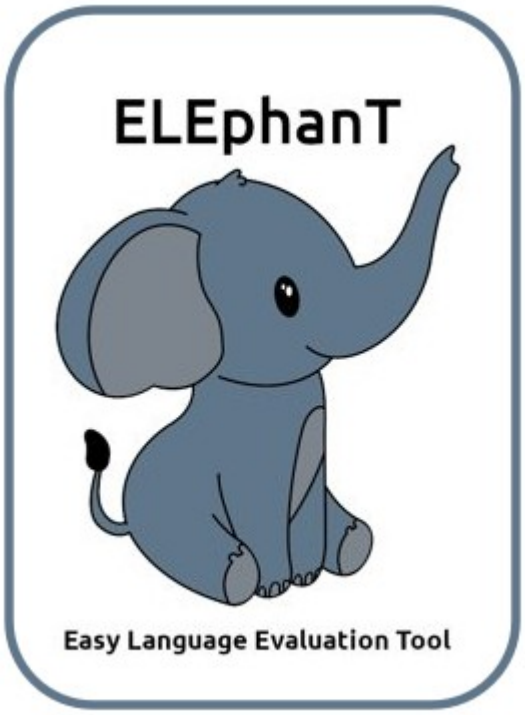
ELEphanT – Easy Language Evaluation Tool

Concept and Implementation

Table of Contents

I. Introduction....2
II. Easy Language....2
III. Pre-Processing....3
IV. Implementation....3
IV.I Easy Language Rules....3
IV.II. Easy Language Results....7
V. Tool Evaluation....7
V.I. Gold Standard: Easy Language Reference Texts....7
V.II. Degree of Easy Language in Easy Language Texts....8
V.III. Discussion....8



1 I. Introduction

The ELEphanT-tool has been developed in the context of a practical scientific internship which is obligatory in the master's program „Digitale Methodik in den Geistes- und Kulturwissenschaften“ (Digital Humanities). At the same time, this project is embedded into the CHYLSA-project which is an interdisciplinary project between the Johannes Gutenberg-Universität Mainz (JGU) and the Freie Universität Berlin (FU). The project’s full name is: „CHYLSA –Advanced sentiment analysis for understanding affective-aesthetic responses to literary texts: A computational and experimental psychology approach to children’s literature“. Thus, CHYLSA focuses on children’s and youth literature (CYL).

The presented student project equals some kind of a first feasibility study in terms of text complexity in CYL. Concretely, the project focuses on the German language variety Easy Language and aims at measuring the degree of Easy Language in a given text. The ELEphanT-tool was tested mainly on CHYLSA-data (CYL-texts) because we are interested in figuring out dimensions of the complexity / readability of children’s books. In terms of ELEphanT, our research question: Do children’s books for younger children comply more to the rules of Easy Language than books for older children? In addition, ELEphanT is tested on a sample of Easy Language texts. This evaluation is described below.

As Easy Language is intended for non-fiction books, this poses not only challenges in terms of a different text genre but also new insights into the complexity of children’s literature with respect to Easy Language. Please note that this part of data can’t be shared due to legal restrictions.

2 II. Easy Language

Easy Language is a German language variety distinguished by simplicity, a clear text formatting and firm rules. In Germany, the influence of Easy Language increased strongly in the first decade of 2000 when the Federal Act on Equality for People with Disabilities (in 2002) was adopted, as well as its realization, the „Barrierefreie-Informationstechnik-Verordnung“ (BITV 2.0) (Accessible Information Technology Regulation) which was adopted in 2011.

In a broader sense, Easy Language targets all people that can’t deal properly with a standard text and would prefer a more simple version. Concretely, this includes groups such as functional illiterate people, people suffering from Dementia, hearing impaired people, foreign language learners etc. Although Easy Language has gained some public attention in the last few years, there is no standard rulebook in Germany. Instead, there are multiple sets of rules, in most cases with either a strongly scientific or practical-political background.

For this project, the rulebook by Netzwerk Leichte Sprache has been chosen, due to its popularity, among other things. The rulebook itself offers no rule numbering, but a separation

regarding different application levels: words, numbers and characters, sentences, texts, design and pictures. Examples for rules are: „Use short words“, „Avoid questions in the text“ or „Use an easy writing font“.

