Creating an out-of-domain corpus with multi-level annotations: processing pipeline, Easy German NLP challenges, and licensing issues

Mehmet Koray Tulgar and Sarah Jablotschkin
University of Hamburg
mehmet.tulgar@studium.uni-hamburg.de

Abstract

In this paper, we report practical experiences and solutions while creating the corpus DE-Lite v2 that may support future work in the area of creating reusable annotated corpora of simplified German. Rather than introducing a new annotation model, we focus on documenting the methodological and technical infrastructure required to build such a resource at scale. In particular, the paper makes three principal contributions: First, we present a scalable and robust corpus-processing pipeline, built on state-of-the-art software, which produces CoNLL-U output including coreference annotation. The pipeline is designed to systematically address errors arising both from noisy raw data and from server-side failures. Secondly, we examine a range of normalisation, segmentation, and annotation difficulties encountered in our data, including challenges associated with harmonising heterogeneous pre-existing collections. Finally, we discuss the practical implications of licensing in corpus dissemination, when the goal is to provide full texts with annotation for the research community. As a concrete outcome of these efforts, a subset of the corpus has been made publicly available as DE-Lite v2.1 via the research data repository of the University of Hamburg.[1]

1 Introduction

Curated, linguistically annotated corpora (cf. Gries and Berez, 2017) have played a major role as training and testing data in Natural Language Processing (NLP). In the area of transformer-based Large Language Models (LLMs), however, their relevance has partly shifted. Jurafsky and Martin (2026, Vol. 2) point out that they now serve a new role as a resource for studying the interpretability of LLMs; while at the same time retaining their relevance as a resource for computational linguistics in the narrow sense, namely for modelling human language structure and human language processing, and, similarly, as we would add, for corpus linguistics, which uses corpora as empirical evidence for linguistic theories. Last but not least, annotated corpora are still relevant for low-resourced languages, as Opitz et al. (2025) point out in their discussion of the relevance of linguistics more generally.

In this paper, we describe the processing and annotation of a corpus of low-resourced simplified German[2] comprising different variants, most prominently texts in Easy German ('Leichte Sprache'). The latter is distinctive in that it is officially recommended in Germany in order to comply with the UN Convention on the Rights of Persons with Disabilities.[3] Easy German has been described as an "umbrella term" (Bock 2018, 9–11, Lindholm and Vanhatalo 2021, 11) referring to texts influenced by a variety of formal specifications, including particular forms of word-internal splitting to compensate for long compounds (see ex. (1)) and the presentation of 'one piece of information at a time', resulting in short sentential units and in an extensive use of lists containing phrasal and clausal items (see ex. (2)).[4]

(1) Alle Daten·sachen sollen bei der Netz·neutralität gleich schnell funktionieren.
With net neutrality, all data things are supposed to work equally fast.

(2) Ein Selbst-Vertreter ist jemand:
• der seine Meinung sagt,

[2]We understand 'simplified' as referring to the language system, that is, to a variant that is particularly easy to understand. It does not necessarily entail a translation relation between a standard source text and a derived target text.

[3]Public sector websites are required to eliminate barriers to accessibility in "information, communication and other services" (Article 9) https://social.desa.un.org/issues/disability/crpd/

[4]Both examples are taken from DE-Lite, see Sec. 3.

- der für sich selbst spricht,
- der sich für seine Rechte einsetzt
- und für Menschen, denen es ähnlich geht wie ihm.

A self-advocate is someone:

- *who says their opinion,*
- *who speaks for themselves,*
- *who advocates for their rights,*
- *and for people who are in a similar situation as them.*

These features, together with other lexical, typographical, and structural peculiarities, result in Easy German texts being out of domain for NLP tools trained on standard German texts.

While the challenges involved in processing out-of-domain corpora have already been well described, the particular requirements of our data give rise to a distinct set of challenges, which we address in this paper. Adapting existing tools to the domain of Easy German is beyond the scope of the present paper. As the project more broadly pursues a cross-linguistic perspective, we rely on state-of-the-art software that can readily be extended to further languages. Our main contributions are threefold: (i) a scalable and robust, processing-lapse-aware corpus pipeline based on state-of-the-art software and capable of handling Easy German texts while accounting for errors caused both by noisy raw data and by server-side failures, aspects of which generalise beyond the processing of simplified language; (ii) a description of normalisation, segmentation, and annotation problems that arise in noisy web-sourced Easy German text—and also in the context of harmonising pre-existing collections; and (iii) an argument in favour of sampling complete texts rather than smaller units, accompanied by a discussion of issues relating to licensing.

The present paper does not introduce a novel annotation model. Rather, it focuses on documenting the methodological and technical infrastructure required to create a reusable, multi-layer annotated corpus of simplified German at scale. Although such infrastructure work may appear primarily technical, it constitutes a necessary foundation for subsequent empirical and theoretical research. In particular, for low-resourced or out-of-domain varieties such as Easy German, transparent documentation of processing decisions, error handling strategies, and licensing constraints is essential for reproducibility and long-term reuse. We therefore position this paper as a methodological contribution that aims to make corpus creation practices explicit and transferable.

