BATS: BenchmArking Text Simplicity 🦇

Christin Katharina Kreutz[1,2], **Fabian Haak**[3], **Björn Engelmann**[3], **Philipp Schaer**[3]
[1]TH Mittelhessen - University of Applied Sciences, Germany
[2]Herder Institute for Historical Research on East Central Europe, Germany
[3]TH Köln - University of Applied Sciences, Germany
christin.katharina.kreutz@mni.thm.de, fabian.haak@th-koeln.de,
bjoern.engelmann@th-koeln.de, philipp.schaer@th-koeln.de

Abstract

Evaluation of text simplification currently focuses on the difference between a source text and its simplified variant. Datasets for this evaluation base on a specific topic and group of readers for which is simplified. The broad applicability of text simplification and specifics that come with intended target audiences (e.g., children compared to adult non-experts) are disregarded. An explainable assessment of the overall *simplicity* of text is missing.

This work is **B**enchm**A**rking **T**ext **S**implicity (BATS): we provide an explainable method to assess practical and concrete rules from literature describing features of simplicity and complexity of text. Our experiments on 15 datasets for text simplification highlight differences in features that are important in different domains of text and for different intended target audiences.

1 Introduction

Text simplification aims to reduce a text's difficulty and adapt it to a reader's abilities (Arfe et al., 2018). Difficulties in fully grasping information excludes individuals from actively participating in society, such as politics, education, or choosing between healthcare options (Štajner et al., 2022). Simplifying text can help overcome this obstacle.

Different groups of people (*target audiences*, short *TAs*) can profit from text simplification, such as children (Štajner et al., 2020b) or non-experts (Kintsch, 1994). These different groups have distinct needs for text in order to be considered simple (Xu et al., 2015). The *domain* (short *D*) of the text that is simplified influences the factors that constitute good simplification (Štajner et al., 2020b, 2022). For example, when simplifying news articles, reducing the level of detail might be helpful as long as the general message stays the same. In contrast, in other domains (e.g., medical texts), preserving all information is imperative (Shardlow and Nawaz, 2019).

Historically, text simplification was performed by schooled experts. However, advances in computer technology, such as the introduction of (transformer-based) large language models, have enabled the development of highly effective automated text simplification (Engelmann et al., 2023) that can compete with manual efforts. Automation and reduced cost, expertise- and time requirements allow for a much broader application of text simplification (Štajner et al., 2022).

Evaluating the quality of simplified text is a task coinciding with the introduction of automation of text simplification. Aside from manual evaluation, text can be automatically evaluated by either benchmarking against reference simplifications or using metrics or models. Readability scores such as the Flesch Reading Ease (Flesch, 1948) are often described as unsuitable for quantifying the level of simplicity of a text as the measure does not take into account sufficient aspects of what constitutes simplicity. While manual evaluation is labor intensive, metrics often lack precision and poorly approximate human judgments (Heineman et al., 2023; Alva-Manchego et al., 2021). Recent automatic evaluation approaches using elaborate language models better correlate with human judgments on texts' simplification level but require a high-quality dataset of labeled simplifications for best training results (Maddela et al., 2023; Heineman et al., 2023).

Current datasets for evaluating metrics do not indicate if a text is simple. Instead, they quantify how much more accessible a text has become by it being simplified compared to its complex form (Sulem et al., 2018; Alva-Manchego et al., 2020; Alva-Manchego et al., 2021; Scialom et al., 2021; Maddela et al., 2023). Therefore, metrics correlating to the human labels from these datasets do not indicate the overall *simplicity* of a text but rather its

Findings of the Association for Computational Linguistics: ACL 2024, pages 11968–11989
August 11-16, 2024 ©2024 Association for Computational Linguistics

degree of *simplification*. All these datasets belong to the same domain and thus potentially ignore specificities found only in certain areas.

This work focuses on **B**enchm**A**rking **T**ext **S**implicity (BATS). We present a reference-free and explainable method to identify characteristics indicating simplicity and complexity of text. Our approach does not directly quantify simplicity but rather provides insights into what properties might make a text simple or complex. These characteristics relate to features of simple text which can be found in the literature. BATS is target audience- and domain-independent. Using BATS, we evaluate three research questions:

RQ_1 Which characteristics from literature reflect the simplicity or complexity of texts?

RQ_2 Which dataset-specific, target audience-specific, and domain-specific characteristics can be found regarding simplicity?

RQ_3 Can BATS be used to quantify simplicity effectively?

Our code and data are publicly available at GitHub[1] under MIT license.

2 Text Simplification Evaluation

2.1 Criteria for Simple Text

A prerequisite for simplifying text and evaluating it is being able to differentiate between complex and simple text (Gooding, 2022). Few studies investigate aspects that constitute simple language. Štajner et al. (2015) explore the correlations between readability measures and linguistic features. In a more theoretically-driven approach, Arfe et al. (2018) describe linguistic aspects of texts accounting for the readability of informational texts.

What makes text simple cannot be universally defined. While expert readers can connect new information to prior knowledge via several routes, information in text not containing context retrieval cues (e.g., definitions of technical terms) stays unavailable for low-knowledge readers (Kintsch, 1994). A straightforward linguistic structure is not helpful for poor readers, but very poor readers need it (Arfe et al., 2018). For language learners, readability and understandability of text depend on their native language and proficiency level of a learned language (Štajner et al., 2022).

2.2 Domain and Target Audience

Aspects which are essential for developing and evaluating text simplification depend on the domain (e.g., medical (Van et al., 2020), news (Vajjala and Lučić, 2018), legislation (Scarton et al., 2018)) and the intended target audience (e.g., children (Barzilay and Elhadad, 2003), non-experts (Kauchak et al., 2022), language learners (Vajjala and Lučić, 2018), persons with language or intellectual impairments (Štajner et al., 2020b)) for which is simplified (Xu et al., 2015; Siddharthan, 2014; Feng et al., 2009). Most text simplification approaches do not consider either (Gooding, 2022).

Domain. The distribution of values of quantifiable features of texts of comparable complexity significantly differs depending on a text's genre (Sheehan, 2013), e.g., preserving the meaning of texts is crucial for medical texts (Shardlow and Nawaz, 2019). However, Shardlow and Nawaz (2019) found that nearly 25 percent of critical information from clinical texts is lost when simplified by automated text simplification. Sheehan et al. (2014) compare 43 textual features across two genres. They found significant differences in measures of academic vocabulary, argumentation, concreteness, lexical cohesion, conversational style, and degree of narrativity between domains.

Target Audience. There are differences in desired properties of simplified text depending on the recipient, e.g., non-native speakers require the simplification of specific words (Štajner et al., 2020b, 2022) whereas children and language-learners require short sentences (Štajner and Hulpus, 2018). For business-oriented readers searching for information, oversimplification is no problem, whereas for children or non-native speakers, oversimplification might lead to disengagement (Štajner et al., 2020b). Feng et al. (2009) consider adults with intellectual disabilities as target audience. They define a list of features of text simplified for this audience. Using datasets with children as the target audience, they calculate the values for the features for simplified and source texts to determine features with significant differences.

2.3 Evaluating Text Simplification

Datasets. Currently, text simplification approaches are only evaluated on very limited data with human labels: simplification-acl (Sulem et al., 2018), ASSET (Alva-Manchego et al., 2020),

[1] github.com/bjoernengelmann/BATS

METAeval (Alva-Manchego et al., 2021), QuestEval (Scialom et al., 2021) as well as SimpEval$_{Past}$ and SimpEval$_{2022}$ (Maddela et al., 2023). All (transitively) base on TurkCorpus (Xu et al., 2015, 2016a) or Wikipedia directly. The human labels have been constructed in pairwise comparisons of complex source texts and their simplified versions. Annotators rated if the simplified text is easier to understand than the original text. Therefore, the datasets do not provide scores indicating the overall simplicity of text but rather describe the degree of simplification that has been performed between the original and simplified version.

TurkCorpus contains text from English Wikipedia and Simple English Wikipedia, as well as manually generated simplifications. Its domain is general knowledge and encyclopedia; the target audiences are the general population, children, and language learners.

Measures and Approaches. Human evaluation would be the most desirable assessment to ensure quality of text simplification approaches but cannot be performed at scale. Current automatic measures for text simplification approximate different aspects of human judgments of the degree of simplification between an original and a modified version of text: SARI (Xu et al., 2016b) describes the overlap between a modified text (e.g., system-simplified text) with its original (e.g., complex text) and ideal modifications (e.g., gold simplifications). The token-based measure considers added, deleted and kept tokens. While it serves as a decent measure of the quality of the simplification, especially if a larger number of ideal texts is provided, it tends to perform worse when the modified text is vastly different from the original one (Alva-Manchego et al., 2020). BERTscore (Zhang et al., 2020) captures the semantic similarity between an original and a modified text. Here, grammar or correctness of text are not considered. LENS (Maddela et al., 2023) quantifies the semantic similarity and performed edits between a modified text, its original and ideal modifications. Its correlation to human judgments is better than SARI and BERTscore. LENS-SALSA (Heineman et al., 2023) extends LENS to an edit-level reference-free simplification metric. SALSA is an evaluation framework based on edit types between original and modified texts.

Several approaches quantify simplicity or complexity of text: TextEvaluator (Sheehan et al., 2014) considers 43 features quantifying complexity and additional ones indicating the genre of a text. CoCo (Štajner et al., 2020a) uses a knowledge graph (here DBpedia Spotlight) to simulate concepts activating neighboring concepts in a reader's working memory following the ideas of Kintsch and van Dijk (1978) and produces a value indicating the text's conceptual complexity. The approach assumes that text is less conceptionally complex and more accessible if more of the text's concepts are activated. Lexile Analyzer[2] is closed-source software describing the reading demand and complexity of texts through a score.

