Overview of the GermEval 2024 Shared Task on Statement Segmentation in German Easy Language (*StaGE*)

Thorben Schomacker[1*], **Miriam Anschütz**[2*], **Regina Stodden**[3*], **Georg Groh**[2], and **Marina Tropmann-Frick**[1]
[1]Hamburg University of Applied Sciences
[2]Technical University of Munich
[3]Heinrich-Heine University Düsseldorf
statements@soc.cit.tum.de

Abstract

Sentences in easy language should only contain a few statements to enhance their readability. However, there exists no common definition of statements or tools that could automatically extract them. With this shared task, we try to close this gap in German Easy Language. We drafted a set of annotation guidelines and created a dataset with manually extracted numbers of statements and statement spans. We attracted three participating teams that tried to automatize the annotation process to automatically annotate the statements. All submissions are based on a BERT classification system and could outperform our naive baselines. However, especially the statement span extraction is challenging and requires further research. Our shared task tries to facilitate and motivate further research in this direction.

1 Introduction

In recent years, a growing community of researchers in computer science and computational linguistics has been working on facilitating writing in German Easy or Plain Language by providing machine learning models and corpora for training or evaluating them, e.g., Säuberli et al. (2020); Deilen et al. (2023); Madina et al. (2023); Jablotschkin et al. (2024) as well as for other languages (Alva-Manchego et al., 2020; Štajner et al., 2023).

German Easy Language (DE: "Leichte Sprache", also called "easy-to-read language") is a controlled language, which increases accessibility and comprehensibility (Bredel and Maaß, 2016). This language is, for example, characterized by using simple and short words, avoiding complex sentence structures (e.g., passive voice, nominal style, or conjunctive mood), and minimizing the statements per sentence. Besides other easy-to-read standards (e.g., Bredel and Maaß 2016 or Netzwerk Leichte Sprache 2022), the DIN SPEC 33429 (DIN-Normenausschuss Ergonomie, 2023) defines the characteristics of the German Easy Language. While currently only available as a draft version, it has the potential to become the ruling guideline in the future.

However, the evaluation of whether a (automatically or manually) generated text is readable and suitable for the target group is still a big challenge (Stajner, 2021; Cumbicus-Pineda et al., 2021). The characteristics or list of requirements for German Easy Language texts are often not considered in the evaluation process. Currently, either the target group is asked whether they face issues with the text (Stajner, 2021) or metrics are used to automatically estimate the texts' simplicity, fluency, and meaning preservation (Alva-Manchego et al., 2020) or all of the aspects at once (Maddela et al., 2023).

In order to integrate the German Easy Language characteristics into the evaluation process more, we are following a modular approach per characteristic respectively. In this work, the characteristic of our interest is the number of statements. At this time, there is only a limited amount of literature available and no practical implementation to solve the above problem. For example, the DIN SPEC 33429 does not further define "one statement". To scientifically operationalize the guidelines, we aim at closing this gap. Therefore, we introduce the "GermEval 2024 Shared Task 2: Statement in German Easy Language (StaGE)"[1] to ignite a scientific debate on how to segment and identify statements in German Easy Language. Our contributions can be summarized as follows:

[*]Equal contribution. Order determined by coin flip.

[1]https://german-easy-to-read.github.io/statements/

1. We provide an annotation guideline on the identification and segmentation of statements in German sentences.

2. We release a gold standard dataset including manual annotations regarding the number and segments of statements in German Easy Language sentences.

3. We organized a shared task that introduced the first approaches regarding the automatic identification and segmentation of statements in German Easy Language sentences.

In the following, we introduce the shared task in more detail and provide an overview of the shared task's results. Our data, evaluation method, and baselines are publicly available for further experiments.[2]

2 Related Work

This shared task contributes in a three-fold manner:

2.1 Corpora

In recent years, many text simplification corpora have been introduced which include German Easy Language, e.g., Simple German Web Corpus '13 (Klaper et al., 2013), GEASY corpus (Hansen-Schirra et al., 2021), Simple German Web Corpus '23 (Toborek et al., 2023), MILS corpus (Schomacker et al., 2023), DEplain-web corpus (Stodden et al., 2023), DE-Lite corpus (Jablotschkin et al., 2024) as well as web harvesters that download texts in German Easy Language from the web, e.g., Klepp (2022), or Anschütz et al. (2023). Most of these corpora include professionally simplified texts, e.g., news texts (see NDR Nachrichten[3], MDR Nachrichten[4]) or texts of public authorities (see Stadt Hamburg[5] or Stadt Köln[6]). But, the web harvester also includes texts written by non-professional translators, i.e., articles of hurraki.de.

Most of the corpora listed above do not contain other linguistic annotation. One exception is the LeiKo corpus (Jablotschkin and Zinsmeister, 2021) in which the authors annotated colon construction in simplified and Standard German. Some sentences of the DEplain-web corpus (Stodden et al., 2023) are also annotated with simplification operations including structural changes, e.g., rewriting from passive-to-active voice, reordering, or splitting. Furthermore, the APA-RST corpus (Hewett, 2023) includes a structural analysis of the texts, but they are written for foreign language learners and not the target group of German Easy Language.

For English, there is the MinWikiSplit corpus (Niklaus et al., 2019b) and the BiSECT corpus that focuses on syntactic simplification. The MinWikiSplit corpus contains 203,000 complex-simple sentence pairs where the simplified parts are automatically segmented into minimal meaningful propositions using the TS framework DISSIM (Niklaus et al., 2019a). The BiSECT corpus (Kim et al., 2021) includes English simplification pairs where a complex sentence is either split into two simple sentences, or two complex sentences are merged into one simple sentence. Another BiSECT version exists with German simplification pairs, but they have major encoding problems (i.e., all diacritical markers are missing).