3 III. Pre-Processing

Due to our use case – literary children's books – some of the pre-processing steps suit better for literature than for non-fiction texts. At the same time, there are probably some missing steps necessary for non-fiction texts (such as enumerations which are common in non-fiction but rare in literature). This will become visible especially when evaluating the tool on non-fiction texts later. Many pre-processing steps is about removing redundant whitespaces or paragraphs etc. When testing spaCy's sentence segmentation, it became evident that the parser has problems with french quotation marks (which are commonly used in German literature). These are replaced by the „normal“ German quotations. Also, it was necessary to modify spaCy's sentence boundaries because there were cases where spaCy separated the sentence although not necessary. This was modified, i.e. for semicolons or commas. The pre-processing is done with spaCy and regular expressions (regex).

4 IV. Implementation

IV.I Easy Language Rules

The rulebook by Netzwerk Leichte Sprache is intended to be used by Easy Language translators. At the same time, it is not targeted to be automatically assessed by computer programs. Thus, many rules are not automatically assessable and excluded. To give an example, all rules from the level of „design and pictures“ were dismissed because rules such as „Use one single, large font“ or „Do not use hyphenation“ are simply not applicable for automatic assessment. Due to the project's time limitation, the rules were separated into must-have and nice-to-have rules. This was decided mainly on the basis of feasibility – there are rules which require more elaborate methods such as web scraping.

Rule Selection, Concretization and Implementation

The following paragraph presents the list of all rules that have been implemented in ELEphanT. They have been translated by the author on the basis of the rulebook by Netzwerk Leichte Sprache. The rule description itself is followed by a deeper explanation on how the rule is interpreted to make it applicable for automatic assessment and on how the rule is implemented. In example, the interpretation often means defining in what case a rule is considered broken or what „not easy“ means. The concretization strongly orients on the rulebook by Netzwerk Leichte Sprache which always gives good and bad examples for each rule. Overall, there are 16 implemented rules.

To achieve a more differentiated result, it was decided to track the degree of Easy Language on sentence level instead of text level. This means that for every sentence it is evaluated if it complies to a rule or violates it. This is tracked in a binary way, by 1 (rule is fulfilled) or 0 (rule is violated). Please note that the tool doesn't track how often a rule is violated per sentence; instead, it is evaluated if a rule is broken or not. This could be further developed in future. With a look on some questions and the text being a literary text, this may be even more important because there are rules that a literary text violates very often, i.e. rule nr. 4.

Character Rules

1. **Avoid high numbers.**

This rule is implemented using regex. High numbers are defined in two ways: 1. Multi-digit numbers that do not end in 0; 2. five-digit numbers in any case. If the sentence contains at least one high number, the tool returns 0 (rule is violated).

2. **Avoid percentage numbers.**

This rule is implemented using regex. A percentage number is defined as the combination of a digit with the percentage sign, either with a space in between or without. If the sentence contains at least one percentage number, the tool returns 0 (rule is violated).

3. **Prefer digits to written out numbers.**

This rule is implemented using regex. The tool is given a pre-specified list of the most common German written-out numbers (such as „eins", „zwei" etc.). If the sentence contains at least one written out number, the tool returns 0 (rule is violated).

4. **Avoid special characters.**

This rule is implemented using regex. The rulebook by Netzwerk Leiche Sprache defines the following special characters as being complicated: quotation marks, percentage sign, ellipsis, semicolon, ampersand, opening round bracket, paragraph sign. Quotation marks and round brackets are tracked by using the opening sign, respectively. If the sentence contains at least one special character, the tool returns 0 (rule is violated).

Word Rules

5. **Always use the same words for the same things.**

This rule is implemented by counting word repetitions. To do so, the text is parsed with spaCy and tokens are counted. Only nouns and verbs are considered for this rule. At first, a list of repetitive words is created (at least 2 repetitions). If a sentence contains one of these repetitive words, the rule is fulfilled and the tool returns 1.

6. **Use short words. If this is not possible: Separate long words by a hyphen.**

This rule is implemented by counting the length of words. Words with 10 or more characters are considered as long because an average German word in the Duden series has a length of roughly 10 characters. Next, it is evaluated if the long word is separated by a hyphen. If this is the case, the tool returns 1 (rule fulfilled). If it doesn't contain a hyphen, the tool returns 0 (rule violated). If there is no long word in the sentence at all, the rule is fulfilled (tool returns 1).

7. **Avoid abbreviations.**

This rule is implemented using regex. This search finds only 2 sequential and with period abbreviated characters (with and without space). If a sentence contains at least on abbreviation, the tool returns 0 (rule is violated).