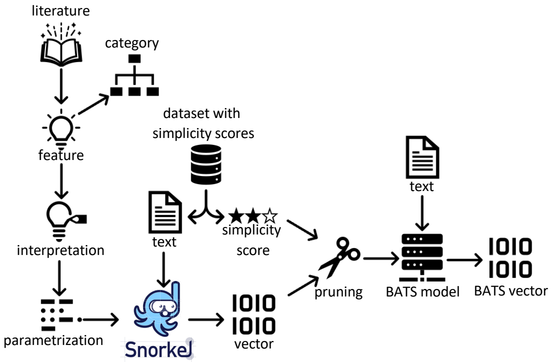

Figure 1: Simplified overview of our approach.

With our approach, we provide another way of assessing the simplicity or complexity of text, fostering characteristics of simple text described in the literature. Our method does not require ideal simplifications or annotated corpora to be used; it can be applied to arbitrary texts.

3 BATS

Figure 1 gives a simplified overview of our approach. We implement rules from literature via Snorkel (Ratner et al., 2017) to determine if these rules appear in a dataset consisting of texts. The rules express heuristics that linguistic experts consider to be salient characteristics of the measure of text simplicity. After applying these rules, binary vectors are obtained. These vectors combined with simplicity scores of texts can then be used to prune the rules to only keep those highly indicative of simplicity and complexity (**BATS model**). We can construct **BATS vectors** for any text with these pruned rules. These vectors indicate which characteristics of simplicity or complexity are satisfied.

[2] https://hub.lexile.com/analyzer

3.1 Data Programming Approach

The data programming paradigm makes it possible to develop a model for a given classification task that does not require any training data by integrating domain knowledge. Instead of generating labels with great effort, this approach defines heuristics that map domain knowledge and use characteristics of the instances to be classified to contribute signals to the classification (Ratner et al., 2016). The definition is made via a so-called *labeling functions* and integrates the basis for the heuristic decision. E.g., heuristics for detecting spam emails could check whether a candidate email contains suspicious words, many grammatical errors, or generic greetings. Individual heuristics of this type alone would not allow satisfactory discrimination. However, combining such partially overlapping heuristics and weighting them leads to better results.

We use Snorkel to define heuristics that distinguish simple from non-simple texts. In our case, the outputs of a single labeling function are either *simple* and *abstain* or *not simple* and *abstain*. Therefore, a single labeling function can only provide a signal for either simple or not simple. The output of a function is 1 if a characteristic was recognized in the text and 0 if not. The intuition here is that a vote should only be given for a class if a particular characteristic appears in the text. In this way, it is possible to create an explainable model that represents signals of the heuristics of texts with a binary vector. Each vector dimension then represents whether or not a predefined labeling function has recognized a characteristic in the text. Dimensions can be combined into groups of similar meaning. Instead of classifying texts based on these vectors, we use this type of modeling as a feature extraction step in texts. This ensures the traceability of the representations and allows for the possibility of integrating knowledge from the literature.

3.2 Characteristics of Simple Text

The literature states numerous rules or guidelines that could indicate the simplicity of a text. We focused on the 37 features mentioned in Table 1. The opposite of the given features could also be seen as an indicator of complexity, i.e., if a text does not have short sentences, it could be complex. Except for the case of *few words containing more than eight characters*, the literature usually does not describe how a feature should be interpreted. For example, *short sentences* could refer to the number of characters, syllables, or words in a sentence. Therefore, we consider multiple **interpretations** of these features to indicate texts possessing characteristics of simplicity and complexity. In total, we consider 135 interpretations of these features. Additionally, an exact **parametrization** of features is mostly missing; *short* could mean less than five, ten, or twenty words. To compensate for this, we consider different thresholds, some relying on numbers found in literature (see Appendix A.1) others shaped through discussion in the development team, resulting in 1,249 parametrizations (560 to identify

average lexical richness (Štajner et al., 2020a)
average number of words before the main verb (Sheehan et al., 2014)
few cases with max distance between 2 appearances of same entity (Štajner et al., 2020b)
few content words (Scarton et al., 2018)
few infrequent words (Štajner and Hulpus, 2018)
few long words (Arfe et al., 2018)
few modifiers (Narayan and Gardent, 2014)
few negations (Sheehan et al., 2014)
few noun phrases (Arfe et al., 2018)
few past perfect verbs (Sheehan et al., 2014)
few past tense aspect verbs (Sheehan et al., 2014)
few punctuation marks (Saggion et al., 2015)
few relative-clauses (Arfe et al., 2018)
few sentences (Arfe et al., 2018)
few third person singular pronouns (Sheehan et al., 2014)
few unique entities (Štajner et al., 2020b)
few words containing more than eight characters (Sheehan et al., 2014)
few words from academic word list (Sheehan et al., 2014)
few words per sentence (Scarton et al., 2018)
grammatical correctness (Xu et al., 2015)
high average distance between consecutive entities (Štajner et al., 2020b)
high concreteness (Scarton et al., 2018)
high Flesch reading ease (Scarton et al., 2018)
high imageability (Scarton et al., 2018)
high percentage of vocabulary learned in initial stages of foreign language learning (Tanaka et al., 2013)
low age of acquisition (Scarton et al., 2018)
low average number of unique entities (Štajner et al., 2020b)
low avg distance between all pairs of same entities (Štajner et al., 2020b)
low depth of the syntactic tree (Štajner et al., 2020a)
low entity to token ratio (Štajner et al., 2020b)
low Flesch-Kincaid Grade Level Index (Narayan and Gardent, 2014)
low unique entities to total number of entities ratio (Štajner et al., 2020b)
no appositions (Narayan and Gardent, 2014)
no conditional clauses (Arfe et al., 2018)
no conjunctions (Arfe et al., 2018)
no passive voice (Arfe et al., 2018)
short sentences (Arfe et al., 2018)

Table 1: Simplicity-inducing features for BATS.

complex texts, 689 to identify simple texts).

Our 37 features describe rules text might follow to be considered simple. We relate these rules to general rules for writing controlled English by considering the descriptions of **categories** provided by O'Brien (2003). Two annotators assigned our features together to the four categories *lexical*, *structural*, *syntactic*, and *pragmatic* with their respective sub-categories if applicable. They discussed each feature until they agreed on a category. Table 4 holds our complete list of features, (sub-)categories, and interpretations; the parametrizations can be found in our code in the supplementary material.

3.3 From Vectors to BATS Vectors

Thus, our vectors resulting from the Snorkel step are 1,249-dimensional binary vectors. Each dimension indicates whether the associated labeling function (indicating simplicity or complexity of text) detected the characteristics implemented in the parametrization or not. The features we built our parametrizations upon stem from literature possibly focusing on specific target groups (e.g., a low number of sentences would be helpful for people with an intellectual disability (Arfe et al., 2018)), our parametrizations contain quasi-arbitrary values (e.g., defining the complexity indicating variant of the simplicity-indicating *few sentences* as texts having five or more sentences).

A pruning then gathers the rules, which are important and worth further consideration. It filters out niche rules and ones that do not discriminate simple from complex texts. We suggest pruning by utilizing a dataset with simplicity scores for texts[3] and excluding all dimensions that are not at least weakly correlated with simplicity quantification of text. The resulting dimensions compose the **BATS model**. We additionally have a mapping of dimensions from one representation to all others, e.g., we know which parametrizations are implementing a specific feature from the literature.

4 Evaluation

We evaluate the overall research questions RQ_{1-3}, described in the introduction.

4.1 Datasets, Pruning, BATS Vectors

For our evaluation, we use the 15 publicly available datasets containing English texts described[4] in Table 2. We only selected datasets with confirmed availability such as OneStopEnglish, for other datasets where access had to be requested we were not able to confirm availability or even acquire them (e.g., Newsela).

Dataset	# texts	Target Audiences (TA)	Domains (D)
ASSET	4718		
AutoMeTS	6994		medical
BenchLS	1856		
Britannica	926	children	encyclopedia
EW-SEW-Turk	1000		encyclopedia
HutSSF	652		news
METAeval	604		encyclopedia
MTurkSF	126	non-experts	medical
NNSeval	478	language learners	encyclopedia
OneStopEnglish	4144	language learners	news
QuestEval	282		encyclopedia
SemEval_2007	598		
SimPA	2204	language learners	administrative
SimpEval	324		encyclopedia
TurkCorpus	4718	children, language learners	encyclopedia

Table 2: Datasets' number of texts, target audiences, and domains other than *general*. Info on audiences and domains stems from datasets themselves.

All datasets consist of pairs of complex texts and their simplified versions. We consider two target audiences: *children* (represented by Britannica and TurkCorpus) and *language learners* (represented by NNSeval, OneStopEnglish, SimPA and TurkCorpus). We consider two domains: *news* (represented by HutSSF and OneStopEnglish) and *encyclopedia* (represented by Britannica, EW-SEW-Turk, METAeval, NNSeval, QuestEval, SimpEval, TurkCorpus and Wiki-Manual). When experimenting with target audiences or domains, we use a merged form of the respective datasets, which contain 200 random simple and the corresponding 200 complex texts of each dataset.

