An Approach Towards Unsupervised Text Simplification on Paragraph-Level for German Texts

Leon Fruth, Robin Jegan, Andreas Henrich
University of Bamberg
An der Weberei 5, 96047 Bamberg, Germany
{leon.fruth, robin.jegan, andreas.henrich}@uni-bamberg.de

Abstract

Text simplification as a research field has received attention in recent years for English and other languages, however, German text simplification techniques are lacking thus far. We present an unsupervised simplification approach for German texts using reinforcement learning (self-critical sequence training). Our main contributions are the adaption of an existing method for English, the selection and creation of German corpora for this task and the customization of rewards for particular aspects of the German language. In our paper, we describe our system and an evaluation, including still present issues and problems due to the complexity of the German language, as well as directions for future research.

1. Introduction

Automatic text simplification (ATS) is a research field in computational linguistics. The objective of text simplification is the modification of texts in a way to make them simpler to read and understand for the target audience. Thus, helping people with low literacy levels, mentally impaired people and children (Al-Thanyyan and Azmi, 2022; Evans et al., 2014; Watanabe et al., 2009). It is closely related to other natural language processing (NLP) tasks such as text summarization.

With the advancements in deep learning, recent research addresses ATS as a mono-lingual machine translation problem (Mallinson et al., 2020): Translating a text with complex linguistic properties into a text with simple linguistic properties in the same language. For this, large-scale simplification datasets are needed. Such parallel datasets are not widely available for most languages, including German.

This work uses the approach from Laban et al. (2021) (referred to as *KiS* in this work) and adapts it to the German language. The approach bypasses the need for parallel datasets by using training based on reinforcement learning (RL) (Sutton and Barto, 2018) and rewards regarding the criteria simplicity, meaning preservation and fluency, that are jointly optimized. Since German text simplification data is limited, the dependency on a large parallel simplification dataset is circumvented using this training method. This work presents the first unsupervised ATS approach for German and one of the first, to the authors' knowledge, that simplifies on a paragraph-level. Source code, model, datasets and evaluation data are available under https://github.com/LFruth/unsupervised-german-ts.

2. Background

Linguistic complexity, a key objective in ATS, consists of lexical simplicity, replacing difficult words with simpler expressions (Carroll et al., 1998; Laban et al., 2021; Keskisärkkä, 2012), and syntactic simplicity, rewriting texts into simpler and more understandable sentences (Saggion, 2017; Alva-Manchego et al., 2019).

ATS can also be addressed through the lens of machine translation (MT), where a complex text is translated into a text of the same language with simpler linguistic properties (Coster and Kauchak, 2011; Specia, 2010).

With the introduction of transformer-based models and large-scale parallel simplification corpora such as WikiLarge (Zhang and Lapata, 2017) and Newsela (Xu et al., 2015) new approaches like ACCESS (Martin et al., 2019) have been proposed. For instance, ACCESS (Martin et al., 2019) presents a sentence simplification methodology wherein the authors introduced a parametrization mechanism to control the compression rate, the paraphrase amount, and the strength of lexical and syntactic simplification. While there also exist some larger datasets for Spanish text simplification (Agrawal and Carpuat, 2019), other languages, including German, only have very limited parallel datasets that are mostly insufficient to train a simplification model in a MT fashion (Naderi et al., 2019; Battisti et al., 2020; Rios et al., 2021; Säuberli et al., 2020; Spring et al., 2021).

An early approach for German used rule-based simplification (Suter et al., 2016), whereas another method chose a zero-shot cross-lingual technique, that was implemented to handle the lack in datasets (Mallinson et al., 2020). Reinforcement learning is applied in unsupervised models, e.g. in Zhang

Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024, pages 77–89

and Lapata (2017), using the framework REINFORCE (Williams, 1992), also deployed in Nakamachi et al. (2020) with an LSTM encoder-decoder model. Newer approaches such as Anschütz et al. (2023) using style-specific pre-training also work on the lack in parallel data.

3. Method

In the following section, we describe our model architecture GUTS, short for **G**erman **U**nsupervised **T**ext **S**implification. We followed the work of KiS from Laban et al. (2021) and adapted it to simplify German paragraphs. We used the same training method k-SCST, an extension of self-critical sequence training (SCST) (Rennie et al., 2017), which is based on the REINFORCE algorithm (Williams, 1992).

3.1. Architecture

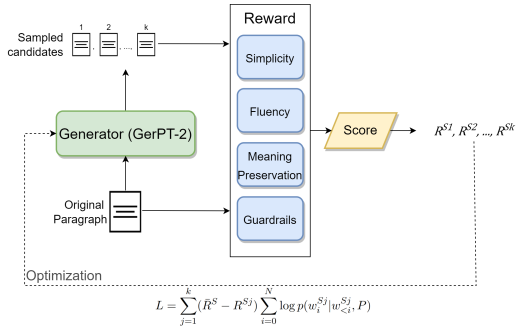

Figure 1: GUTS Learning architecture with k-SCST

Figure 1 displays how the generator learns: First k simplification candidates are sampled from the generator model, conditioned on an original paragraph. These candidates are then scored according to the reward. From the resulting rewards $R^{S1}, R^{S2}, ..., R^{Sk}$ the mean reward $\bar{R}^S$ is calculated as a baseline for the loss. The loss is computed as the difference between individual candidate rewards and the baseline, with candidates having rewards above the baseline contributing more to optimization. The probability of generating a word $p(w_i^{Sj}|...)$ is conditioned on the input paragraph P and previously generated words $w_{<i}^{Sj} = w_1^{Sj}, ...w_{i-1}^{Sj}$.