After sketching the related work in the next section, we briefly introduce the DE-Lite corpus in Section 3, before addressing the three main topics of this paper in the subsequent sections: the processing pipeline (Section 4), normalisation, segmentation and annotation (Section 5), and licensing issues (Section 6).

2 Related Work

Stodden (2026, chap.,4, pp.,63–99) provides a comprehensive overview of corpus resources for simplified German as of July 2024. In Table 4.16 (p.,94) she summarises the available corpora with the exception of her own resource, DEplain (Stodden et al., 2023). Schomacker et al. (2024) provide a concise overview in the introduction to the GermEval shared task on statement segmentation in German Easy Language (StaGE). They note that only few corpora contain linguistic annotation at sentence level, with notable exceptions including LeiKo (Jablotschkin and Zinsmeister, 2021), DEplain (Stodden et al., 2023), and APA-RST (Hewett, 2023). LeiKo is a comparable corpus consisting of 108 articles crawled from two German news websites in Easy German and two in Plain German. The corpus is automatically annotated for syntax and coreference. A core subset of 20 texts was manually corrected and additionally annotated for text structure and discourse relations. DEplain is a web-crawled and manually aligned parallel corpus containing Easy and Plain German news texts from different domains (approx. 500 document pairs and 13,000 sentence pairs), as well as a web-domain corpus comprising approx. 150 aligned documents and 2,000 aligned sentence pairs. Some sentence pairs are additionally "annotated with simplification operations including structural changes" (Schomacker et al., 2024, Sect. 2.1). APA-RST is a web-crawled corpus of 75 German-language newspaper articles from the Austrian Press Agency, "covering different topics such as politics, culture and sport" (Hewett, 2023). The corpus contains 25 articles at each of three complexity levels, with the simplified levels roughly corresponding to Plain German and Easy German respectively. The articles were preprocessed, sentence-aligned, and annotated with discourse relations. Finally, the Geasy corpus (Hansen-Schirra et al., 2021) is a linguisti-

cally annotated parallel corpus of Easy and Standard German compiled for translation studies. It comprises 276 text pairs on different topics, which are automatically annotated with syntactic dependencies and partly manually aligned at the sentence level.

Beyond linguistically annotated corpora, the Simple German Corpus (Toborek et al., 2023) is also worth mentioning from a methodological perspective. The corpus consists of 712 articles in standard German and 708 corresponding articles in Simple German collected from eight web sources spanning different domains. The articles are automatically aligned at sentence level. Rather than redistributing text files directly, the corpus publication provides URLs and code, which mitigates potential copyright issues. The original webpages are archived using the Internet Archive's Wayback Machine in order to support sustainability. This corpus may serve as an example of how subsequent users can beneficially reuse processing pipelines such as the one presented in Section 4 to ease processing the reconstructed webpages.

More recent resources include the partially synthetic German4All corpus (Anschütz et al., 2025), a parallel corpus of aligned paragraphs representing five levels of complexity based on the Common European Framework of Reference for Languages (CEFR). The simplified versions were generated with GPT-4 through paraphrasing standard German Wikipedia articles, which were subsequently thoroughly evaluated.

ELGEPA (Schomacker et al., 2025) is a further recent parallel corpus consisting of Standard German and Easy German webpages from the German public sector. The corpus is used for fine-tuning large language models for paraphrasing standard texts into Easy German.

While Jakubíček et al. (2020) and Barbaresi (2021) discuss general challenges involved in the construction of web-crawled corpora, Stodden (2026, pp. 64–72) addresses these issues specifically in the context of scraping simplified German resources.

3 The DE-Lite corpus

The corpus of this study, DE-Lite v2, comprises about 9,600 texts, which means that it is a small corpus compared to genuine web corpora (Schäfer and Bildhauer, 2013), there are also larger collections of simplified German, as described in Section 2.

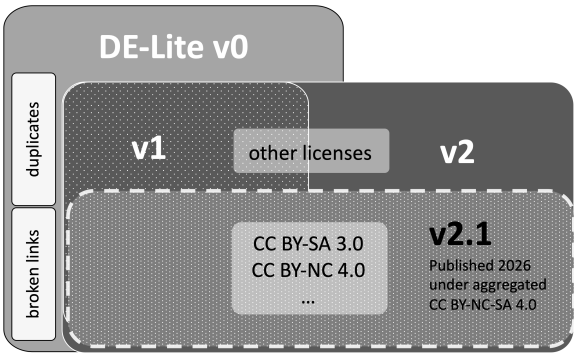

Figure 1: Different sampling states of DE-Lite and text overlap between the resulting corpus versions. DE-Lite v0 and v1 are described in Jablotschkin et al. (2024). The current pipeline described in this paper was applied to the enlarged textbase of DE-Lite v2. A subset of v2 with appropriate licences was published for reuse as DE-Lite v2.1.

