Everybody likes short sentences - A Data Analysis for the Text Complexity DE Challenge 2022

Ulf A. Hamster
Berlin-Brandenburgische Akademie der Wissenschaften, Berlin, Germany
{hamster}@bbaw.de

Abstract

The German Text Complexity Assessment Shared Task in KONVENS 2022 explores how to predict a complexity score for sentence examples from language learners' perspective. Our modeling approach for this shared task utilizes off-the-shelf NLP tools for feature engineering and a Random Forest regression model. We identified the text length, or resp. the logarithm of a sentence's string length, as the most important feature to predict the complexity score. Further analysis showed that the Pearson correlation between text length and complexity score is about $\rho \approx 0.777$. A sensitivity analysis on the loss function revealed that semantic SBert features impact the complexity score as well.

1 Introduction

We create and extract features from pre-trained NLP models and train a random forest model to predict scores of the TextComplexityDE dataset (Naderi et al., 2019) because we want to find out what evaluation criteria the annotators, here language learners, used. Using handcrafted features was the common approach before the breakthrough and wide adoption of deep learning models. For example, Lee et al. (2021) combine transformer models with random forest models based on 255 manually specified features for readability assessments. Xia et al. (2016) predict the CEFR-level of a text with support-vector machines and linguistic features, e.g., lexical, syntactic, discourse-based. Beinborn et al. (2014) and Lee et al. (2019) measure text difficulty with word familiarity, false cognates, morphological inflections, and phonetic complexity in C-Tests. Feng et al. (2009) handcraft linguistic features assuming these may be relevant due to human cognition, or resp., working memory limits. The advantage of manual feature engineering is that it allows to assess the impact of each feature or group of features later on, e.g., sensitivity on the loss function, and feature importance in random-forest. In other words, the model becomes partially explainable, and allows deriving feedback for practitioners such as language teachers.

2 Feature Engineering

We use sentence-level features addressing different language levels by using different types of features generated by or derived from off-the-shelf NLP tools (Table 1).

language level	types of features
semantics	Contextual sentence embeddings
syntax	Node distances in dependency trees
morphosyntax	Part-of-Speech tag distribution Lexical & grammatical properties
phonetics	IPA-based consonant clusters
morphology	Lexeme statistics Char- & Bi-gram frequencies
lexicology	Word frequencies
-	Text length

Table 1: Types of features and their language level.

Contextual sentence embeddings. We use feature vectors from the pretrained Sentence-BERT model `paraphrase-multilingual-MiniLM-L12-v2` what is trained on parallel corpora (Reimers and Gurevych, 2019). Using a multilingual contextualized sentence embeddings for German may help with code-switching phenomena and adoption of neologisms.

Node distances in dependency trees. We parse sentences with Trankit v1.1.1 `german-hdt` (Nguyen et al., 2021), what is trained on the Hamburg Treebank (Foth et al., 2014), to retrieve the dependency tree, PoS tags, and other morphosyntactic properties. We compute the adjusted node distance as the shortest path between each word token in the dependency tree minus their distance

in the token sequence. We, finally, compute the empirical distributions over adjusted node distances between $[-5, 15]$ whereas fat tail occurrences are assigned to -5 and 15.

Part-of-Speech (PoS) tag distribution. We compute the empirical distribution over the 17 Universal Dependency PoS tags for the word tokens of each sentence, i.e., the percentage of tokens of a specific PoS tag within a sentence.

Other lexical & grammatical properties. We compute the percentage of word tokens that have specific lexical and grammatical properties.