Nevertheless, we are not aware of a high-quality German corpus including syntactic simplification or any other kind of annotations regarding the number of statements in Easy Language texts.

2.2 German Text Simplification Approaches

Most existing German text simplification systems focus on generating texts for foreign language learners (e.g., Säuberli et al. 2020 or Spring et al. 2021), for laypeople (e.g., Trienes et al. 2022), for a mixed group (e.g., Ryan et al. 2023, Klöser et al. 2024 or Fruth et al. 2024), but only a few address the German Easy Language target group (e.g., Siegel et al. 2019 or Deilen et al. 2023).[7] However, due to the lack of corpora for syntactical German simplification, there are no German text simplification systems that focus on syntactic simplification.

But, we are aware of English TS systems that bring syntactic and semantic separation into focus that are related to our approach of statement segmentation. DISSIM (Niklaus et al., 2019a) aims at

[2] https://github.com/german-easy-to-read/statements

[3] https://www.ndr.de/fernsehen/barrierefreie_angebote/leichte_sprache/Nachrichten-in-Leichter-Sprache,nachrichtenleichtesprache100.html

[4] https://www.mdr.de/nachrichten-leicht/nachrichten-in-leichter-sprache-114.html

[5] https://www.hamburg.de/barrierefrei/leichte-sprache

[6] https://www.stadt-koeln.de/service/leichte-sprache/index.html

[7] For a more detailed overview, we refer to Madina et al. (2023) or Stodden (2024).

splitting English sentences into "minimal propositions" based on hand-crafted rules. Niklaus et al. define minimal propositions as utterances that cannot be further split into meaningful propositions and that represent a minimal semantic unit. In comparison, Jamelot et al. (2024) aim at separating so-called "rheses" in French and English texts. Rheses are also defined as small and syntactically meaningful units of a sentence but are additionally based on the prosody of the text. Their system is intended to split the segments based on syntactic dependency trees, punctuation marks, and prosodic features. Our annotation procedure of "statements" is closer to the definition of minimal propositions than rheses.

2.3 Evaluation of simplified texts

Evaluating the quality of simplified texts is a hard task to solve, which is rooted in the challenge of defining a standard simplified output. Grabar and Saggion (2022) name two reasons behind this root challenge: (1) it is not factual since it relies on transformations, and (2) it is heavily based on the own knowledge and opinion of people and thereby not consensual. Furthermore, unlike for standard language, a *native simplified-language speaker does not exist* (Siddharthan, 2014). There have been several attempts to solve this challenge:

Deilen (2020) analyzed eye-tracking data collected from people with impairments to investigate the effects of certain simplified language stylistics, such as separating longer nouns via a hyphen. Säuberli et al. (2024) conducted a survey of people with and without cognitive impairment using various questionnaires to measure the comprehensibility of the texts. Cumbicus-Pineda et al. (2021) proposed a checklist-based evaluation, but each of the items is verified manually. In contrast, in the automatic evaluation of text simplification, the focus is on measuring simplicity, fluency, and meaning preservation (Alva-Manchego et al., 2020). But, neither of these aspects considers characteristics of Easy (German) Language. Evaluations based on existing guidelines for simplified language have not yet been implemented but have been theorized (Schomacker et al., 2024; Madina et al., 2024).

2.4 Shared Tasks

Recent shared tasks in the scope of German text simplification have focused on the level of lexical simplification. For example, the identification of complex words in sentences (Yimam et al., 2018), the prediction of the complexity of German sentences for foreign language learners (Mohtaj et al., 2022) or the prediction of the complexity of a word within a German sentence and the suggestion of a simpler synonym (Shardlow et al., 2024). To the best of our knowledge, there is no shared task that focuses on syntactic simplification of German.

3 Task Description

The goal of the shared task is to identify the number of statements and the statement spans in the sentence. We do not aim to force authors to write one-statement sentences exclusively but rather make the current statement distribution per sentence transparent to them. In addition, we allow authors to re-think their texts and re-phrase them to increase readability with reference to German Easy Language guidelines.

3.1 Sub Tasks

The shared task consists of two subtasks and the participants can submit their systems to both of the tasks independently. In subtask 1, we only estimate the number of statements that are present in the sample. For the second subtask, the respective statement spans are annotated. Figure 1 shows how our annotated data looks line. The two subtasks refer to two separate columns. In the first column, we only state the number of statements in the sentence, which corresponds to subtask 1. For the second subtask, we tackle the statement spans as distinct sets of words. For the example in Figure 1, the sentence "Diese Flugzeuge transportierten Nahrung und Brenn-material." (*These planes carried food and fuel.*) contains two statements: "The planes carried food" and "The planes carried fuel". There are two different things carried by the planes that the readers have to understand. Even though they might be related, they are two separate chunks of information that the readers have to digest.

3.2 Use Cases

We envision the following use cases for statement and statement span annotations:

Complexity assessment as part of a holistic German Easy Language assessment (Mohtaj et al., 2022). German E2R gives clear rules about the number of statements in sentences. Therefore, the assessments can be utilized for editorial purposes or direct feedback for writers.

Phrase_tokenized	#Statements	Statement spans
0:=Diese 1:=Flugzeuge 2:=transportierten 3:=Nahrung 4:=und 5:=Brenn-material.	2	[[3] , [5]]

Figure 1: Example Annotations from our trial data set. We enumerate the tokens in the sentence to simplify referring to them in the statement span annotations. In this case, we annotate the tokens that belong to the different spans as a set of statements spans. For more information on the annotation process, refer to Appendix A in the Appendix. Our data is open source and can be downloaded on our homepage.

Rule-based simplification is actively researched for German (Suter et al., 2016) and could also profit from incorporating statements. Using discourse-aware statement splits, a similar approach achieved promising results for the English language (Niklaus et al., 2019a).