3.2. Rewards

The approach used in this work – adopted from (Laban et al., 2021) – can be described as a non-differentiable reward maximization problem. For each original paragraph P and its corresponding generated simplification S, scores in the range of $[0, 1]$ are obtained for simplicity, meaning preservation, fluency, and some guardrails. These individual reward scores are then combined into a single reward using a scoring function.

$$R = \sum_{i=0}^{N} W_i \log(s_i) \tag{1}$$

Here R denotes the total reward for a simplification. N is the number of individual scores of the reward, and s_i describes an individual score with its assigned weight W_i. This way, not every score has the same impact on the overall reward. A drawback of this scoring function is that the guardrail scores cannot zero the score since $log(0)$ is undefined. To work around this, these scores are either set to 0.0001 or 0.9999 instead.

We used the reward scores from Laban et al. (2021) and adapted them to the German language. In the following, only the scores that function differently are explained. Small changes and adaptations of the other scores are outlined in A.2.

3.2.1. Meaning Preservation

To measure how well the meaning is preserved in the generated simplification, a novel approach is presented. First, each sentence from the simplification S is aligned to the most similar sentence from the original paragraph P using sentence transformer representations (Reimers and Gurevych, 2019). By aligning these sentences, operations like sentence splitting are considered. The aligned sentences are then compared and scored with BERTScore (Zhang et al., 2020), to make use of contextual similarity between them and consider synonymity. For every sentence of P, the F1 BERTScore is computed for the aligned sentences of S. $s_{meaning}$ is calculated as follows:

$$\frac{\sum_{(se^P, se^S) \in aligned} F_{BERT}(se^P, se^S)}{|aligned| + |unaligned|} \tag{2}$$

where $aligned$ denotes the set of aligned sentence pairs from the original paragraph and the system's simplification, with the original se^P and simplified sentence se^S. The sum of the F1 BERTScores of each sentence-pair is divided by the number of aligned sentences $|aligned|$ and unaligned sentences $|unaligned|$. The set of unaligned sentences contains original and simplified sentences that were not semantically related to another sentence. Sentences from P that had no matching simplification sentence are penalized because it is assumed that information was lost during simplification. Unaligned simplified sentences that were not semantically related to any original sentence are also penalized since they are assumed to contain unnecessary or hallucinated content.

3.2.2. Hallucination Detection

A common problem for text generation tasks like ATS or summarization are factual inconsistencies. An important requirement for these tasks is that the facts from the generated text match the source text (Fischer, 2021), also referred to as faithfulness (Cao et al., 2018). In this work, we only focus on detecting the addition of named entities. First, all named entities from the generated simplification are extracted. Second, the BERTScore library (Zhang et al., 2020) is used to obtain the words from P with the highest similarity to each extracted entity. Next, the similarity value from the most related word in the original paragraph is selected for each detected entity. This value is then compared against a threshold. If the BERTScore similarity falls below this threshold, a hallucination is detected and the score $s_{hallucination}$ returns 0. Otherwise, it returns 1. Figure 3 in the appendix shows an example of the described score.

3.2.3. Article Repetition Penalty

To counter the cheating of the language model fluency score described in section A.2.3, another guardrail score was introduced that detects and penalizes the repetition of German articles like “der”, “die”, “das”. This score was introduced for this approach since the generator was abusing the repetition of high probable articles to artifically increase $s_{fluency}$. The score is set to 0 if three or more articles appear in a sequence, else it returns 1.

4. Experiments

For the generator, a German version of the medium GPT-2 model GerPT-2 (Minixhofer, 2020) was used for the experiments. More details about the training process are presented in the appendix.

4.1. Data

To test and tune the parameters of the reward scores two datasets have been used as a reference. We used the *TextComplexityDE dataset* (Naderi et al., 2019), which contains a total of 1019 sentences with simplifications, and a manually collected dataset of parallel articles from the website “Gemeinnützige Werkstätten und Wohnstätten” (Gemeinnützige Werkstätten und Wohnstätten - GWW, 2023). The latter is referred to as the *GWW dataset* and was created for this work. The GWW dataset was manually created by the authors for this work by aligning original articles with their simplified versions from the website. The dataset consists of 52 parallel articles, mainly texts for disabled people containing information and help about topics like work or living.

For the training of the generator, a dataset of short paragraphs extracted from Wikipedia articles has been generated. The raw Wikipedia articles were extracted from German Wikipedia dumps.

For the evaluation we used a dataset based on TextComplexityDE that was manually assembled, where the authors combined individual sentences to create 52 paragraphs. Besides the TextComplexityDE dataset, 300 paragraphs from the training dataset of Wikipedia articles have been randomly selected. This subset contains articles that are linguistically more diverse and difficult than those from the TextComplexityDE dataset, but have no reference simplification.

4.2. Evaluation

Since there were no comparable German models available that can simplify on a paragraph-level, a *Pivot model* is introduced for evaluation, consisting of two machine translation models (Tiedemann and Thottingal, 2020) and one simplification model (Laban et al., 2021). This Pivot model is inspired by a similar model introduced by (Mallinson et al., 2020), which the authors used as a comparison in their evaluation. First, the paragraph is translated from German to English (de-en). The KiS model can then simplify the English paragraph, before it is translated back to German by the second translation model (en-de).