8. **Use verbs.**

This rule is implemented with spaCy. The rule is interpreted (on the basis of the examples) as „avoid nominalisated verbs". As there is no list of all German nominalisations, this rule is approximated as follows: Generally, a search for nominalisations with ending „ung" is performed. Then, a spaCy matcher object is used to find nouns ending on „ung" with a preceding article. A second approach searches for titlecase tokens ending on „ung" or „ungen" (plural). If their ending can be replaced by the German verb ending „en" **and** if this results in a word that is part of the language model's vocabulary, the found word is considered as being a nominalisation. Overall, if a sentence contains at least one nominalisations, the rule is considered violated (tool returns 0).

9. **Use active words.**

This rule is implemented with spaCy. The rule is interpreted as „avoid passive voice". To find passive voice in a sentence, a spaCy matcher object is used. The matcher searchs for the lemma of the verb „werden" (to be) and another verb in participle form. If a sentence contains at least one passive voice phrase, the tool returns 0 (rule violated).

10. **Avoid genitive case.**

This rule is implemented with spaCy. With the help of spaCy's token attributes and the morph tag, a search for morph = „Case: Gen" is performed. If a sentence contains at least one word in genitive case, the tool returns 0 (rule violated).

11. **Avoid subjunctive case.**

This rule is implemented with spaCy. SpaCy offers a token-tag for subjunctive case but the first test showed poor results and a high number of false-positive matches. To find subjunctive case, a list with the lemma of German verbs that build unique forms in subjunctive is created. These verbs are i.e. „können" („can"), „müssen" („must") or „dürfen" („may"). To define the results of the formerly named tag, a search for the morph tag „Mood: Sub" is performed. Then, it is checked whether the

word is part of the former list. This comparison is based on the words' lemmata. If a sentence contains a subjunctive form, the tool returns 0 (rule is violated).

12. Use positive language.

This rule is implemented with spaCy. According to the rulebooks' examples, negations should be avoided. To track this, a search for tokens with the lemma „kein" („no") or „nicht" („not") is performed. If a sentence contains a negation, the tool returns 0 (rule is violated).

Sentence Rules

13. Write short sentences.

This rule is implemented with spaCy. A long sentence is interpreted as containing 9 or more words. This decision goes back to a study by Best (2002) who states that the minimum sentence length amounts to 6 and the maximum length to 12 words per sentence for children's and youth' literature. To track the compliance, the number of words per sentence (all tokens that are not punctuation) is counted. If the sentence contains 9 or more words, the tool returns 0 (rule violated).

14. Use one statement per sentence.

This rule is implemented using spaCy. There are two definitions of „one statement per sentence": 1. All sentences containing a direct speech phrase (i.e.: „The weather is nice", he said.) are considered as having multiple statements. To track this, the sentence is searched for a closing quotation mark that is followed by a comma. If this is the case, the rule is considered violated and returns 0. If not, the 2nd search is performed: 2. POS-tagging. All sentences with at least 2 verbs (token.pos_ == „VERB") and 2 associated subjects (token.dep_ == „sb") are considered as violating the rule. If this is the case, the tool returns 0; if not, the tool returns 1.

15. Use a simple sentence structure.

This rule is implemented using spaCy. There are two considerations for defining a „simple sentence structure": 1. All sentences with a subordinate clause are considered complicated. To track this, the sentence is searched for existing commas. If this is the case, the rule is considered violated and the tool returns 0. 2. The rulebook's example focuses on the chronology of subject and verb. If the verb precedes the corresponding subject, the sentence is considered complex. This is picked up. To evalutate the chronology of subject and verb in a sentence, a spaCy dependency matcher object is created. If the verb precedes the subject, the tool returns 0 (rule is violated).

16. The sentences may begin with: Or, If, Because, And, But.

This rule is implemented with spaCy. It is checked, whether the first word (not token!) in a sentence equals one of the allowed sentence beginnings. If this is the case, the rule is considered fulfilled and the tool returns 1.

IV.II. Easy Language Results

After ELEphanT has evaluated all Easy Language rules, these results are saved in an excel file – if multiple text files are evaluated, an excel file for each text file is created. These files contain all rule results on sentence level – one excel sheet row equals one sentence of the input text. In addition, another excel file is created that summarizes all single text results. This is especially interesting if you plan to compare multiple texts.

Sentence Level Results

This excel file contains multiple information about the analyzed text file: the number of tokens, words, and characters per sentence, the average word length per sentence in characters, the number of satisfied rules per sentence (absolute), the number of satisfied rules per sentence (in %), and the results of the 16 Easy Language rules. Rule nr. 6 is implemented in two different functions (long words and long words with hyphens) and then merged. Thus, there are three different associated columns for this rule.