Statement-aware evaluation could increase the performance of overlap-based metrics like BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004). A similar metric is described and implemented in Sulem et al. (2018).

Fact checking of statements as atomic facts. The extracted statements can be considered atomic facts and checked individually for truthfulness. This statement-wise evaluation facilitates the detection of partially-correct information (Min et al., 2023).

4 Data Set and Evaluation

4.1 Data Set

Our corpus contains texts from the Hurraki wiki site, a German online dictionary. We used the scraper by Anschütz et al. (2023) and scraped the articles as of January 2024. The texts are supposed to be written in German Easy Language by people who are not obligatorily trained in writing in this controlled language. The texts are not verified before publication, and thus, they may contain errors such as punctuation or grammatical errors. However, we do not correct any typos manually, but if the errors are severe, e.g., if two sentences are merged into one due to missing punctuations, these samples are manually annotated as erroneous.

In terms of pre-processing, we have split the texts into sentences at punctuation marks, and hence, they can span over several lines. The tokens in the sentence are separated by white spaces and consecutively numbered (see *Phrase_tokenized* in Figure 1). A statement only contains full tokens; we don't split tokens for sub-token annotations. If a sentence spans several lines (as typical for LS), the line breaks are displayed with `\newline`.

Figure 1 exemplifies a data sample from our trial dataset. We manually annotated the sentences regarding the number of statements contained in them (see *#Statements* column). The distinct statements of a sentence are annotated at the word level (see *Statement spans* column).

4.1.1 Annotation Guideline

Our annotation process consists of two parts: the statement splitting and the annotation of the statement spans. To determine the number of statements, we split the sentences at conjunctions. Additional pieces of information like adjectives, adverbs, or prepositional constructions form separate statements. The detailed rules for statement splitting are explained in Appendix A. If a statement does not include a verb or if it contains multiple sentences, we annotate it as 0 statements to indicate it is erroneous. However, we do not include samples with 0 statements in our test and eval splits.

For the statement span annotations, we enumerated the tokens in the sentence in ascending order. With this, we could easily refer to them in the statement span annotations. The statement spans are only annotated for samples with more than 1 statement. We annotate the tokens that belong to the different spans as a set of statements spans. In addition, we define the following rules:

- The order of the spans and the order of tokens within the spans is not important and can be altered.
- Articles, coordinating conjunctions, and "\newline" are never included in the spans. In contrast, subordinating conjunctions, relative pronouns, or filler words are included.
- Only the mutually exclusive parts are annotated, i.e., tokens that are contained in all statements are not annotated.

Example annotations can be found in Figure 1 and Table 6 in the Appendix.

For the annotation process itself, we split the full dataset into chunks of around 500 samples. Articles should be complete and should not span over multiple chunks, and thus, the chunks vary in size. For each chunk, two of the authors annotated the data independently. If the two annotators agreed on their annotation, we kept it for the final dataset. Otherwise, if the two annotators disagreed, we showed the samples to a third annotator who chose the correct annotation. We had three different annotators. All annotators are German native speakers and have experience with German Easy Language. The inter-annotator agreements range from 64.38% to 73.21%. Overall, we annotated 4,553 samples in 9 chunks and removed erroneous and ambiguous samples. Then, we split the data into train (2,944 samples), test (416 samples), and final evaluation (878 samples) splits. For the test and evaluation sets, we additionally removed the 0-statement samples.

4.1.2 Data Structure

In Hurraki, articles consist of three different elements. The first element is the short abstract that contains the most relevant information about the topic. The second element is a picture, and the last element is a more detailed description with further information. Hurraki provides metadata for their articles, such as the article's category or the timestamp of the last edit. We kept this metadata to construct a rich dataset. In addition, we pseudonymized the author to enable analyses of differences in writing styles among different authors. Table 1 gives an overview of the information available in our dataset.

4.1.3 Statistics

The basis of the task is a dataset (Figure 2) consisting of over 4000 entries, and it is available with an open license[8]. We have included sentences with 0 statements in the train data, which means that those sentences are incomplete or erroneous. In this case, even sentences with multiple statements can be annotated by 0. Nonetheless, we left them in the data to create real-world experiences but removed them from the test and eval datasets. Furthermore, we achieved a similar distribution of 1-statement sentences (52%, 62%, 50%), which formed the largest class of sentences in each of the three sub-dataset. This distribution similarity was only achieved for 2-,3- and 4-statement sentences. Since 5- and 6-statement sentences are very rare and could almost be considered outliers, their distribution differed across the sub-datasets.

Column name	Description
topic	Defined by the lower-case title of the Hurraki article title.
phrase	Original phrase.
phrase_ number	Number of the phrase. "_long" indicates, that the phrase belongs to the detailed explanation
genre	Genre(s) extracted based on Hurraki's categories
timestamp	Time of the article's last modification.
phrase_ tokenized	Phrase separated in tokens. Each token is given an index for referencing.
num_ statements	Number of statements.
statement_ spans	List of statements. Each statement is a span of words, represented by their indices.
author	Pseudonymized (md5-hash) author id
notes	Optional notes by the annotators.

Table 1: Column names and description of the available information in our dataset.

4.2 Evaluation

We evaluate the two subtasks independently and use different metrics for each of the subtasks. Our evaluation scripts are published so that the performances of future systems can be reported easily[9].

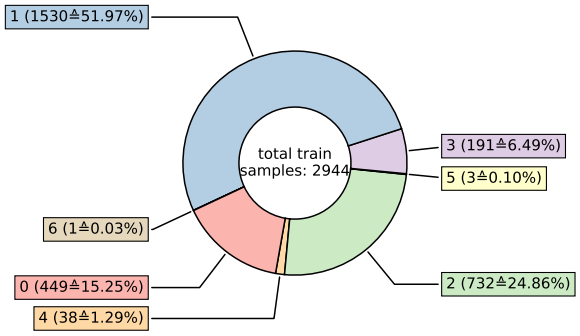
4.2.1 Subtask 1: Number of Statements

The first task is a classification problem where the participants should determine the number of statements in the sentence.