Because there is no single agreed-upon measurement for simplicity (Alva-Manchego et al., 2021), a combination of reference-based and reference-less metrics has been used. SARI was integrated as a reference-based simplification metric. SARI showed the best correlation with human judgements on simplicity gain compared to other automatic metrics (Alva-Manchego et al., 2021). To measure the syntactic simplicity, the *Flesch Reading Ease* (FRE) for German has been used (Amstad, 1978). The mean FRE of the models' outputs *FRE(S)* and the average difference between the FRE value of the original text and the simplification, referred to as *FRE diff*, are calculated. For measuring the lexical simplicity improvement *Zipf diff*, the difference of the average Zipf values of all non-stop words between the original paragraph P and the simplification S are calculated. The score $s_{meaning}$ is used to capture the meaning adequacy of the simplifications. With this score, the models are rated on how well the contents from the original paragraph are preserved. Lastly, the compression rate (*Comp.*) is measured.

Table 1 displays the automatic results on the adapted TextComplexityDE dataset and on the Wikipedia paragraphs. On the TextComplexityDE dataset GUTS is slightly outperformed by the Pivot model on SARI. Both models improve on FRE and achieve, arguably, therefore syntactic simplification.

TextComplexityDE						
Model	SARI	FRE(S)	FRE diff	Zipf diff	Meaning	Comp.
manual reference	-	46.847	21.194	0.274	0.896	0.933
GUTS	0.348	37.448	11.795	0.059	**0.875**	0.789
Pivot	**0.370**	**38.712**	**13.059**	**0.206**	0.727	0.863
Wikipedia Paragraphs						
GUTS	-	**53.130**	**9.376**	-0.001	**0.819**	0.731
Pivot	-	50.187	6.402	**0.243**	0.549	0.766

Table 1: Automatic results of TextComplexityDE and Wikipedia

The Pivot model outperforms GUTS on both metrics. Both models performed reasonably well on meaning preservation. GUTS even comes close to the reference baseline, since it was directly trained on this score. The Pivot model lags behind in this area, indicating that its simplifications did not capture as much information from the original paragraph, according to $s_{meaning}$. All models tend to shorten the texts during simplification, shown by the compression values.

The evaluation on the Wikipedia paragraphs is performed with only reference-less metrics. GUTS achieves better FRE values than the pivot model for this dataset, but has worse results on the Zipf scores, showing no gain for this metric. GUTS achieves the best meaning preservation scores on this dataset.

To further evaluate the performance of GUTS, a limited manual evaluation has been conducted outlined in the following section.

4.3. Observations

In the following, the simplifications produced by the models are manually evaluated. Note that these are observations by the authors, focusing on simplification phenomena and common problems with GUTS. This is done to guide future work to improve the system.

4.3.1. Simplification Phenomena

With GUTS, some lexical simplifications in the form of substitutions with synonyms could be observed, but most of the examples were not necessarily simpler. Sometimes words that do not exist in German were used as substitutes. Many lexical changes in simplifications were not synonyms but involved shortening of words. A part of a composed word was deleted during simplification and the rest was kept. This sometimes resulted in arguably simpler words without changing the content of the text. For instance, GUTS replaced the word “Schlossräume” (English: “palace rooms”), with a Zipf value of 1.08, with “Räume” (English: “rooms”) with a value of 4.4, indicating a lexical simplification that did not significantly change the meaning of the sentence. Most of these word shortenings removed important information from the sentence and result in a misleading simplification. For example: The word “Präsidentenflugzeug” (English: “presidential plane”) was reduced to only “Präsident” (English: “president”).

For structural changes of the paragraphs rarely any sentence splittings were observed with GUTS or the Pivot model. Both models tend to delete parts of the text to make shorter sentences rather than splitting them. In many observations the arguably most important statement of the sentence is preserved. Deletions can help the reader understand texts better by removing non-essential information that may be confusing to a low literacy reader.

4.3.2. Problems

Guaranteeing the fluency and readability of a text is one of the most critical aspects of natural language generation tasks such as text simplification. One big limitation of GUTS were non-fluent text and grammatical mistakes in the generated simplifications that occurred in most of the evaluated outputs. Many of these were minor errors, like confusing German articles, e.g. using “das” instead of “der” or making mistakes with the tense of a word, for example, using the present instead of past tense. GUTS regularly produced some of the previously mentioned grammatical issues but rarely had completely incoherent outputs. The Pivot model showed the least amount of grammatical mistakes.

Another common issue were problems with faithfulness. Factual inconsistencies between source and generated texts were frequently observed with the simplifications of GUTS. One of the most common inconsistencies were numeric values, such as dates or measurements. These inconsistencies with numbers were not considered by any scoring method for the reward. For future work, the score for hallucination detection $s_{hallucination}$ could be extended to take numbers and dates into account, like Laban et al. (2021) did in their approach. The results of the Pivot model rarely contained the faithfulness issues from above. However, it rather introduced new sentences or phrases to the simplification, that were hallucinated.

5. Discussion

To bypass the data scarcity for German text simplification datasets, this work showed the first unsupervised text simplification approach for the German language. Furthermore the system is able to simplify on a paragraph-level. While many simplification phenomena happen on a paragraph-level (Alva-Manchego et al., 2019), most of the previous research on ATS has been performed on a sentence-level.

Another contribution in this work has been the novel hallucination detection method. This method is arguably implemented more dynamically than the implementation in KiS (Laban et al., 2021, §3.4.2), which directly matches the named entities in the source text and the generated text. However, their score also identifies false and hallucinated numeric values that our scoring function $s_{hallucination}$ could not do.

The meaning preservation score in this work is also a novel contribution. The score in this work presents a combination of sentence alignment and similarity measuring using BERTScore (Zhang et al., 2020), in order to rate how well the content of the original paragraph is preserved in the generated simplification.

Different problems and limitations were detected during the analysis of the rewards, the conducted experiments, and the evaluation of GUTS. For further exploration of the approach presented in this work, different parameters and settings need to be explored. Also, the individual reward scores should be investigated and improved further. GUTS lacked lexical and syntactic simplification phenomena, e.g. simpler vocabulary or sentence splitting.

