Reproduction & Benchmarking of German Text Simplification Systems

Regina Stodden
Department of Computational Linguistics
Faculty of Arts and Humanities
Heinrich Heine University Düsseldorf, Germany
regina.stodden@hhu.de

Abstract

The paper investigates the reproducibility of various approaches to automatically simplify German texts and identifies key challenges in the process. We reproduce eight sentence simplification systems including rules-based models, fine-tuned models, and prompting of autoregressive models. We highlight three main issues of reproducibility: the impossibility of reproduction due to missing details, code, or restricted access to data/models; variations in reproduction, hindering meaningful comparisons; and discrepancies in evaluation scores between reported and reproduced models. To enhance reproducibility and facilitate model comparison, we recommend the publication of model-related details, including checkpoints, code, and training methodologies. Our study also emphasizes the importance of releasing system generations, when possible, for thorough analysis and better understanding of original works. In our effort to compare reproduced models, we also create a German sentence simplification benchmark of the eleven models across seven test sets. Overall, the study underscores the significance of transparency, documentation, and diverse training data for advancing reproducibility and meaningful model comparison in automated German text simplification.

Keywords: Text Simplification, Reproduction Study, German Sentence Simplification Benchmark

1. Introduction

Text simplification (TS) is a Natural Language Processing (NLP) task that aims to enhance the accessibility and understandability of textual content for a diverse audience. This process involves the transformation of complex language structures into simpler and more straightforward forms to be better understandable for a specific target group, e.g., people with varying linguistic abilities, cognitive impairments, or those learning a new language (Alva-Manchego et al., 2020).

In recent years, German TS has also gained more attraction resulting in a few sentence simplification models, e.g., ZEST (Mallinson et al., 2020), sockeye-APA-LHA (Spring et al., 2021), mBART-DEplain-APA (Stodden et al., 2023), or custom-decoder-ats (Anschütz et al., 2023). But even if the NLP community has increased efforts in better reproducibility of (new) research by designing checklists on responsibility[1] or asking for reproducibility studies (Branco et al., 2020), some NLP models are still not easily reproducible. This has also hampered German TS because the access to resources is often restricted, not enough information are named for reproduction, models and code are unavailable, or system outputs are not made accessible to other researchers.

Therefore, in this work, we try to reproduce existing German TS models and re-generate their system outputs to facilitate analysing different German TS approaches or creating evaluation methods for German TS. Further, we discuss whether the reproduced models match or differ from the original models by analysing automatic TS metrics. To compare the reproduced models with each other, we also create a German sentence simplification benchmark on 7 test sets, including the system outputs of all 11 TS models. We make the code, the system outputs (if permitted by license), and the system evaluation reports available to increase the reproduction of this work in future German TS research. All materials are provided in `https://github.com/rstodden/easse-de`.

2. Related Work

The most similar works to ours are reproduction studies of English text simplification systems. Cooper and Shardlow (2020) and Arvan et al. (2022), for example, both reproduced the work on English TS by Nisioi et al. (2017): they trained a TS model using the provided code on a to-be-processed dataset and evaluate whether they can simulate the original findings. In our work, we will do the same for 7 German TS models.

Popović et al. (2022), in comparison, do not focus on the reproduction of a TS model but tried to repeat the human evaluation study proposed

[1] `https://aclrollingreview.org/responsibleNLPresearch/`

in Nisioi et al. (2017). Unfortunately, we cannot replicate this for German TS as human evaluation is rarely performed, and insufficient information would be available to repeat the process.

3. Method

For our reproduction study, we first describe the selection of models (see subsection 3.1) and then explain on which data we have trained and evaluated them (see subsection 3.2). Afterwards, we explain more about how we check the extent of the reproduction, whether the model seems totally different, rather close or identical to the original model (see subsection 3.3).

3.1. Models

Based on a literature review, we found some sentence simplification models for German, which have been proposed in recent years. We split the lines of research into

(i) rule-based models, e.g., rule-based model by Suter et al. (2016), *DISSIM* (Niklaus et al., 2019), and **hda-etr** (Siegel et al., 2019) (see subsection 4.1),

(ii) training sequence-to-sequence generation models, e.g., **sockeye-APA-LHA** (Spring et al., 2021) and other sockeye variants (Ebling et al., 2022) (see subsection 4.2),

(iii) fine-tuning sequence-to-sequence generation models, e.g., **mBART_DEplain-APA** (Stodden et al., 2023), **mBART_DEplain-APA+web** (Stodden et al., 2023), or *mT5-MultiSim* (Ryan et al., 2023) (see subsection 4.3),

(iv) zero shot simplification, e.g., *ZEST* (Mallinson et al., 2020),

(v) prompting autoregressive language models, e.g., **BLOOM** in Ryan et al. (2023) or Ponce et al. (2023) (see subsection 4.4), or *ChatGPT* in Manning (2023) or Deilen et al. (2023) (see subsection 4.4), and

(vi) combining autoregressive language models and sequence-to-sequence models, e.g., **custom-decoder-ats** (Anschütz et al., 2023) (see subsection 4.5).

We tried to reproduce all of the listed models. Unfortunately, for some models (see models highlighted in italics), neither the code nor the prompts are available, and they require too much computing power (i.e., mT5-MultiSim) to reproduce the model. Further, no system generations are available for these models, which could have been used for comparisons. Hence, we could only reproduce 6 models and their corresponding system outputs (see models highlighted in boldface). In section 4, we will describe each of the (reproduced) models in more detail and describe how we have reproduced them.

For the German TS benchmark, we also propose three new TS systems, i.e., mT5 (Xue et al., 2021) fine-tuned on a manually aligned news corpus, i.e., DEplain-APA (Stodden et al., 2023a), and mT5 fine-tuned on an automatically aligned web corpus, i.e., Simple German Corpus (Toborek et al., 2023) (and Toborek et al. 2023 for more corpus description paper). For both models, i.e., mT5-DEplain-APA[2] and mT5-SGC[3], we use the same hyperparameters (see Appendix A), the code and the system outputs also available in the Github repository. Additionally, we train sockeye on DEplain-APA with the same parameters as those used for sockeye-APA-LHA. This model is further called sockeye-DEplain-APA.

3.2. Training & Test Data

For training, fine-tuning, or prompting the models, we used the same training and evaluation data as named in the original work (if available).

hda-etr is a rule-based system which require no training data and was not evaluated on any test data yet. The training and/or evaluation data to reproduce trimmed_mbart_sent (i.e., DEplain-APA (Stodden et al., 2023a) and DEplain-web (Stodden et al., 2023b))[4], BLOOM (i.e., TextComplexityDE (TCDE19) (Naderi et al., 2019) and GEOlino (Mallinson et al., 2020)[5]), and encoder-decoder-ats (i.e., 20Minuten (Rios et al., 2021)[6]) are available, pre-split into training, development, and test set which enhanced the reproduction process. The APA-LHA data (Spring et al., 2021) to reproduce sockeye-APA-LHA is available upon request[7], but the data is randomly split into training and test sets each time when pre-processing the data. Hence, our experiments for sockeye-APA-LHA are conducted on a different split than in the original paper.

Additionally, we evaluate the models on the Simple German Corpus (Toborek et al., 2023); a manually aligned test set of web texts corresponding to the training data for mT5-SGC.

[2] https://huggingface.co/DEplain/mt5-DEplain-APA

[3] https://huggingface.co/DEplain/mt5-simple-german-corpus

[4] https://github.com/rstodden/DEPlain

[5] https://github.com/XenonMolecule/MultiSim

[6] https://github.com/ZurichNLP/20Minuten

[7] https://zenodo.org/records/5148163

3.3. Evaluation

Strategies on how to evaluate the similarity of reproduced models to the original models are (i) similarity of system outputs, (ii) comparison of automatic metrics measured on the new system outputs and the reported scores in the original papers, or (iii) comparison of human judgements on the system outputs.

