Universität Zürich^UZH

Seminar
Media Accessibility
Fall term 2025

Highlighting Simplified Text

Author: Corina Julia Raclé
Student ID: 21-932-603

Course instructor: Prof. Dr. Sarah Ebling
Department of Computational Linguistics

Submission date: 04.01.2026

Abstract

Text simplification has enabled wider access to complex information for target groups ranging from individuals with mental impairments to older adults. Affective captions have proven successful in aiding comprehension for Deaf or Hard-of-Hearing individuals. Combining the typographic modulations, also called highlights, used in affective captioning with text simplification is a promising approach for further increasing the comprehensibility of simplified text. In an initial investigation of this topic, this paper explores both a pipeline and an end-to-end approach for the generation of highlighted simplified text. Special consideration is given to accessibility, computational cost and the environmental impact of the models.

1 Introduction

Simplified language is valuable to diverse target groups, ranging from individuals with mental impairments to older adults. It allows access to complex texts and other media that would otherwise be difficult to understand. Simplified language is not standardized and continuously being refined and revised. In this paper, we attempt to add another aspect to simplified language: automatically adding typographic modulations that have proven successful in affective captions (de Lacerda Pataca et al., 2024).
This paper represents a first step towards investigating the usefulness and efficiency of such typographic modulations. To suitably narrow the scope, we focus specifically on bolding and enlarging headings and bolding keywords or phrases in text. Collectively, these typographic changes are referred to as highlighting. Highlighting is added directly to texts via Markdown, as illustrated with an example from SimpleGerman V2.0 (Battisti et al., 2020) in Figure 1.

Original Text

Bauen und Umbauen
Barrierefrei gestaltete Lebensbereiche sind eine der maßgeblichen Voraussetzungen für eine selbstbestimmte und gleichberechtigte Teilhabe von Menschen mit Behinderungen am Leben in der Gesellschaft.

Highlighted Simple Text

```
## Barrierefreie Orte für alle!
**Ziel:** Jeder Mensch soll ohne Hindernisse in Gebäude eintreten und sich dort wohlfühlen können. Das bedeutet barrierefreies Bauen!
```

Figure 1: Example of an original text and corresponding highlighted simple text

To explore the generation of highlighted simplified text, two approaches were designed. Both approaches simplify the problem by assuming there are no distinctions between different levels of simplification.
The first approach is a **pipeline**, where the output of a model trained for simplification is fed into a secondary model trained to highlight simplified text. A pipeline approach is a traditional methodology in NLP and remains useful even in the current era of end-to-end approaches, for example for a task that requires complex processing or step-wise quality control. From this point onward, the model trained to simplify text is referred to as the *simplifier* and the model trained to highlight as the *highlighter*.
The second approach is an **end-to-end** model, such as are used in contemporary automatic speech recognition (ASR) systems. The *end-to-end model* receives the original text and is trained to directly produce highlighted simplified text. While this approach may appear more challenging, it has been shown to be highly effective, since the model is able to learn both tasks in parallel and capture interactions (Desot et al., 2022). Interactions between the two objectives of highlighting and simplifying text may lead to both influencing each other to produce a more unified output, tailored to target groups.

The objective of this paper is to investigate whether a pipeline or end-to-end approach is preferable for generating highlighted simplified text, with careful consideration of computational and environmental resources.

Following this introduction, a review of related work will be presented. The methods section describes the models, dataset, and experimental setup, while the experiments and results section presents evaluation metrics and findings. This is followed by a discussion and limitations, then a conclusion with future outlook. Appendices provide all prompts used and additional technical details.

2 Related Work

Simplified language has been a topic of research that is slowly gaining momentum. While currently no global definitions or standards exist, several guidelines have been developed. Simplified language is considered a variety of standard language with low lexical and syntactic complexity and a clearly structured layout. Additionally, it includes definitions of complex terms and assumes a reduced world knowledge in readers, which distinguishes it from simplified summarization or paraphrasing (Bredel and Maaß, 2016).

Understanding the full meaning and intention of a speaker purely from transcribed text can often be challenging. Kim et al. (2023, p. 1) have succinctly summarized the problem: "*'how' something is said determines the full meaning of 'what' is being said*". Emphasizing keywords or keyphrases is not only useful for improving comprehension, but can even help disambiguate the meaning of a text. Table 1 demonstrates an example of how the meaning of a sentence can change depending on which word is emphasized.