The ARTS datasets[5] (Engelmann et al., 2024) capture humans' and ChatGPT's perceptions of simplicity in the form of ARTS scores. Scores are derived from votes on the simpler texts out of pairs of two unrelated texts. ARTS datasets consist of texts from 26 datasets from different domains for different target audiences with a numeric description of their simplicity between 0 (simple) and 1 (complex). The simplicity scores are calculated through an Elo-based algorithm. We use three datasets: ARTS$_{94}$ contains 94 texts and simplicity scores given by humans, ARTS$_{300}$ contains 300 texts and scores resulting from assessment by ChatGPT 4, ARTS$_{3000}$ holds 3000 texts and ratings from ChatGPT 4.

We prune our 1,249-dimensional vectors with ARTS$_{3000}$. We set the minimal threshold for correlation between found characteristics of simplic-

[3] Alternative ways of pruning are discussed in the Appendix in Section A.4.

[4] More details are in the Appendix in Section A.5.

[5] See Appendix A.6.

ity/complexity contained in the initial vector and simplicity scores to 0.25[6]. This step resulted in 98 dimensions indicating simplicity and 120 dimensions indicating the complexity of texts. We construct BATS vectors for all (merged) datasets.

4.2 RQ_1: From Literature to Practice

We investigate the importance of characteristics indicating simplicity or complexity of texts in the four levels described in Section 3.2: parameterizations, interpretations, features, and categories. We observe the correlation between the occurrences of characteristics in texts and their real simplicity score using ARTS.

Setting. We identify the most important parametrizations, interpretations, features, and categories that indicate simplicity or complexity by observing BATS vectors for texts combined with their real simplicity ratings contained in ARTS. ARTS depicts a cross-section over all domains and target audiences and is thus a suitable way to observe which characteristics might reflect simplicity or complexity. High ARTS scores indicate complexity, low ARTS scores indicate simplicity. High values in BATS vectors indicate simplicity (analogous: complexity), if encountered in a dimension describing a simplicity-inducing (analogous: complexity-inducing) parametrization. We expect positive (analogous: negative) correlations between ARTS scores and dimensions of BATS vectors indicating complexity (analogous: simplicity). The more pronounced the correlations, the more important the dimension. In higher levels - for interpretations, features, and categories - we observe the average of the correlations of the respective parametrizations belonging to the respective interpretation, feature, and category. Thus, we define importance as correlation or average correlation scores.

Results. In general, we find numerous parametrizations, interpretations, and features focusing on the same characteristic from opposing sides being the most important ones per level[7]. E.g. for all datasets, the feature *short sentences* is important in simplicity (so, actually, few sentences per text) and complexity (the complex variant of the feature describes the opposite - many sentences per text).

We seem to have found parametrizations for distinguishing between simple and complex texts - the maximum number of words in sentences being lower than 20 indicates simplicity, and higher than 22 indicates complexity. In the case of features, in addition to the length of sentences, we also encounter the Flesch-Kincaid Grade Level being important, which considers the number of words, sentences, and syllables.

Discussion. As the most important parametrizations, interpretations, and features, we found traditional characteristics, such as the length of sentences, being preferable to reflect the simplicity or complexity of texts over possibly newer or less straightforward characteristics, such as the number of entities in a text or the imageability of text. Additionally, we encountered the need for a high resolution of characteristics in order to analyze the data meaningfully. Only observing categories (see Figure 5 in the Appendix) does not clearly highlight essential characteristics. The level of interpretations offered a good balance between detail and generalizability. By answering this RQ, we show the explainability of BATS.

4.3 RQ_2: Characteristics of Simplicity

Following insights from RQ_1 we observe the level of interpretations across datasets, target audiences, and domains. We analyze which interpretations are the most different in terms of text containing characteristics between texts from sources and simplified parts of datasets.

Setting. We use BATS vectors to calculate how often a characteristic is found for source and simplified texts of parallel corpora. The larger the difference in occurrences between source and simplified text, the more selective a characteristic is. We observe selectivities for all 15 datasets, target audiences, and domains.

Results. Figure 2 depicts the selectivities of all datasets separated by characteristics identifying simplicity and complexity. There are vast differences between datasets. While several datasets (BenchLS, EW-SEW-Turk, MTurkSF, NNSeval, SemEval_2007) do not seem to differ much between texts in the source and simplified part, a closer look into the specifics of these datasets (see

[6] -0.25 for simplicity-inducing dimensions as a high ARTS score indicates complexity.

[7] Figure 4 and Figure 5 in the Appendix show correlation of ARTS scores and average values for the different levels. Table 5, Table 6 and Table 7 hold parametrizations, interpretations, and features with most prominent correlations to ARTS scores per dataset.

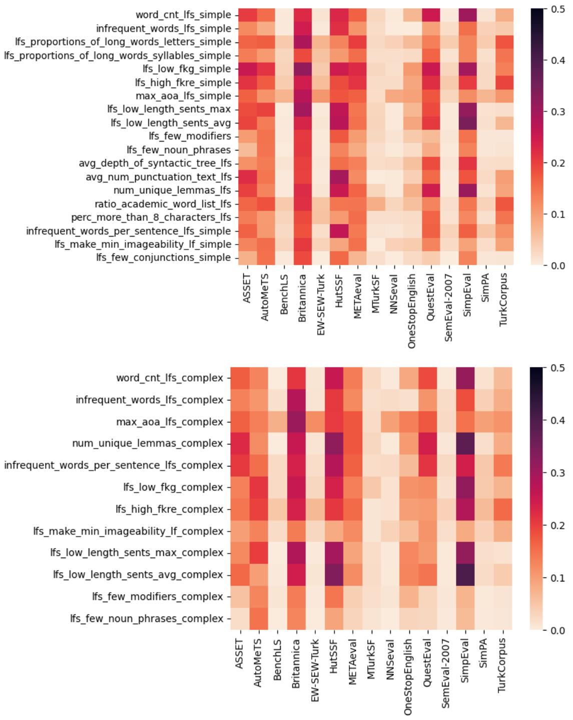

Figure 2: Average difference between source and simplified texts in parametrizations from different interpretations.

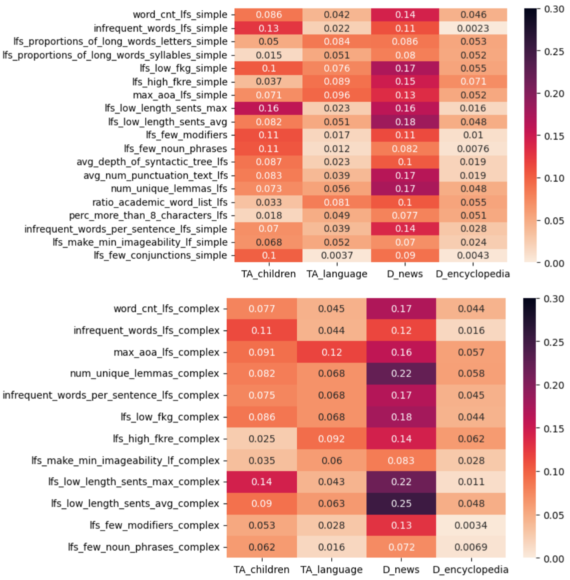

Figure 3: Average difference between source and simplified texts in parametrizations from different interpretations of TAs and Ds.

Table 4) reveals them to be datasets where source texts were simplified by substituting one difficult word for another. Among the most important characteristics of complexity in these cases is the maximum age of acquisition of a word. When replacing a complex word, possibly the most difficult one in a text, it makes sense for the replacement to have a lower age of acquisition. This would, in turn, result in the simplified text having a lower maximum age of acquisition. SimPA shows similarities to these aforementioned datasets. Here, several words were replaced between complex and simplified texts. Some datasets show a more notable difference between source and simplified texts (AutoMeTS, Britannica, HutSSF, SimpEval). In these datasets, humans wrote both versions, which can be the case in encyclopedias, where simpler versions of some articles exist. The sentence length, FKG/FKRE score, and the number of unique lemmas seem to be the most selective characteristics overall.

Figure 3 depicts the selectivities for our two target audiences and domains. There are vast differences. Texts simplified for children seem different in sentence length and the number of infrequent words compared to the respective source texts. For language learners, the maximum age of acquisition seems to be important, while sentence length does not considerably change between the two settings. We found the most differences in the news domain. Simplified text in this domain seems to be highly different from its source texts in multiple perspectives, such as sentence length, punctuation, number of used unique lemmas, FKRE, and, again, maximum age of acquisition of words. Contrasting this, we did not find major differences between source and simplified texts for the encyclopedia domain.

Discussion. In the observation of dataset-specific, target audience-specific and domain-specific characteristics regarding simplicity (RQ_2) we again found the traditional characteristics identified in RQ_1 being a discriminating factor between source and simplified texts.

Additionally, we found considerable differences between datasets (potentially stemming from their construction), target audiences, and domains. The seemingly low differences between texts in the domain of encyclopedias could stem from the heterogeneous datasets that are part of the domain. In addition to datasets with pronounced differences between source and simplified texts, there are other datasets from this domain where sources differ only in one word from simplified texts.