Features	Properties
Verb form	VerbForm={Fin, Inf, Part, Mod}
Finite verb forms	Mood={Ind, Imp}
Aspect	Aspect=Perf
Verb tense	Tense={Pres, Past}
Gender	Gender={Fem, Masc, Neut}
Number	Number={Sing, Plur}
Person	Person={1, 2, 3}
Case	Case={Nom, Dat, Gen, Acc}
Adposition	AdpType={Post, Prep, Circ}
Conjunction	ConjType=Comp
Comparison	Degree={Pos, Cmp, Sup}
Cardinal number	NumType=Card
Particle type	PartType={Res, Vbp, Inf}
Pronominal type	PronType={Art, Dem, Ind, Prs, Rel, Int}
Negation	Polarity=Neg
Possessive words	Poss=Yes
Reflexive words	Reflex=Yes
Alternative form	Variant=Short
Foreign word	Foreign=Yes
Hyphenated	Hyph=Yes
Punctation	PunctType={Brck, Comm, Peri}

Table 2: List of counted lexical and grammatical features and properties.

IPA-based consonant clusters. We convert the sentences to IPA symbols with Epitran v1.18 `deu-Latn` (Mortensen et al., 2018) and a) count the number of IPA consonants, b) consonant clusters of two, and c) consonant clusters of three or more divided by the number of IPA symbols.

Lexeme statistics. We parse lexemes of words with SMOR (Schmid et al., 2004; Schmid, 2006). SMOR returns all possible morphological variants that can be inferred from the surface form of a word. We count a) syntactical ambivalent variants for each word, b) ambivalent lexeme combinations of a word, and c) take the variant with the most lexemes for a word as approximation for the working memory requirement to comprehend composites. Each of the three frequencies are divided by the number of words in the sentence.

Char- & Bi-gram frequencies. DeReChar contains the character and bi-gram frequencies of the DeReKo corpus (IDS, 2022). We apply max-scaling to each, the character frequency list, and bi-gram frequency list, to values between 0 and 1. For each sentence, we look up all scaled character frequencies, sum them up, and divide by the string length of the sentence example. In case of bi-gram, we window-slide over the string and divided the looked up frequencies by the string length minus one.

Word frequencies. The COW16 list contains the frequencies approx. 42 Mio. words from the COW web corpus (Schäfer and Bildhauer, 2012; Schäfer, 2015),[1] and we removed $\sim$ 97 % of the least frequent words for faster lookup. Max-scaling is applied to the logarithm of 1 plus the COW frequencies. For each sentence example, the scaled word frequencies are assigned to one of six bins if their values falls within brackets $[0, 1/6, 1/3, 1/2, 2/3, 5/6, 1]$. The bin counts are divided by the number of words of the sentence, and used as features.

Text length. We measure the text length in two ways. First, the logarithm of 1 plus the number of words per sentence. Second, the logarithm of 1 plus the string length.

3 Experiments

Dataset. The subject of this shared task is the TextComplexityDE dataset by Naderi et al. (2019). Its training set contains 1000 German sentence example from Wikipedia. Each sentence example had 3 items with Likert-scale from 1 to 7 resulting in a) complexity, b) understandability, and c) lexical difficulty scores. And 369 German language learners provided, 10650 valid sentence ratings.

Random-Forest Feature Importance. We trained the multi-output random-forest (Breiman, 2001) implementation of Scikit-Learn package (Pedregosa et al., 2011) with 100 trees, max. tree depth of 16, and at least 10 samples per leaf, as well as bootstrap aggregation with subsample size of 50% and out-of-bag errors. Table 3 shows the Gini or impurity-based feature importance scores of the trained random-forest model. The text

[1] https://github.com/olastor/german-word-frequencies

length, or logarithm of the number of characters per sentence ($length_1$), appears to be the single most important feature of the model.

feature	fi score
$length_1$	.6042
$sbert_{156}$	.0170
$frequency_2$	.0151
$sbert_{173}$	.0095
$sbert_{69}$	.0077

Table 3: Top-5 feature importance scores of the fully trained Random Forest model.

The text length. The linear relationship between complexity score and the logarithm of the number of characters per sentence has a Pearson correlation coefficient of $\rho \approx 0.777$ with a p-value $< 10^{-202}$.

