Conference on Machine Learning, pages 6642–6653. PMLR, July 2021. URL https://proceedings.mlr.press/v139/lin21b.html. ISSN: 2640-3498.

Tal Linzen and Marco Baroni. Syntactic Structure from Deep Learning. Annual Review of Linguistics, 7(1):195–212, 2021. doi: 10.1146/annurev-linguistics-032020-051035. URL https://doi.org/10.1146/annurev-linguistics-032020-051035. _eprint: https://doi.org/10.1146/annurev-linguistics-032020-051035.

Peter J. Liu, Mohammad Saleh*, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating Wikipedia by Summarizing Long Sequences. In Proceedings of the International Conference on Learning Representations, page 18, February 2018.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs], July 2019. URL http://arxiv.org/abs/1907.11692. arXiv: 1907.11692.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual Denoising Pre-training for Neural Machine Translation. Transactions of the Association for Computational Linguistics, 8:726–742, 2020. doi: 10.1162/tacl_a_00343. URL https://aclanthology.org/2020.tacl-1.47. Place: Cambridge, MA Publisher: MIT Press.

Marella Magris and Dolores Ross. Barrierefreiheit auf Webseiten von Gebietskörperschaften: ein Vergleich zwischen Deutschland, Italien und den Niederlanden. 8:8–39, 2015. ISSN 1867-4844. URL http://www.trans-kom.eu/bd08nr01/trans-kom_08_01_02_Magris_Ross_Barrierefrei.20150717.pdf.

Benjamin Marie. BLEU: A Misunderstood Metric from Another Age, November 2022. URL https://towardsdatascience.com/bleu-a-misunderstood-metric-from-another-age-d434e18f1b37.

Louis Martin, Samuel Humeau, Pierre-Emmanuel Mazaré, Éric de La Clergerie, Antoine Bordes, and Benoît Sagot. Reference-less Quality Estimation of Text Simplification Systems. In Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA), pages 29–38, Tilburg, the Netherlands, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-7005. URL https://aclanthology.org/W18-7005.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. ∞-former: Infinite Memory Transformer, March 2022. URL http://arxiv.org/abs/2109.00301. arXiv:2109.00301 [cs].

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13, pages 3111–3119, Red Hook, NY, USA, December 2013. Curran Associates Inc.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A Generative Model for Raw Audio, September 2016. URL http://arxiv.org/abs/1609.03499. arXiv:1609.03499 [cs].

Gustavo Paetzold and Lucia Specia. Benchmarking Lexical Simplification Systems. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 3074–3080, Portorož, Slovenia, May 2016. European Language Resources Association (ELRA). URL https://aclanthology.org/L16-1491.

Gustavo Paetzold, Fernando Alva-Manchego, and Lucia Specia. MASSAlign: Alignment and Annotation of Comparable Documents. In Proceedings of the IJCNLP 2017, System Demonstrations, pages 1–4, Tapei, Taiwan, November 2017. Association for Computational Linguistics. URL https://aclanthology.org/I17-3001.

Sinno Jialin Pan and Qiang Yang. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, October 2010. ISSN 1041-4347.

doi: 10.1109/TKDE.2009.191. URL http://ieeexplore.ieee.org/document/5288526/.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pages 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image Transformer. In Proceedings of the 35th International Conference on Machine Learning, pages 4055–4064. PMLR, July 2018. URL https://proceedings.mlr.press/v80/parmar18a.html. ISSN: 2640-3498.

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

Thang Pham, Trung Bui, Long Mai, and Anh Nguyen. Out of Order: How important is the sequential order of words in a sentence in Natural Language Understanding tasks? In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 1145–1160, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.98. URL https://aclanthology.org/2021.findings-acl.98.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. page 12, 2018.

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. arXiv:1910.10683 [cs, stat], July 2020. URL http://arxiv.org/abs/1910.10683. arXiv: 1910.10683.

Vinay Venkatesh Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of Catastrophic Forgetting: Hidden Representations and Task Semantics. In Proceedings of Conference on Learning Representations, pages 1–31. International Conference on Learning Representations, March 2021. URL https://openreview.net/forum?id=LhY8QdUGSuw.