Sentence	**Interpretation**
I never said you did that.	It wasn't me who said that.
I never said **you** did that.	It wasn't you I said did that.
I never said you did **that**.	It wasn't that specific thing I said you did.

Table 1: Changes in sentence meaning depending on emphasis

Adding an additional dimension to texts through typography facilitates understanding and allows readers to interpret meaning and intention more easily. Expressive or affective captions serve as an excellent example of this principle. Affective captions are captions with special typographic modulations designed to convey emotions more effectively than standard captions. This topic has been investigated in depth by de Lacerda Pataca et al. (2024), who identified two specific design styles preferred by Deaf or Hard-of-Hearing individuals for affective captioning. The most effective designs combined font-color modulations to represent valence with either font-weight or font-size modulations to convey arousal.

3 Methods: Pipeline vs End-to-End Approach

3.1 Models and Architectures

The end-to-end, simplifier and highlighter models are based on Gemma-2-9B-Instruct (Google, 2024) with merged LoRA adapters (Hu et al., 2021) fine-tuned for their specific purpose. Gemma-2-9B-Instruct is a decoder-only transformer model with 9B parameters, instruction-tuned, and achieving 71.3% accuracy on MMLU. It is easily accessible via HuggingFace or GitHub.

Several considerations influenced the choice of this relatively small model. Within the limited computational resources of a seminar project, SOTA or large models were not feasible. Moreover, since the goal of this paper is to compare two approaches rather than achieve high quality, a smaller model is even preferable. A SOTA model would likely perform so well in both approaches that no meaningful differences could be discerned.

The use of LoRA adapters was motivated by computational cost and environmental concern (Dauner and Socher, 2025). By training only 3 epochs with a learning rate of 1e-4, almost 4.5M parameters were updated, which corresponds to 0.05% of the total 9B parameters of Gemma-2-9B-Instruct.

The end-to-end model, as well as the simplifier and highlighter components in the pipeline approach, all use the same basic Gemma-2-9B-Instruct architecture combined with LoRA adapters (see Figure 2).

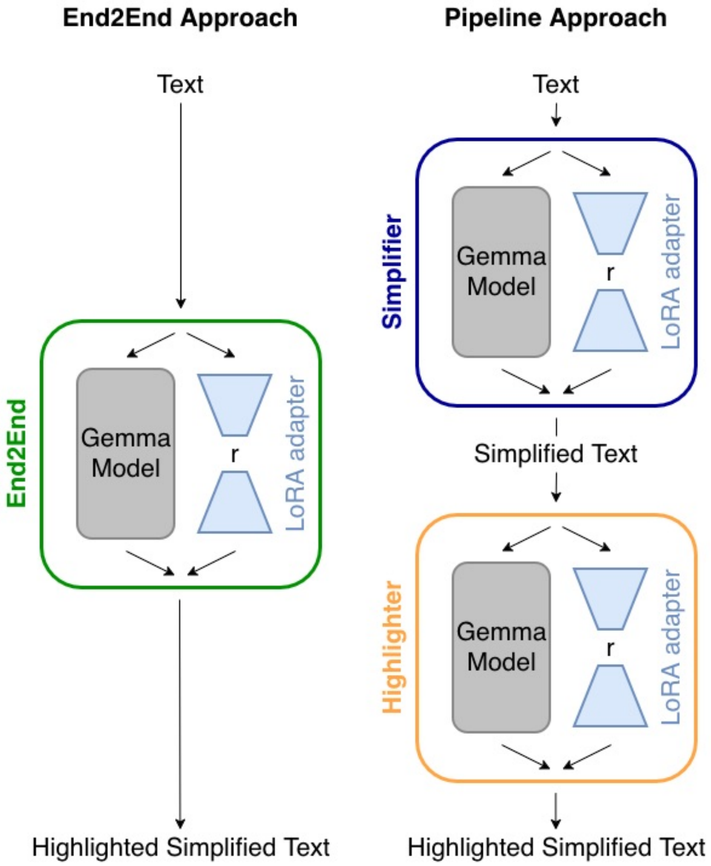

Figure 2: The architecture of both approaches.