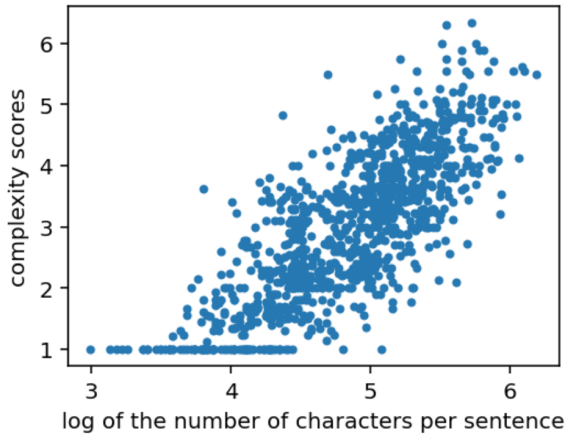

Figure 1: Complexity score versus the log of the number of characters per sentence, or text length ($length_1$).

Sensitivity Analysis. We systematically replaced each of the nine types of inputs with random numbers, computed the RMSE and subtracted the training loss. Table 4 shows the impact of the two text length features, and that semantic SBert features still have some influence on the *complexity score*. The text length has less impact on the *understandability score*, and the semantic SBert features more impact on the *lexical score*.

We also trained a Random Forest model without the text length features. The impact of morphological features and word frequencies seems more visible. The semantic SBert features have still an impact on the loss function. The impact of node distance feature can be explained by text length because larger node distances require longer sentences.

input type	complex.	underst.	lexical
Sentence semantic	**0.2174**	0.2874	**0.3426**
Node distances	0.0039	0.0043	0.0049
PoS tags	0.0157	0.0160	0.0179
lex. & syntact. prop.	0.0078	0.0079	0.0081
IPA consonant clusters	0.0008	0.0011	0.0012
Lexeme stat.	0.0038	0.0050	0.0055
Word freq.	0.0211	0.0226	0.0354
Char & Bi-gram freq.	0.0203	0.0199	0.0229
Text length	**2.3412**	**1.5246**	2.1846

Table 4: Losses with pertubated inputs per input types subtracted by the training loss.

input type	complex.	underst.	lexical
Sentence semantic	0.1580	0.1810	**0.2023**
Node distances	**0.3309**	0.2308	0.2753
PoS tags	0.0131	0.0136	0.0155
lex. & syntact. prop.	**0.1095**	0.0847	0.0969
IPA consonant clusters	0.0030	0.0031	0.0037
Lexeme stat.	0.0075	0.0067	0.0089
Word freq.	0.0859	0.0812	**0.1006**
Char- & Bi-gram freq.	0.0281	0.0281	0.0322

Table 5: Sensitivity analysis for the Random Forest model without text length features.

4 Discussion

An explanation for the text length as the dominant feature for the TextComplexityDE dataset could be the working memory (Miller, 1956; Cowan, 2001), or cognitive load theory for sentence comprehension (Mikk, 2008). Foreign language texts are new to a language learner to varying degrees. Dealing with new things can require more conscious and analytical information processing, which is more cognitively demanding. Respondents may have developed and applied text length as a heuristic while answering the survey, what can be explained by the effort-reduction framework (Shah and Oppenheimer, 2008). In extreme cases, a study participant could only measure the black and white contrast of the dark letters on a light background as an approximation for the text length, i.e., a person do not even have to read the text to assign a score. However, some part of the complexity score is related to semantic SBert features, i.e., the text content still mattered to the survey participants. The other proposed evaluation criteria (e.g., node distance, consonant cluster, word frequency) cannot explain the dependent variables of the TextComplexityDE dataset.

5 Conclusion

Although the study designer can ask for thoughtful responses, this does not prevent study participants

or annotators from using or developing heuristics such as text lengths. We suggest two solutions to prevent annotators from using text length as scoring heuristic. First, use text length as a control variable during the survey, i.e., a participant assess a set of sentence examples of a similar text length. This would force the participant to consider other evaluation criteria related to the survey question. Although the implementation is easy, the annotation time would increase because participants might develop more differentiated sets of evaluation criteria. Second, ask the participant to translate each German sentence example into their native language before assigning a score. This countermeasure would ensure that participants spend time for details, and may weight less obvious evaluation criteria higher, e.g., they became aware of the syntactic or lexical similarity between both languages. The drawback is that the annotation time would increase considerably when survey participants create a parallel corpus.