Annette Rios. longmbart, 2020. URL https://github.com/a-rios/longmbart.

Annette Rios, Nicolas Spring, Tannon Kew, Marek Kostrzewa, Andreas Säuberli, Mathias Müller, and Sarah Ebling. A New Dataset and Efficient Baselines for Document-level Text Simplification in German. In Proceedings of the Third Workshop on New Frontiers in Summarization, pages 152–161, Online and in Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.newsum-1.16. URL https://aclanthology.org/2021.newsum-1.16.

Raz Rotenberg. What is Gradient Accumulation in Deep Learning?, January 2020. URL https://towardsdatascience.com/what-is-gradient-accumulation-in-deep-learning-ec034122cfa.

Sebastian Ruder. The 4 Biggest Open Problems in NLP, January 2019. URL https://ruder.io/4-biggest-open-problems-in-nlp/.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. In Neurocomputing: foundations of research, pages 696–699. MIT Press, Cambridge, MA, USA, January 1988. ISBN 978-0-262-01097-9.

Markus Sagen. Large-Context Question Answering with Cross-Lingual Transfer. page 51, March 2021.

Thorben Schomacker and Marina Tropmann-Frick. Language Representation Models: An Overview. Entropy, 23(11):1422, November 2021. ISSN 1099-4300. doi: 10.3390/e23111422. URL https://www.mdpi.com/1099-4300/23/11/1422. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, Alex Wang, and Patrick Gallinari. QuestEval: Summarization Asks for Fact-based Evaluation. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 6594–6604, Online and Punta Cana, Dominican Republic, November 2021a. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.529. URL https://aclanthology.org/2021.emnlp-main.529.

Thomas Scialom, Louis Martin, Jacopo Staiano, Éric Villemonte de la Clergerie, and Benoît Sagot. Rethinking Automatic Evaluation in Sentence Simplification, April 2021b. URL http://arxiv.org/abs/2104.07560. arXiv:2104.07560 [cs].

Abigail See, Peter J. Liu, and Christopher D. Manning. Get To The Point: Summarization with Pointer-Generator Networks. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1073–1083, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1099. URL https://aclanthology.org/P17-1099.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. BLEURT: Learning Robust Metrics for Text Generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7881–7892, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.704. URL https://aclanthology.org/2020.acl-main.704.

Noam Shazeer. GLU Variants Improve Transformer, February 2020. URL http://arxiv.org/abs/2002.05202. arXiv:2002.05202 [cs, stat].

Advaith Siddharthan. A survey of research on text simplification. ITL - International Journal of Applied Linguistics, 165(2):259–298, January 2014. ISSN 0019-0829, 1783-1490. doi: 10.1075/itl.165.2.06sid. URL https://www.jbe-platform.com/content/journals/10.1075/itl.165.2.06sid. Publisher: John Benjamins.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. A Study of Translation Edit Rate with Targeted Human Annotation. In Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, pages 223–231, Cambridge, Massachusetts, USA, August 2006. Association for Machine Translation in the Americas. URL https://aclanthology.org/2006.amta-papers.25.

Matthew G. Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. TER-Plus: paraphrase, semantic, and alignment enhancements to Translation Edit Rate. Machine Translation, 23(2-3):117–127, September 2009. ISSN 0922-6567, 1573-0573. doi: 10.1007/s10590-009-9062-9. URL http://link.springer.com/10.1007/s10590-009-9062-9.

Lucia Specia. Translating from Complex to Simplified Sentences. In Thiago Alexandre Salgueiro Pardo, António Branco, Aldebaro Klautau, Renata Vieira, and Vera Lúcia Strube de Lima, editors, Computational Processing of the Portuguese Language, Lecture Notes in Computer Science, pages 30–39, Berlin, Heidelberg, 2010. Springer. ISBN 978-3-642-12320-7. doi: 10.1007/978-3-642-12320-7_5.

Nicolas Spring, Annette Rios, and Sarah Ebling. Exploring German Multi-Level Text Simplification. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), pages 1339–1349, Held Online, September 2021. INCOMA Ltd. URL https://aclanthology.org/2021.ranlp-1.150.