4.4 RQ_3: Representation of Simplicity

To assess BATS vectors' suitability to capture discriminating factors useful in quantifying the sim-

		MSE (lower values are better)					R^2 (higher values are better)				
		RF		GB			RF		GB		
Train	Predict	BATS	OAI	BATS	OAI	FRE	BATS	OAI	BATS	OAI	FRE
ARTS$_{94}$	ARTS$_{300}$	**.068**	.073	.08	.078	.078	**.188**	.133	.045	.074	.075
ARTS$_{94}$	ARTS$_{3000}$	**.059**	.076	.067	.081	.086	**.296**	.089	.192	.029	-.026
ARTS$_{300}$	ARTS$_{94}$	**.052**	.068	.059	.061	.065	**.393**	.203	.313	.282	.236
ARTS$_{300}$	ARTS$_{3000}$	**.055**	.07	.06	.07	.086	**.336**	.164	.286	.155	-.026
ARTS$_{3000}$	ARTS$_{94}$	.044	.06	**.039**	.055	.065	.485	.297	**.541**	.354	.236
ARTS$_{3000}$	ARTS$_{300}$	.048	.057	**.047**	.052	.078	.426	.325	**.435**	.378	.075

Table 3: Regression (Random Forest (RF) and Gradient Boosting (GB)) performance using BATS vectors or OpenAI embeddings (OAI) and ARTS scores. We report FRE as a baseline.

plicity of text, we compare using them to the current state of the art (OpenAI embeddings).

Setting. We use the three ARTS datasets in this experiment. Two types of regressors (random forest and gradient boosting with unchanged hyperparameters) are trained on either BATS vectors or OpenAI embeddings of ARTS texts and ARTS scores. The state-of-the-art text embeddings are generated using the OpenAI embeddings client[8] with the *text-embedding-3-small* model. The maximum number of input tokens is 8,191, and the length of the embedding vectors is 1,536.

The trained regressors then predict simplicity scores for unseen data (vectors/embeddings of texts from a different ARTS dataset). We compare these predicted scores against the real simplicity (ARTS) scores. We report the mean squared error (MSE) over the mean absolute error for an increased penalization of larger errors between predicted and actual simplicity scores. In our case, the predicted scores do not matter as much (*is a text's simplicity 0.342 or 0.37*), but the scores should instead indicate if a text can be considered simple or complex and should not be vastly different. We report R^2 to quantify how well a regressor predicts the actual data. R^2 gives the proportion of the variance for the complexity score that the input vectors or embeddings can explain.

As a baseline, we additionally report the MSE and R^2 of the Flesch Reading Ease (FRE) score of texts from ARTS. Since the FRE score is mapped on a scale from 0 to 100 and a high score indicates high simplicity, we transform it to make it comparable with the ARTS score. We achieve this by applying a MinMax-Scaler[9] and calculating the difference to 1 so that the transformed score is in $[0, 1]$ and small scores indicate high simplicity.

Results. Table 3 shows the outcome of the experiment; values in rows need to be compared to each other. Using BATS vectors outperforms using OpenAI embeddings and FRE in all rows. Row 5 holds the most important setting: a model trained on the most available data and tested on the least amount of available data.

Discussion. BATS vectors outperform OpenAI embeddings for training a regressor to predict ARTS scores. We conclude that BATS vectors are highly suitable for capturing characteristics of simplicity or complexity[10]. Thus, we showed the effectivity of BATS vectors for quantification of simplicity (RQ_3).

5 Conclusion

We presented BATS, a method to evaluate straightforward and concrete rules that can be used in quantifying the simplicity or complexity of text. Our approach is inherently explainable. The potential of BATS vectors becomes apparent in their comparison to the state of the art in text representation (OpenAI textual embeddings). Through our evaluations we showed the possibility for nuanced evaluation of text simplification for different target audiences or domains while shedding light on their different needs in properties of simplified text.

Future work should focus on the incorporation of more features into BATS and in-depth evaluations of simplification approaches and more datasets. As a direct consequence of our findings, simplification of text should be evaluated in a target audience- and domain-specific manner. Here, datasets which contain simplifications that differ in more than one word only from the source texts would be desirable.

[8] https://platform.openai.com/docs/guides/embeddings/embedding-models

[9] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler

[10] More general applications of BATS vectors in practice are discussed in the Appendix in Section A.8.

6 Limitations

Even though we started out with 1,249 parametrizations, we cannot ensure having utilized the most meaningful parameters for interpretations or the most suitable interpretations for features. Additionally, even though we chose 37 features describing the simplicity of text in the literature, there might be other and more important features that we disregarded in this analysis. Independent of features, interpretations, and parametrizations, our approach holds, is extensible and applicable. By utilizing a multi-domain, multi-target audience dataset for pruning the labeling functions, we ensure that rules that are not discriminatory or too niche are filtered out. Text simplification, in general, lacks a focus on meaning preservation. Our parametrizations currently do not capture this aspect (e.g., via BERTscore (Zhang et al., 2020)).

Our approach is currently limited by the literature in that we only implement well-known characteristics of simplicity or complexity.

In our evaluation of the characteristics in different datasets, TAs, and Ds, we used datasets of heterogeneous quality. Even though we used 15 datasets in total, the datasets became comparably small after excluding duplicates such that each source text was only combined with one simplification. In our analysis of TAs and Ds, we only compared two of each.

Currently, pruned labeling functions are chosen in a data-driven way, and the seed set of our labeling functions can be applied to parallel corpora from different languages. In general, the approach is extensible to other languages by adding new features/functions for language-specific aspects or implementing features in a language-independent fashion, e.g., by using a multi-lingual embedding of n-grams for imaginability, but this requires additional work. It remains open to investigate how simplicity might have different contributing aspects for different languages and if language-specific aspects still hold for other languages from the same language family.

Aside from ARTS, there is a lack of datasets quantifying the simplicity of text. It would be desirable to test our approach with another type of simplicity score to ensure generalizability.

We use pruning to identify only the dimensions that are important, excluding nonsensical parametrizations or ill-fitting interpretations of features. It is unclear if our method of pruning dimensions from vectors produces optimal results. We offer alternatives for pruning labeling functions in our discussion of additional methods in the Appendix (Section A.4).

References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. Asset: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. *Preprint*, arXiv:2005.00481.

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

Barbara Arfe, Lucia Mason, and Inmaculada Fajardo. 2018. Simplifying informational text structure for struggling readers. *Reading and Writing*, 31.

Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *EMNLP 2003*.

Casper Da Costa-Luis, Stephen Karl Larroque, Kyle Altendorf, Hadrien Mary, Mikhail Korobov, Noam Yorav-Raphael, Ivan Ivanov, Marcel Bargull, Nishant Rodrigues, Guangshuo CHEN, Charles Newey, James, Martin Zugnoni, Matthew D. Pagel, Mjstevens777, Mikhail Dektyarev, Alex Rothberg, Alexander, Daniel Panteleit, Fabian Dill, FichteFoll, HeoHeo, Hugo Van Kemenade, Jack McCracken, Max Nordlund, Orivej Desh, RedBug312, Socialery, Staffan Malmgren, and Todd. 2020. tqdm: A fast, Extensible Progress Bar for Python and CLI.

Björn Engelmann, Fabian Haak, Christin Katharina Kreutz, Narjes Nikzad-Khasmakhi, and Philipp Schaer. 2023. Text simplification of scientific texts for non-expert readers. In *Working Notes of the Conference and Labs of the Evaluation Forum (CLEF 2023), Thessaloniki, Greece, September 18th to 21st, 2023*, volume 3497 of *CEUR Workshop Proceedings*, pages 2987–2998. CEUR-WS.org.

Björn Engelmann, Christin Katharina Kreutz, Fabian Haak, and Philipp Schaer. 2024. ARTS Datasets - ARTS94, ARTS300, ARTS3000.

Lijun Feng, Noemie Elhadad, and Matt Huenerfauth. 2009. Cognitively motivated features for readability assessment. In *EACL 2009, 12th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, Athens, Greece, March 30 - April 3, 2009*, pages 229–237. The Association for Computer Linguistics.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221–233. Place: US Publisher: American Psychological Association.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. 2012. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 759–765. European Language Resources Association (ELRA).

I. J. Good. 1955. On the marking of chess-players. *The Mathematical Gazette*, 39(330):292–296.

Sian Gooding. 2022. On the Ethical Considerations of Text Simplification. arXiv. Version Number: 1.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

David Heineman, Yao Dou, Mounica Maddela, and Wei Xu. 2023. Dancing between success and failure: Edit-level simplification evaluation using SALSA. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3466–3495. Association for Computational Linguistics.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using wikipedia. volume 2, pages 458–463.

J. D. Hunter. 2007. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95.

David Kauchak, Jorge Apricio, and Gondy Leroy. 2022. Improving the quality of suggestions for medical text simplification tools. *AMIA Jt Summits Transl Sci Proc*, 2022:284–292.

Walter Kintsch. 1994. Text comprehension, memory, and learning. *American psychologist*, 49(4):294.

Walter Kintsch and Teun A van Dijk. 1978. Toward a model of text comprehension and production. *Psychological Review*, 85:363–394.

Camilla Lindholm and Ulla Vanhatalo. 2021. *Handbook of easy languages in Europe*. Frank & Timme.

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.

Diana McCarthy and Roberto Navigli. 2007. SemEval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic. Association for Computational Linguistics.

Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *ACL 2014*, pages 435–445.

Peter Norvig. English letter frequency counts: Mayzner revisited or etaoin srhldcu. http://norvig.com/mayzner.html.

Sharon O'Brien. 2003. Controlling controlled english. In *European Association for Machine Translation Conferences/Workshops*.

Gustavo Paetzold and Lucia Specia. 2016a. Benchmarking lexical simplification systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3074–3080, Portorož, Slovenia. European Language Resources Association (ELRA).

Gustavo H. Paetzold and Lucia Specia. 2016b. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 3761–3767. AAAI Press.