Text Level Results

Independently from the number of input text files, the tool always creates one summarizing excel file on text level. This contains the following information: the total number of sentences, tokens, characters and words per text, the average sentence length per text (in words), the 16 rule evaluations, each in absolute and relative numbers, and 3 different Easy Language scores. The 1. Easy Language score is calculated in building the average number of satisfied rules per text (absolute and relative score). The latter relative score is also called „unweighted Easy Language score“. The 2. score is a weighted Easy Language score. The weighting builds on a reference test corpus of 10 Easy Language texts, mostly from German public authorities (see below). The tool was applied to the corpus and average rule compliance scores were extracted as thresholds for the weighting. The 3. score counts the number of „perfect“ sentences per text in absolute and relative numbers. A perfect sentence is defined as a sentence that violates 1 rule at maximum. This score is also given both in absolute and relative numbers.

5 V. Tool Evaluation

V.I. Gold Standard: Easy Language Reference Texts

To be able to verify the tool’s results, two groups of texts were collected. The first group is a corpus of 10 Easy Language texts. These texts have been randomly selected from the Internet when searching for „Leichte Sprache Beispiele“ („Easy Language examples“). The texts were processed by the tool: According to pre-processing rules, the texts were modified and separately saved. Then, the Easy Language scores were evaluated for each text.

On the basis of the summarizing text_level_result.xlsx-file, the rules' average scores were calculated and then used to implement the 2. Easy Language score (the weighted score). Assuming that a low average score of a single rule means that this rule is not very important for Easy Language, the weighted score is created in multiplying the relative rule results with the average score. In example: On average, the first two rules are never violated in the Easy Language reference corpus. This leads to the assumption that these rules are very important. On the contrary, rule nr. 16 is violated very often in all texts. Therefore, it is assumed that this rule is not of high importance for Easy Language.

These sample texts are mainly non-fictional Easy Language texts by public authorities. One text is a fairy tale translation. The 2nd group consists of only two texts: a 1:1 translation, meaning a standard German and Easy Language version of the same text: the general terms and conditions by the Bundeszentrale für politische Bildung (bpb).

V.II. Degree of Easy Language in Easy Language Texts

For both the Easy Language Reference Corpus and the 1:1 parallel translation, the average scores of the three Easy Language scores were calculated. The following table presents the results of ELEphanT on real-world Easy Language texts.

	Unweighted EL Score	Weighted EL Score	Perfect Sentences
Easy Language Corpus	0.836	0.768	0.13
1:1 Easy Language	0.798	0.727	0.059
1:1 Standard German	0.697	0.650	0.005

V.III. Discussion

It may seem strange that even the Easy Language texts from the corpus have a weighted Easy Language score of about 0.768 (for the terms and conditions-text the weighted Easy Language score is 0.727, respectively), whereas one would expect a higher score. There are multiple possible explanations for this interesting result:

1. Even Easy Language texts do not always adhere to all rules which is even more interesting because Easy Language is known to be firmly rule-based and one would expect it to strictly stick to all rules.

2. When collecting the Easy Language texts, in most cases it was not possible to find out to which rulebook the texts adhere to. In 2015, the German researcher Maaß highlighted that although different rulebooks show some intersections, there are many rules that are unique in one set of rules. This results in a situation where the choice of a rulebook highly effects the results of ELEphanT. Even more, because ELEphanT focuses on one specific rulebook. If a text from the corpus follows other rules, this may cause lower Easy Language scores.

3. Even if the corpus texts adhere to the same rulebook as ELEphanT, they may have been implemented differently. As has been noted in section IV.I, concretization is necessary for many

rules. If the translator of one of the EL corpus texts used a different definition for the rules – or a different priority –, this may also cause lower Easy Language scores with ELEphanT. To give an example: For rule nr. 13, I decided that a long sentence is defined as containing 9 or more words. But this threshold is not determined by the rulebook, and if the translator decides differently, this effects the results.

4. Of course, ELEphanT doesn't evalaute all rules that are part of the rulebook by Netzwerk Leichte Sprache. At the same time, some rules can be evaluated approximatively only. In example, not all nominalisations are found with ELEphanT's implementation. If the tool returns a score of 1, this simply means that a word ending on „ung" hasn't been found. But there are other nominalisated word forms that are not tracked by ELEphanT.

Thus, the tool's Easy Language score should not be interpreted as an absolute score but rather as an approximation. Still, this tendency may be valuable for assessing a text's complexity. At the same time, the single results for the rules may be helpful in determining where to improve a text to make it more accessible.