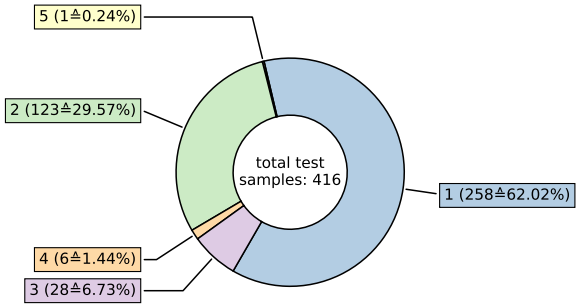
F1-Score For this classification problem, we evaluate the submissions according to precision, recall, and F1-score. As our data is heavily unbalanced,

[8] https://github.com/german-easy-to-read/statements/tree/master/data

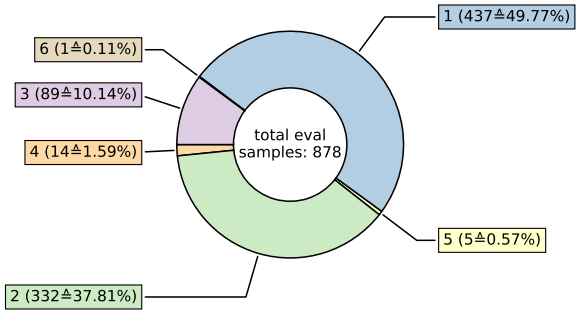
[9] https://github.com/german-easy-to-read/statements/tree/master/data/scoring



(a) train



(b) test



(c) eval

Figure 2: Distribution of the number of statements per sentence in the train, test, and eval datasets.

we averaged the F1-scores between the different classes by weighting the number of samples in the respective class.

MAE & MSE However, this evaluation judges all mispredictions equally and does account for less severe off-by-one errors. Therefore, we add an evaluation on a continuous scale, which is the mean absolute error (MAE) and the mean squared error (MSE). We considered the MAE as the most meaningful and, thus, ranked the submissions based on this metric.

4.2.2 Subtask 2: Segmentation of Statements

The evaluation of subtask 2 is more complex. If we only rewarded exact matches, the scores would be very low and could not account for almost correct annotations. Therefore, we selected chrF, which evaluates the spans based on character n-gram overlaps, and the Jaccard index, which interprets the annotations as sets and measures the overlap with the ground truth.

chrF (Popović, 2015) is designed as a machine translation evaluation metric that measures the F1-score of n-gram character overlaps between the candidate and the reference translation. For our evaluation setup, we interpret the statement spans as string sequences and calculate the overlap between the proposed statement spans and the reference annotation. For samples with more than ten tokens, the corresponding indices contain two digits that would be interpreted as separate characters during evaluation. Therefore, we replaced all numbers by single letter characters and thus obtained an equal weighting of all indexes. In addition, we sorted the spans in an ascending order to avoid lower scores due to the index order.

For the score calculation, we use the standardized implementation of the sacrebleu package (Post, 2018). We ignored whitespaces and set the character n-gram order to 6 and the word n-gram order to 0.

Jaccard index, or the Jaccard similarity coefficient, is a statistical mean used for gauging the similarity and diversity of sample sets. We adapted it, so that it is applicable for a list of sets. We iterate over all sets in the gold truth Y (with length=N) and the sets in the prediction $\hat{Y}$ (with a variable length) to calculate the Jaccard index for each of the tuples, which summed up to build the J_{list} score. To increase the robustness, we sort the sets by min, max, and median value and average the different J_{list}-values.

$$J_{list}(Y, \hat{Y}) = \sum_{n}^{N} J(y_n, \hat{y}_n) \quad (1)$$

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (2)$$

Note that by design, $0 \leq J(A, B) \leq 1$. If A intersection B is empty, then $J(A, B) = 0$.

5 Results

Three different teams participated in our shared task. While two teams submitted predictions for both subtasks, one team only submitted their system to subtask 1. The results can be seen in Table 2

for task 1 and Table 3 for task 2. All teams outperformed our baselines for both tasks. When comparing the predictions with a paired t-test against the all-1 baseline, all systems, but also our string-match baseline, achieve statistically significant improvements[10].

Team	MAE↓
KlarTextCoder	0.35
StaGE FriGHt	0.40
CUET_Big_O	0.40
Baseline: string-match	0.60
Baseline: all-1	0.65
Baseline: random	0.92

Table 2: Results Final Phase – Subtask 1

Team	chrF↑	jaccard↑
KlarTextCoder	0.36	0.29
StaGE FriGHt	0.30	0.38
CUET_Big_O	-	-
Baseline: random	0.24	0.27
Baseline: string-match	0.05	0.04
Baseline: all-1	0.00	0.00

Table 3: Results Final Phase – Subtask 2

5.1 Baselines

We provide three baselines to rank and evaluate the participant submissions. The first baseline assumes that all phrases only have one statement. Since single statement samples are the most common in our data (see Figure 2), this baseline can be seen as a majority voting system. According to our task design, we only annotate the statement spans if the number of statements in the sample is greater than one. Therefore, all statement spans are empty for this baseline.

The second baseline is a random baseline that randomly assigns a number between one and three to the number of statements. If there are at least two statements, we split the sample into the corresponding number of statement spans at random indices. The spans must be non-empty, and we split them into equal or near-equal sizes.

[10]P-values are $1.6e^{-12}$ for the string-match baseline, $9.8e^{-119}$ for CUET_Big_O, $9.16e^{-64}$ for StaGE FriGHt, and $1.1e^{-97}$ for KlarTextCoder.