The pandas development team. 2020. pandas-dev/pandas: Pandas.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 11, page 269. NIH Public Access.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. 2016. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Horacio Saggion, Sanja Štajner, Stefan Bott, Simon Mille, Luz Rello, and Biljana Drndarevic. 2015. Making it simplext: Implementation and evaluation of a text simplification system for spanish. *ACM Trans. Access. Comput.*, 6(4).

Carolina Scarton, Gustavo Paetzold, and Lucia Specia. 2018. SimPA: A sentence-level simplification corpus for the public administration domain. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Max Schwarzer, Teerapaun Tanprasert, and David Kauchak. 2021. Improving human text simplification with sentence fusion. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15)*, pages 106–114, Mexico City, Mexico. Association for Computational Linguistics.

Thomas Scialom, Louis Martin, Jacopo Staiano, Eric Villemonte de la Clergerie, and Benoît Sagot. 2021. Rethinking automatic evaluation in sentence simplification. *ArXiv*, abs/2104.07560.

Amir Sepehri, Mitra Sadat Mirshafiee, and David M. Markowitz. 2023. Passivepy: A tool to automatically identify passive voice in big text data. *Journal of Consumer Psychology*, 33(4):714–727.

Matthew Shardlow and Raheel Nawaz. 2019. Neural text simplification of clinical letters with a domain specific phrase table. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 380–389, Florence, Italy. Association for Computational Linguistics.

Kathleen Sheehan, Irene Kostin, Diane Napolitano, and Michael Flor. 2014. The textevaluator tool: Helping teachers and test developers select texts for use in instruction and assessment. *The Elementary School Journal*, 115:184–209.

Kathleen M. Sheehan. 2013. Measuring cohesion: An approach that accounts for differences in the degree of integration challenge presented by different types of sentences. *Educational Measurement: Issues and Practice*, 32(4):28–37.

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

Bengt Sigurd, Mats Eeg-Olofsson, and Joost Van Weijer. 2004. Word length, sentence length and frequency – zipf revisited. *Studia Linguistica*, 58(1):37–52.

Robyn Speer. 2022. rspeer/wordfreq: v3.0.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018. Simple and effective text simplification using semantic and neural methods. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 162–173, Melbourne, Australia. Association for Computational Linguistics.

Hideki Tanaka, H. Mino, M. Shibata, S. Ochi, and Tadashi Kumano. 2013. News service in simplified japanese and its production support systems. pages 5.1–5.1.

Subhadra Vadlamannati and Gözde Şahin. 2023. Metric-based in-context learning: A case study in text simplification. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 253–268, Prague, Czechia. Association for Computational Linguistics.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.

Hoang Van, David Kauchak, and Gondy Leroy. 2020. AutoMeTS: The autocomplete for medical text simplification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1424–1434, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. 2020. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.

Michael L. Waskom. 2021. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021.

Wes McKinney. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56 – 61.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

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

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with BERT. In *ICLR 2020*. OpenReview.net.

Sanja Štajner, Daniel Ferrés, Matthew Shardlow, Kai North, Marcos Zampieri, and Horacio Saggion. 2022. Lexical simplification benchmarks for english, portuguese, and spanish. *Frontiers Artif. Intell.*, 5.

Sanja Štajner and Ioana Hulpus. 2018. Automatic assessment of conceptual text complexity using knowledge graphs. In *COLING 2018*, pages 318–330.

Sanja Štajner, Ruslan Mitkov, and Gloria Corpas Pastor. 2015. Simple or Not Simple? A Readability Question. In Núria Gala, Reinhard Rapp, and Gemma Bel-Enguix, editors, *Language Production, Cognition, and the Lexicon*, Text, Speech and Language Technology, pages 379–398. Springer International Publishing, Cham.

Sanja Štajner, Sergiu Nisioi, and Ioana Hulpus. 2020a. Coco: A tool for automatically assessing conceptual complexity of texts. In *LREC 2020*, pages 7179–7186. European Language Resources Association.

Sanja Štajner, Sergiu Nisioi, and Daniel Ibanez. 2020b. Is simple english wikipedia as simple and easy-to-understand as we expect it to be? In *DSAI 2020*, pages 66–70. ACM.

A Appendix

A.1 Example Parametrization

With the average length of words in English being 4.79 (Norvig) characters, words with more, so e.g., 5, 6, 7, 8, 9 letters could be regarded as long or complex words. Words with less characters could be considered short or simple words as the length of words can be a burden to readers (Sigurd et al., 2004). Goldhahn et al. (2012)'s analysis of corpora gives an average English sentence length as 117.542 characters. Together this would mean an average English sentence would consist out of around 20 words (117.542 characters / (4.79 characters + 1 blank symbol) = 20.3) when only containing shorter than average words. Another perspective could be introduced by Lindholm and Vanhatalo (2021), who describe long or harder sentence as those with at least 15 words. Combined this would lead to sentences with at least 87 characters (15 words * (4.79 characters + 1 blank symbol) = 86.85) being considered as complex. These numbers could be used in the parametrization of the labeling function *short sentences* (Arfe et al., 2018) in implementations either deciding on complexity of a text based on its number of words or characters.

A.2 Factors Characterizing Simplicity of Text

This work considers 37 descriptions of features from the literature. Table 4 gives an overview of the features as well as their (sub) categorization following O'Brien (2003) and our 135 interpretations of features.

Feature	C	Sub C	Interpretations

average lexical richness (Štajner et al., 2020a)	L	VU/Sy	num_unique_lemmas_label=NS_thresh=θ,
			num_unique_lemmas_label=S_thresh=θ,
			num_unique_lemmas_norm_label=NS_thresh=θ,
			num_unique_lemmas_norm_label=S_thresh=θ

average number of words before the main verb (Sheehan et al., 2014)	St	IS	avg_num_words_before_main_verb_label=NS_thresh=θ,
			avg_num_words_before_main_verb_label=S_thresh=θ

few content words (Scarton et al., 2018)	L	VU	lf_content_ratio_thresh=θ_label=NS,
			lf_content_ratio_thresh=θ_label=S

few infrequent words (Štajner and Hulpus, 2018)	L	VU	lf_infrequent_words_cnt=θ_$\$animal$_NS,
			lf_infrequent_words_cnt=θ_$\$animal$_S,
			lf_infrequent_words_per_sentence=θ_$\$animal$_NS,
			lf_infrequent_words_per_sentence=θ_$\$animal$_S

few long words (Arfe et al., 2018)	L	VU	low_prop_long_words_syllables_long=θ_prop=η_label=S,
			low_prop_long_words_letters_long=θ_prop=η_label=S

few modifiers (Narayan and Gardent, 2014)	Sy	MU	few_modifiers_thres=θ_label=NS, few_modifiers_thres=θ_label=S,
			low_modifier_ratio_thres=θ_label=NS,
			low_modifier_ratio_thres=θ_label=S

few negations (Sheehan et al., 2014)	L	Ne	freq_negations_label=S_thresh=θ, freq_negations_label=NS_thresh=θ,
			freq_negations_ratio_label=NS_thresh=θ,
			freq_negations_ratio_label=S_thresh=θ

few noun phrases (Arfe et al., 2018)	Sy	NC	few_noun_phrases_ratio_thres=θ_label=NS,
			few_noun_phrases_ratio_thres=θ_label=S,
			few_noun_phrases_thres=θ_label=NS,
			few_noun_phrases_thres=θ_label=S

few past perfect verbs (Sheehan et al., 2014)	Sy	Tense	num_past_perfect_label=NS_thresh=θ,
			num_past_perfect_label=S_thresh=θ,
			perc_past_perfect_label=NS_thresh=θ,
			perc_past_perfect_label=S_thresh=θ

few past tense aspect verbs (Sheehan et al., 2014)	Sy	Tense	num_past_tense_label=NS_thresh=θ,
			num_past_tense_label=S_thresh=θ, perc_past_tense_label=NS_thresh=θ,
			perc_past_tense_label=S_thresh=θ

few punctuation marks(Saggion et al., 2015)	Sy	Pn	avg_num_punctuation_text_label=NS_thresh=θ,
			avg_num_punctuation_text_label=S_thresh=θ