However, it is unique in its breadth of source publishers and text genres. It is also unique in being annotated on multiple linguistic levels. Furthermore, it aims at being a resource for computational linguistics and corpus linguistic studies for the interaction of sentence-level simplification and text-level coherence, so it includes full text with their annotations. And substantial efforts have been made to obtain the licensing rights to distribute the annotated texts. Figure 1 visualises different sampling states of DE-Lite.

In particular, DE-Lite v2 consists of 2.5 million tokens of German texts optimised for comprehensibility. Most of these data (approx. 1.9 million tokens) have been labelled Easy German by the publishers. The rest (approx. 600,000 tokens) carry labels such as Plain German ('Einfache Sprache') or are taken from text products created specifically for children. Texts originally created for children are explicitly marked in the metadata and can therefore be distinguished from Easy ('Leicht') and Plain ('Einfach') German texts in downstream analyses. We acknowledge that these categories reflect different communicative settings and stylistic conventions; however, they share the overarching goal of enhanced comprehensibility and are therefore included within the broader scope of the corpus. While most of the texts in DE-Lite have originally been written in a simplified version of German, some of the Easy and Plain German texts are translations of more complex original texts. That is why part of DE-Lite is a parallel corpus, aligned at the text level, and also contains 1.7 million tokens of

non-simplified German.

Since simplified text is not readily identifiable through systematic web domains,[5] DE-Lite v0 first draw on unifying texts from a number of pre-existing corpora, see Section 2. In addition, targeted search for further websites with texts labelled Easy or Plain German were scraped, for example https://www.mdr.de/nachrichten-leicht and https://www.behindertenbeauftragter.de/DE/LS. Table 1 summarises the distribution of texts in DE-Lite v2 across its different sources.

Source	Monolingual	Parallel	Total
WebCorpus	3,900	344	4,244
DEplain	815	1,350	2,165
LeiKo	169	81	250
Geasy	0	301	301
APA-RST	0	75	75
Targeted search	1,577	999	2,576
Total	6,461	3,150	9,611

Table 1: Distribution of DE-Lite v2 texts across source corpora including targeted search. The subcorpora 'monolingual' and 'parallel' are based on our process of metadata documentation. 'Parallel' comprises all source and target texts that are aligned at the text level, irrespective of their level. There is no strict one-to-one correspondence between the texts of different levels.

To obtain a more balanced composition in terms of text genres, when sufficient web text was not available, texts were also extracted from selected PDFs (e.g., for lexicons and election programmes). Table 3 in Appendix A provides an overview of the distribution of texts across collection methods. Since parts of the corpus were collected via web scraping, not all original URLs are still accessible at the time of writing; however, the metadata preserve the original source information for transparency and traceability.

A distinguishing feature of DE-Lite is the compilation of a comprehensive text-level metadata collection, such as source URL, level of simplification, type of assessment (if available), publisher, author, publication date, licensing information, corpus provenance, and text genre. Table 2 summarises the distribution of texts across simplification levels. The original German labels are provided in parenthesis. More detailed aggregated statistics derived from the metadata tables, including distributions by text genre and licence type, are provided in Appendix A.

Level	Monolingual	Parallel	Total
easy ('Leicht')	4,450	904	5,354
plain ('Einfach')	1,374	339	1,713
children ('Kinder')	633	320	953
no_label ('unklar')	4	0	4
standard	0	1,587	1,587
Total	6,461	3,150	9,611

Table 2: Distribution of DE-Lite v2 texts by simplification level ('Sprachtyp').

In line with our goal of creating a reusable resource for the research community, a subset of the corpus and its metadata has been made publicly available as DE-Lite v2.1 via the research data repository of the University of Hamburg; see Section 6 for details.

4 Processing Pipeline

We first describe the environment and a chunking strategy for splitting the corpus in manageable meta-units as a prerequisite for processing. Then we detail the individual processing stages for these units (aka 'chunks'). The process is orchestrated by a master shell script that integrates three primary NLP tools, as illustrated in Figure 2: spaCy (Honnibal et al., 2020) for pre-tokenisation, UDPipe 2 (Straka, 2018) for UD annotation, and CorPipe (Straka, 2024) for coreference resolution. The final output for each source document is a single, self-contained CoNLL-U file containing all annotation layers and detailed metadata for provenance.

4.1 Prerequisites

Computing Environment The pipeline is developed for a standard High-Performance Computing (HPC) environment equipped with a workload manager (e.g., SLURM), multi-core CPU nodes, and GPU accelerators. While it was implemented and tested on a particular HPC cluster, its core logic is portable. The design assumes the ability to request specific resources (CPUs, a single GPU), manage software via an environment module system, and access distinct filesystems for software and data. Adapting the pipeline to another HPC cluster would primarily involve updating environment variables for file paths and modifying the specific module load commands.