The first rule of our annotation guidelines (Appendix A) separates statements via conjunctions. Our third baseline considers this rule and counts the occurrences of the conjunctions "und"(*and)*, "oder"(*or*), and "aber"(*but*). In addition, the statement spans are determined by splitting at these conjunctions without annotating the conjunctions themselves.

5.2 Proposed Systems

KlarTextCoder (Ramarao et al., 2024) integrated part-of-speech information with pre-trained BERT models. The authors also showcased alternative approaches, including a rule-based system, LLMs, and traditional machine-learning models. These machine learning models used a comprehensive feature set, combining Abstract Meaning Representation features to capture deep semantic structures, part-of-speech (POS) tags for syntactic information, and other linguistic features.

StaGE FriGHt (Säuberli and Bodenmann, 2024) combined sequence labeling with dependency parsing. They used a fine-tuned BERT model to predict the head token of each statement span and expands the span using dependency relations.

CUET_Big_O upload a final submission entitled "Fine Tuned BERT". Unfortunately, we did not receive any information from the participants on how their system worked.

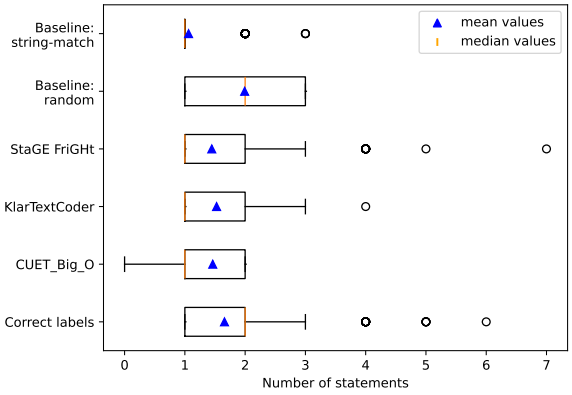


Figure 3: Distribution of the predicted number of statements.

5.3 Error Analysis

We further analyzed the submitted predictions. Figure 3 shows their distribution of the number of statements. The correct samples and the random baseline have the highest means, being the only

systems with a median value of two. Especially the submission by CUET_Big_O overfits on the single statements samples, but also the other submissions have a bias toward the lower number of samples. Only the submission by StaGE FriGHt predicts more than four statements.

Furthermore, we analyzed in which cases the systems failed to predict the correct number of samples. For this, we compared the cases with our annotator agreement. Since two annotators reviewed the samples independently, we can identify challenging or non-trivial cases by their disagreement. For KlarTextCoder, 68.87% of the misclassified samples had a disagreement within the annotators. For the other two teams, these numbers were even higher, with 79.80% for StaGE FriGHt and 76.60% for CUET_Big_O. These numbers are higher than the overall annotator disagreement, indicating that the systems especially fail at the harder samples.

6 Discussion & Limitations

This shared task does not claim to solve the problem of statement segmentation conclusively. The main aim was to create the basis for practical applications and to ignite a scientific discussion. The most apparent limitation is the focus on mostly syntactic structures and often neglegting semantic relations and interpretation. This weakness becomes evident through a few examples.

Our statements focus mostly on chunks of information that need to be understood by the readers. Hence, we split sentences like “Eine Angel ist eine Schnur mit einem Haken.” (*A fishing rod is a string with a hook.*) into two statements. From a content perspective, a fishing rod always consists of a string and a hook, so splitting these into two statements makes no sense. However, the reader still has to parse two separate statements independently: 1) there is a string, and 2) there is a hook. Therefore, we decided to split the sentence into two statements.

In contrast to this, we also discussed a few specifics, such as separated compound nouns, named entities, and dates, and combined them under the umbrella term "semantic units". For instance, “Berliner Fernsehturm ist ein Wahrzeichen der Stadt.” (*Berlin’s TV tower is a landmark of the city.*) in which only the combination of the two terms "Berlin’s" and "TV tower" combined form the semantic unit behind the statement, i.e., one noun phrase.

Similarly, separated compound nouns, such as “Die Religion der Aleviten” (*Religion of the alevites.*) are not considered separate but as one named entity since it is synonymously with “Alevitentum” (*alevism*). Another case where semantics become very prominent is the case of idiomatic expressions and whether they should be treated as independent statements, as compound nouns, or as regular phrases. We made the practical decision to postpone this discussion, and we, therefore, annotated every sentence in which idioms occur as components with "0" (e.g., “Das A und O ist eine Redewendung” (*The alpha and omega is an idiom.*)).

Our shared task focuses on German Easy Language. While there are many similarities between Easy Language and standard German, our annotations cover the specifics of Easy Language. In Easy Language, writers are motivated to add explanations (section 5.2.2 in DIN-Normenausschuss Ergonomie 2023) to more complex terms. Therefore, there is an overamount of phrases like “das heißt” (*that means*) and “some say” (*manche sagen*). They most often function as an introductory phrase before the actual explanation, and thus, they don’t convey information on their own. Hence, we ignored them in our statement annotations. Due to the focus on Easy Language specifics, the results cannot be directly transferred to standard German without further adaptions.

7 Conclusion

In the course of this shared task, we have presented both data and possible solutions to the problem of statement segmentation. Even if these results do not solve the problem conclusively, they nevertheless provide a valuable initial contribution to the discussion on the problem. We hope that future researchers will be able to draw further insights based on this foundation. The competition on codabench[11] is re-opened for further experiments and an easy comparison to the previous systems.

Ethics statement

Since Easy Language has a vulnerable target group, special care should always be taken in this research field. As we do not create new texts in this task but merely conduct analyses on texts that have already been written, we do not see any direct risk of misuse or ethical implications. The results can

[11] https://www.codabench.org/competitions/3244/

be applied directly to German Easy Language, but we expect they could be generalized and used for simple languages from other nations as well.