few relative-clauses (Arfe et al., 2018)	L	PU	low_relative_clauses_ratio_label=NS_thresh=θ, low_relative_clauses_ratio_label=S_thresh=θ, no_relative_clauses_label=S_thresh=θ, no_relative_clauses_label=NS_thresh=θ, low_relative_sub_clauses_label=NS_thresh=θ, low_relative_sub_clauses_ratio_label=NS_thresh=θ, low_relative_sub_clauses_label=S_thresh=θ, low_relative_sub_clauses_ratio_label=S_thresh=θ, no_relative_sub_clauses_label=S_thresh=θ, no_relative_sub_clauses_label=NS_thresh=θ
few sentences (Arfe et al., 2018)	St	SL	num_sents_num_thres=θ_label=NS, num_sents_num_thres=θ_label=S
few third person singular pronouns (Sheehan et al., 2014)	L	PU	freq_third_person_singular_pronouns_label=S_thresh=θ, freq_third_person_singular_pronouns_label=NS_thresh=θ, freq_third_person_singular_pronouns_ratio_label=NS_thresh=θ, freq_third_person_singular_pronouns_ratio_label=S_thresh=θ
few unique entities (Štajner et al., 2020b)	St	IL	unique_entities_text_label=NS_thresh=θ, unique_entities_text_label=S_thresh=θ, unique_entities_text_ratio_label=NS_thresh=θ, unique_entities_text_ratio_label=S_thresh=θ
few words containing more than eight characters (Sheehan et al., 2014)	L	VU	perc_more_than_8_characters_label=NS_thresh=θ, perc_more_than_8_characters_label=S_thresh=θ
few words from academic word list (Sheehan et al., 2014)[11]	L	VU/DU	ratio_academic_word_list_label=NS_thresh=θ, ratio_academic_word_list_label=S_thresh=θ
few words per sentence (Scarton et al., 2018)	St	SL/PL	lf_words_cnt_wcount=θ_NS, lf_words_cnt_wcount=θ_S
grammatical correctness (Xu et al., 2015)	Sy		few_gram_errors_ratio_thres=θ_label=NS, few_gram_errors_ratio_thres=θ_label=S, thresh=θ_label=NS, thresh=θ_label=S
high average distance between consecutive entities (Štajner et al., 2020b)	St	IS	avarage_distance_entities_para_consec_θ_S, avarage_distance_entities_para_same_θ_S, avarage_distance_entities_sent_consec_θ_S, avarage_distance_entities_sent_same_θ_S
high concreteness (Scarton et al., 2018)	P	SI	lf_avg_concreteness=θ_NS, lf_avg_concreteness=θ_S, lf_max_concreteness=θ_NS, lf_max_concreteness=θ_S, lf_median_concreteness=θ_NS, lf_median_concreteness=θ_S
high Flesch reading ease (Scarton et al., 2018)	St	IL	high_fkre_threshold=θ_NS, high_fkre_threshold=θ_S
high imageability (Scarton et al., 2018)	L	DU	lf_avg_imageability=θ_NS, lf_avg_imageability=θ_S, lf_min_imageability=θ_NS, lf_min_imageability=θ_S, lf_med_imageability=θ_NS, lf_med_imageability=θ_S
high percentage of vocabulary learned in initial stages of foreign language learning (Tanaka et al., 2013)	L	VU/DU	perc_vocab_initial_forLang_learn_label=NS_thresh=θ, perc_vocab_initial_forLang_learn_label=S_thresh=θ
low age of acquisition (Scarton et al., 2018)	L	VU	lf_avg_age_of_acquisition=θ_NS, lf_avg_age_of_acquisition=θ_S, lf_max_age_of_acquisition=θ_NS, lf_max_age_of_acquisition=θ_S, lf_median_age_of_acquisition=θ_NS, lf_median_age_of_acquisition=θ_S
low average number of unique entities (Štajner et al., 2020b)	St	IL	average_entities_paragraph_label=NS_thresh=θ, average_entities_paragraph_label=S_thresh=θ, average_entities_sentence_label=NS_thresh=θ, average_entities_sentence_label=S_thresh=θ
low avg distance between all pairs of same entities (Štajner et al., 2020b)	St	IS	avarage_distance_appearance_same_entities_paragraph_label=NS_thresh=θ, avarage_distance_appearance_same_entities_label=S_thresh=θ, avarage_distance_appearance_same_entities_sentence_label=NS_thresh=θ, avarage_distance_appearance_same_entities_sentence_label=S_thresh=θ
low depth of the syntactic tree (Štajner et al., 2020a)	St	IL	depth_of_syntactic_tree_label=NS_thresh=θ, depth_of_syntactic_tree_label=S_thresh=θ
low entity to token ratio (Štajner et al., 2020b)	St	IL	entity_token_ratio_text_label=NS_thresh=θ, entity_token_ratio_text_label=S_thresh=θ, entity_token_ratio_paragraph_label=NS_thresh=θ, entity_token_ratio_paragraph_label=S_thresh=θ, entity_token_ratio_sentence_label=NS_thresh=θ, entity_token_ratio_sentence_label=S_thresh=θ

[11] https://www.eapfoundation.com/vocab/academic/awllists/

low Flesch-Kincaid Grade Level Index (Narayan and Gardent, 2014)	St	IL	low_thresh=θ_NS, low_thresh=θ_S
low number of cases with max distance between 2 appearances of same entity (Štajner et al., 2020b)	St	IS	distance_appearance_same_entities_sentence_dist_label=S_thresh_dist=θ_numη, distance_appearance_same_entities_paragraph_dist_label=S_thresh_dist=θ_numη

low unique entities to total number of entities ratio (Štajner et al., 2020b)	St	IL	unique_entity_total_entity_ratio_text_label=NS_thresh=θ,
			unique_entity_total_entity_ratio_text_label=S_thresh=θ,
			unique_entity_total_entity_ratio_paragraph_label=NS_thres=θ,
			unique_entity_total_entity_ratio_paragraph_label=S_thres=θ,
			unique_entity_total_entity_ratio_sentence_thresh=θ_label=NS,
			unique_entity_total_entity_ratio_sentence_thresh=θ_label=S

no appositions (Narayan and Gardent, 2014)	Sy	Ap	no_apposition, percentage_appositions=θ_NS, percentage_appositions=θ_S
no conditional clauses (Arfe et al., 2018)	Sy	Co	no_conditional, percentage_conditional=θ_NS, percentage_conditional=θ_S

no conjunctions (Arfe et al., 2018)	L	CU	few_conjunctions_thres=θ_label=NS,
			few_conjunctions_thres=θ_label=S, few_conjunctions_ratio_thres=θ_label=NS, few_conjunctions_ratio_thres=θ_label=S

no passive voice (Arfe et al., 2018)	Sy	Voice	no_passive_voice, percentage_passive_voice=θ_NS, percentage_passive_voice=θ_S

short sentences (Arfe et al., 2018)	St	SL	low_num_words_in_sents_avg_thres=θ_label=NS,
			low_num_words_in_sents_avg_thres=θ_label=S,
			low_num_words_in_sents_max_thres=θ_label=NS,
			low_num_words_in_sents_max_thres=θ_label=S

Table 4: Features from literature, Clusters (C), Sub cluster and interpretations of labeling functions. Clusters: Lexical (L), Structural (St), Syntactic (Sy), Pragmatic (P). Sub cluster: Vocabulary Usage (VC), Dictionary Usage (DU), Synonymy (Sy), Information Structure (IS), Noun Cluster (NC), Pronoun Usage (PU), Modifier Usage (MU), Sentence Length (SL), PL (PL), Specificity of Information (SI), Negation (Ne), Apposition (Ap), Coordination (Co), Information Load (IL), Punctuation (Pn), Conjunction Usage (CU). Abbreviations in interpretations: NOT_SIMPLE (NS), SIMPLE (S); variables: thresholds (θ, η), $ animal.

A.3 Used Python Packages

In our implementation, we utilize the following packages:

- allennlp (Gardner et al., 2017)
- bs4 (https://www.crummy.com/software/BeautifulSoup/)
- dotenv (https://saurabh-kumar.com/python-dotenv/)
- language_tool_python (https://pypi.org/project/language-tool-python/)
- Levenshtein (https://rapidfuzz.github.io/Levenshtein/)
- matplotlob (Hunter, 2007)
- numpy (Harris et al., 2020)
- openai (https://platform.openai.com/docs/api-reference?lang=python)
- pandas (Wes McKinney, 2010; pandas development team, 2020)
- PassivePy (Sepehri et al., 2023)
- py7zr (https://py7zr.readthedocs.io/en/latest/index.html)
- requests (https://requests.readthedocs.io/en/latest/)
- scipy (Virtanen et al., 2020)
- seaborn (Waskom, 2021)

- sklearn (Pedregosa et al., 2011)
- snorkel (Ratner et al., 2017)
- spacy (Honnibal and Montani, 2017)
- textstat (`https://github.com/textstat`)
- tqdm (Da Costa-Luis et al., 2020)
- wortfreq (Speer, 2022)

A.4 Alternatives for Pruning Dimensions

We assume not all dimensions in vectors contribute meaningfully when assessing if the text should be considered simple or complex. A pruning of dimensions can be performed in multiple ways, we present four different methods for selecting the dimensions to compose the BATS model: A **naive approach** would be using those dimensions with the highest difference in the percentage of texts from source and simplified origins to contain a feature. Another method of determining the kept dimensions is selecting the best ones with a **chi**^{2} test. A third method is the selection of the top n dimensions, which have the highest average permutation importance (see sklearn[12]). Lastly, a **random** choice could also identify the kept dimensions.

A.5 Description of Datasets

This work considers the following 15 publicly available parallel corpora in the English language:

- **ASSET** (Alva-Manchego et al., 2020) is based on TurkCorpus (Xu et al., 2015, 2016a) and is a dataset specifically constructed for text simplification. It contains 2,000 validation and 359 test source sentences with ten simplifications each, so 23,590 pairs in total. Texts are primarily single sentences. Sources are, on average, 116.77 characters, while simplifications are, on average, 98.52 characters. The dataset was published under CC BY-NC license.
- **AutoMeTS** (Van et al., 2020) is based on EW-SWE. It contains single sentences and 4,280 pairs of texts in total. Sources are, on average, 205.14 characters, while simplifications are, on average, 154.07 characters. The dataset was published under an MIT license.
- **BenchLS** (Paetzold and Specia, 2016a) is based on sources from LexMTurk and LSeval. It contains 929 source sentences with multiple simplifications. Simplifications differ in one word from the sources. Human annotators gave the substitutions for complex words. Sources are, on average, 152.76 characters, while simplifications are, on average, 150.08 characters. The dataset was published under a CC BY-SA 4.0 license.
- **Britannica** (Barzilay and Elhadad, 2003) is based on Encyclopedia Britannica and Britannica Elementary. We consider the non-empty sentences from the training and test set annotated by humans. This leaves us with 6,846 pairs of texts. Sources are, on average, 88.15 characters, while simplifications are, on average, 147.35 characters. For the dataset, no license was specified.
- **EW-SEW-Turk** (Horn et al., 2014) is based on EW-SEW. Source sentences each have multiple simplifications, which differ in one word from the original ones. This produces 7,330 pairs of texts. Sources are, on average, 146.97 characters, while simplifications are, on average, 147.29 characters. For the dataset, no license was specified.
- **HutSSF** (Schwarzer et al., 2021) is based on news. The data is split into train, development, and test sets. There are 5,245 pairs of texts in total. Sources are, on average, 88.15 characters, while simplifications are, on average, 147.35 characters. For the dataset, no license was specified.