In our study, the first strategy only applies to one model (i.e., trimmed_mbart_sent), as for the other models, the system outputs of the original TS models have not been made available. Hence, only for the trimmed_mbart_sent model we can compare how similar the published system generations are to our reproduced system generations. As similarity measurement, we check for exact matches in both sets of generations and apply the BERT-Score-F1 (Zhang* et al., 2020)[8]. Further, we did not validate the reproduced models by manual evaluation, as evaluation using manual judgements is often not conducted. If conducted, for example in Mallinson et al. (2020), no system generations (also no reproduced generations) are available to be analyzed.

Our strategy for validation of the reproduced models is to compare the reported scores of TS metrics, e.g., SARI (Xu et al., 2016), BLEU (Papineni et al., 2002), BERT-Score Precision (BS_P) (Zhang* et al., 2020), or the German adaptation of Flesch Reading Ease (FRE) (Amstad, 1978) with the scores measured for the system generations of the reproduced systems.

Most of the German TS papers describe that they are evaluating their systems using the implementation of the metrics in EASSE (Alva-Manchego et al., 2019)[9], i.e., Trienes et al. (2022), Ryan et al. (2023), Stodden et al. (2023)[10], and Ponce et al. (2023).

Mallinson et al. (2020) use their own version of SARI, BLEU, and FRE-BLEU, Anschütz et al. (2023) did not use EASSE as it does not include ROUGE. Hence, they use the implementations of BLEU, SARI, and ROUGE provided in Huggingface (Wolf et al., 2020). In other papers, e.g., Spring et al. (2021) or Rios et al. (2021), it is not mentioned which implementation of SARI or BLEU has been used. We have generated the metric scores for all models using the metrics implementation described in the original paper. If no details on the implementation were provided, we have generated the scores with the EASSE-DE package (Stodden, 2024).

4. TS Models & Reproduction

In the following, we briefly summarize the TS systems for which we can reproduce results and argue why we couldn't or haven't reproduced the other models (see subsection 4.6). See Table 1 for an overview of all reproduced models.

4.1. Rule-based Models

Siegel et al. (2019) implement some rules of easy-to-read guidelines ("Leichte Sprache") as a rule-based simplification model. In more detail, it contains the following two rules: substitution of complex words and compound splitting. Their model, called hda-etr, focuses only on lexical simplification. Siegel et al. include their rules into LanguageTool[11], a re-writing tool that assists in giving recommendations on how to correct or improve a given input text. For hda-etr a working code is provided[12], containing also a graphical interface for highlighting infringements against easy guidelines. For better performance, we re-implemented the code without the infringements and interface. The updated code can be found at `https://github.com/rstodden/easy-to-understand_language`.

4.2. Training Sequence-to-sequence Models

In recent years, the same department, i.e., the computational linguistics department of the University of Zurich, has published a few research papers including very similar TS models (see (Säuberli et al., 2020; Spring et al., 2021; Ebling et al., 2022)). They trained a sequence-to-sequence model with a transformer architecture using the Sockeye framework (Domhan et al., 2020) among others on APA-LHA-OR-B1 and APA-LHA-OR-A2 (Spring et al., 2021).

(Säuberli et al., 2020) experimented with a former and smaller version of APA-LHA and Sockeye. They report results of their base Sockeye architecture as well as additional experiments with, e.g., smaller batch sizes or extensions with linguistic features. They also experimented with data augmentation strategies, i.e., adding non-parallel simplifications (NULL2TRG), adding identical pairs with the simplifications on both sides of the pair (TRG2TRG), and adding pairs including

[8] For both metrics we have used their Huggingface implementation, i.e., `https://huggingface.co/spaces/evaluate-metric/exact_match` and `https://huggingface.co/spaces/evaluate-metric/bertscore`.

[9] EASSE is a Python package (Alva-Manchego et al., 2019) which is designed for the ease of evaluation of English sentence simplification.

[10] They are using the German version of EASSE, i.e., EASSE-DE (Stodden, 2024).

[11] `https://github.com/languagetool-org/languagetool`

[12] `https://github.com/hdaSprachtechnologie/easy-to-understand_language`

System Name	Reference	Type	Training Data	# Simp. Pairs	URL
hda-etr	Siegel et al. (2019)	rule-based	-	-	https://github.com/hdaSprachtechnologie/easy-to-understand_language
sockeye-APA-LHA	Spring et al. (2021) & Ebling et al. (2022)	seq2seq	APA-LHA OR-A2 & APA-LHA OR-B1	8,455 & 9,268	https://github.com/ZurichNLP/RANLP2021-German-ATS
sockeye-DEplain-APA	-	seq2seq	DEplain-APA	10,660	https://huggingface.co/DEplain
mBART-DEplain-APA	Stodden et al. (2023)	fine-tuned seq2seq	DEplain-APA	10,660	https://huggingface.co/DEplain/trimmed_mbart_sents_apa
mBART-DEplain-APA+web	Stodden et al. (2023)	fine-tuned seq2seq	DEplain-APA+web	10,660 + 1,594	https://huggingface.co/DEplain/trimmed_mbart_sents_apa_web
mT5-DEplain-APA	-	fine-tuned seq2seq	DEplain-APA	10,660	https://huggingface.co/DEplain
mT5-SGC	-	fine-tuned seq2seq	SGC	4,430	https://huggingface.co/DEplain
BLOOM-zero	Ryan et al. (2023)	zero-shot AR model	-	-	https://github.com/XenonMolecule/MultiSim
BLOOM-sim-10	Ryan et al. (2023)	few-shot AR model	TCDE19 & GEOlino	200 & 959	https://github.com/XenonMolecule/MultiSim
BLOOM-random 10	Ryan et al. (2023)	few-shot AR model	TCDE19 & GEOlino	200 & 959	https://github.com/XenonMolecule/MultiSim
custom-decoder-ats	Anschütz et al. (2023)	AR model + fine-tuned seq2seq	Simplified, monolingual German data & 20Minuten	544,467 & 17,905	https://huggingface.co/josh-oo/custom-decoder-ats

Table 1: Overview of German TS models including training details (i.e., training data and size of training samples). Each line separates different model types. The models in italics are newly proposed in this work.

back-translated simplifications and original simplifications (BT2TRG). Compared to their base model, adding more data decreased their SARI and BLEU scores, except for adding TRG2TRG, their overall best-performing system. As far as we know, these (and the ones by Anschütz et al. (2023)) are the only experiments with augmented data for German TS. Unfortunately, the experiments cannot be reproduced as neither the corpus, the models, the code, nor enough details regarding building the models are available.

However, we reproduce another Sockeye variant for German TS, which was proposed by (Spring et al., 2021; Ebling et al., 2022). However, we ran into a few issues which made our results incomparable to the original results and reported scores. First, due to non-solvable conflict errors of required Python packages, we need to update the sockeye version from 2.3.8 to 3.1.14[13]. The technical differences between both implementations are listed in Appendix B. Further, a data split into training, development and test set were neither provided nor fixed through parameters in the code.

4.3. Fine-tuning Sequence-to-sequence models (transfer-learning)

Rios et al. (2021) are the first who used mBART (Liu et al., 2020) for German document simplification. The main improvements of their approach compared to the standard mBART are to maximize the input length (to 4096), reduce the vocabulary to the 20k most frequent German tokens, and add a special language tag to specify the target language level (de_A1, de_A2, or de_B1). This approach has also been adapted for document simplification of news and web texts (Stodden et al., 2023), for paragraph simplification of clinical notes (Trienes et al., 2022), and for sentence simplification of news and web texts (Stodden et al., 2023). An overview of the adaptations and the different hyperparameters can be found in Appendix H.

As this work focuses on sentence simplification, we will just include the models proposed in Stodden et al. (2023), i.e., trimmed_mbart_sents_DEplain-APA (further called mBART-DEplain-APA) and trimmed_mbart_sents_DEplain-APA+web (further called mBART-DEplain-APA+web). Compared to Rios et al. (2021), they reduce the vocabulary to 35k and use one universal language tag (de_SI). As the names suggest the models are trained on DEplain-APA (Stodden et al., 2023a) or DEplain-APA plus DEplain-web (Stodden et al., 2023b). The checkpoints of the models, instructions on how to use them and their system generations for three test sets are available on Huggingface[14] and GitHub[15]. Hence, for reproduction we could use the Huggingface's text-to-text-generation pipeline to generate the system outputs on all test sets.