3.2 Data

We were kindly provided with the dataset SimpleGerman V2.0, introduced in Battisti et al. (2020). Under the condition of including some type of typographic information, a subset of 379 document pairs of original and corresponding texts in simplified language were extracted from this corpus. These texts are mostly sourced from websites of non-profit organizations, specialized institutions and government agencies. They cover 92 domains, such as politics, culture and health. The dataset includes annotations for headings and bolded keywords and -phrases. This makes it a prime candidate for training models to learn simplification and higlighting behaviour.

Unfortunately, this dataset was not suitable for fine-tuning a small model directly. The original and simplified texts do not align well enough, neither at the paragraph nor document level. The length of some texts posed another problem (a significant portion of the text exceeded 40,000 characters). Additionally, the typographic annotations were insufficient for effective training of highlighting.

To address these issues, a synthetic dataset was created. SOTA models have shown an acceptable performance in simplifying language in Deilen et al. (2023). Within the the scope of this paper, highlighting can be framed as keyword extraction, which is a task SOTA models excel in. Again in consideration of computational and environmental cost, the model Leo-mistral-hessian-ai-7b-chat (Björn Plüster and Christoph Schuhmann in collaboration with LAION and HessianAI, 2023) was chosen to generate highlighted simplified texts given suitable original texts from SimpleGerman V2.0 (prompts are included in Appendix A). This model is trained on German text and has a context length of 32k tokens which suits the original texts well. Texts of excessive length, with preexisting annotation errors or other noise were excluded, which resulted in 250 original and highlighted simplified text pairs. This dataset was directly used to train the end-to-end model.

For the pipeline approach, separate training sets were constructed. The original texts were paired with their simplified counterparts, with highlights removed, for the training of the *simplifier*. These simplified texts without highlights were then paired with highlighted simplified texts to train the *highlighter*.

3.3 Code Availability

All code used for the generation of the synthetic dataset and the training and evaluation of the models is publicly available.[1]

4 Experiments and Results

4.1 Experiment Design

Given the small size of our dataset and models, we did not expect high quality. The models have indeed often failed to adhere to the task. Frequently they paraphrased or summarized instead of simplifying the original text, and there were even cases where model diverged entirely from the given task, starting to translate to English for example. Nonetheless, high quality of outputs was not necessary for the comparison of the two approaches. It did however complicate the evaluation of the models considerably. A common problem in evaluation of text simplification is that reference and model output may diverge severely but still both be of good quality. This is a problem even with SOTA models. Typically, it is mediated by using several references, which increases complexity of evaluation and the scale of the dataset significantly. While the limited scope of this project does not allow for this approach, automatically computed metrics still provide valuable information since the two approaches of the experiment were designed to be comparable rather than optimal.

[1] GitHub repository: `https://github.com/c-racle/Highlighting-Simplified-Text`

4.2 Automatic evaluation metrics

Two commonly used simplified language evaluation metrics are SARI and FKGL.
SARI was introduced in Xu et al. (2016) as one of the first comprehensive evaluation metrics for simplified language. SARI specifically addresses the problem that models often just paraphrase instead of simplify. Unlike other metrics, SARI not only compares the system output with the references, but also the input, which is necessary to determine whether the model simplified or just paraphrased. For simplification, a system will need to delete certain parts of the input, keep others and add some parts that were not in the input. SARI specifically rewards these changes.
The Flesch-Kinaid Grade Level (FKGL) was developed to evaluate readability of training material for the US Navy(Kincaid et al., 1975) and is based on relating text difficulty to US school grades. As explained by Ehara (2024), the popularity of FKGL mainly stems from being easily interpreted and very robust due to the simplistic approach of combining the average sentence length with the average number of syllables in a word.
Evaluating the highlighting of texts proved difficult, suffering from the same evaluation difficulties as simplified language does. The markdown nature of the highlights does however allow for computing partial span overlaps.
Both approaches were evaluated on SARI, FKGL and partially overlapping spans for evaluation of highlights on a test set of 27 examples, as shown in Table 2. The pipeline approach outperforms the end-to-end approach in SARI. Notably, competitive systems usually score above 40 (Xu et al., 2016). The FKGL scores all range in a university graduate level (Prieto et al., 2025), but there is some simplification visible comparing input and model output. The end-to-end approach outperforms the pipeline by two points in this regard. The highlighting scores are predictably low, but it is still notable that the pipeline approach outperforms the end-to-end approach by 30 times.