[5] Asghari et al. (2023) used a "mix of web measurements and qualitative information" to evaluate the proportion of Easy German web pages in websites of the German public sector. They also developed a classifier to identify simplified text.

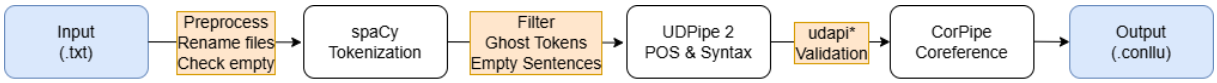

Figure 2: NLP annotation pipeline: raw texts are processed through spaCy tokenization, UDPipe 2 POS tagging and dependency parsing, CoNLL-U structural validation (udapi), and CorPipe coreference resolution, producing fully annotated output in CoNLL-U format.

A key design choice is the strategic allocation of CPU and GPU resources to prevent Out-Of-Memory (OOM) errors and job failures: The GPU, a high-demand resource, is reserved exclusively for the most computationally intensive task, the Transformer-based coreference model in CorPipe, see Section 4.7, while the processes of spaCy and UDPipe, see Section 4.5, are explicitly restricted to the allocated CPU cores.

To enforce this separation, we leverage CUDA (Compute Unified Device Architecture), NVIDIA's parallel computing platform that enables software to access and utilize GPU hardware (Nickolls et al., 2008). CUDA provides fine-grained control over which GPUs are visible to a running process. Within our execution script, we set the environment variable `CUDA_VISIBLE_DEVICES` to an empty string for the CPU-only stages. This configuration effectively hides all GPUs from the spaCy and UDPipe processes, preventing them from attempting to load or access GPU memory and ensuring that these stages run exclusively on the allocated CPU cores.

Corpus Chunks To ensure robustness, manage job time limits, and facilitate error recovery, the corpus is not processed in a single, monolithic run. Instead, it is partitioned into four manageable chunks. This is implemented using SLURM's array job feature (`-array=0-3`), which launches our main script four times. Each job instance is assigned a unique task ID by SLURM, which the script uses to select its corresponding chunk of files to process. Crucially, we constrain the array jobs to run sequentially (`%1`). This prevents race conditions and file-writing conflicts that would otherwise occur in the shared output directories without complex file-locking mechanisms.

While the pipeline was developed and executed in an HPC environment, such an infrastructure is not strictly required for processing the corpus. In principle, the pipeline can be executed on a sufficiently powerful local workstation. However, our preliminary experiments on a standard notebook computer showed that sustained processing of large corpora under high CPU load is impractical. Extended full-load operation increases the risk of thermal stress, reduced system stability, and potential data loss in case of crashes, and it prevents parallel work on the same device.

In addition, the Transformer-based coreference resolution component (CorPipe) benefits substantially from GPU acceleration. Running this stage on CPU alone considerably increases processing time and may require additional memory management strategies. An HPC or server-class environment allows the workload to be distributed across dedicated CPU cores and GPU resources, ensuring stable execution and reasonable turnaround times.

Therefore, while the pipeline is technically executable without HPC resources, we strongly recommend access to a server-class machine or HPC infrastructure for large-scale corpora in order to ensure efficiency, stability, and reproducibility.

4.2 Preprocessing

As the first step within a job, a script—depicted in the first orange box in Figure 2—performs basic file-level preparations. It scans for files with spaces in their names and normalises them by replacing spaces with underscores to ensure command-line compatibility. More importantly, it checks each file for actual text content. If a source file is empty or contains only whitespace, it is flagged, and a minimal `.conllu` file containing only metadata is generated. This file is then excluded from all subsequent, computationally expensive annotation steps, saving resources and preventing potential errors.

4.3 Tokenisation

While UDPipe offers built-in tokenisation, initial tests showed it was inadequate for the specific orthographic conventions of Easy German. These texts frequently use a hyphenation point (`·`) to segment compound nouns for better readability (e.g., `Wahl·ergebnis` instead of standard *Wahlergebnis*). The default UDPipe tokeniser incorrectly split such words into three separate tokens (`Wahl`,

·, ergebnis), leading to cascading errors in the downstream syntactic analysis.

To resolve this, we employ a two-stage approach. First, we use spaCy (with the de_core_news_sm model), which correctly identifies these segmented compounds as single tokens. This initial step performs both sentence segmentation and tokenisation. The output is a temporary, pre-tokenised CoNLL-U file which serves as the input for the next stage, ensuring that the token boundaries established by spaCy are respected throughout the rest of the pipeline.

4.4 Filter Whitespaces and Empty Sentences

The spaCy-based tokenisation introduced a critical side effect: the de_core_news_sm model occasionally generated "ghost tokens" consisting solely of whitespace characters. These tokens corrupted the tab-separated CoNLL-U format, as a tab within the FORM column would cause a column shift, rendering the file unparsable. We modified the spaCy output generation script to include two filters:

(a) Any token consisting only of whitespace is silently discarded.