[13] Sockeye 3 (Hieber et al., 2022) is a neural machine translation pipeline.

[14] https://huggingface.co/DEplain/trimmed_mbart_sents_apa and https://huggingface.co/DEplain/trimmed_mbart_sents_apa_web

[15] https://github.com/rstodden/DEPlain/tree/main/G__Automatic_Text_Simplification_Experiments/generated_outputs

4.4. Autoregressive Language Models

Ryan et al. (2023) experimented with few-shot and zero-shot learning on a multi-lingual simplification corpus (including German) (Ryan et al., 2023b) using the autoregressive language model BLOOM (with 176 billion parameters) (Workshop, 2023). As examples in the few-shot setting they used either k random sentences pairs or k pairs in which source sentences are most similar to the to-be-tested sentence. We reproduced their experiments with the provided code and data.

4.5. Autoregressive Language Models + Sequence-to-sequence Models

For custom-decoder-ats (Anschütz et al., 2023), first, Anschütz et al. have fine-tuned an autoregressive language model on simplified language and then have combined it with a fine-tuned sequence-to-sequence model.

For custom-decoder-ats[16] (Anschütz et al., 2023) the checkpoint of the model and instructions on how to use it are available on Huggingface. Hence, we could use the provided code and Huggingface's text-to-text-generation pipeline to generate the system outputs on all test sets.

4.6. No Reproduction

We have not reproduced some of the models, the reasons for that are as follows: (Mallinson et al., 2020) propose a zero-shot cross-lingual sentence simplification model called *ZEST*. Although the code is available, we could not reproduce the ZEST model and regenerate its outputs.

Ryan et al. (2023) have proposed a multilingual sentence simplification model named mT5-MultiSim. They fine-tuned mT5 (Xue et al., 2021) on several corpora, including three German corpora, i.e., GEOlino (Ryan et al., 2023a), TCDE19 (Ryan et al., 2023c), and German News[17]. Due to limited computing power, we could not reproduce mT5-MultiSim as it was originally trained on 3 GPUs with the size of 48 GB for each.

Schlippe and Eichinger (2023) also used a T5 model for training their German TS model, but they use the multilingual model Flan-T5 (Chung et al., 2022). Their training and evaluation data is not available. Hence, we haven't included this model in our reproduction study.

Ponce et al. (2023) also experiment with BLOOM, but with the version with 7 billion parameters[18] and on structural simplification, i.e., split and rephrase. They do not provide enough information to reproduce their approach (e.g., prompt missing, few-shot or zero-shot?) as it is only a small side project of their work.

Some researchers experiment on German TS with ChatGPT, e.g., (Deilen et al., 2023), (Manning, 2023) or Schlippe and Eichinger (2023), but we do not include this approaches as we are focusing on open, non-proprietary language models.

5. Reproduction Results

To check whether the reproduced models are identical to the models described in the original work, we compare the newly measured scores with those reported in the original papers.

5.1. hda-etr

For hda-etr, unfortunately, no automatic scores are provided in the original paper, hence, we cannot compare whether our re-implementation works as expected. However, to enable comparisons in future work, in section 6, we report results of hda-etr on a few test sets.

5.2. Sockeye-APA-LHA

As previously mentioned, our reproduction of sockeye-APA-LHA was trained on a different model version with different training data and will also be evaluated on different test sets of APA-LHA. The comparison of reported and reproduced results also reflects this (see Appendix C): the BLEU scores differ between roughly 1.0 and for SARI, even between 4.0 and 9.0 points. Hence, unfortunately, our reproduced Sockeye-APA-LHA model is not comparable to the original model, and the conclusions we can draw from the reproduction might not be the same as the original model.

5.3. BLOOM

For the three different approaches using BLOOM, i.e., zero-shot BLOOM, random 10-shot BLOOM and similarity 10-shot BLOOM, our reproduced system generations seem to be slightly different than the original system generations (see Appendix D). For all approaches on GEOlino, the SARI scores differ by less than 1.5 points. However, for TCDE19, the gap between the SARI scores is up to 2.5 points. If we also compare the baseline results, we can see that these are identical. Hence, we can exclude different data splits as a possible reason. It remains unclear why the numbers are that different, either the provided code is slightly different from the one used for the reported experiments, the evaluation method is different, or predictions of BLOOM are not fixed.

[16] https://huggingface.co/josh-oo/custom-decoder-ats

[17] Unfortunately, although it should be available on request, we do not yet have access to this corpus.

[18] https://huggingface.co/bigscience/bloom-7b1

5.4. custom-decoder-ats

To check whether custom-decoder-ats still meets the results reported in the original paper, we reproduced the results on the 20min corpus. The reported results of Anschütz et al. (2023) differ only slightly from the reproduced results (BLEU: roughly 0.3 and SARI: roughly 2.0, see Appendix E). Hence, we argue that the reproduced model is fairly comparable to the described model.

5.5. mBART-DEplain-APA & mBART-DEplain-APA+web

Even if the checkpoints of the mBART models are provided to reproduce the system generations, the scores of the reproduced models differ from the reported scores in the original paper (see Appendix F). For example, the SARI scores of mBART-DEplain-APA differ by roughly 2 points on DEplain-web or roughly 4 points on DEplain-APA when comparing reproduced and reported scores. However, the scores of the reproduced baseline are identical to the reported baseline using the EASSE-DE evaluation framework. Hence, we can argue that the same evaluation approach and the same test data have been used, and these are not the reasons for the differences.

We also compared the similarity of the reproduced system generations of mBART-DEplain-APA and mBART-DEplain-APA+web with the provided system generations of the original models by measuring their exact match and BERTScore-F1. As can be seen in Table 2, the exact matches for mBART-DEplain-APA are on each test set lower than 50% and for mBART-DEplain-APA+web varying between 49% and 74%. However, the BERTScores show that the predictions per instance are quite similar for both models, even if, again, the scores are higher for DEplain-APA+web. This confirms the previous findings; thus, the uploaded model must be slightly different from the one used to report the results in the original paper.

	exact↑	BS mean↑	BS min	BS std
DEplain-APA	42.24	0.9589	0.6587	0.0546
DEplain-web	17.98	0.9163	0.4885	0.0740
TCDE19	9.20	0.8889	0.7253	0.0739

(a) mBART-DEplain-APA

	exact↑	BS mean↑	BS min	BS std
DEplain-APA	73.68	0.9827	0.7328	0.0391
DEplain-web	56.99	0.9628	0.4623	0.0694
TCDE19	48.80	0.9593	0.7360	0.0600

(b) mBART-DEplain-APA+web

Table 2: Similarity between copied system generations and reproduced system generations by exact match (in %), and BERT-Score F1 values (mean, minimum, and standard deviation).

6. German TS Benchmark

The previous results show that most of the reproduced models are similar to the results of the original models. However, the results of the models are not comparable to each other as they are evaluated on different test sets and with different metrics implementations. To unify the evaluation reports and build a German sentence simplification benchmark, we evaluate the reproduced models and three new models on seven German sentence simplification test sets, i.e., APA-LHA-OR-A2, APA-LHA-OR-B1, DEplain-APA, DEplain-web, Simple German Corpus (SGC), TCDE19, and GEOlino. We first describe the evaluation approach, then report the models' results per domain of the test set, and finally compare the results across all test sets.

6.1. Method

All models are automatically evaluated against one reference[19] and on the same evaluation metrics, i.e., SARI (Xu et al., 2016), BLEU (Papineni et al., 2002), BS_P (Zhang* et al., 2020), and FRE (Amstad, 1978). Although the metrics have been criticized regarding their suitability for text simplification evaluation (e.g., see Sulem et al. 2018, Tanprasert and Kauchak 2021, or Alva-Manchego et al. 2021), we are reporting them due to missing alternatives. Following the recommendation of Alva-Manchego et al. (2021), we use BS_P as the main evaluation metric. If the score is high, we verify it with other metrics, such as SARI, BLEU, and FRE. In addition, as recommended by Tanprasert and Kauchak (2021) and Alva-Manchego et al. (2019), we also report linguistic features to get more insights into the system-generated simplifications, i.e., compression ratio and sentence splits.