Acknowledgments

I dedicate this paper to my late nephew, Max Joshua Hamster († June 18, 2022).

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 433249742.

References

Lisa Beinborn, Torsten Zesch, and Iryna Gurevych. 2014. Predicting the difficulty of language proficiency tests. *Transactions of the Association for Computational Linguistics*, 2:517–530.

Leo Breiman. 2001. Random Forests. *Machine Learning*, 45(1):5–32.

Nelson Cowan. 2001. The magical number 4 in short-term memory: A reconsideration of mental storage capacity. *Behavioral and Brain Sciences*, 24(1):87–114.

Lijun Feng, Noémie Elhadad, and Matt Huenerfauth. 2009. Cognitively motivated features for readability assessment. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 229–237, Athens, Greece. Association for Computational Linguistics.

Kilian A. Foth, Arne Köhn, Niels Beuck, and Wolfgang Menzel. 2014. Because size does matter: The hamburg dependency treebank. In *LREC*, pages 2326–2333. European Language Resources Association (ELRA).

Ulf A. Hamster. 2021a. node-distance: Tree node distances as features.

Ulf A. Hamster. 2021b. A simple json database to lookup the properties of ipa symbols.

IDS. 2022. DeReWo – Korpusbasierte Grund-/Wortformenlisten.

Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10669–10686, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ji-Ung Lee, Erik Schwan, and Christian M. Meyer. 2019. Manipulating the difficulty of C-tests. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 360–370, Florence, Italy. Association for Computational Linguistics.

Jaan Mikk. 2008. Sentence length for revealing the cognitive load reversal effect in text comprehension. *Educational Studies*, 34(2):119–127.

George A. Miller. 1956. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63(2):81–97.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Paris, France. European Language Resources Association (ELRA).

Babak Naderi, Salar Mohtaj, Kaspar Ensikat, and Sebastian Möller. 2019. Subjective assessment of text complexity: A dataset for german language.

Minh Van Nguyen, Viet Dac Lai, Amir Pouran Ben Veyseh, and Thien Huu Nguyen. 2021. Trankit: A light-weight transformer-based toolkit for multilingual natural language processing. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Roland Schäfer. 2015. Processing and querying large web corpora with the cow14 architecture. Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora (CMLC-3), Lancaster, 20 July 2015, pages 28 – 34, Mannheim. Institut für Deutsche Sprache.

Roland Schäfer and Felix Bildhauer. 2012. Building large corpora from the web using a new efficient tool chain. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 486–493, Istanbul, Turkey. European Language Resources Association (ELRA).

Helmut Schmid. 2006. A Programming Language for Finite State Transducers. In Anssi Yli-Jyrä, Lauri Karttunen, and Juhani Karhumäki, editors, *Finite-State Methods and Natural Language Processing*, volume 4002, pages 308–309. Springer Berlin Heidelberg, Berlin, Heidelberg. Series Title: Lecture Notes in Computer Science.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German computational morphology covering derivation, composition and inflection. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal. European Language Resources Association (ELRA).

Anuj K Shah and Daniel M. Oppenheimer. 2008. Heuristics made easy: an effort-reduction framework. *Psychological bulletin*, 134 2:207–22.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. 2016. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22, San Diego, CA. Association for Computational Linguistics.

A Appendices

A.1 Developed Software

- Code for the experiments: github.com/ulf1/study-370b
- Node versus token distances in a dependency tree: pypi.org/project/node-distance (Hamster, 2021a).
- JSON database with IPA symbol properties, and routines to count IPA-based consonant clusters: pypi.org/project/ipasymbols (Hamster, 2021b)