(b) If this filtering results in a sentence with no tokens, the entire empty sentence block is omitted from the output.

This ensures that only valid, non-empty tokens are passed to the subsequent annotation tools.

4.5 UDPipe Annotation

Morpho-syntactic and syntactic annotation is performed using UDPipe 2 with the pre-trained de_hdt-ud-2.17-251125 model. By providing the pre-tokenised files (--input=conllu), we instruct UDPipe to only add linguistic information without altering token boundaries. It generates a lemma, UPOS/XPOS tags, morphological features, and a dependency parse for each token.

To optimise performance, we avoid the significant overhead of loading the UDPipe model for each file. Instead, at the beginning of each job, UDPipe is started once as a persistent, local HTTP server. A client script then sends each file to this server for annotation. This server-client architecture, running exclusively on CPU as enforced by our resource allocation strategy (see Section 4.1), dramatically reduces the cumulative processing time for a chunk of several thousand files.

4.6 Validation and Error Recovery

To guarantee the integrity of our data, our quality assurance process consists of two key steps: an intermediate validation check within the pipeline and a final error recovery procedure after the run is complete.

First, an intermediate validation step is performed after the UDPipe annotation stage. To ensure that the input for the computationally expensive coreference stage is structurally valid, we use the udapi library to check each CoNLL-U file. Any file that fails this check is flagged in its metadata (skipped_invalid_conllu) and excluded from coreference annotation, preventing it from causing errors in the final stage of the pipeline.

Second, we have a procedure to handle errors that are only discovered after processing. A final check of all output files revealed that a small number were incomplete, typically missing essential # sent_id and # text headers. Our investigation identified the cause as transient, non-deterministic errors common in HPC environments, not bugs in the pipeline logic. As noted in the literature on HPC fault tolerance (Montezanti et al., 2020), soft errors and transient failures are frequent in large-scale systems and require automatic recovery strategies.

Rather than attempting fragile manual fixes, our approach leverages an idempotent repair script—one that can be safely executed multiple times without causing side effects. The script identifies failed files, finds their original plain-text source, and re-runs them through the entire annotation pipeline. Crucially, since transient HPC failures persist as a systemic issue, the repair script may need to be invoked multiple times until all files are successfully annotated. This idempotent design ensures that repeated executions gradually resolve failures without corrupting previously successful annotations. This approach guarantees that all files in the final corpus are created using the exact same methodology, ensuring consistency and reproducibility.

4.7 CorPipe Annotation

The final annotation layer is coreference resolution, handled by CorPipe with the trained model (corpipe24-corefud1.2-240906). This step adds coreference annotations (mentions/entities) and is accelerated on the HPC cluster using a GPU. CorPipe takes the validated .conllu file

as input and adds the coreference information in the `MISC` column, following the CoNLL-U format standards. The resulting file, now containing all three annotation layers, overwrites the intermediate file.

To ensure stability during the GPU-intensive coreference step, we configure CorPipe with a `batch_size` of 1. In neural network processing, the batch size determines how many documents are processed together in a single forward pass through the model. While CorPipe's default batch size is 8, processing multiple documents simultaneously can cause memory spikes on the GPU, potentially leading to Out-Of-Memory (OOM) errors. By setting `batch_size=1`, we ensure that each document is processed individually, minimizing peak memory usage and robustly handling documents of varying lengths and complexity without risking GPU memory exhaustion.

4.8 Cleanup and Provenance

In addition to the pipeline depicted in Figure 2, for each chunk a cleanup process terminates the UDPipe server and removes temporary files. A key feature of our pipeline is its emphasis on provenance. Each final `.conllu` file is augmented with metadata headers documenting the exact models and versions used in the pipeline. For example:

```
# pipeline_date = 2026-03-13
# pipeline_spacy = de_core_news_sm
# pipeline_udpipe = de_hdt-ud
    -2.17-251125
# pipeline_corpipe = corpipe24-corefud1
    .2-240906
# source_file = text/example.txt
```

These headers make every output file self-documenting: You can always tell which models were used and when the file was processed. The detailed logging and status tracking from the validation and repair steps ensure that the entire process is auditable.

5 Normalisation, Segmentation and Annotation: Known Problems

Despite careful compilation and processing, there are still some issues in DE-Lite v2 that affect the quality of the data. These issues partly derive from the unification of a variety of corpus resources and formats and partly from the fact that traditional NLP tools have not been trained on Easy German data so far.

5.1 Different Corpus Resources and Formats

In order to reduce syntactic complexity, Easy German uses a lot of lists (see ex. (2)). In html code, lists are often created by using the tag `<li>`. When scraping websites and parsing the html files, we normalised all bullet points by replacing the `<li>` tag with • ('bullet point'). However, as we also included text from various other formats (tcf, docx, PDF, txt) in DE-Lite, this normalisation step did not take place for all of the corpus texts. As a result, there are a number of different bullet forms in DE-Lite v2 (e.g., points of different sizes, dashes, squares, triangles, arrows).