Dataset	SARI	FKGL Input	FKGL Model	FKGL Reference	Markdown F1 (bold+headings)
End-to-End	0.387	17.15	12.11	8.73	0.0044
Pipeline	0.409	17.15	10.18	8.73	0.1331

Table 2: Evaluation of simplification performance on End2end and Pipeline test sets.

The FKGL metric is very vulnerable to artificial manipulation, as shown by Tanprasert and Kauchak (2021). In this experiment, FKGL scores could be manipulated while affecting other metrics only minimally, proving that optimizing FKGL scores is possible without improving simplification. Therefore, FKGL is not reliable enough for simplification evaluation.
Generally, automatic metrics will struggle to capture exact model performance on such a small data set as is used in this paper without even considering the challenging nature of simplification evaluation. Highlighting, as defined in this paper, is equally difficult to evaluate and additionally does not have a specific established automatic metric that would allow for comparison with other projects.
Consequently, only automatically computed metrics do not seem to be a sufficient evaluation of the performance of the two approaches. The gold standard of evaluation is a detailed human evaluation by the target groups, which was unfortunately not possible due to the limited scope of this paper. A very limited version of rating is preference voting by LLMs, which we used to complement the above computed automatic metrics.

4.3 Preference voting

Lacking human resources, preference voting is best be done completely by SOTA models, but this approach was beyond the computational cost limit of this paper. Reducing the scope of this evaluation method, the two models already introduced were used as well as GPT-4.1 (OpenAI, 2023) as a SOTA model. The models were asked to rate which highlighted simplified text they preferred in three categories: purely regarding simplification, purely regarding highlighting and in combination. Details on the prompts used are provided in Appendix C and in the accompanying code repository.
As shown in Table 3, the pipeline approach is preferred in 62.32% of cases. In both the *Simplified Language* and *Combined* categories, its outputs are chosen roughly twice as often as those of the end-to-end model. In contrast, in the *Highlighting* category, the two approaches perform more comparably.

Category	**End-to-End**	**in %**	**Pipeline**	**in %**
Simplified Language	25	36.23%	44	63.77%
Highlights	31	44.93%	38	55.07%
Combined	22	31.88%	47	68.12%
In total	78	37.69%	**129**	**62.32%**

Table 3: Preference voting on a test set of 27 texts.

To gain more detailed insight into the models' voting behavior, Figure 4.3 presents their preferences for each test-set example. The non-uniform distribution highlights the challenges of evaluating performance on tasks as difficult as simplification and highlighting, particularly when the underlying outputs are of relatively low quality.