Acknowledgments

We are grateful to all the participants, whose enthusiastic participation made our shared task a great success.
This research has been partially funded by the German Federal Ministry of Education and Research (BMBF) through grant 01IS23069 Software Campus 3.0 (Technical University of Munich) as part of the Software Campus project "LIANA".
Thorben Schomacker received external funding from the IFB Hamburg as well as from the European Education and Culture Executive Agency.

References

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

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

Ursula Bredel and Christiane Maaß. 2016. *Leichte Sprache theoretische Grundlagen, Orientierung für die Praxis*. Sprache im Blick. Dudenverlag.

Oscar M. Cumbicus-Pineda, Itziar Gonzalez-Dios, and Aitor Soroa. 2021. Linguistic Capabilities for a Checklist-based evaluation in Automatic Text Simplification. In *Proceedings of the First Workshop on Current Trends in Text Simplification (CTTS 2021)*, pages 70–83. CEUR-WS.

Silvana Deilen. 2020. Using eye-tracking to evaluate language processing in the easy language target group deilen/schiffl. *Easy language research: Text and user perspectives*, 2:273.

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.

DIN-Normenausschuss Ergonomie. 2023. Empfehlungen für Deutsche Leichte Sprache (DIN SPEC 33429).

Leon Fruth, Robin Jegan, and Andreas Henrich. 2024. An approach towards unsupervised text simplification on paragraph-level for German texts. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024*, pages 77–89, Torino, Italia. ELRA and ICCL.

Natalia Grabar and Horacio Saggion. 2022. Evaluation of automatic text simplification: Where are we now, where should we go from here. In *Actes de la 29e Conférence sur le Traitement Automatique des Langues Naturelles. Volume 1 : conférence principale*, pages 453–463, Avignon, France. ATALA.

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*, 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, Toronto, Canada. 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.

Antoine Jamelot, Solen Quiniou, and Sophie Hamon. 2024. Improving text readability through segmentation into rheses. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 8925–8930, Torino, Italia. ELRA and ICCL.

Joongwon Kim, Mounica Maddela, Reno Kriz, Wei Xu, and Chris Callison-Burch. 2021. BiSECT: Learning to split and rephrase sentences with bitexts. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6193–6209, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

David Klaper, Sarah Ebling, and Martin Volk. 2013. Building a German/simple German parallel corpus for automatic text simplification. In *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 11–19, Sofia, Bulgaria. Association for Computational Linguistics.

Ruben Klepp. 2022. Klassifizierung der Textkomplexität von Chatbot-Antworten mittels Transformer-Modellen. Master thesis, Hochschule Darmstadt, Germany.

Lars Klöser, Mika Beele, Jan-Niklas Schagen, and Bodo Kraft. 2024. German text simplification: Finetuning large language models with semi-synthetic data. In *Proceedings of the Fourth Workshop on Language Technology for Equality, Diversity, Inclusion*, pages 63–72, St. Julian's, Malta. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. LENS: A learnable evaluation metric for text simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16383–16408, Toronto, Canada. Association for Computational Linguistics.

Margot Madina, Itziar Gonzalez-Dios, and Melanie Siegel. 2023. Easy-to-read language resources and tools for three european languages. In *Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '23, page 693–699, New York, NY, USA. Association for Computing Machinery.

Margot Madina, Itziar Gonzalez-Dios, and Melanie Siegel. 2024. Towards Reliable E2R Texts: A Proposal for Standardized Evaluation Practices. In *Computers Helping People with Special Needs: 19th International Conference, ICCHP 2024, Linz, Austria, July 8–12, 2024, Proceedings, Part II*, pages 224–231, Berlin, Heidelberg. Springer-Verlag.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

Salar Mohtaj, Babak Naderi, and Sebastian Möller. 2022. Overview of the GermEval 2022 shared task on text complexity assessment of German text. In *Proceedings of the GermEval 2022 Workshop on Text Complexity Assessment of German Text*, pages 1–9, Potsdam, Germany. Association for Computational Linguistics.

Netzwerk Leichte Sprache. 2022. Die Regeln für Leichte Sprache. `https://www.leichte-sprache.org/wp-content/uploads/2017/11/Regeln_Leichte_Sprache.pdf`. [Online; Last Update: *n/a*; Last Access: 2024-07-29].

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

Christina Niklaus, André Freitas, and Siegfried Handschuh. 2019b. MinWikiSplit: A sentence splitting corpus with minimal propositions. In *Proceedings of the 12th International Conference on Natural Language Generation*, pages 118–123, Tokyo, Japan. Association for Computational Linguistics.

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

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Akhilesh Kakolu Ramarao, Wiebke Petersen, Anna Sophia Stein, Emma Stein, and Hanxin Xia. 2024. KlarTextCoders at StaGE: Automatic Statement Annotations for German Easy Language. In *Proceedings of the GermEval 2024 Shared Task on Statement Segmentation in German Easy Language (StaGE)*, Vienna, Austria. Association for Computational Linguistics.

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

Andreas Säuberli and Niclas Bodenmann. 2024. Statement Segmentation for German Easy Language Using BERT and Dependency Parsing. In *Proceedings of the GermEval 2024 Shared Task on Statement Segmentation in German Easy Language (StaGE)*, Vienna, Austria. Association for Computational Linguistics.

Andreas Säuberli, Sarah Ebling, and Martin Volk. 2020. Benchmarking data-driven automatic text simplification for German. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with*

REAding DIfficulties (READI), pages 41–48, Marseille, France. European Language Resources Association.

Andreas Säuberli, Franz Holzknecht, Patrick Haller, Silvana Deilen, Laura Schiffl, Silvia Hansen-Schirra, and Sarah Ebling. 2024. Digital Comprehensibility Assessment of Simplified Texts among Persons with Intellectual Disabilities. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, CHI '24, New York, NY, USA. Association for Computing Machinery.