[12] `https://scikit-learn.org/stable/modules/generated/sklearn.inspection.permutation_importance.html`

- **METAeval** (Alva-Manchego et al., 2021) is based on TurkCorpus (Xu et al., 2015, 2016a). It contains sentence pairs with simplifications from six systems using the TurkCorpus (Xu et al., 2015, 2016a) dataset: PBMT-R, Hybrid, SBMT-SARI, Dress-Ls, DMASS-DCSS, ACCESS. It contains 302 original source sentences and 600 pairs in total. Sources are, on average, 161.89 characters, while simplifications are, on average, 136.43 characters. The dataset was published under a CC BY-NC-SA 4.0 license.

- **MTurkSF** (Kauchak et al., 2022) is based on Wikipedia. It contains sentence pairs where the simplification differs in one word from the source. We only consider pairs, where the replacement words in sentences were rated as good substitutes by MTurkers, resulting in 221 pairs of texts. Sources are, on average, 168.26 characters, while simplifications are, on average, 171.24 characters. For the dataset, no license was specified.

- **NNSeval** (Paetzold and Specia, 2016b) is based on Wikipedia. It contains 239 source sentences with multiple simplifications, each differing in one word from the sources that stem from LSeval and LexMTurk. Substitution suggestions are from LEXenstein and vetoed by human annotators. In total, there are 1,791 pairs of texts. Sources are, on average, 145.27 characters, while simplifications are, on average, 143.54 characters. The dataset was published under a CC BY-SA 4.0 license.

- **OneStopEnglish** (Vajjala and Lučić, 2018) is based on the website onestopenglish.com. It contains 2,107 human-written triples of advanced (close to its source's complexity), intermediate, and elementary simplifications of texts on 189 topics from the English language learning resource website. Considering advanced-intermediate, advanced-elementary, and intermediate-elementary pairings produced 6,321 pairs of source-simplified texts. Sources are, on average, 241.11 characters, while simplifications are, on average, 285.96 characters. The dataset was published under a CC BY-SA 4.0 license.

- **QuestEval** (Scialom et al., 2021) is based on a subset of ASSET (Alva-Manchego et al., 2020) but contains more simplifications. There are 366 pairs in total in the dataset. Sources are, on average, 116.55 characters, while simplifications are, on average, 93.08 characters. For the dataset, no license was specified.

- **SemEval-2007** (McCarthy and Navigli, 2007) is based on the English Internet Corpus of English. In this, base sentences were crawled from the web; annotators suggested simpler replacement words for complex ones. The dataset contains 1,208 pairs of texts. Sources are, on average, 150.6 characters, while simplifications are, on average, 153.42 characters. For the dataset, no license was specified.

- **SimPA** (Scarton et al., 2018) is based on the Sheffield City Council website and contains 6,600 pairs of complex-simplified text pairs. Sources are, on average, 165.76 characters, while simplifications are, on average, 160.5 characters. For the dataset, no license was specified.

- **SimpEval** (Maddela et al., 2023) consists of four parts. It is partially based on Wikipedia entries from revisions or new entries between October and November 2022 and on TurkCorpus (Xu et al., 2015, 2016a). Combined, the four parts contain 2,570 pairs of texts. Sources are, on average, 155.85 characters, while simplifications are, on average, 135.17 characters. For the dataset, no license was specified.

- **TurkCorpus** (Xu et al., 2015, 2016a) is based on PPDB. It contains 2,359 sentence pairs from English Wikipedia and Simple English Wikipedia. Each source sentence also contains eight simplified versions from Amazon Mechanical Turkers, so there are 21,231 pairs in total. Simplifications are lowercase. Sources are, on average, 118 characters, while simplifications are, on average, 110.98 characters. The dataset was published under a GPL 3.0 license.

From all datasets, we only consider non-duplicated texts, so each complex text is only included once, meaning if a source text has multiple simplifications, we only consider one of the simplifications.

ARTS	Simple Parametrizations
∩	low_num_words_in_sents_max_thres=20_label=0
94	∩, low_num_words_in_sents_avg_thres=20_label=0, lf_words_cnt_wcount=20_SIMPLE, lf_words_cnt_wcount=17_SIMPLE, lf_words_cnt_wcount=19_SIMPLE
300	∩, low_num_words_in_sents_max_thres=24_label=0, low_num_words_in_sents_avg_thres=22_label=0, lf_words_cnt_wcount=22_SIMPLE, low_num_words_in_sents_max_thres=22_label=0
3000	∩, low_num_words_in_sents_max_thres=17_label=0, low_num_words_in_sents_max_thres=24_label=0, low_num_words_in_sents_max_thres=26_label=0, low_num_words_in_sents_max_thres=22_label=0
ARTS	Complex Parametrizations
∩	low_num_words_in_sents_max_thres=22_label=1
94	∩, lf_words_cnt_wcount=20_NOT_SIMPLE, lf_words_cnt_wcount=17_NOT_SIMPLE, low_num_words_in_sents_avg_thres=20_label=1, lf_words_cnt_wcount=19_NOT_SIMPLE
300	∩, low_num_words_in_sents_max_thres=22_label=1, lf_words_cnt_wcount=22_NOT_SIMPLE, low_num_words_in_sents_max_thres=24_label=1, low_num_words_in_sents_avg_thres=22_label=1
3000	∩, low_num_words_in_sents_max_thres=22_label=1, low_num_words_in_sents_max_thres=24_label=1, low_num_words_in_sents_max_thres=26_label=1, low_num_words_in_sents_max_thres=28_label=1

Table 5: Five most important simple and complex parametrizations in ARTS, ∩ indicates the parametrizations, which are present in all three versions of ARTS: ARTS$_{94}$, ARTS$_{300}$, and ARTS$_{3000}$.

A.6 ARTS: Assessing Readability & Text Simplicity

To evaluate our BATS approach we require a dataset consisting of texts and their numeric simplicity indication to prune our binary vectors to obtain the BATS model. All existing datasets used for evaluating simplicity evaluation measures quantify the difference of a source text and a simplified version of the same text (i.e., „how much simpler has a text gotten compared to the original version?“) instead of quantifying the simplicity of text without comparison to its original version (i.e., „how simple is text?“). Datasets estimating the first question cannot be used in our case to prune dimensions as there is only one value for a combination of two texts. If we intend to correlate values in our vectors with simplicity scores we require a dataset consisting of single texts and their simplicity

The ARTS dataset (Engelmann et al., 2024) provides texts with an associated simplicity score. This score, on a scale from 0 to 1, indicates how simple a text is. ARTS is based on the idea that annotators can easily decide which of two given texts is easier. Based on 394 consecutive decisions, an order was formed concerning simplicity. This ordering results from applying the Elo algorithm used in chess (Good, 1955). Judging that one text is more complex than another is analogous to winning a chess game. Each text is assigned a score based on the order of the texts. Therefore, the text rated as the most difficult has a score of 1, and the easiest one has a score of 0. This way, high-reliable simplicity scores can be assessed for a given set of texts. Based on five annotators, the result scores achieved an average rank correlation of 0.82 with the majority vote. This dataset was also automated by replacing the annotators with GPT4. The GPT-based scores exhibit a high rank correlation (0.80) with the human annotations.

A.7 Additional Material related to RQ_1

Figure 4 depicts the correlations of BATS vectors with ARTS scores while Figure 5 holds correlations of BATS vectors with ARTS scores in higher levels: interpretations, features, and categories.

Table 5, Table 6 and Table 7 describe the most important parametrizations, interpretations and features indicating simple and complex text.

ARTS	Simple Interpretations
∩	word_cnt_lfs_simple, lfs_low_fkg_simple, lfs_low_length_sents_max
94	∩, **lfs_low_length_sents_avg**, **num_unique_lemmas_lfs**
300	∩, **lfs_low_length_sents_avg**, **num_unique_lemmas_lfs**
3000	∩, lfs_make_min_imageability_lf_simple, **lfs_few_noun_phrases**
ARTS	Complex Interpretations
∩	-
94	**num_unique_lemmas_complex**, word_cnt_lfs_complex, **lfs_low_length_sents_avg_complex**, lfs_low_fkg_complex, infrequent_words_per_sentence_lfs_complex
300	**num_unique_lemmas_complex**, lfs_make_min_imageability_lf_complex, word_cnt_lfs_complex, **lfs_low_length_sents_avg_complex**, lfs_low_length_sents_max_complex
3000	max_aoa_lfs_complex, infrequent_words_lfs_complex, lfs_low_length_sents_max_complex, **lfs_few_noun_phrases_complex**, lfs_low_fkg_complex

Table 6: Five most important simple and complex interpretations in ARTS, ∩ indicates the interpretations, which are present in all three versions of ARTS: ARTS$_{94}$, ARTS$_{300}$, and ARTS$_{3000}$. Interpretations marked in bold indicate cases where corresponding ones were important for simplicity and complexity.