Grammatical mistakes and non-fluent samples during the experiments were also an issue in this work. This is one of the most important criteria and needs to be reliable for an ATS system. Unfortunately, there is no research for measuring fluency of German texts to the authors' knowledge.

Non-factual content in the produced simplifications was another dominant issue with GUTS. This limitation is an ongoing research field for text generation tasks, such as summarization (Fischer, 2021; Cao et al., 2018; Falke, 2019). The GUTS model regularly generated simplifications with incorrect numbers and dates. Furthermore, the models sometimes even introduced hallucinations to the simplifications, which led to disinformation.

We hope that future research addresses the problems and challenges identified in this work by building upon this contribution.

6. Bibliographical References

Sweta Agrawal and Marine Carpuat. 2019. Controlling text complexity in neural machine translation. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 1549–1564. Association for Computational Linguistics.

Suha Al-Thanyyan and Aqil M. Azmi. 2022. Automated text simplification: A survey. *ACM Comput. Surv.*, 54(2):43:1–43:36.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2019. Cross-sentence transformations in text simplification. In *Proceedings of the 2019 Workshop on Widening NLP@ACL 2019, Florence, Italy, July 28, 2019*, pages 181–184. Association for Computational Linguistics.

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

Toni Amstad. 1978. *Wie verständlich sind unsere Zeitungen?* Abhandlung: Philosophische Fakultät I. Zürich. 1977. Studenten-Schreib-Service.

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.

Giuseppe Attardi. 2015. Wikiextractor. `https://github.com/attardi/wikiextractor`.

Alessia Battisti, Dominik Pfütze, Andreas Säuberli, Marek Kostrzewa, and Sarah Ebling. 2020. A corpus for automatic readability assessment and text simplification of german. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 3302–3311. European Language Resources Association.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

Hunter M Breland. 1996. Word frequency and word difficulty: A comparison of counts in four corpora. *Psychological Science*, 7(2):96–99.

Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2018. Faithful to the original: Fact aware neural abstractive summarization. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4784–4791. AAAI Press.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. Practical simplification of english newspaper text to assist aphasic readers. *Proc. of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*.

William Coster and David Kauchak. 2011. Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, Portland, Oregon, USA. Association for Computational Linguistics.

Richard Evans, Constantin Orasan, and Iustin Dornescu. 2014. An evaluation of syntactic simplification rules for people with autism. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations, PITR@EACL 2014, Gothenburg, Sweden, April 27, 2014*, pages 131–140. Association for Computational Linguistics.

Tobias Falke. 2019. *Automatic Structured Text Summarization with Concept Maps*. Ph.D. thesis, Darmstadt University of Technology, Germany.

Tim Fischer. 2021. *Finding Factual Inconsistencies in Abstractive Summaries*. Ph.D. thesis, Universität Hamburg.

Gemeinnützige Werkstätten und Wohnstätten - GWW. 2023. Gemeinnützige Werkstätten und Wohnstätten - GWW. https://www.gww-netz.de/de/. Last visited on May 22, 2023.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Robin Keskisärkkä. 2012. Automatic text simplification via synonym replacement. Master's thesis, Linköping UniversityLinköping University, Department of Computer and Information Science, Faculty of Arts and Sciences.

Philippe Laban, Andrew Hsi, John F. Canny, and Marti A. Hearst. 2020. The summary loop: Learning to write abstractive summaries without examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 5135–5150. Association for Computational Linguistics.

Philippe Laban, Tobias Schnabel, Paul N. Bennett, and Marti A. Hearst. 2021. Keep it simple: Unsupervised simplification of multi-paragraph text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 6365–6378. Association for Computational Linguistics.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

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 2020, Online, November 16-20, 2020*, pages 5109–5126. Association for Computational Linguistics.

Louis Martin, Benoît Sagot, Éric de la Clergerie, and Antoine Bordes. 2019. Controllable sentence simplification. *CoRR*, abs/1910.02677.

Benjamin Minixhofer. 2020. GerPT2: German large and small versions of GPT2.

Babak Naderi, Salar Mohtaj, Kaspar Ensikat, and Sebastian Möller. 2019. Subjective assessment of text complexity: A dataset for german language. *CoRR*, abs/1904.07733.

Akifumi Nakamachi, Tomoyuki Kajiwara, and Yuki Arase. 2020. Text simplification with reinforcement learning using supervised rewards on grammaticality, meaning preservation, and simplicity. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 153–159, Suzhou, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese

BERT-Networks. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Steven J. Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1179–1195. IEEE Computer Society.

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 *Third Workshop on New Frontiers in Summarization*, pages 152–161. ACL Anthology.

Horacio Saggion. 2017. *Automatic Text Simplification*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Andreas Säuberli, Sarah Ebling, Martin Volk, Nuria Gala, and Rodrigo Wilkens. 2020. Benchmarking data-driven automatic text simplification for german.

Lucia Specia. 2010. Translating from complex to simplified sentences. In *Computational Processing of the Portuguese Language, 9th International Conference, PROPOR 2010, Porto Alegre, RS, Brazil, April 27-30, 2010. Proceedings*, volume 6001 of *Lecture Notes in Computer Science*, pages 30–39. Springer.

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.

Julia Suter, Sarah Ebling, and Martin Volk. 2016. Rule-based automatic text simplification for german. In *Proceedings of the 13th Conference on Natural Language Processing, KONVENS 2016, Bochum, Germany, September 19-21, 2016*, volume 16 of *Bochumer Linguistische Arbeitsberichte*.

Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction*, second edition. The MIT Press.