Another issue concerns hyphenation due to linebreaks, which also stems from file formats such as PDF. We removed line-break hyphenation for most of the texts extracted from PDFs, manually with smaller files and with the Python library PyMuPDF (Artifex Software, Inc., 2025) with larger files. Still, an explorative analysis reveals that there are still corpus texts that contain unwanted hyphenation (see ex. (3)).

(3) Auch an Familien gewährte Hilfen für den Besuch öffentlicher Schulen oder Leistungen der **Ausbil- dungsförderung** fallen unter den Begriff der „ **öf- fentlichen** Fürsorge " (gemäß Artikel 74 Absatz 1 Nr. 7 GG) mit der Begründung , dass Jugendliche in ihrer ganzen Existenz auf die Hilfe ihrer **Um- welt** angewiesen sind [...]

Web text is formally different than text in PDFs and therefore poses different challenges during data cleaning: It is often interactive and contains multimodal elements as well as hyperlinks. Even though our research interest concerns running text only, some of the corpus files still contain textual elements extracted from images, or references to other websites or downloadable documents (see ex. (4)).

(4) [...] Erklärung zum Thema Intensivpflege- und Rehabilitationsstärkungsgesetz (IPREG) vom 13. Dezember 2019 (Alltagssprache) Dokument vorlesen Herunterladen (PDF , 235 KB , Datei ist nicht barrierefrei) [...]

5.2 Performance of NLP Tools on Easy German Data

The frequent use of bullet points in Easy German data affects POS tagging as well as sentence seg-

mentation. In STTS (Schiller et al., 1999), a widely used POS tagset for German, there is no label for bullet points. In our data, UDPipe assigns the label FM (foreign-language material) to approx. 60% (n=13,639) of bullet point tokens, and $. (sentence-final punctuation mark) to approx. 20 % (n=4,111) tokens. A variety of other labels (e.g., ADV, NE, APPR) is assigned to bullet point items with lower frequencies.

The inconsistent POS tagging consequently affects dependency parsing and sentence identification. This can be illustrated by having a look at one-token sentences, which make up more than 5% of all sentences in DE-Lite v2: Approx. 40% (n=8,606) of these one-token sentences consist of a bullet point. However, there are also sentences longer than one token that contain bullet points, as in ex. (2), which is identified as one complete sentence and where the first two bullet points are labelled FM and the last two ADV.

We are also aware that coreference resolution has not been adapted to Easy German data yet, which affects annotation quality. As has been shown by Jablotschkin et al. (2025), Easy German frequently uses linguistic structures that are difficult for CorPipe to resolve, e.g., event anaphora, split antecedents or lexical substitutions. These structures arise as a consequence of simplification efforts on other levels, e.g., syntactic simplification or the explanation of concepts and difficult words.

5.3 Raw Text Normalization and Ghost Token Cleaning

While the annotation pipeline filters ghost tokens during processing (see Section 4.4), we also created a cleaned version of the raw text corpus to ensure consistency between input texts and annotated outputs. Using spaCy with the same tokenization logic as the pipeline, we identified and removed ghost tokens—whitespace-only tokens including special Unicode characters such as non-breaking spaces (\xa0) and various line-break characters (\n).

Our analysis revealed that 5,367 out of 9,611 texts (55.8%) contained ghost tokens, totalling 279,827 replacements. However, this cleaning process introduced a trade-off: Paragraph boundaries encoded as whitespace sequences were collapsed, resulting in single-line text files. While this cleaned version is fully compatible with the annotation pipeline, it loses the original document structure.

An additional challenge emerged: Some non-breaking space characters (\xa0) appear within word boundaries rather than as separate tokens. Since spaCy does not tokenize these as separate tokens, they cannot be automatically identified or removed without manual inspection. Thus, the final corpus includes both the original raw texts (with preserved paragraph structure but containing ghost tokens) and the cleaned, single-line version (with ghost tokens removed but paragraph structure lost). Researchers using DE-Lite v2 (or the published version v2.1) should be aware of these trade-offs depending on their use case.

6 Licensing Issues

We created DE-Lite with the aim of reusability to provide an easily accessible and sound data base for research on the area of Easy German. In particular, we want to support research that looks beyond the sentence level and takes phenomena of textual coherence into account. That is why we systematically collected the publishing license as a metadata along with each text. For a lot of texts, this information is directly derivable from the respective website or the PDF document. For the remaining texts, we contacted the publishers and requested a CC BY-NC 4.0 license.