ARTS	Simple Features
∩	**few words per sentence**, **short sentences**, **low Flesch-Kincaid Grade Level Index**
94	∩, high Flesch reading ease, **average lexical richness**
300	∩, **high imageability**, **average lexical richness**
3000	∩, **few noun phrases**, high imageability
ARTS	Complex Features
∩	**few words per sentence**, **short sentences**, **low Flesch-Kincaid Grade Level Index**
94	∩, few infrequent words, **average lexical richness**
300	∩, **high imageability**, **average lexical richness**
3000	∩, **few noun phrases**, low age of acquisition

Table 7: Five most important simple and complex features in ARTS, ∩ indicates the features, which are present in all three versions of ARTS: ARTS$_{94}$, ARTS$_{300}$, and ARTS$_{3000}$. Features marked in bold are important for both simplicity and complexity.

A.8 Description of Example Applications

Case 1: Evaluating Text Simplification Approaches. Consider the case of evaluating text simplification. To evaluate if texts that have been simplified can be considered simpler compared to the original ones, we can analyze the difference in the specificity of found characteristics. We construct BATS vectors on both types of texts (original and simplified). If more characteristics indicating simplicity are found in BATS vectors constructed from simplified texts, the simplification seems to have been successful.

In this scenario, it would also make sense to compare two different ways of simplifying text to identify the text simplification method out of several implementations. Here, one would consider the simplification method as better, for which more characteristics indicating simplicity can be found.

By having a pre-selection of potentially good simplifications conducted automatically the workload of human annotators could be reduced.

Case 2: Building a Text Simplification Approach. Consider the case of composing a text simplification approach, e.g., by using LLMs via prompting. Here the outcome of the text simplification needs to be assessed in order to modify the current prompt. Constructing BATS vectors for the text indicates the parametrizations and thus rules for simplicity, which the text satisfies and ones that are still hinting at complexity. These found characteristics of complex text can be considered as untapped potential, which can be also included in the approach.

Another perspective could be the application of BATS for in-context learning. Asides from evaluation, scores such as SARI (Xu et al., 2016b) or the compression ratio between a source and a simplified text can be used to select the best pairs of examples for text simplification approaches via LLMs (Vadlamannati and Şahin, 2023). Our approach could provide an option to select single texts instead of pairs to demonstrate what a text fulfilling requirements for a specific domain or target audience could look like.

Case 3: Assisted Text Simplification. The BATS model can help with the targeted simplification of a given text. If a person wants to simplify a text for a specific target audience, the BATS model can assist with the various interpretations. The explainability of the BATS model supports people in their work and enables them to make informed changes to the text with respect to their specific needs.

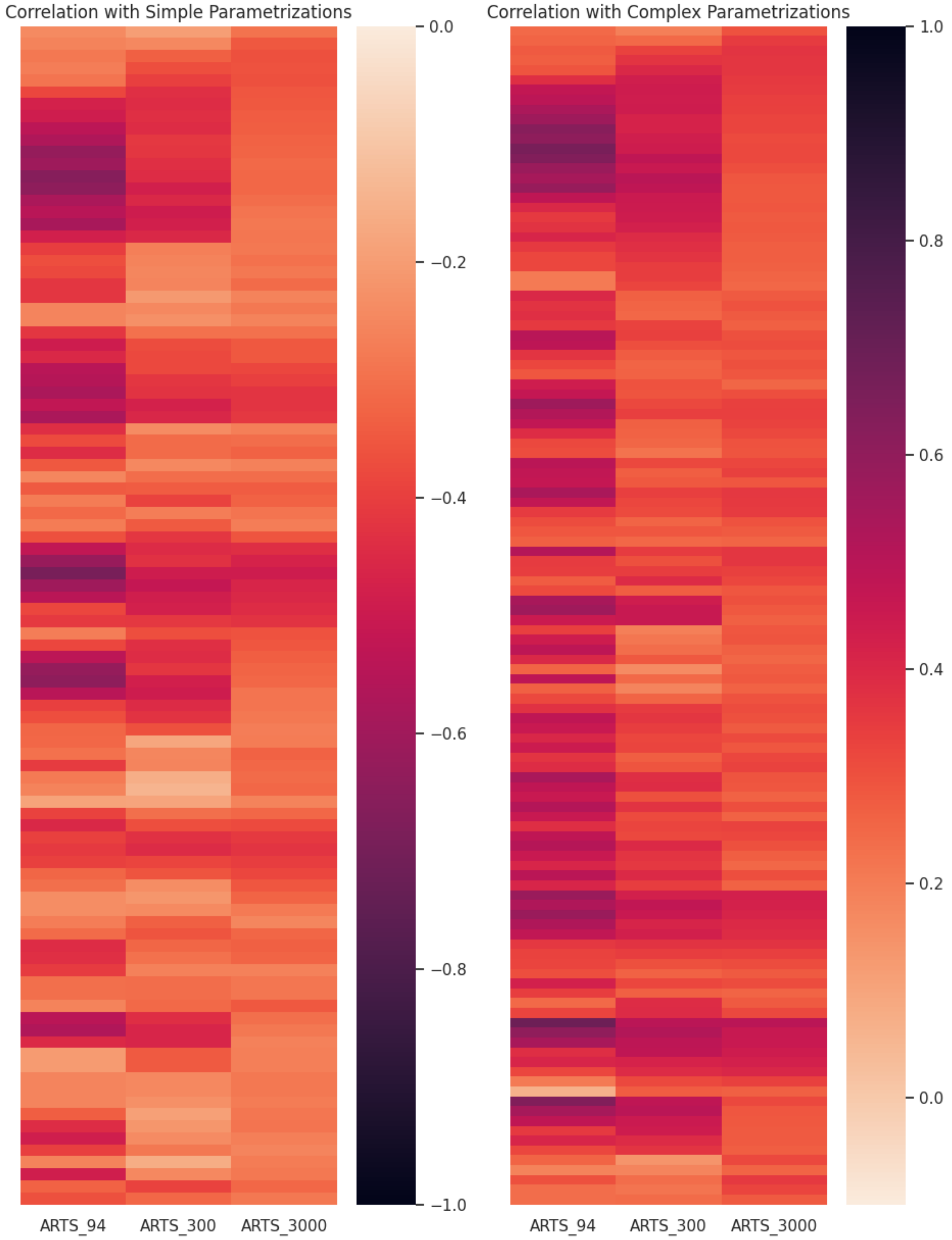

Figure 4: Correlation analysis with the color bars indicating the correlation of dimensions in BATS vectors with ARTS scores for the three datasets in parametrization. The more the correlation deviates from 0, the more intense the color.

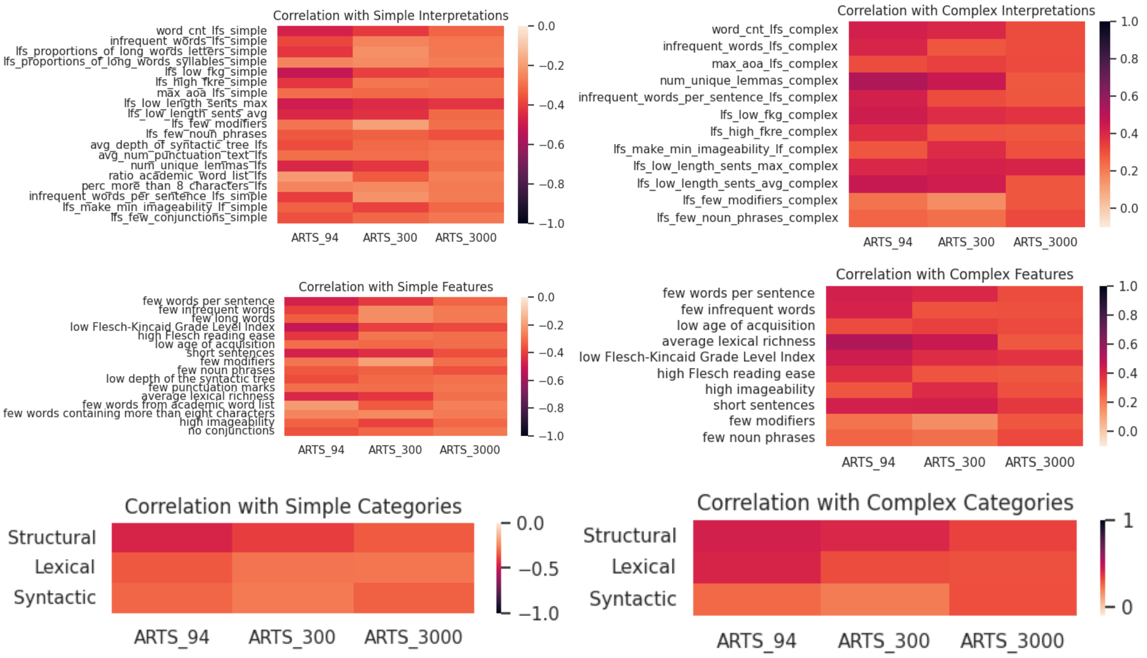

Figure 5: Correlation analysis with the color bars indicating the average correlation of interpretations, features, and categories in BATS vectors with ARTS scores for the three datasets in parametrizations. The more the correlation deviates from 0, the more intense the color.