Jörg Tiedemann and Santhosh Thottingal. 2020. OPUS-MT - building open translation services for the world. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation, EAMT 2020, Lisboa, Portugal, November 3-5, 2020*, pages 479–480. European Association for Machine Translation.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. Facilita: Reading assistance for low-literacy readers. SIGDOC '09, page 29–36, New York, NY, USA. Association for Computing Machinery.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8:229–256.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Trans. Assoc. Comput. Linguistics*, 3:283–297.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 584–594. Association for Computational Linguistics.

A. Appendix

A.1. Datasets and Preprocessing

The German Wikipedia dump[1] from the 21st of January 2022 was downloaded and processed as follows:

1. The dump is preprocessed into articles using WikiExtractor (Attardi, 2015).
2. Empty articles are removed.
3. Articles are split into individual paragraphs at each new line ("\n"), resulting in 10.8 million paragraphs.
4. The paragraphs are further cut down into a length between 80 and 175 tokens.

The resulting dataset consists of 1,080,000 paragraphs with an average number of 4.6 sentences and 93.8 words. The data is available in the Github repository. Even though the transformer models used for this approach can handle a sequence length of at least 512 tokens, the paragraphs are cut down to a maximum of 175 tokens. This has been done to speed up each training step and limit the GPU memory consumption by the models.

The TextComplexityDE contains 23 articles split into sentences with their corresponding simplification. Since this approach aims to simplify on a paragraph-level, the individual sentences from the same Wikipedia articles were manually combined to form paragraphs. Notably, in some occasions the sentences in the composed articles were not logically sequential. While the GWW dataset contains simpler simplifications than TextComplexityDE, the information contained in the complex article and its simplification differs more. For tuning the individual reward scores the TextComplexityDE dataset and the GWW dataset have been used as a reference (see table 2 for more details).

A.2. Reward Scores

A.2.1. Lexical Simplicity

For determining lexical simplicity, the approach from Laban et al. (2021) has been used. The score relies on the observation that word frequency and difficulty are correlated (Breland, 1996). First, we strip all stop-words from the texts, as they should not be considered. Next, all remaining words are lemmatized, to have a more accurate comparison between morphologically different words with the same base form. Two sets of words are created: One set contains all words that have been removed, and one set with words that have been added during simplification. The most complex or to be precise least frequent 15% of words from both of these sets are kept, all other words are filtered out. Then the average Zipf value for each set is computed: $\overline{Zipf}_{add}$ for the added words and $\overline{Zipf}_{rem}$ for the removed words. With these values, the lexical shift $shift_{lexical}$ between the simplification S and the original paragraph P can be calculated. The score is clipped between 0 and 1 and has a ramp shape, where the score $s_{lexical}$ falls off when achieving a $shift_{lexical}$ above the target value of 0.8. An example is given in Figure 2.

A.2.2. Syntactic Simplicity

To measure the readability of their generator's output Laban et al. (2021, §3.1.1) used the readability metric FKGL. Since there was no German adaption for this metric, the adaption FRE was chosen for this score instead (Amstad, 1978). Short sentences with short words are scored well with these metrics. The objective is to reward the model for generating shorter sentences. For the syntactic score $s_{syntactic}$, the approach from KiS has been adapted. Laban et al. (2021, §3.1.1) argue that an already syntactically simple paragraph should not require any further simplification and define the target FKGL conditioned on the original paragraph's FKGL score. To calculate the score we use the same scoring function as for $s_{lexical}$.

A.2.3. Language-Model Fluency

Again, we follow the work of KiS which is based on Lau et al. (2017) showing that grammaticality of a text can be measured by observing a language models probability. The score was constructed by taking the likelihood of the original and simplified paragraph:

$$s_{fluency} = \left[\frac{\lambda + LM(P) - LM(S)}{\lambda}\right]^{+} \quad (3)$$

where $LM(P)$ and $LM(S)$ stand for the likelihood of the original and simplified paragraph that are obtained by a masked language model. If the loss of a generated simplification $LM(S)$ is higher than $LM(P)$ by λ or more, $s_{fluency}$ is set to 0. The score is clipped between 0 and 1; if $LM(S)$ is above or equal to $LM(P)$, the score is 1 otherwise the score is a linear interpolation between 0 and 1. (Laban et al., 2021, §3.2.1) For more details on the model and training used for this score see section A.3.2.

Unfortunately, adapting the LM-Fluency score $s_{fluency}$ to the German language came with new problems: The reward seemed to encourage shorter and more probable words, especially articles like "der", "die", "das" (English: "The"). This

[1] https://dumps.wikimedia.org/dewiki/

Dataset	parallel articles	avg number of sentences		avg number of words	
		original	simplification	original	simplification
TextComplexityDE	23	11.00	23.43	286.48	282.52
GWW	52	5.52	8.98	82.31	67.29

Table 2: Statistics of reference datasets TextComplexityDE and GWW

might be because articles are relatively frequent words and therefore overall very probable in German, which results in a smaller loss. It was found that just adding repeating articles to a text often decreases the overall loss of a text, therefore scoring it as more fluent. To mitigate this problem the Article Repetition Penalty was employed for this, see section 3.2.3.

A.2.4. Discriminator Fluency

The Language-Model Fluency score can be limiting as it is static and deterministic (Laban et al., 2021, §3.2.2). Therefore it can be exploited by the generator. To counter this we incorporate a score s_{discr} based on a dynamic discriminator which they used in KiS. In this case, the generator simplifies the examples and the discriminator tries to predict if a given paragraph is a generated simplification or an original paragraph written by a human. During the generator's training process, both the simplification outputs and the original paragraphs are added to the discriminators training buffer. The original paragraphs are assigned a label of 1, and the generator outputs a label of 0. When the buffer reaches n samples, the discriminator is trained and the buffer is emptied again. More details are available in section A.3.2.