Creative Commons (CC) is a non-profit organisation promoting shared knowledge and therefore issuing standardised and easily applicable licenses allowing to reuse and distribute published data under certain conditions (see https://creativecommons.org/cc-licenses/). The least restrictive CC license is CC BY. If a creator publishes their work under this license, anyone is allowed to reuse and distribute this work in any edited or original form, provided they give credit to the creator. The license CC BY can be combined with different additional conditions. For example, in addition to giving credit to the creator CC BY-SA implies that edited material be shared under the same or a compatible license. CC BY-NC, in contrast, allows only non-commercial uses, while CC BY-ND prohibits any adaptations of the shared material.

The most recent version of all CC licenses is 4.0. However, some of the material we included in DE-Lite v2 (e.g., hurraki.de/wiki) was published under one of the older CC 3.0 licenses. As a consequence, the individual texts keep their original licence, even if the collection as a whole is published under version 4.0. An unresolved issue concerns texts equipped with a CC BY-ND license

which prohibits remixing or adapting the original work. We still have to confirm that the comprehensive annotations of our corpus do not count as 'remix', if they do, this license would not be appropriate for our purpose.

While these licensing constraints prevent us from releasing the entire corpus under a single unified license, a substantial subset of DE-Lite v2 has already been made publicly available: DE-Lite v2.1, released via the research data repository of the University of Hamburg (Jablotschkin and Zinsmeister, 2026), comprises those texts for which publishers granted redistribution rights under compatible licenses. This release includes both raw text and CoNLL-U-annotated versions, along with comprehensive metadata. The public availability of DE-Lite v2.1 ensures the reproducibility of the processing pipeline described in Section 4 and allows the research community to build upon our work.

7 Conclusion

We discussed a range of challenges encountered during the processing and annotation of DE-Lite v2. While we were able to propose solutions and heuristics for some of these problems, others remain unresolved. We nevertheless believe that these observations may prove useful for other projects dealing with web-scraped simplified German, particularly Easy German texts that are "born simplified", rather than derived through the simplification of standard German source texts. Our focus was on providing full texts enriched with linguistic annotation and comprehensive metadata. This metadata collection could furthermore serve as a source for the creation of a "data statement" (Bender and Friedman, 2018).

Limitations

The pipeline makes use of state-of-the-art tools that have not been adapted to our data, although we argue that Easy German constitutes an out-of-domain variety for tools developed for standard German. Some of our solutions are necessarily idiosyncratic and may not generalise to other projects.

Acknowledgments

This research was funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 232722074 – SFB 1102. We would like to thank Kai Wörner for helpful advice on licensing issues. We would also like to thank the reviewers for their insightful comments.

References

Miriam Anschütz, Thanh Mai Pham, Eslam Nasrallah, Maximilian Müller, Cristian-George Craciun, and Georg Groh. 2025. German4All – a dataset and model for readability-controlled paraphrasing in German. In *Proceedings of the 18th International Natural Language Generation Conference*, pages 390–407, Hanoi, Vietnam. Association for Computational Linguistics.

Artifex Software, Inc. 2025. PyMuPDF 1.26.5.

Hadi Asghari, Freya Hewett, and Theresa Züger. 2023. On the Prevalence of Leichte Sprache on the German web. In *Proceedings of the 15th ACM Web Science Conference (WebSci '23)*, pages 1–6, Austin, TX, USA. ACM.

Adrien Barbaresi. 2021. Trafilatura: A web scraping library and command-line tool for text discovery and extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 122–131, Online. Association for Computational Linguistics.

Emily M. Bender and Batya Friedman. 2018. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604.

Bettina M. Bock. 2018. *"Leichte Sprache" - kein Regelwerk. Sprachwissenschaftliche Ergebnisse und Praxisempfehlungen aus dem LeiSA-Projekt*. Universität Leipzig, LeiSA.

Stefan Th. Gries and Andrea L. Berez. 2017. Linguistic annotation in/for corpus linguistics. In Nancy Ide and James Pustejovsky, editors, *Handbook of Linguistic Annotation*, pages 379–409. Springer.

Silvia Hansen-Schirra, Jean Nitzke, and Silke Gutermuth. 2021. An intralingual parallel corpus of translations into German Easy Language (Geasy corpus): What sentence alignments can tell us about translation strategies in intralingual translation. In Vincent X. Wang, Lily Lim, and Defeng Li, editors, *New Perspectives on Corpus Translation Studies*, pages 281–298. Springer Singapore, Singapore.

Freya Hewett. 2023. APA-RST: A text simplification corpus with RST annotations. In *Proceedings of the 4th Workshop on Computational Approaches to Discourse (CODI 2023)*, pages 173–179. Association for Computational Linguistics.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength natural language processing in python.

Sarah Jablotschkin, Ekaterina Lapshinova-Koltunski, and Heike Zinsmeister. 2025. Coreference in simplified German: Linguistic features and challenges of

automatic annotation. In *Proceedings of the Eighth Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 12–23, Suzhou, China. Association for Computational Linguistics.

Sarah Jablotschkin, Elke Teich, and Heike Zinsmeister. 2024. DE-Lite - a new corpus of Easy German: Compilation, exploration, analysis. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 106–117, St. Julian's, Malta. Association for Computational Linguistics.