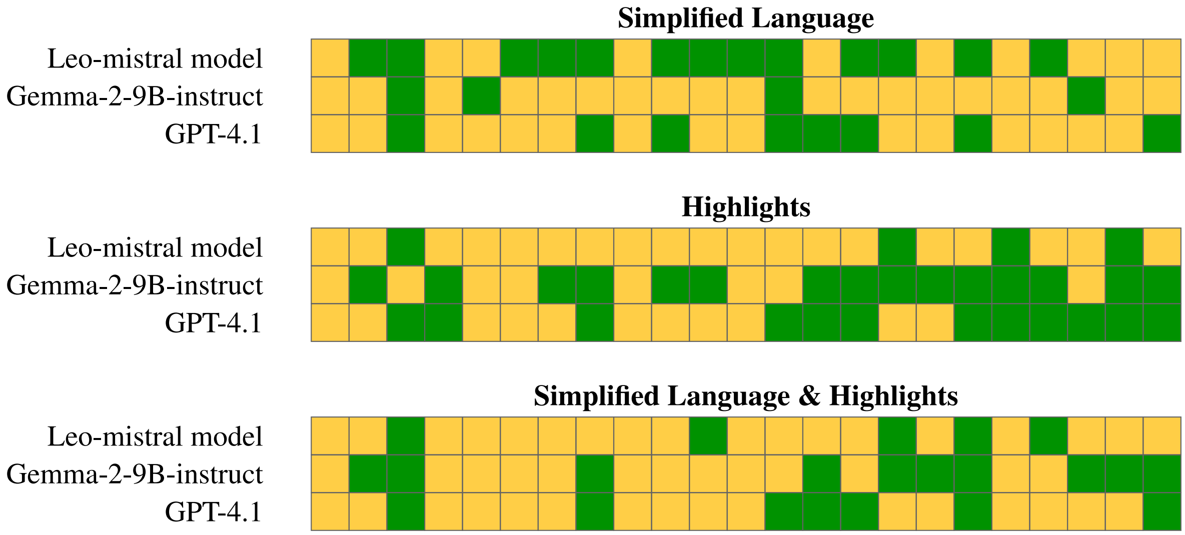

Figure 3: Preference voting behavior of the three LLMs.

Legend 🟩 End2End 🟨 Pipeline

5 Discussion

Against our initial expectations, the pipeline approach proved superior in this experiment. The pipeline approach performed especially well in simplification and in producing highlighted simplified text, whereas its advantage was less decisive in pure highlighting performance.

This outcome indicates that pipeline approaches continue to have practical relevance in modern, small-scale tasks. However, it must also be emphasized that this experiment was conducted on a very limited scale and level of complexity, employing small models and a very small dataset. Results might differ substantially in larger-scale settings.

Our initial expectation was that the end-to-end approach would perform significantly better than the pipeline approach, paralleling trends in speech technology applications such as automatic speech recognition (Pham et al., 2019). In the ASR domain, end-to-end models led to dramatic improvements compared to earlier pipeline approaches, where the task was solved sequentially in multiple subtasks. This reflects a broader trend of pipeline approaches losing prominence relative to end-to-end models.

A larger model, especially a SOTA model, might well outperform a pipeline approach in the task of generating highlighted simplified text. Such a model would likely be able to make use of the dual learning required in the end-to-end approach much better, similar to how dual learning significantly improved performance in Desot et al. (2022). Presumably, jointly learning highlighting and simplification as one would produce highlighting that is specifically tailored to the needs of individuals that rely on simplified language. For instance, a model might choose to emphasize the connection to previous text sections over highlighting of keywords in the current section, which helps perceive the text as a whole instead of disconnected paragraphs.

Scaling up these experiments to a SOTA model would therefore most likely lead to a superior end-to-end system. In contrast, a small model trained on a small dataset would struggle to perform both tasks simultaneously, much less learn beneficial behaviors from doing so. Still, not all applications are capable of accessing or running a SOTA model, nor is such access necessary in many practical contexts. Especially considering recent research into the environmental effects of large models (Dauner and Socher, 2025), one should aim to use the smallest model that still provides satisfactory performance.

As demonstrated in this experiment, and especially in regard to small models, the traditional pipeline approach is still viable. A small model may perform sufficiently well in one task but a more complicated multifaceted task may prove too challenging. In this case, instead of employing an end-to-end model that struggles and is computationally costly, a sequence of small single-task models may be a more efficient and practical solution.

5.1 Limitations

A primary limitation of this seminar paper is the relatively small size of the dataset, consisting only of 379 original texts with corresponding simplified texts. Any further research on this topic will require a larger dataset, even if the domains remain the same. Scaling up the dataset could lead not only to improved performance overall but might also change the relative performance of the different approaches examined in this study.

The synthetic nature of the simplified and highlighted simplified texts is another significant limitation. Naturally, authentic highlighted texts in simplified language are far preferable to synthetic texts in the context of training a model intended for a realistic application. Without authentic data, these models have limited practical utility.

Another limitation is the size of the Gemma-2-9B-Instruct model, which was chosen for minimizing computational cost but it offers lower performance in general compared to larger models. Whilst sufficient for this experiment, a slightly larger model would likely lead to deeper insights into the highlighting and simplification processes. The trained models often struggled to adhere to their given tasks, despite their inherent instruction tuning and the fine-tuned LoRA adapters. Often, the texts were summarized more than simplified. The level of simplification also differed vastly, ranging from barely any simplification to losing significant pieces of information due to over-simplification. These problems would likely be minimized by employing a larger model.