A.2.5. Brevity

The brevity guardrail is a score that ensures that the length of a generated simplification falls into the range of the original paragraph. The brevity score was configured to return 0.9999 if $0.6 \leq C \leq 1.3$, otherwise it returns 0.0001.

A.3. Training Details

A.3.1. Generator

A German version of the medium GPT-2 model GerPT-2 (Minixhofer, 2020) with 345M parameters was used for the generator. The training was performed on a workstation with 64 GB of RAM, an I9-9900K processor, and two RTX 2080 Ti GPUs with 11GB memory. All training tasks performed in this work used Automatic Mixed Precision (AMP) to save memory during training and increase the speed. For optimization, AdamW was used (Loshchilov and Hutter, 2017). For experiment tracking and visualization, Weights & Biases has been utilized (Biewald, 2020).

First, the model was pre-trained on the *copy task*. Using this task, the generator learns to output an exact or close copy of the input. This is a good baseline to start the simplification process. When the generator was trained for too long on the copy task, the sampled simplification candidates during simplification training were often too similar or even an exact copy of the original text. This low diversity resulted in very similar rewards, which limited the training signal for the generator. For the *copy task*, the training script from the Summary Loop Github repository has been used (Laban et al., 2020). The generator was fine-tuned with a learning rate of $2 \cdot 10^{-5}$, with a batch size of eight examples. The model was trained on this task for about 1800 training steps (25 minutes).

For the simplification training with k-SCST a learning rate of $4 \cdot 10^{-5}$ was chosen. A batch-size of one example was applied, meaning after sampling and scoring $k = 8$ simplification candidates conditioned on one original paragraph, the generator is then optimized. The simplifications were sampled using nucleus sampling with $p = 0.95$, combined with a top-K value of $K = 5$. Additionally a setting suppressing the repetition of 5-grams in a sequence was employed during sampling to avoid repeating phrases. The p value was chosen based on the research of Holtzman et al. (2020). They argue that values between 0.9 and 1 are the most reliable, and lower values tend to generate repetitions. The value $K = 5$ was selected relatively low, as it produced the most reliable results considering the meaning preservation, hallucination and brevity scores in the beginning. In retrospect, the top-K value may have been chosen too low, limiting the diversity of the candidates and restricting the nucleus sampling capabilities.

Our main model GUTS was trained for over 110,000 steps (roughly five days). Table 3 shows how the reward scores during training were weighted.

	GUTS
$s_{lexical}$	0.5
$s_{syntactic}$	3.0
$s_{meaning}$	4.0
$s_{fluency}$	0.5
s_{discr}	0.5
$s_{brevity}$	1.0
$s_{hallucination}$	1.0
s_{ngram}	1.0
$s_{articles}$	1.0

Table 3: Score weights used for training

A.3.2. Fluency Models

The model used for $s_{fluency}$ is a German BERT base model[2], with 110M parameters. It was fine-tuned on Wikipedia articles to better capture the linguistic properties of the domain. The model was trained for roughly 20,000 steps using AdamW (Loshchilov and Hutter, 2017) as an optimizer with a learning rate of 10^{-5} and a batch size of eight examples.

The Discriminator for the score s_{discr} was trained on a buffer consisting of original paragraphs and generated simplifications, collected during the training process. When the buffer reaches 4000 samples, the discriminator is trained with the data. Afterwards the buffer gets emptied again. The same German BERT base model mentioned in the previous paragraph is used here again. It is trained using 90% of the training buffer for the discriminator. The discriminator is trained for five epochs. The end of each epoch is used as a checkpoint, where the discriminator model is saved along with the F1 performance tested on the last 10% of the training buffer. The best model of the five checkpoints is kept as the new discriminator until the training buffer reaches 4000 samples again. The model is trained using AdamW (Loshchilov and Hutter, 2017) as an optimizer with a learning rate of 10^{-5}, a batch size of 6 and AMP.

A.3.3. Further models

For the meaning preservation score $s_{meaning}$ a fine-tuned distilbert model[3], with 66M parameters, has been used for the sentence alignment. For calculating the BERTScore for $s_{meaning}$ and $s_{hallucination}$ the same German BERT base model described in section A.2.3) was utilized. For the NER tagging in $s_{hallucination}$ another finetuned distilbert model[4] was applied.

[2] https://huggingface.co/dbmdz/bert-base-german-cased

[3] https://huggingface.co/sentence-transformers/msmarco-distilbert-multilingual-en-de-v2-tmp-lng-aligned

[4] https://huggingface.co/Davlan/distilbert-base-multilingual-cased-ner-hrl

A.4. Simplification examples

A.4.1. Simplification example 1

Original Halvarsson gewann bei der Junioreneuropameisterschaft 1979 in Achenkirch die Bronzemedaille im Slalom. Die ersten Punkte in einem Weltcuprennen holte er am 10. März 1980 als Zehnter des Slaloms von Cortina d'Ampezzo. Im selben Jahr gewann er den schwedischen Riesenslalom-Meistertitel. In den zwei folgenden Wintern fuhr er im Weltcup je zwei Mal unter die besten 15, während er beim Slalomrennen der Weltmeisterschaft 1982 ausschied. In der Saison 1982/83 gelangen ihm sechs Platzierungen unter den besten 15, darunter ein sechster Platz in Kranjska Gora.