Regina Stodden and Laura Kallmeyer. TS-ANNO: An Annotation Tool to Build, Annotate and Evaluate Text Simplification Corpora. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 145–155, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-demo.14. URL https://aclanthology.org/2022.acl-demo.14.

Yixuan Su and Nigel Collier. Contrastive Search Is What You Need For Neural Text Generation, October 2022. URL http://arxiv.org/abs/2210.14140. arXiv:2210.14140 [cs].

Hong Sun and Ming Zhou. Joint Learning of a Dual SMT System for Paraphrase Generation. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pages 38–42, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/P12-2008.

Julia Suter, Sarah Ebling, and Martin Volk. Rule-based Automatic Text Simplification for German. page 9, 2016.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to Sequence Learning with Neural Networks. arXiv:1409.3215 [cs], December 2014. URL http://arxiv.org/abs/1409.3215. arXiv: 1409.3215.

Andreas Säuberli, Sarah Ebling, and Martin Volk. Benchmarking Data-driven Automatic Text Simplification for German. In Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI), pages 41–48, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-45-0. URL https://aclanthology.org/2020.readi-1.7.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient Transformers: A Survey. ACM Computing Surveys, April 2022. ISSN 0360-0300. doi: 10.1145/3530811. URL https://doi.org/10.1145/3530811. Just Accepted.

Lewis Tunstall, Leandro von Werra, and Thomas Wolf. Natural Language Processing with Transformers. O'Reilly Media, Inc., Erscheinungsort nicht ermittelbar, revised edtion edition, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems 30, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Tu Thanh Vu, Giang Binh Tran, and Son Bao Pham. Learning to Simplify Children Stories with Limited Data. In Ngoc Thanh Nguyen, Boonwat Attachoo, Bogdan Trawiński, and Kulwadee Somboonviwat, editors, Intelligent Information and Database Systems, Lecture Notes in Computer Science, pages 31–41, Cham, 2014. Springer International Publishing. ISBN 978-3-319-05476-6. doi: 10.1007/978-3-319-05476-6_4.

Susanne Wagner. Im Spannungsfeld von fachlichen Anforderungen und sprachlichen Barrieren - Einfache Sprache in der beruflichen Bildung. In Barrierefreie Kommunikation in interdisziplinärer Perspektive Proceedings, page 10, October 2015.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. Do NLP Models Know Numbers? Probing Numeracy in Embeddings. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5307–5315, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1534. URL https://aclanthology.org/D19-1534.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/hash/4496bf24afe7fab6f046bf4923da8de6-Abstract.html.

Shuohang Wang, Luowei Zhou, Zhe Gan, Yen-Chun Chen, Yuwei Fang, Siqi Sun, Yu Cheng, and Jingjing Liu. Cluster-Former: Clustering-based Sparse Transformer

for Long-Range Dependency Encoding, June 2021. URL http://arxiv.org/abs/2009.06097. arXiv:2009.06097 [cs].

Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-Attention with Linear Complexity, June 2020. URL http://arxiv.org/abs/2006.04768. arXiv:2006.04768 [cs, stat].

Zarah Weiß and Detmar Meurers. Modeling the Readability of German Targeting Adults and Children: An empirically broad analysis and its cross-corpus validation. In Proceedings of the 27th International Conference on Computational Linguistics, pages 303–317, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1026.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural Text Generation With Unlikelihood Training. In International Conference on Learning Representations, pages 1–18. International Conference on Learning Representations, March 2020. URL https://openreview.net/forum?id=SJeYe0NtvH.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data. In Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), page 10, May 2020.

Wikipedia. List of writing genres, November 2022. URL https://en.wikipedia.org/w/index.php?title=List_of_writing_genres&oldid=1120060692. Page Version ID: 1120060692.

Garrett Wilson and Diane J. Cook. A Survey of Unsupervised Deep Domain Adaptation, February 2020. URL http://arxiv.org/abs/1812.02849. arXiv:1812.02849 [cs, stat].