Thorben Schomacker, Michael Gille, Marina Tropmann-Frick, and Jörg von der Hülls. 2023. Data and approaches for German text simplification – towards an accessibility-enhanced communication. In *Proceedings of the 19th Conference on Natural Language Processing (KONVENS 2023)*, pages 63–68, Ingolstadt, Germany. Association for Computational Lingustics.

Thorben Schomacker, Michael Gille, Marina Tropmann-Frick, and Jörg von der Hülls. 2024. Self-regulated and Participatory Automatic Text Simplification. In *Applied Machine Learning and Data Analytics*, Communications in Computer and Information Science, pages 264–273, Cham. Springer Nature Switzerland.

Matthew Shardlow, Fernando Alva-Manchego, Riza Batista-Navarro, Stefan Bott, Saul Calderon Ramirez, Rémi Cardon, Thomas François, Akio Hayakawa, Andrea Horbach, Anna Hülsing, Yusuke Ide, Joseph Marvin Imperial, Adam Nohejl, Kai North, Laura Occhipinti, Nelson Peréz Rojas, Nishat Raihan, Tharindu Ranasinghe, Martin Solis Salazar, Sanja Štajner, Marcos Zampieri, and Horacio Saggion. 2024. The BEA 2024 shared task on the multilingual lexical simplification pipeline. In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 571–589, Mexico City, Mexico. Association for Computational Linguistics.

Advaith Siddharthan. 2014. A survey of research on text simplification. *ITL - International Journal of Applied Linguistics*, 165(2):259–298. Publisher: John Benjamins.

Melanie Siegel, Dorothee Beermann, and Lars Hellan. 2019. Aspects of Linguistic Complexity: A German – Norwegian Approach to the Creation of Resources for Easy-To-Understand Language. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*, pages 1–3. IEEE.

Nicolas Spring, Annette Rios, and Sarah Ebling. 2021. Exploring German multi-level text simplification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1339–1349, Held Online. INCOMA Ltd.

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

Sanja Štajner, Horacio Saggio, Matthew Shardlow, and Fernando Alva-Manchego, editors. 2023. *Proceedings of the Second Workshop on Text Simplification, Accessibility and Readability*. INCOMA Ltd., Shoumen, Bulgaria, Varna, Bulgaria.

Regina Stodden. 2024. Reproduction of German text simplification systems. In *Proceedings of the Workshop on DeTermIt! Evaluating Text Difficulty in a Multilingual Context @ LREC-COLING 2024*, pages 1–15, Torino, Italia. ELRA and ICCL.

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

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Semantic structural evaluation for text simplification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 685–696, New Orleans, Louisiana. Association for Computational Linguistics.

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*, pages 279–287.

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

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

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

Team	Task 1					Task 2	
	MAE↓	MSE↓	Prec↑	Rec↑	F1↑	chrF↑	jaccard↑
KlarTextCoder	**0.35**	**0.41**	**0.65**	**0.68**	**0.66**	**0.36**	0.29
StaGE FriGHt	0.40	0.55	**0.65**	0.66	0.64	0.30	**0.38**
CUET_Big_O	0.40	0.5	0.57	0.64	0.61	-	-
Baseline: string-match	0.60	0.91	0.53	0.53	0.41	0.05	0.04
Baseline: random	0.92	1.41	0.38	0.31	0.33	0.24	0.27
Baseline: all-1	0.66	1.05	0.25	0.50	0.33	0.00	0.00

Table 4: Detailed results. The F1-scores averages are weighted by the support of the respective class

A Annotation guidelines v0.1

Our data samples are single sentences. We split sentences at punctuation marks. Hence, they can span over several lines. We annotate the sentences by the number of statements contained in them. The tokens in the sentence are separated by whitespaces. A statement only contains full tokens; we don't split tokens for subtoken annotations. If a sentence spans several lines (as typical for Leichte Sprache), the line breaks are displayed with "\newline". Newlines could help classification models in the future, and thus, they are given as separate tokens but should be ignored during annotation.

We do not correct any typos or parsing errors. Thus, the samples can contain punctuation or grammatical errors. If the errors are severe, e.g., if two sentences are merged into one due to missing punctuations, these samples are annotated as erroneous. The annotations process is split into two parts, similar to our subtasks. First, we determined the number of statements and then annotated the statement spans.

A.1 Determining the number of statements / Statement separation

Ideally, a statement is formed by a Subject-Verb-Object (SVO) combination or additional adjectives or prepositional constructions that could form a separate sentence. Example: "Die moderne Sportart heißt: \newline Splashdiving." (*The modern sport is called: \newline Splashdiving*) contains 2 statements, one for the main SVO combination "Die Sportart heißt Splashdiving", and one for the additional adjection "moderne". The sentences don't have to be in proper SVC order, as subclauses can form a statement without any main clause. As a rule of thumb, one statement contains only one full verb at maximum (yet auxiliary and full verbs can be together).

A sentence is split into multiple sentences based on the criteria discussed in the following subsections.

A.1.1 Separating via conjunctions

Conjunctions are used to connect clauses within a sentence. There are two types of conjunctions: Coordinating and subordinating conjunctions. Coordinating conjunctions connect two main clauses. Subordinating conjunctions, in contrast, connect a main clause with a subordinate clause, emphasizing the main clause more than the subordinate clause. Both clauses form separate statements in our annotations.

Special case: "Manche sagen" (*Some say*) and "Das heißt" (*That means*) "Das heißt, .." is a common construction in Leichte Sprache. These phrases are not counted as separate statements. However, if the phrase contains additional information, such as "Lügen heißt.." (*Lying means*) or "Manche Politiker sagen.." (*Some politicians say*), these phrases are annotated as usual. Only the very specific formulations above are an exception.

Special case: Parentheses Information in parentheses is always a separate statement.