GUTS Hvar gewann bei der Junioren Europameisterschaft 1979 in Achenkirchen die Bronzemed Medaille im Slalom. Der ersten Punkte in einem Wettkampf holte er am 10." Bis zum Zehnten des Slaloms" von Cortina d'Ampezzo. Beim selben Jahr gewann er die schwedischen Riesenslalom. In den zwei Jahren fuhr er im Weltcup jeweils zwei Mal unter die guten 15, während er bei beim Slalom gegeneinander ausschied. In den Saison 1982/83 gelang ihm sechs Platzierungen unter dem besten 15, darunter einen sechster Platz in Ku.

Pivot Halvarsson gewann die Bronzemedaille bei der Junioren-Europameisterschaft 1979. Er war einer der ersten Spieler, der die Slalombarriere durchbrach, und er gewann die ersten beiden Punkte in einem Rennen am 10. März 1980. Zwei Jahre zuvor hatte er den schwedischen Riesen-Slalom gewonnen, aber dieses Mal war er am Rande des Sieges. In den folgenden zwei Wintern gewann er die schwedische Meisterschaft, einschließlich des Slaloms, und zog sich in der Saison 1982/83.

A.4.2. Simplification example 2

Original Mizuo Peck wuchs in New York auf und stand bereits in jungen Jahren auf der Theaterbühne. Sie besuchte die Theaterabteilung der State University of New York und machte ihr Filmdebüt im Jahr 2000 in dem deutschen Fernsehfilm "Nicht heulen, Husky" neben Heiner Lauterbach und Barbara Rudnik. Bekannt wurde sie aber vor allem durch ihre Rolle als Sacajawea in der dreiteiligen Filmreihe "Nachts im Museum" zwischen 2006 und 2014.

GUTS Mizuo Pecks wuchs in New York und stand bereits in den jungen Jahren auf der Bühne. Sie besuchte die Schauspielabteilung der State University von New York und machte sich ihr Filmdebüt. Bekannt wurde sie doch vor allem durch ihre Rollen als Sacajawe in der dreiteiligsten Filmreihe "Nacht" zwischen 2006 und 2015.

Pivot Mizuo Peck wurde in New York geboren und hatte bereits in jungen Jahren eine Karriere als Theaterdarstellerin. Sie besuchte die Theaterabteilung der State University of New York und debütierte Ende 2000 in einer deutschen TV-Serie mit dem Titel "Nicht heulen, Husky". Doch bald wurde sie in ihrem eigenen Film Sacajawea zur Vollzeitdarstellerin,

A.4.3. Simplification example 3

Original König Dom Manuel I. nahm ab dem Ende des 15. Jahrhunderts weitreichende Neugestaltungen in der Stadt Sintra vor. Ab dem 16. Jahrhundert nahm die Beliebtheit Sintras bei den Oberschichten zu, und adlige Familien errichteten im Kreis Sintra einige Herrenhäuser. Das Erdbeben von Lissabon 1755 rief auch in Sintra erhebliche Zerstörungen hervor, denen in der Folge umfangreiche Arbeiten des Wiederaufbaus folgten. Ende 18. Jahrhundert entstand mit der Textildruckerei und -färberei "Fábrica de Estamparia de Rio de Mouro" die erste industrielle Einrichtung im Kreis.

GUTS König Dom Manuel I nahm ab dem Ende der 15. Jahrhundert weitreichende Neugestaltungen. In der Stadt Sintra ab dem 16. Jahrhundert gab die Beliebtheit Sintras. Ab dem 16 Jahr nahm die Beliebtheit Sintra bei den Oberschichten bei, und adlige Familie errichteten im Kreis Sint. Das Erdbeben von Liss 1755 rief auch noch in Sintra erhebliche Schäden hervor, denen in den Folge umfangreiche Arbeiten des Aufbaus folgten. Ende 19. Jahrhundert entstand mit dem Textilddruckerei und -firberei "Fébrica de Estaparia de Rio de" die erste industrielle Organisation im Kreis.

Pivot Seit dem Ende des 15. Jahrhunderts wurde Sintra umfassend renoviert. Ab dem 16. Jahrhundert war die Stadt für ihre hohe Lebensqualität bekannt geworden. Von dort aus begannen Adelsfamilien, im Kreis Sintra Villen zu bauen, die eine große Anzahl von Geschäften und Restaurants umfassten. Das Erdbeben in Lissabon im Jahr 1755 verursachte auch erhebliche Schäden, was zu umfangreichen Wiederaufbauarbeiten führte. Ende des 18. Jahrhunderts wurde die erste Industrieanlage in der Gegend

Original

Seit einigen Jahren finden Rasiermesser jedoch auch zunehmend im **Privatbereich** wieder eine **wachsende** Verwendung. Die Klinge muss vor jeder Rasur auf einem **Streichriemen abgeledert** und in regelmäßigen Abständen nachgeschliffen werden, um die **Schärfe** der Schneide zu erhalten.

Simplification

Seit einigen Jahren werden auch zuhause **öfter** Rasiermesser **benutzt**. Die Klinge muss vor jeder Rasur auf einem **Lederriemen abgestrichen** werden. In regelmäßigen Abständen muss die Klinge nachgeschliffen werden, damit sie **scharf** bleibt.