For the measurement of the metrics and features, we are using the evaluation framework, i.e., EASSE-DE, a multi-lingual adaptation of the EASSE evaluation framework (Stodden, 2024). In comparison to EASSE, EASSE-DE includes, for example, German tokenization, German readability metrics, and a multi-lingual version of BERTScore. In Appendix G, more details are provided regarding the settings used for evaluation with EASSE-DE. We do not manually evaluate the models as this is out of the scope of this work.

	BLEU↑	SARI↑	BS_P↑	FRE↑	Compr. ratio↓	Sent. splits↑
hda_LS	3.02	14.02	0.12	37.55	1.14	1.04
sockeye-APA-LHA	**13.59**	**51.77**	**0.35**	68.65	0.64	0.99
sockeye-DEplain-APA	4.79	40.32	0.25	**70.25**	0.71	1.25
mBART-DEplain-APA	4.73	30.28	0.23	57.55	0.85	1.33
mBART-DEplain-APA+web	4.56	25.89	0.23	56.35	0.84	1.16
mT5-DEplain-APA	4.65	34.47	0.24	58.10	0.58	1.09
mT5-SGC	2.78	39.79	0.28	**70.25**	**0.48**	1.00
BLOOM-zero	2.44	26.83	0.19	51.85	0.82	1.29
BLOOM-10-random	2.64	33.05	0.24	57.95	0.64	0.98
BLOOM-10-similarity	5.10	38.05	0.29	64.60	0.59	0.98
custom-decoder-ats	0.28	37.05	0.08	52.60	3.16	**2.91**
Identity baseline	3.50	3.90	0.18	44.70	1.00	1.00
Reference baseline	100	100	1.00	69.55	0.60	0.97
Truncate baseline	2.60	17.49	0.19	54.25	0.79	1.00

Table 3: Evaluation on APA-LHA-OR-A2.

	BLEU↑	SARI↑	BS_P↑	FRE↑	Compr. ratio↓	Sent. splits↑
hda_LS	4.54	15.49	0.15	36.15	1.15	1.10
sockeye-APA-LHA	**11.00**	**44.93**	**0.32**	61.90	0.70	0.97
sockeye-DEplain-APA	3.57	39.4	0.25	**70.65**	0.68	1.26
mBART-DEplain-APA	5.32	30.94	0.26	57.65	0.86	1.37
mBART-DEplain-APA+web	5.81	26.61	0.25	56.05	0.85	1.19
mT5-DEplain-APA	4.92	35.70	0.26	57.70	0.57	1.10
mT5-SGC	2.54	39.36	0.29	70.45	**0.48**	1.00
BLOOM-zero	3.41	27.56	0.21	56.80	0.84	1.34
BLOOM-10-random	5.18	32.43	0.26	56.25	0.71	0.98
BLOOM-10-similarity	6.21	37.22	0.27	62.00	0.72	0.98
custom-decoder-ats	0.52	37.59	0.07	49.70	3.78	**3.51**
Identity baseline	5.47	4.89	0.22	43.70	1.00	1.00
Reference baseline	100	100	1.00	62.60	0.68	0.98
Truncate baseline	4.59	18.36	0.22	53.85	0.79	1.00

Table 4: Evaluation on APA-LHA-OR-B1.

6.2. News Test Sets: APA-LHA-OR-A2 & APA-LHA-OR-B1 & DEplain-APA

Although, mBART-DEplain-APA, mT5-DEplain-APA, sockeye-DEplain-APA, and sockeye-APA-LHA are trained on alignments of the same source, i.e., news of the Austrian Press Agency, sockeye-APA-LHA achieves clearly better BS_P (difference > 5), SARI (difference > 9) and BLEU scores (difference > 5) on both APA-LHA test sets (see Table 3 and Table 4). In contrast, sockeye-DEplain-APA, mBART-DEplain-APA and mT5-DEplain-APA perform much better on DEplain-APA than sockeye-APA-LHA (see Table 5) with respect to BS_P (difference > 16), SARI (difference > 4), and BLEU (difference > 8). Hence, as expected, the models are most suitable on the test set of the corpus that they have been trained on (APA-LHA vs. DEplain-APA). Besides computational reasons, this might also be due to the different alignment strategies (APA-LHA: automatically vs. DEplain-APA: manually) or the different extent of the complex-simple pairs (APA-LHA: OR to A2 or B1 vs. DEplain-APA: B2 to A2) of both corpora.

	BLEU↑	SARI↑	BS_P↑	FRE↑	Compr. ratio↓	Sent. splits↑
hda_LS	22.3	26.06	0.55	64.60	1.00	1.00
sockeye-APA-LHA	11.84	40.16	0.37	63.70	0.94	0.97
sockeye-DEplain-APA	19.58	**44.14**	0.53	71.45	0.94	1.09
mBART-DEplain-APA	**28.49**	38.72	**0.64**	65.30	0.99	1.07
mBART-DEplain-APA+web	28.03	33.81	**0.64**	65.20	0.98	1.05
mT5-DEplain-APA	22.32	39.41	0.61	63.20	0.87	1.04
mt5-SGC	8.12	37.92	0.48	**71.65**	**0.74**	1.00
BLOOM-zero	16.14	35.43	0.53	65.10	0.87	1.14
BLOOM-10-random	17.97	35.93	0.57	65.50	0.91	1.00
BLOOM-10-similarity	20.97	41.27	0.57	65.70	0.93	1.07
custom-decoder-ats	1.24	36.42	0.16	53.00	7.41	**5.07**
Identity baseline	26.89	15.25	0.63	58.75	1.00	1.00
Reference baseline	100.00	100.00	1.00	65.80	1.03	1.20
Truncate baseline	16.11	27.20	0.55	66.10	0.80	1.01

Table 5: Evaluation on DEplain-APA.

However, mBART-DEPlain-APA, mT5-DEplain-APA, and sockeye-DEplain-APA are all trained on the same training data. Hence, their differences in performance seem to be due to their system architectures. When evaluating on DEplain-APA, sockeye-DEplain-APA splits the sentences most often, whereas mT5-DEplain-APA compresses most sentences. Further, the mBART model achieves the best results concerning BS_P and BLEU, but sockeye-DEplain-APA achieves the highest SARI score and a much lower BS_P score (difference = 11). More experiments with different hyperparameters and training sets are required to confirm this finding.

Further, we can compare the mBART models with respect to a data augmentation strategy because both models are trained in an identical setting except for additional training data in mBART-DEplain-APA+web. The augmented data (automatically aligned and from different domains) seems to reduce the quality of the system generations on the news domain as on all three test sets: the BLEU, SARI and BS_P scores are lower for mBART-DEplain-APA+web than mBART-DEplain-APA.

6.3. Web Test Sets: DEplain-web & Simple-German-Corpus

	BLEU↑	SARI↑	BS_P↑	FRE↑	Compr. ratio↓	Sent. splits↑
sockeye-APA-LHA	0.24	32.41	0.13	69.55	0.74	0.90
sockeye-DEplain-APA	3.44	36.24	0.24	76.7	0.76	1.32
mBART-DEplain-APA	13.50	33.11	0.40	69.65	0.90	1.30
mBART-DEplain-APA+web	**17.99**	34.07	**0.44**	69.05	0.85	1.16
mT5-DEplain-APA	6.80	**37.15**	0.36	70.90	0.63	1.10
mt5-SGC	2.50	36.56	0.37	**78.10**	**0.47**	0.93
BLOOM-zero	10.88	30.58	0.35	70.30	0.85	1.28
BLOOM-10-random	11.06	30.90	0.39	68.55	0.69	0.98
BLOOM-10-similarity	11.62	37.03	0.42	70.05	0.63	0.98
custom-decoder-ats	0.72	34.92	0.10	57.15	5.41	**3.79**
Identity baseline	20.85	11.93	0.42	62.95	1.00	1.00
Reference baseline	100.00	100.00	1.00	77.90	0.94	1.84
Truncate baseline	17.28	24.58	0.40	67.05	0.82	1.02

Table 6: Evaluation on DEplain-web.