The minimal evaluation is another limitation. Proper automatic evaluation of text simplification requires multiple reference solutions due to the nature of simplification, which was not feasible in the context of this seminar project. Evaluation of highlights would also vastly profit from this. Although preference voting is an efficient and informative method of evaluation, performing this by using LLMs instead of human participants is suboptimal. An application designed for a specific target group necessitates a human evaluation by members of said target group. Ideally, the human evaluation should even be more detailed than simple preference voting, evaluating the text across multiple categories and aspects would be hugely valuable.

6 Conclusion and Outlook

In this paper, we have explored two distinct approaches for the generation of highlighted simplified text. The end-to-end approach directly generates highlighted simplified text from a given input text, while the pipeline approach first employs a simplification model to produce simplified text, which is then processed by a highlighter model to finally output highlighted simplified text. For these experiments, LoRA adapters for Gemma-2-9B-Instruct were trained for simplification, highlighting and a combined process on a synthetic dataset consisting of original texts and corresponding simplified texts with and without highlights.

In this small-scale experiment, the pipeline approach proved slightly superior. It is anticipated that this would hold for other small datasets and small models, which are environmentally preferable compared to larger, more computationally expensive models. In contrast, a SOTA model would likely perform better in the end-to-end approach due to the increased ability to learn and interconnect multifaceted tasks simultaneously. For future research, a larger, human-created dataset is absolutely necessary. Employing a slightly larger model would also be beneficial, provided the computational cost increase is acceptable. Finally, a human evaluation, ideally involving members of the target groups, is necessary to evaluate models and ensure their real-world applicability.

Declaration of Use of Text Generation Models

GPT-4.1 (OpenAI, 2023) was extensively used in the fall semester of 2025 to generate code for dataset creation, model training and evaluation for this paper. Additionally, we used the model for help with LaTeX figures and management of references. No LLM was used to write the text of this paper, but GPT-5 mini was used for correction of grammar and stylistic errors.

References

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 Twelfth Language Resources and Evaluation Conference*, pages 3302–3311, Marseille, France. European Language Resources Association.

Björn Plüster and Christoph Schuhmann in collaboration with LAION and HessianAI. 2023. leo-mistral-hessianai-7b-chat. `https://huggingface.co/LeoLM/leo-mistral-hessianai-7b-chat`. Accessed: 2025-11-17.

Ursula Bredel and Christiane Maaß. 2016. *Leichte Sprache: Theoretische Grundlagen. Orientierung für die Praxis*. Duden, Berlin.

M. Dauner and G. Socher. 2025. Energy costs of communicating with ai. *Frontiers in Communication*, 10:1572947.

Caluã de Lacerda Pataca, Saad Hassan, Nathan Tinker, Roshan Lalintha Peiris, and Matt Huenerfauth. 2024. Caption royale: Exploring the design space of affective captions from the perspective of deaf and hard-of-hearing individuals. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

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.

Thierry Desot, François Portet, and Michel Vacher. 2022. End-to-end spoken language understanding: Performance analyses of a voice command task in a low resource setting. *Computer Speech Language*, 75:101369.

Yo Ehara. 2024. An analytical study of the flesch-kincaid readability formulae to explain their robustness over time. In *Proceedings of the 38th Pacific Asia Conference on Language, Information and Computation*, pages 989–997, Tokyo, Japan. Tokyo University of Foreign Studies.

Google. 2024. gemma-2-9b-it. `https://huggingface.co/google/gemma-2-9b-it`. Accessed: 2025-11-18.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

JooYeong Kim, SooYeon Ahn, and Jin-Hyuk Hong. 2023. Visible nuances: A caption system to visualize paralinguistic speech cues for deaf and hard-of-hearing individuals. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, New York, NY, USA. Association for Computing Machinery.

Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

OpenAI. 2023. Gpt-4.1. `https://platform.openai.com/docs/models/gpt-4`. Accessed: 2025-11-19.

Ngoc-Quan Pham, Thai Son Nguyen, Jan Niehues, Markus Müller, and Alexander H. Waibel. 2019. Very deep self-attention networks for end-to-end speech recognition. *ArXiv*, abs/1904.13377.