A.1.2 Separating via single adjectives/adverbials

Nouns are often further described by adjectives. If these adjectives give additional information or restrict the noun to a subgroup, these adjectives form a separate statement. In the first example in Table 5, the adjective modern gives additional information about the sport, so it is extracted as a new statement. In contrast, if the adjective/adverb gives no further restrictions, it is not counted as a separate statement. If multiple adjectives/adverbs are concatenated into a sequence, each of them forms a single statement (see Table 5).

Sentence	Translation	# stmts	Explanation
Die moderne Sportart heißt: \newline Splashdiving.	The modern sport is called: \newline Splashdiving.	2	"modern" gives additional information about the sport
Der Vorschlag wurde ohne Gegenstimmen angenommen.	The proposal was accepted without any opposing votes.	2	Modal adverbial "ohne Gegenstimmen"
Die Einwohner konnten ihre Lebensmittel ganz normal kaufen.	The inhabitants were able to buy their food as normal.	1	"ganz normal" gives no restrictions
Die Mitarbeiter reparieren gemeinsam kaputte Dinge.	The employees repair broken things together.	2	Sequence: The employees repair broken things. The employees repair things together.

Table 5: Example sentences and explanations for their annotated number of statements.

Special case: Quantifiers Quantifiers like "viele" (*many*), "alle" (*all*), "mache" (*some*) or counts like "50 Menschen" (*50 people*) are no separate statements.

Special case: Filler words Filler words like "sehr" (*very*), "oft" (*often*), "darum" (*therefore*), or "auch" (*also*) are no separate statements.

Special case: Comparatives and superlatives Comparatives and superlatives in constructions like "größte Stadt" (*biggest city*) or "weniger Menschen" (*fewer people*) are closely connected to their noun and, thus, form no separate statement.

A.1.3 Separating via prepositional phrase

Prepositional often specify things and actions, e.g., by indicating ownership ("von .." (*by*)), a direction ("nach .." (*to*)), or modality ("mit .." (*with*)). Similar to adjectives, this additional information forms a new statement. For example, the sentence "Man kann mit einem Fahr-stuhl nach oben fahren." (*You can go to the top in an elevator.*) contains the prepositional phrases "mit einem Fahrstuhl" and "ach oben", thus containing two statements. If multiple prepositional phrases are stacked after another, each of them forms a separate statement.

Special case: Composites The German language gives the option to stack multiple pieces of information into one word (so-called composite words). Prepositional phrases that can be rephrased to a single word (e.g., "Religion der Christen" (*religion of the Christians*) → "Christentum" (*Christianity*); "Mitglied in der Partei" (*member of the party*) → "Parteimitglied" (*party member*)) are no additional statement.

Speical case: Trivial prepositions Sometimes, the verb already contains the information in the prepositional phrase (e.g., "in die Luft sprengen" (*lit: blow up into the air*)). Then, we don't annotate the prepositional phrases as separate statements.

Special case: Date and year specifications Dates are annotated as separate statements except if the remaining statement only states that something exists or was born. For example, in the sentence "Im Jahr 1877 heiraten Alexander Graham Bell und Mabel Hubbard." (*In 1877, Alexander Graham Bell and Mabel Hubbard marry.*), the year and the marriage itself are two different statements. However, in the sentence "Angela Merkel wurde am 17. Juli 1954 geboren." (*Angela Merkel was born on July 17, 1954.,*), the information that she was born is trivial. Thus, the sentence contains only 1 statement. In contrast, abstract time information like "manchmal"sometimes or "immer" (*always*) is no separate statement.

A.1.4 4. 0-statement sentences or un-parseable sentences (= erroneous samples)

There can be samples that don't contain a statement, e.g., if there are problems with the preprocessing and there are multiple sentences in the sample. Also, sentences that don't contain a verb, like "An der Universität Boston." (*At the university of Boston.*), are annotated with 0.

Sentence	Translation	# stmts	Spans	Explanation
0:=Die 1:=moderne 2:=Sportart 3:=heißt: 4:= 5:=\newline 6:=Splashdiving.	0:=The 1:=modern 2:=sport 3:=is called: 5:=\newline 6:=Splashdiving.	2	[[1], [3,6]]	Tokens 0, 4, 5 are never annotated. Token 2 (Sportart) is in both spans, so it is not stated explicitly.
0:=Der 1:=Vorschlag 2:=wurde 3:=ohne 4:=Gegenstimmen 5:=angenommen.	0:=The 1:=proposal 2:=was 5:=accepted 3:=without 4:=opposing votes.	2	[[1,2,5], [3,4]]	No overlaps between statements
0:=Die 1:=Einwohner 2:=konnten 3:=ihre 4:=Lebensmittel 5:=ganz 6:=normal 7:=kaufen.	0:=The 1:=inhabitants 2:=could 7:=buy 3:=their 4:=food 5:=as 6:=normal.	1		No spans for sentences with one statement
0:=Die 1:=Mitarbeiter 2:=reparieren 3:=gemeinsam 4:=kaputte 5:=Dinge.	0:=The 1:=employees 2:=repair 4:=broken 5:=things 3:=together.	2	[[3], [4,5]]	We identified two statements in this sentence (“0:=Die 1:=Mitarbeiter 2:=reparieren 3:=gemeinsam.” and “0:=Die 1:=Mitarbeiter 2:=reparieren 4:=kaputte 5:=Dinge.”). The tokens 2 and 3 (“Mitarbeiter reparieren”) are contained in both statements, so we don’t include them in the annotations. The tokens 3 (“gemeinsam”) and 4,5 (“kaputte Dinge”) are independant, and hence, form the two statement spans.

Table 6: Example sentences and their statement span annotations. The indexes of the English translations are use the same as the German original. Hence, they are not in ascending order and more than one English word may be related to one German word.