Focusing on the web test sets, mBART-DEplain-APA+web performs best on DEplain-web (wrt. BS_P and BLEU, see Table 6) and BLOOM-10-similarity best on SGC (wrt. BS_P, SARI, and BLEU, see Table 7). Although mt5-SGC and mBART-DEplain-APA+web are both trained on complex-simple pairs of the web domain, both achieve comparable low BS_P scores on SGC. A reason for that might be the mix of topics, different alignment types (automatic vs. manual), or

[19] Unfortunately, no test set contains more than one reference. Therefore, the results should be considered with caution as the suitability of the evaluation metrics has been checked on (English) test sets with multiple references.

a mix of language varieties (Easy German, Plain German, and others) in their training data.

customer-decoder-ats and sockeye-APA-LHA perform the worst on both datasets (wrt. BS_S). Following the compression ratio and sentence split values, customer-decoder-ats seems to hallucinate by extending the original text with many additional sentences. This might be because customer-decoder-ats is originally built to simplify longer texts. Sockeye-APA-LHA appears to underperform on test sets for other target groups or domains other than its training data.

	BLEU↑	**SARI**↑	**BS_P**↑	**FRE**↑	**Compr. ratio**↓	**Sent. splits**↑
hda_LS	6.34	20.22	0.25	41.15	1.00	1.03
sockeye-APA-LHA	0.33	35.50	0.13	63.70	0.80	0.82
sockeye-DEplain-APA	1.35	37.86	0.18	**71.05**	0.79	1.01
mBART-DEplain-APA	5.70	32.77	0.31	58.15	0.97	1.00
mBART-DEplain-APA+web	6.56	29.80	0.33	44.95	1.61	1.09
mT5-DEplain-APA	2.81	35.92	0.30	51.45	0.76	0.88
mt5-SGC	3.30	43.62	0.37	58.55	**0.61**	0.85
BLOOM-zero	3.76	31.95	0.25	53.55	0.81	1.07
BLOOM-10-random	4.64	33.16	0.30	51.50	0.75	0.92
BLOOM-10-similarity	**13.32**	**44.66**	**0.38**	58.65	0.92	1.13
custom-decoder-ats	0.44	36.53	0.06	32.05	8.83	**3.68**
Identity baseline	7.46	6.51	0.29	41.15	1.00	1.00
Reference baseline	100.00	100.00	1.00	65.40	1.25	1.81
Truncate baseline	4.66	20.12	0.28	50.50	0.81	0.87

Table 7: Evaluation on SGC.

6.4. Knowledge Acquiring Test Sets: GEOlino & TCDE19

	BLEU↑	**SARI**↑	**BS_P**↑	**FRE**↑	**Compr. ratio**↓	**Sent. splits**↑
hda_LS	55.22	34.20	0.76	61.50	1.00	1.00
sockeye-APA-LHA	0.69	18.94	0.15	69.45	1.05	0.92
sockeye-DEplain-APA	7.27	24.71	0.33	77.3	0.96	1.15
mBART-DEplain-APA	50.56	**44.29**	0.74	70.75	1.04	1.15
mBART-DEplain-APA+web	**55.35**	44.28	**0.79**	64.60	0.97	1.08
mT5-DEplain-APA	28.43	36.93	0.65	67.95	0.80	1.04
mt5-SGC	11.92	28.75	0.55	**78.30**	**0.70**	0.94
BLOOM-zero	28.18	32.15	0.59	67.85	0.87	1.26
custom-decoder-ats	0.77	22.05	0.08	46.55	14.61	**4.76**
Identity baseline	67.12	26.81	0.86	61.50	1.00	1.00
Reference baseline	100.00	100.00	1.00	66.00	0.95	1.32
Truncate baseline	45.39	29.78	0.75	63.80	0.83	1.00

Table 8: Evaluation on GEOlino (n=663).

We also evaluate on two test sets with simplification of knowledge-acquiring platforms, i.e. GEOlino simplification of science for children and TCDE19 with simplifications of Wikipedia texts for non-native German speakers. For both corpora, only test sets and no training sets exist. Therefore, BLOOM-10-random and BLOOM-10-similarity cannot be evaluated as no samples exist that could be added during prompting.

Further, currently, no training data for sentence simplification in the same domain or for the same target group of these test sets exists. Therefore, the presented results in Table 8 and Table 9 can be seen in the out-of-domain evaluation of the TS systems. mBART-DEplain-APA+web performs best on both test sets with respect to BLEU and BERTScore whereas mBART-DEplain-APA achieves best SARI scores.

	BLEU↑	**SARI**↑	**BS_P**↑	**FRE**↑	**Compr. ratio**↓	**Sent. splits**↑
hda_LS	20.66	26.92	0.45	33.65	1.00	1.01
sockeye-APA-LHA	0.13	29.87	0.14	**69.05**	0.43	0.97
sockeye-DEplain-APA	0.68	31.79	0.19	65.0	0.51	1.42
mBART-DEplain-APA	13.69	**39.14**	0.50	51.10	0.76	1.57
mBART-DEplain-APA+web	**17.75**	37.37	**0.55**	43.65	0.74	1.29
mT5-DEplain-APA	2.84	35.09	0.40	46.60	0.40	1.14
mt5-SGC	1.05	32.98	0.38	64.40	**0.31**	0.97
BLOOM-zero	9.46	34.96	0.42	45.55	0.78	1.75
custom-decoder-ats	1.73	32.87	0.22	27.70	1.54	**4.22**
Identity baseline	27.31	14.99	0.55	28.10	1.00	1.00
Reference baseline	100.00	100.00	1.00	51.20	0.95	2.04
Truncate baseline	20.17	26.45	0.52	37.65	0.81	1.00

Table 9: Evaluation on TCDE19 (n=250).

6.5. Comparison Across Domains

In this section, we analyse the reproduced models' results across all test sets. For a better overview of the capabilities of the models across the test sets, in Appendix I, we provide the BS_P scores of all models on all test sets and in Appendix J for SARI. The tables also include the rank of the model per test set.[20]

Comparing the performance of the models across all test sets, the scores of hda_LS are always close to the scores of the identity baseline, which might be due to only minimal changes in the original sentences. Of all models, custom-decoder-ats still produces the most complex sentences with respect to FRE and compression ratio. On all test sets, the readability seems even lower for custom-decoder-ats than for the original complex texts (see identity baselines). The reason for that is hallucination in the system outputs, which could be explained by the model's design as it is trained for document simplification, in which the texts are, by nature, longer than in sentence simplification corpora. mt5-SGC has the lowest compression ratio on all test sets, possibly due to the very short sentences in its training data, which are mostly texts in Easy German.

Overall, no system ranks best across all test sets (wrt. BS_P and SARI). On average, BLOOM-10-similarity performs best (wrt. BS_P) if similar examples are available, whereas mBART-DEplain-APA+web achieves on average, the best ranks following BS_P on all seven test sets, and sockeye-DEplain-APA performs best on both settings wrt. SARI. The additional data, i.e., massive data during pre-training in BLOOM and additional web data for mBART, seems to have a positive effect on the system generations or at least the evaluation

[20] BLOOM-10-random and BLOOM-10-similarity require training samples each time when generating a simplified sentence, which is not available for all test sets (e.g., TCDE19 or GEOlino). In addition, when simplifying texts in practice, i.e., as an intra-lingual translation tool, also no simplification examples would be made available. In order to integrate this limitation, BLOOM-10-random and BLOOM-10-similarity will be penalized in our evaluation on TCDE19 and GEOlino with the highest rank equal to the worst result.

scores. In comparison, sockeye-DEplain-APA is only trained on simple-complex pairs of DEplain-APA, and, therefore, the model cannot transfer well to other domains as it only performs very well on the news test sets.