Javier Prieto, Dianelis Gonzalez Pupo, Raeven Grant, Ellie Mehrara, and Kymora B. Scotland. 2025. An assessment of the quality and readability level of online content on urinary tract infection treatment in spanish and english. *Translational Andrology and Urology*, 14(7).

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

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.

A Prompts for the Generation of the synthetic dataset

This prompt was used to simplify the original texts of SimpleGermanV2.0.

Simplification Prompt

```
<|im_start|>system
Du bist ein hilfreicher Assistent, der Texte bereinigt und
 ↪  vereinfacht, wobei wichtige Wörter hervorgehoben werden.
<|im_end|>
<|im_start|>user
Du erhältst einen Text.
Deine Aufgabe: Erstelle ein Textpaar mit einer sauberen Version des
 ↪  Originaltextes und einer vereinfachten Version dieses Textes.

Regeln:
- Das Textpaar besteht aus (orig, simple)
- Zuerst bereinige den Originaltext: Entferne alle Links,
 ↪  merkwürdige Formatierungen oder andere Artefakte aus der
 ↪  Vorverarbeitung. Es soll nur noch sauberer Text übrig bleiben,
 ↪  aber nichts umgeschrieben werden.
- Nun erstelle aus diesem sauberen Text einen entsprechenden
 ↪  vereinfachten Text.
- Der Text soll für Menschen mit kognitiven Beeinträchtigungen
 ↪  vereinfacht sein: Verwende sehr einfaches Vokabular, einfache
 ↪  Grammatik und wirklich sehr kurze Sätze.
- Jeder Satz sollte in einer neuen Zeile stehen.

WICHTIG:
- Gib nur ein JSON-Array aus.
- Format für das Textpaar genau wie folgt:

[
  {{"orig": sauberer Originaltext, "simple": vereinfachter Text}},
  ...
]

- Gib sonst nichts aus. Keine Erklärungen, keine Kommentare, kein
 ↪  zusätzlicher Text.
- Verwende doppelte Anführungszeichen für alle Schlüssel und
 ↪  String-Werte.
- Trenne mehrere Paare durch Kommata.

Originaler Text:
{orig}

Textpaar:
<|im_end|>
<|im_start|>assistant
```

This prompt was used to highlight the simplified text.

Highlighting Prompt

```
<|im_start|>system
Du bist ein hilfreicher Assistent, der Text in Markdown annotiert.
<|im_end|>
<|im_start|>user
Du erhältst einen Text.
Deine Aufgabe: Füge Markdown-Annotationen hinzu, ohne den Text
 ↪  inhaltlich zu verändern.

Anweisungen:
- Annotiere Überschriften im Text mit Markdown (#, ## oder ###).
- Markiere alle wichtigen Wörter oder Phrasen fett (**so**).
- Halte dich strikt an das Markdown-Format.
- WICHTIG: Ändere oder schreibe den Text nicht um!
- Füge nur die Markdown-Annotationen hinzu.
- Dein Output soll ausschließlich der originale Text mit den
 ↪  Markdown-Annotationen sein.
- Gib keine Erklärungen, Kommentare oder zusätzlichen Text aus.

Originaler Text:
{simple}

Textpaar:
<|im_end|>
<|im_start|>assistant
```

B Prompts used in the Pipeline and End-to-End Approaches

B.1 Prompts used in the Pipeline

This is the prompt given to the simplifier.

Simplifier Prompt

Du erhältst einen Text. Deine Aufgabe ist es, diesen Text zu bereinigen und zu vereinfachen.

- Zuerst bereinige den Originaltext: Entferne alle Links, merkwürdige Formatierungen oder andere Artefakte aus der Vorverarbeitung. Es soll nur noch sauberer Text übrig bleiben, aber nichts umgeschrieben werden.
- Nun erstelle aus diesem sauberen Text einen entsprechenden vereinfachten Text.
- Der Text soll für Menschen mit kognitiven Beeinträchtigungen vereinfacht sein: Verwende sehr einfaches Vokabular, einfache Grammatik und wirklich sehr kurze Sätze.
- Jeder Satz sollte in einer neuen Zeile stehen.
- Gib sonst nichts aus. Keine Erklärungen, keine Kommentare, kein zusätzlicher Text.

This is the prompt given to the highlighter.