Felix Wu, Angela Fan, Alexei Baevski, Yann N. Dauphin, and Michael Auli. Pay Less Attention with Lightweight and Dynamic Convolutions, February 2019. URL http://arxiv.org/abs/1901.10430. arXiv:1901.10430 [cs].

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. Sentence Simplification by Monolingual Machine Translation. In Proceedings of the 50th Annual Meeting of

the Association for Computational Linguistics (Volume 1: Long Papers), pages 1015–1024, Jeju Island, Korea, July 2012. Association for Computational Linguistics. URL https://aclanthology.org/P12-1107.

Jin Xu, Yinuo Guo, and Junfeng Hu. Incorporate Semantic Structures into Machine Translation Evaluation via UCCA, October 2020. URL http://arxiv.org/abs/2010.08728. arXiv:2010.08728 [cs].

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. Learning to Break the Loop: Analyzing and Mitigating Repetitions for Neural Text Generation. In Proceedings of the 36th Conference on Neural Information Processing Systems, pages 1–36. 36th Conference on Neural Information Processing Systems, October 2022. URL http://arxiv.org/abs/2206.02369. arXiv:2206.02369 [cs].

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing Statistical Machine Translation for Text Simplification. Transactions of the Association for Computational Linguistics, 4:401–415, 2016. doi: 10.1162/tacl_a_00107. URL https://aclanthology.org/Q16-1029. Place: Cambridge, MA Publisher: MIT Press.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, 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, pages 483–498, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.41. URL https://aclanthology.org/2021.naacl-main.41.

Tiezheng Yu, Zihan Liu, and Pascale Fung. AdaptSum: Towards Low-Resource Domain Adaptation for Abstractive Summarization. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5892–5904, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.471. URL https://aclanthology.org/2021.naacl-main.471.

Karolina Zaczynska, Nils Feldhus, Robert Schwarzenberg, Aleksandra Gabryszak, and Sebastian Möller. Evaluating German Transformer Language Models with Syntactic Agreement Tests, July 2020. URL http://arxiv.org/abs/2007.03765. arXiv:2007.03765 [cs].

Hang Zhang, Yeyun Gong, Yelong Shen, Weisheng Li, Jiancheng Lv, Nan Duan, and Weizhu Chen. Poolingformer: Long Document Modeling with Pooling Attention. In Proceedings of the 38th International Conference on Machine Learning, pages 12437–12446. PMLR, July 2021. URL https://proceedings.mlr.press/v139/zhang21h.html. ISSN: 2640-3498.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating Text Generation with BERT, February 2020a. URL http://arxiv.org/abs/1904.09675. arXiv:1904.09675 [cs].

Xuan Zhang, Huizhou Zhao, KeXin Zhang, and Yiyang Zhang. SEMA: Text Simplification Evaluation through Semantic Alignment. In Proceedings of the 6th Workshop on Natural Language Processing Techniques for Educational Applications, pages 121–128, Suzhou, China, December 2020b. Association for Computational Linguistics. URL https://aclanthology.org/2020.nlptea-1.17.

Wei Zhao, Maxime Peyrard, Fei Liu, Yang Gao, Christian M. Meyer, and Steffen Eger. MoverScore: Text Generation Evaluating with Contextualized Embeddings and Earth Mover Distance. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 563–578, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1053. URL https://aclanthology.org/D19-1053.

Chen Zhu, Wei Ping, Chaowei Xiao, Mohammad Shoeybi, Tom Goldstein, Anima Anandkumar, and Bryan Catanzaro. Long-Short Transformer: Efficient Transformers for Language and Vision. In Advances in Neural Information Processing Systems, volume 34, pages 17723–17736. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/hash/9425be43ba92c2b4454ca7bf602efad8-Abstract.html.

Sanja Štajner, Marc Franco-Salvador, Paolo Rosso, and Simone Paolo Ponzetto. CATS: A Tool for Customized Alignment of Text Simplification Corpora. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), pages 3895–3903, Miyazaki, Japan, May 2018. European Language Resources Association (ELRA). URL https://aclanthology.org/L18-1615.