However, the transfer learning of pre-trained models appears to be more effective for BLOOM than for mBART or mT5, which might be due to its larger pre-training data size. Further, BLOOM has been prompted with only a few samples, but it still outperforms the smaller language models, even though they have been fine-tuned on many task-relevant samples. We can also confirm the findings of (Ryan et al., 2023) that BLOOM-10-similarity generates better simplifications than BLOOM-10-random and better than BLOOM-zero on all test sets with respect to BS_P and SARI. For more comparisons of mT5 and BLOOM (also including the capability of TS models across multiple languages), we refer the interested reader to Ryan et al. (2023).

For the sockeye models, we assume that the size of APA-LHA and DEplain-APA is too small to train a model from scratch. It could be a promising approach to combine similar training data with each other to increase the training size for sockeye, e.g., a combination of APA-LHA, DEplain-APA, and/or SGC because a positive effect of data combination has been revealed for mBART-DEplain-APA+web compared to mBART-DEplain-APA.

7. Conclusion & Discussion

We have reproduced different approaches on how to simplify German texts automatically. However, we have also revealed some new issues regarding models' reproduction and have confirmed previously named problems with respect to the training data and the evaluation process.

We found the following three main issues with the models, i.e.,

(i) impossibility of reproduction, e.g., due to missing details, missing code, not-available or restricted-access data, or restricted-access language models,

(ii) differences in reproduction and, therefore, less comparison, e.g., due to different data splits, and

(iii) differences in evaluation scores for reported scores and scores of reproduced models due to different system outputs or different implementations of metrics.

For better reproducibility and better comparison between ATS models, we recommend publishing as many details and materials related to the models as possible with respect to copyright and licenses, e.g., publishing (i) the checkpoints of the trained or fine-tuned models and code how to reuse them, or (ii) the code and a description of how to rebuild and re-train the model, including model versions and used prompts.

Additionally, we also recommend publishing the system generations (if not restricted by copyright) to enable further analysis of the results. In our reproduction study, in the comparison of the reported and reproduced scores, we have seen that even if the ATS models or the code are available for reproduction, the system generations seem to be different from those described in the original works. Hence, some analysis of the original work might not hold when reproducing.

We have also shown that, for example, due to limited computing resources, system generations cannot always be reproduced even if the code or the model is provided. We argue that the system generations are helpful for understanding the original work better and can also be valuable for building better evaluation metrics.

To compare the reproduced models with each other, we have built a German sentence simplification benchmark on 7 test sets. We found, as expected, that models achieve the best scores if they are evaluated and trained on the same corpus. We have also shown that some models, especially mBART-DEplain-APA+web (wrt. SARI and BERT-Score), achieve good scores on test sets on which domain or target group they were not trained. Hence, the models seem to have learned some universal simplification. Nevertheless, we want to emphasize that simplicity is subjective. Hence, for each person and each target group, a text is easier or more difficult to read. Following this, a text simplification model should also learn to simplify for a specific target group and not for many target groups at the same time (Gooding, 2022; Stajner, 2021). Therefore, we recommend not mixing training data from texts written for different target groups but evaluating the models only on texts written for the target group of interest. Due to limited resources, this is currently impractical. Hence, we have presented approaches with mixed training data and evaluated across texts of different target groups.

However, the analysis with respect to SARI or BERT-Score allows us to draw different conclusions: Following their scores, different models are ranked as best models. More work regarding the suitability and interpretability of evaluation metrics (especially regarding test sets with only one reference) is required for a more reliable interpretation of this German TS benchmark.

8. Bibliographical References

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. 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. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (un)suitability of automatic evaluation metrics for text simplification. *Computational Linguistics*, 47(4):861–889.

Toni Amstad. 1978. *Wie verständlich sind unsere Zeitungen?* Studenten-Schreib-Service. PhD Thesis.

Miriam Anschütz, Joshua Oehms, Thomas Wimmer, Bartłomiej Jezierski, and Georg Groh. 2023. Language models for German text simplification: Overcoming parallel data scarcity through style-specific pre-training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1147–1158, Toronto, Canada. Association for Computational Linguistics.

Mohammad Arvan, Luís Pina, and Natalie Parde. 2022. Reproducibility of exploring neural text simplification models: A review. In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, pages 62–70, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.

António Branco, Nicoletta Calzolari, Piek Vossen, Gertjan Van Noord, Dieter van Uytvanck, João Silva, Luís Gomes, André Moreira, and Willem Elbers. 2020. A shared task of a new, collaborative type to foster reproducibility: A first exercise in the area of language science and technology with REPROLANG2020. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5539–5545, Marseille, France. European Language Resources Association.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models. ArXiv preprint, arXiv:2210.11416.

Michael Cooper and Matthew Shardlow. 2020. CombiNMT: An exploration into neural text simplification models. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 5588–5594, Marseille, France. European Language Resources Association.

Silvana Deilen, Sergio Hernández Garrido, Ekaterina Lapshinova-Koltunski, and Christiane Maaß. 2023. Using ChatGPT as a CAT tool in easy language translation. In *Proceedings of the Second Workshop on Text Simplification, Accessibility and Readability*, pages 1–10, Varna, Bulgaria. INCOMA Ltd., Shoumen, Bulgaria.

Tobias Domhan, Michael Denkowski, David Vilar, Xing Niu, Felix Hieber, and Kenneth Heafield. 2020. The sockeye 2 neural machine translation toolkit at AMTA 2020. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 110–115, Virtual. Association for Machine Translation in the Americas.

Sarah Ebling, Alessia Battisti, Marek Kostrzewa, Dominik Pfütze, Annette Rios, Andreas Säuberli, and Nicolas Spring. 2022. Automatic text simplification for german. *Frontiers in Communication*, 7.

Sian Gooding. 2022. On the ethical considerations of text simplification. In *Ninth Workshop on Speech and Language Processing for Assistive Technologies (SLPAT-2022)*, pages 50–57, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2020. Zero-shot crosslingual sentence simplification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5109–5126, Online. Association for Computational Linguistics.

Sabine Manning. 2023. KI-Tools für Einfache Sprache: (3) Bard und GPT-4 im Vergleich. https://multisprech.org/2023/11/16/ki-tools-fuer-einfache-sprache-3-bard-und-gpt-4-im-vergleich/. Last change: 2023-11-16; Last access: 2024-01-11.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019. DisSim: A discourse-aware syntactic text simplification framework for English and German. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 504–507, Tokyo, Japan. Association for Computational Linguistics.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

David Ponce, Thierry Etchegoyhen, Jesús Calleja Pérez, and Harritxu Gete. 2023. Split and rephrase with large language models. ArXiv preprint, arXiv:2312.11075.

Maja Popović, Sheila Castilho, Rudali Huidrom, and Anya Belz. 2022. Reproducing a manual evaluation of the simplicity of text simplification system outputs. In *Proceedings of the 15th International Conference on Natural Language Generation: Generation Challenges*, pages 80–85, Waterville, Maine, USA and virtual meeting. Association for Computational Linguistics.

Annette Rios, Nicolas Spring, Tannon Kew, Marek Kostrzewa, Andreas Säuberli, Mathias Müller, and Sarah Ebling. 2021. 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. Association for Computational Linguistics.

Michael Ryan, Tarek Naous, and Wei Xu. 2023. Revisiting non-English text simplification: A unified multilingual benchmark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4898–4927, Toronto, Canada. Association for Computational Linguistics.

Andreas Säuberli, Sarah Ebling, and Martin Volk. 2020. 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. European Language Resources Association.

Tim Schlippe and Katharina Eichinger. 2023. Multilingual text simplification and its performance on social sciences coursebooks. In *Artificial Intelligence in Education Technologies: New Development and Innovative Practices*, pages 119–136, Singapore. Springer Nature Singapore.

Melanie Siegel, Dorothee Beermann, and Lars Hellan. 2019. Aspects of linguistic complexity: A german - norwegian approach to the creation of resources for easy-to-understand language. In *11th International Conference on Quality of Multimedia Experience QoMEX 2019, Berlin, Germany, June 5-7, 2019*, pages 1–3. IEEE.

Nicolas Spring, Annette Rios, and Sarah Ebling. 2021. 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. INCOMA Ltd.