Sarah Jablotschkin and Heike Zinsmeister. 2021. Annotating Colon constructions in Easy and Plain german. In *Proceedings of the 3rd Swiss Conference on Barrier-Free Communication (BfC 2020)*, pages 125–134. ZHAW Zürcher Hochschule für Angewandte Wissenschaften.

Sarah Jablotschkin and Heike Zinsmeister. 2026. *DE-Lite*. Corpus Version v2.1; Forschungsdatenrepositorium der Universität Hamburg. https://doi.org/10.25592/uhhfdm.18676.

Miloš Jakubíček, Vojtěch Kovář, Pavel Rychlý, and Vit Suchomel. 2020. Current challenges in web corpus building. In *Proceedings of the 12th Web as Corpus Workshop*, pages 1–4, Marseille, France. European Language Resources Association.

Daniel Jurafsky and James H. Martin. 2026. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, with Language Models*, 3rd edition. Online manuscript released January 6, 2026.

Camilla Lindholm and Ulla Vanhatalo. 2021. Introduction. In Camilla Lindholm and Ulla Vanhatalo, editors, *Handbook of Easy Languages in Europe*, pages 11–26. Frank & Timme, Berlin.

Diego Montezanti, Enzo Rucci, Armando De Giusti, Marcelo Naiouf, Dolores Rexachs, and Emilio Luque. 2020. Soft errors detection and automatic recovery based on replication combined with different levels of checkpointing. *Future Generation Computer Systems*, 113:240–254.

John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. 2008. Scalable parallel programming with CUDA. *ACM Queue*, 6(2):40–53.

Juri Opitz, Shira Wein, and Nathan Schneider. 2025. Natural language processing RELIES on linguistics. *Computational Linguistics*, 51(3):1009–1032.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. *Guidelines für das Tagging deutscher Textcorpora mit STTS (Kleines und großes Tagset)*. Stuttgart/Tübingen.

Thorben Schomacker, Miriam Anschütz, Regina Stodden, Georg Groh, and Marina Tropmann-Frick. 2024. Overview of the GermEval 2024 shared task on statement segmentation in German easy language (StaGE). In *Proceedings of GermEval 2024 Shared Task on Statement Segmentation in German Easy Language (StaGE)*, pages 1–14, Vienna, Austria. Association for Computational Linguistics.

Thorben Schomacker, Burak Tinman, Chris Biemann, and Marina Tropmann-Frick. 2025. LLMs for Easy Language Translation: A Case Study on German Public Authorities Web Pages. In *German Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 252–261. Springer.

Roland Schäfer and Felix Bildhauer. 2013. *Web Corpus Construction*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool, San Francisco.

Regina Stodden. 2026. *Automatic German Text Simplification: Data, Evaluation, and Models*, 1 edition, volume 21 of *Easy – Plain – Accessible*. Frank & Timme.

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.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, Brussels, Belgium. Association for Computational Linguistics.

Milan Straka. 2024. CorPipe at CRAC 2024: Predicting zero mentions from raw text. In *Proceedings of the Seventh Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 97–106, Miami. 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.

A Appendix: Corpus Composition Details

This appendix provides aggregated statistics derived from the corpus metadata table.

A.1 Distribution by collection method

Format	Monolingual	Parallel	Total
TXT via HTML	2,439	2,401	4,840
TCF	3,900	344	4,244
PDF	122	61	183
DOCX	0	269	269
TXT	0	75	75
Total	6,461	3,150	9,611

Table 3: Collection methods in DE-Lite v2.

A.2 Distribution by text genre

Text genre	Easy	Plain	Children	Standard	Total
wiki	3,017	0	0	0	3,017
lexicon	532	166	953	396	2,047
lay comm.	737	262	0	691	1,690
news	645	292	0	416	1,353
blog	158	957	0	2	1,119
org. profile	74	0	0	30	104
other	189	40	0	52	281
Total	5,354	1,717	953	1,587	9,611

Table 4: Distribution of texts in DE-Lite v2 by level and text genre. 'Lay comm(unication)" refers to information for lay audiences ('Fachexterne Kommunikation'), "org(anisation) profile" to profiles of institutions and charities ('Organisationsportät'). Four texts with level 'no_lable' are subsumed under 'plain'.

A.3 Distribution by license type

License type	Easy	Plain	Child.	Stand.	Total
CC BY-SA *	3,033	198	0	75	3,306
CC BY-NC *	895	206	0	395	1,466
CC BY-NC-ND *	12	0	953	331	1,296
other	9	20	0	18	47
no free license	493	1,111	0	375	1,979
NA	912	182	0	423	1,517
Total	5,354	1,717	953	1,587	9,611

Table 5: Distribution of texts in DE-Lite v2 by level and original license. '*' is a placeholder for different versions. Four texts with level 'no_label' are subsumed under 'plain'.