Highlighter Prompt

Du erhältst einen Text. Deine Aufgabe ist es, Markdown-Annotationen hinzuzufügen, ohne den Text inhaltlich zu verändern.

- Annotiere Überschriften im Text mit Markdown (#, ## oder ###).
- Markiere alle wichtigen Wörter oder Phrasen fett (**so**).
- Halte dich strikt an das Markdown-Format.
- WICHTIG: Ändere oder schreibe den Text nicht um!
- Füge nur die Markdown-Annotationen hinzu.
- Dein Output soll ausschließlich der originale Text mit den Markdown-Annotationen sein.
- Gib keine Erklärungen, Kommentare oder zusätzlichen Text aus.

B.2 Prompt used in the End-to-End model

This is the prompt given to the end-to-end model.

End-to-End Prompt

Du erhältst einen Text. Deine Aufgabe ist es, diesen Text zu vereinfachen und mit Markdown zu annotieren.

- Zuerst bereinige den Originaltext: Entferne alle Links, merkwürdige Formatierungen oder andere Artefakte aus der Vorverarbeitung. Es soll nur noch sauberer Text übrig bleiben, aber nichts umgeschrieben werden.
- Nun erstelle aus diesem sauberen Text einen entsprechenden vereinfachten Text.
- Der Text soll für Menschen mit kognitiven Beeinträchtigungen vereinfacht sein: Verwende sehr einfaches Vokabular, einfache Grammatik und wirklich sehr kurze Sätze.
- Jeder Satz sollte in einer neuen Zeile stehen.
- Annotiere Überschriften im Text mit Markdown (#, ## oder ###).
- Markiere alle wichtigen Wörter oder Phrasen fett (**so**).
- Halte dich strikt an das Markdown-Format und füge nur die Markdown-Annotationen hinzu.
- Gib keine Erklärungen, Kommentare oder zusätzlichen Text aus.

C Prompt used in Preference Voting

This is the prompt used for all models in preference voting.

Preference Voting Prompt

```
        <|im_start|>system
Du bist ein hilfreicher Assistent, der zwei Texte in einfacher Sprache mit Markdow
<|im_end|>
<|im_start|>user
Du erhältst zwei Texte in einfacher Sprache. Die Texte enthalten:
- Markdown-Titel (#, ##, ###)
- Markdown-Fettmarkierungen (**) für wichtige Wörter oder Phrasen
Ausserdem erhältst du den originalen Text in üblicher Sprache und ohne Annotatione

Deine Aufgabe:
1. Bewerte, welcher Text die einfachere Sprache verwendet.
2. Bewerte, welcher Text die besseren Titel und wichtigen Wörter hat.
3. Bewerte, ob Text 1 dem Original ähnlich ist.
4. Bewerte, ob Text 2 dem Original ähnlich ist.
3. Bewerte, welchen Text du insgesamt besser findest.

Kriterien für die Bewertung:
- Sprache:
  - Sehr einfaches Vokabular und Grammatik
  - Sehr kurze Sätze
  - Jeder Satz steht in einer neuen Zeile
- Annotationen:
  - Titel sind kurz und sinnvoll
  - Wichtige Wörter sind korrekt markiert
  - Es sind nicht zu viele Wörter hintereinander fett markiert
- Gesamte Verständlichkeit für Menschen mit kognitiven Beeinträchtigungen

Instruktionen:
- Wähle für Sprache einen Text (Text 1 oder Text 2)
- Wähle für Annotationen einen Text (Text 1 oder Text 2)
- Bewerte, ob Text 1 dem Original ähnlich ist {TRUE oder FALSE}
- Bewerte, ob Text 2 dem Original ähnlich ist {TRUE oder FALSE}
- Wähle welchen Text du allgemein besser findest (Text 1 oder Text 2)
- Deine Ausgabe muss genau dieses Format haben:

AUSGABE:
Sprache: {Text 1 oder Text 2}
Annotationen: {Text 1 oder Text 2}
Allgemein: {Text 1 oder Text 2}
Text 1 ist dem Original ähnlich: {TRUE oder FALSE}
Text 2 ist dem Original ähnlich: {TRUE oder FALSE}

Begründung:
```

```
Hier sind die Texte:
{...}
<|im_end|>
<|im_start|>assistant
```