Sanja Stajner. 2021. Automatic text simplification for social good: Progress and challenges. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2637–2652, Online. Association for Computational Linguistics.

Regina Stodden. 2024. EASSE-DE: Easier Automatic Sentence Simplification Evaluation for German. ArXiv preprint, arXiv:2404.03563.

Regina Stodden, Omar Momen, and Laura Kallmeyer. 2023. DEplain: A German parallel corpus with intralingual translations into plain language for sentence and document simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16441–16463, Toronto, Canada. Association for Computational Linguistics.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*

Processing, pages 738–744, Brussels, Belgium. Association for Computational Linguistics.

Julia Suter, Sarah Ebling, and Martin Volk. 2016. Rule-based automatic text simplification for german. In *13th Conference on Natural Language Processing (KONVENS 2016)*.

Teerapaun Tanprasert and David Kauchak. 2021. Flesch-kincaid is not a text simplification evaluation metric. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 1–14, Online. Association for Computational Linguistics.

Vanessa Toborek, Moritz Busch, Malte Boßert, Christian Bauckhage, and Pascal Welke. 2023. A new aligned simple German corpus. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11393–11412, Toronto, Canada. Association for Computational Linguistics.

Jan Trienes, Jörg Schlötterer, Hans-Ulrich Schildhaus, and Christin Seifert. 2022. Patient-friendly clinical notes: Towards a new text simplification dataset. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 19–27, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

BigScience Workshop. 2023. Bloom: A 176b-parameter open-access multilingual language model. Version 4.

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

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

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

9. Language Resource References

Alva-Manchego, Fernando and Martin, Louis and Scarton, Carolina and Specia, Lucia. 2019. *EASSE: Easier Automatic Sentence Simplification Evaluation*. GitHub. PID https://github.com/feralvam/easse.

Felix Hieber and Michael Denkowski and Tobias Domhan and Barbara Darques Barros and Celina Dong Ye and Xing Niu and Cuong Hoang and Ke Tran and Benjamin Hsu and Maria Nadejde and Surafel Lakew and Prashant Mathur and Anna Currey and Marcello Federico. 2022. *Sockeye 3: Fast Neural Machine Translation with PyTorch*.

Mallinson, Jonathan and Sennrich, Rico and Lapata, Mirella. 2020. *GEOlino*. GitHub. PID https://github.com/Jmallins/ZEST-data.

Babak Naderi and Salar Mohtaj and Kaspar Ensikat and Sebastian Möller. 2019. *TextComplexityDE*. GitHub. PID https://github.com/babaknaderi/TextComplexityDE.

Rios, Annette and Spring, Nicolas and Kew, Tannon and Kostrzewa, Marek and Säuberli, Andreas and Müller, Mathias and Ebling, Sarah. 2021. *20Minuten*. GitHub. PID https://github.com/ZurichNLP/20Minuten.

Ryan, Michael and Naous, Tarek and Xu, Wei. 2023a. *GEOlino-small*. GitHub. PID https://github.com/XenonMolecule/MultiSim/tree/main/data/German.

Ryan, Michael and Naous, Tarek and Xu, Wei. 2023b. *MultiSim*. GitHub. PID https://github.com/XenonMolecule/MultiSim.

Ryan, Michael and Naous, Tarek and Xu, Wei. 2023c. *TextComplexityDE-small*. GitHub. PID https://github.com/XenonMolecule/MultiSim/tree/main/data/German.

Spring, Nicolas and Rios, Annette and Ebling, Sarah. 2021. *LHA Sentence Alignments Extracted From the Austria Press Agency Corpus*. Zenodo. PID https://doi.org/10.5281/zenodo.5148163.

Regina Stodden. 2024. *EASSE-DE: Easier Automatic Sentence Simplification Evaluation for German*. GitHub. PID https://github.com/rstodden/easse-de.

Regina Stodden and Omar Momen and Laura Kallmeyer. 2023a. *DEplain-APA*. Zenodo. PID https://doi.org/10.5281/zenodo.8304430.

Stodden, Regina and Momen, Omar and Kallmeyer, Laura. 2023b. *DEplain-web*. GitHub. PID https://github.com/rstodden/DEPlain.

Toborek, Vanessa and Busch, Moritz and Boßert, Malte and Bauckhage, Christian and Welke, Pascal. 2023. *Simple German Corpus*. GitHub. PID https://github.com/buschmo/Simple-German-Corpus.

A. Hyperparamter mT5

parameter name	value
epochs	10
model	mt5-base
prefix	"simplify to plain German: "
max length	128:128
learning rate	0.001
batch size	4
metric	SARI
optimzer	adafactor

Table 10: Hyperparameter for fine-tuning mT5

B. Hyperparameter Sockeye

C. Reproduction results of sockeye-APA-LHA

D. Reproduction results of BLOOM

E. Reproduction results of customer-decoder-ats

F. Reproduction Results of mBART-DEplain-APA and mBART-DEplain-APA+web

G. EASSE-DE settings

• lowercasing: False, • tokenizer: spacy, • test set: custom, • metrics: bleu,sari,bertscore,fre • language: DE

	Sockeye-APA-LHA	Spring et al. (2021)
Sockeye version	3.1.34	< 2.3.17
num_layers	6	6
optimized_metric	'bleu'	'bleu'
max_num_checkpoint_not_improved	10	10
checkpoint_improvement_threshold	0.001	
seed	42	1
batch_type	'sentence'	word
batch_size	256	2048
optimizer	'adam'	'adam'
max_seq_len	95	95
label_smoothing	0.3	0.3
transformer_model_size	512	512
transformer_attention_heads	4	4
transfromer_feed_forward_num_hidden	2048	2048
transformer_dropout_attention	0.1	0.1
transformer_dropout_act	0	0
transformer_dropout_prepost	0.1	0.1
embed_dropout	0.3	0.3
transformer_positional_embedding_type	'fixed'	'fixed'
initial_learning_rate	0.0002	0.0002
learning_rate_reduce_factor	0.9	0.9
learning_rate_schedule_type	'plateau-reduce'	'plateau-reduce'
update_interval	1	2
vocabular size	20000	20000
init		xavier
Init-scale		3
Init-xavier-factor-type		avg
architecture		transformer

Table 11: Hyperparameters of our reproduction and the ones reported in Spring et al. (2021).

	copied		**reproduced**	
System	**BLEU↑**	**SARI↑**	**BLEU↑**	**SARI↑**
Sockeye-APA-LHA	12.3	40.73	11.40	45.20

(a) APA-LHA OR-B1

	copied		**reproduced**	
System	**BLEU↑**	**SARI↑**	**BLEU↑**	**SARI↑**
Sockeye-APA-LHA	15.20	42.04	14.15	52.17

(b) APA-LHA OR-A2

Table 12: Reproduced and copied results for sockeye on APA-LHA (Spring et al., 2021).

H. Hyperparameter mBART

I. Overview of BERT-Score Precision per model and test set

J. Overview of SARI results per model and test set

	TCDE19 (n=25)		**GEOlino (n=25)**	
	copied	**repro.**	**copied**	**reprod.**
System	**SARI↑**	**SARI↑**	**SARI↑**	**SARI↑**
zero-shot	32.26	34.96	29.59	28.75
random 10-shot	38.07	35.49	35.42	36.92
similarity 10-shot	38.93	39.86	39.7	40.36
Identity Baseline	15.42	15.42	27.45	27.44
Truncate Baseline	26.81	26.81	30.7	30.74

Table 13: Reproduced and copied results for BLOOM. The identity baseline results are taken from the code, all other copied scores are taken from the original paper (Ryan et al., 2023)
.

	BLEU↑	**SARI↑**	**ROUGE-L↑**
german_gpt FT	4.8	42.74	17.93

(a) 20Minuten (copied)

	BLEU↑	**SARI↑**	**ROUGE-L↑**
german_gpt FT	4.12	41.85	17.23

(b) 20Minuten (reproduced)

Table 14: Reproduced and copied results for 20Min and custom-decoder-ats (Anschütz et al., 2023).