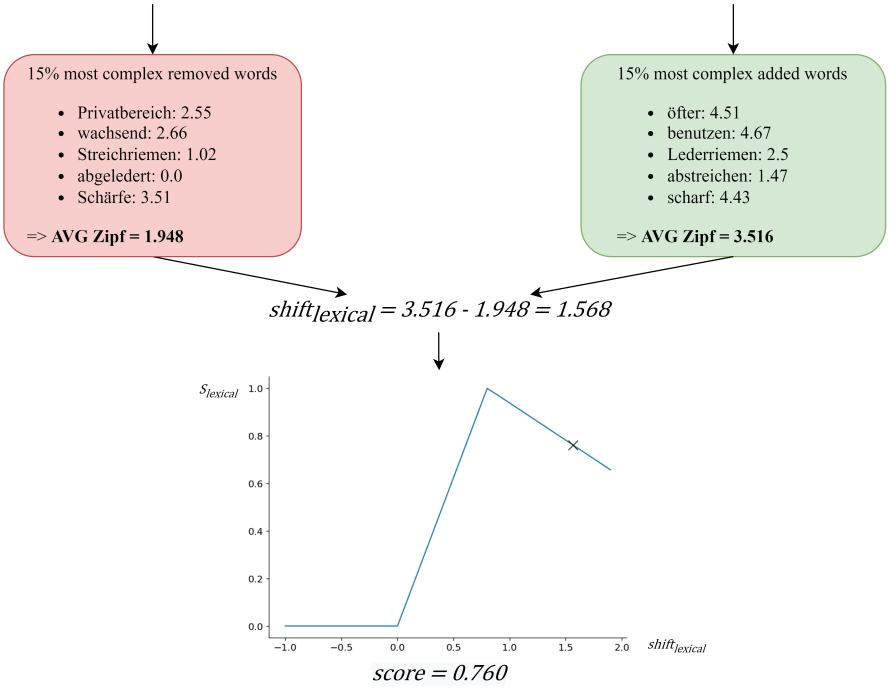

Figure 2: Example for the calculation of $s_{lexical}$

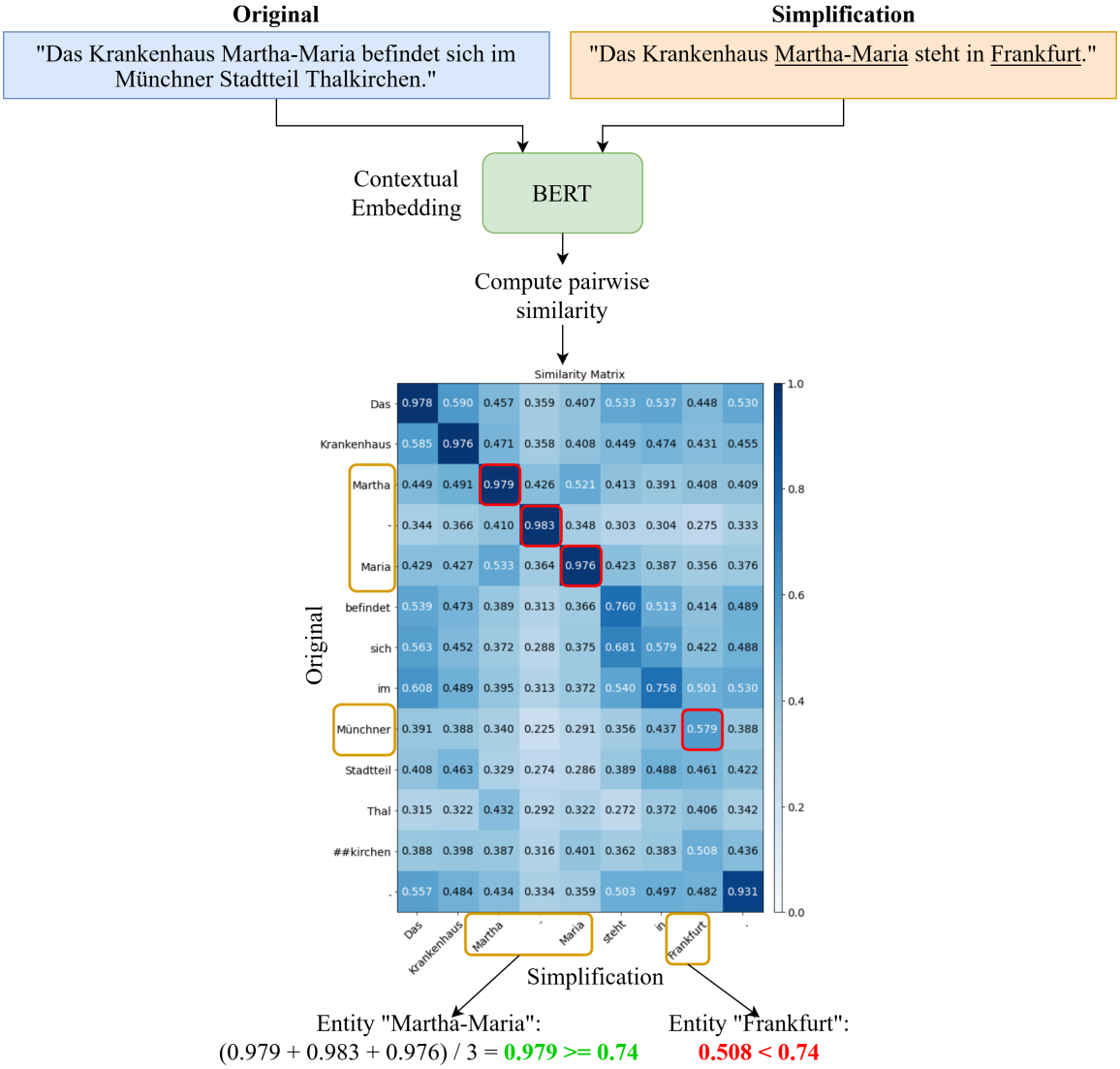

Figure 3: Hallucination detection algorithm. The confusion matrix is calculated with BERTScore (Zhang et al., 2020) using the original text and the simplification. Then, the entities in the simplification are detected: In this case “Martha-Maria” and “Frankfurt”. For each of the entities the highest similarity value in the matrix is selected. If the value is below the threshold of 0.74, it is assumed that a hallucination is present.