System	**BLEU↑**	**SARI↑**	**BS_P↑**	**FRE↑**
mBART-DEplain-APA	28.25	34.818	0.639	63.072
mBART-DEplain-APA+web	28.506	34.904	0.64	62.669
Identity baseline	26.89	15.25	0.63	58.75

(a) copied

System	**BLEU↑**	**SARI↑**	**BS_P↑**	**FRE↑**
mBART-DEplain-APA	30.01	39.12	0.48	-
mBART-DEplain-APA+web	29.62	34.44	0.47	-
Identity baseline	28.50	15.88	0.45	-

(b) reproduced & EASSE

System	**BLEU↑**	**SARI↑**	**BS_P↑**	**FRE↑**
mBART-DEplain-APA	28.49	38.72	0.64	65.3
mBART-DEplain-APA+web	28.03	33.81	0.64	65.2
Identity baseline	26.89	15.25	0.63	59.23

(c) reproduced & EASSE-DE

Table 15: Reproduced and copied results for mBART-DEplain-APA and mBART-DEplain-APA+web (Stodden et al., 2023) on DEplain-APA.

System	**BLEU↑**	**SARI↑**	**BS_P↑**	**FRE↑**
mBART-DEplain-APA	15.727	30.867	0.413	64.516
mBART-DEplain-APA+web	17.88	34.828	0.436	65.249
Identity baseline	20.85	11.931	0.423	60.825

(a) copied

System	**BLEU↑**	**SARI↑**	**BS_P↑**	**FRE↑**
mBART-DEplain-APA	14.41	33.15	0.20	-
mBART-DEplain-APA+web	18.95	34.11	0.25	-
Identity baseline	21.65	12.34	0.23	-

(b) reproduced & EASSE

System	**BLEU↑**	**SARI↑**	**BS_P↑**	**FRE↑**
mBART-DEplain-APA	13.5	33.11	0.4	69.65
mBART-DEplain-APA+web	17.99	34.07	0.44	69.05
Identity baseline	20.85	11.93	0.42	62.95

(c) reproduced & EASSE-DE

Table 16: Reproduced and copied results for mBART-DEplain-APA and mBART-DEplain-APA+web (Stodden et al., 2023) on DEplain-web.

	Rios et al. (2021)	Rios et al. (2021)	Trienes et al. (2022)	Stodden et al. (2023)	Stodden et al. (2023)
model	standard mbart	small mbart	mBART-large-cc25	mBART	long-mbart
model-url			facebook/mbart-large-cc25	facebook/mbart-large-cc25	facebook/mbart-large-cc25
max length	1024:1024	1024:4096		256:256	2048:1024
learning rate			0.00003	0.00003	0.00003
lr_schedule_type	'plateau-reduce'	'plateau-reduce'		'plateau-reduce'	'plateau-reduce'
batch size	1024:1024	4	4	16	1
optimizer			adamW	adam	adam
warm-up			10% of train + linear decay		
beam size			5	6	6
vocabulary size	250k	20k		35k	35k
attention window	x	512		512	512
attention dropout	0.1	0.1		0.1	0.1
dropout	0.3	0.3		0.3	0.3
label smoothing	0.2	0.2		0.2	0.2
early stopping	rougeL	rougeL		rougeL	rougeL
language tags	de_DE:[de_A1|de_A2|de_B1]		yes, but not specified	de_DE:de_SI	de_DE:de_SI

Table 17: Hyperparamters of mBART in different papers.

	APA-LHA-OR-B1		**APA-LHA-OR-A2**		**DEplain-APA**		**DEplain-web**		**SGC**		**GEOlino**		**TCDE19**		**AVG rank**	
	BS_P	**Rank**	**BS_P**	**Rank**	**BS_P**	**Rank**	**BS_P**	**Rank**	**BS_P**	**Rank**	**BS_P**	**Rank**	**BS_P**	**Rank**	**(5 sets)**	**(7 sets)**
hda_LS	0.12	10	0.15	10	0.55	6	*n/a*	11	0.25	8	0.76	2	0.45	3	9	7.14
sockeye-APA-LHA	**0.35**	**1**	**0.32**	**1**	0.37	10	0.13	9	0.13	10	0.15	8	0.14	9	6.2	6.86
sockeye-DEplain-APA	0.25	4	0.25	8	0.53	8	0.24	8	0.33	9	0.19	7	0.18	8	7.4	7.43
mBART-DEplain-APA	0.23	8	0.26	6	**0.64**	**2**	0.4	3	0.31	4	0.74	3	0.5	2	4.6	4
mBART-DEplain-APA+web	0.23	8	0.25	8	**0.64**	**2**	**0.44**	**1**	0.33	3	**0.79**	**1**	**0.55**	**1**	4.4	**3.43**
mT5-DEplain-APA	0.24	6	0.26	6	0.61	3	0.36	6	0.3	6	0.65	4	0.4	5	5.4	5.14
mT5-SGC	0.28	3	0.29	2	0.48	9	0.37	5	0.37	2	0.55	6	0.38	6	4.2	4.71
BLOOM-zero	0.19	9	0.21	9	0.53	8	0.35	7	0.25	8	0.59	5	0.42	4	8.2	7.14
BLOOM-10-random	0.24	6	0.26	6	0.57	5	0.39	4	0.3	6	*n/a*	11	*n/a*	11	5.4	7
BLOOM-10-similarity	0.29	2	0.27	3	0.57	5	0.42	2	**0.38**	**1**	*n/a*	11	*n/a*	11	**2.6**	5
custom-decoder-ats	0.08	11	0.07	11	0.16	11	0.1	10	0.06	11	0.08	9	0.22	7	10.8	10

Table 18: Overview of BERT-Score Precision values per model and test set including ranks per test set. The last two columns contain the averages across all test sets (n=7) and all test sets with available training data (n=5).

	APA-LHA-OR-B1		**APA-LHA-OR-A2**		**DEplain-APA**		**DEplain-web**		**SGC**		**GEOlino**		**TCDE19**		**AVG rank**	
	SARI	**Rank**	**SARI**	**Rank**	**SARI**	**Rank**	**SARI**	**Rank**	**SARI**	**Rank**	**SARI**	**Rank**	**SARI**	**Rank**	**(5 sets)**	**(7 sets)**
hda_LS	14.02	11	15.49	11	26.06	11	*n/a*	11	20.22	11	34.2	4	26.92	9	11	9.71
sockeye-APA-LHA	**51.77**	**1**	**44.93**	**1**	40.16	3	32.41	8	35.5	6	18.94	9	29.87	8	3.8	5.14
sockeye-DEplain-APA	40.32	2	39.4	2	**44.14**	**1**	36.24	4	24.71	3	31.79	7	37.86	7	**2.4**	**3.71**
mBART-DEplain-APA	30.28	8	30.94	8	38.72	5	33.11	7	32.77	8	**44.29**	**1**	**39.14**	**1**	7.2	5.43
mBART-DEplain-APA+web	25.89	10	26.61	10	33.81	10	34.07	6	29.8	10	44.28	2	37.37	2	9.2	7.14
mT5-DEplain-APA	34.47	6	35.7	6	39.41	4	**37.15**	**1**	35.92	5	36.93	3	35.09	3	4.4	4
mT5-SGC	39.79	3	39.36	3	37.92	6	36.56	3	43.62	2	28.75	6	32.98	5	3.4	4
BLOOM-zero	26.83	9	27.56	9	35.43	9	30.58	10	31.95	9	32.15	5	34.96	4	9.2	7.86
BLOOM-10-random	33.05	7	32.43	7	35.93	8	30.9	9	33.16	7	n/a	11	n/a	11	7.6	8.57
BLOOM-10-similarity	38.05	4	37.22	5	41.27	2	37.03	2	**44.66**	**1**	n/a	11	n/a	11	2.8	5.14
custom-decoder-ats	37.05	5	37.59	4	36.42	7	34.92	5	36.53	4	22.05	8	32.87	6	5	5.57

Table 19: Overview of SARI scores per model and test set including ranks per test set. The last two columns contain the averages across all test sets (n=7) and all test sets with available training data (n=5).