Proceedings of the GermEval 2022 Workshop on Text Complexity Assessment of German Text

18th Conference on Natural Language Processing
KONVENS 2022

University of Potsdam
September 12, 2022

Edited by

Salar Mohtaj
Babak Naderi
Sebastian Möller

Preface

Text forms an integral part of exchanging information and interacting with the world. Along with the other types of content (e.g., image and video), textual content has been increased drastically in amount and importance during recent years. Text complexity (in the following used interchangeably with text readability) is one of the factors which affects a reader's understanding of text. A readability score is the mapping of a body of text to mathematical unit quantifying the degree of readability. We present the GermEval 2022 Workshop on Text Complexity Assessment of German Text. The task included developing Natural Language Processing (NLP) models to automatically assign a complexity score in the range of 1 to 7 to German texts. In other words, the shared task is a text regression task in which the output is continuous variables between 1 and 7. We received 84 submissions in the test phase from 10 teams. The results and the data sets can be found at the shared task website at https://qulab.github.io/text_complexity_challlenge/.

We are grateful to the *KONVENS* 2022 conference organizers for their support on the shared task. We would also like to show our gratitude to the participants of *GermEval* 2022 whose participation and effort made *GermEval* 2022 a great success. Finally we thank Kaspar Ensikat for his support for data preparation and Faraz Maschhur, Chuyang Wu, and Max Reinhard for developing the baseline model.

Potsdam, September 2022

The organizing committee

Organizers:
Salar Mohtaj (Technische Universität Berlin)
Babak Naderi (Technische Universität Berlin)
Sebastian Möller (Technische Universität Berlin)

Overview of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text

Salar Mohtaj[1,2], Babak Naderi[1], and Sebastian Möller[1,2]

[1]Quality and Usability Lab, Technische Universität Berlin, Berlin, Germany
[2]German Research Centre for Artificial Intelligence (DFKI), Labor Berlin, Germany
{salar.mohtaj|babak.naderi|sebastian.moeller} @ tu-berlin.de

Abstract

In this paper we present the GermEval 2022 shared task on Text Complexity Assessment of German text. Text forms an integral part of exchanging information and interacting with the world, correlating with quality and experience of life. Text complexity is one of the factors which affects a reader's understanding of a text. The mapping of a body of text to a mathematical unit quantifying the degree of readability is the basis of complexity assessment. As readability might be influenced by representation, we only target the text complexity for readers in this task. We designed the task as text regression in which participants developed models to predict complexity of pieces of text for a German learner in a range from 1 to 7. The shared task was organized in two phases; the development and the test phases. Among 24 participants who registered for the shared task, ten teams submitted their results on the test data.

1 Introduction

Text forms an integral part of exchanging information and interacting with the world. Along with the other types of content (e.g., image and video), textual content has been increased drastically in amount and importance during recent years. Text complexity (in the following used interchangeably with text readability) is one of the factors which affects a reader's understanding of text (Dale and Chall, 1949). Readability is concerned with the relation between a given text and the cognitive load of a reader to comprehend it. This complex relation is influenced by many factors, such as a degree of lexical and syntactic sophistication, discourse cohesion, and background knowledge (Crossley et al., 2017; Martinc et al., 2021). A readability score is the mapping of a body of text to a mathematical unit quantifying the degree of readability. It is the basis of readability assessment. Readability assessment has diverse use cases and applications, such as helping people with disabilities and also facilitate choosing of learning material for second language learners (Aluisio et al., 2010).

In this paper, we present the challenge and results from the task of German text complexity assessment in *GermEval* 2022. The task includes developing Natural Language Processing (NLP) models to automatically assign a complexity score in the range from 1 to 7 to German texts, where 1 represent an easy to understand (i.e., simple) text/sentence and 7 shows a complex text for German learners. In other words, the shared task is a text regression task in which the output is a continuous variable between 1 and 7.

GermEval is a series of shared task evaluation campaigns that focus on Natural Language Processing for the German language. It started in 2014 with a shared task on German Named Entity Recognition (Benikova et al., 2014) and continued in the years after with different tasks from lexical substitution (Miller et al., 2015) to the task of identification of toxic, engaging, and fact-claiming comments (Risch et al., 2021) and German scene segmentation (Zehe et al., 2021).

The rest of the paper is organized as follow; Section 2 presents recent research on text readability and complexity assessment and related tasks. An overview of the shared task and the data set, resources and the evaluation metrics that have been used in the shared task are presented in Sections 3 and 4, respectively. We briefly review the submitted models and discussed the results in Sections 5. Finally, we conclude the paper and the *German Text Complexity Assessment* shared task in Section 6.

2 Related Work

In this section we provide an overview of related shared tasks in different languages, and also highlight a number of the recent approaches for the task of text complexity assessment.

2.1 Shared Tasks

To the best of our knowledge, no shared task has been held so far on text complexity assessment at a sentence level. However, there are a few competitions on word level complexity assessment.

Paetzold and Specia organized the a shared task on complex word identification as a *SemEval* 2016 task (Paetzold and Specia, 2016). The task was to develop systems that can predict whether a target word is complex for a non-native English speaker, knowing the context sentence. In other words, it was a binary classification task in which 1 means the target word is complex in the given context sentence, and 0 means it's a simple word for a non-native English speaker.

The next complex word identification shared task was organized at the BEA workshop in 2018 for different languages including English, German, Spanish and French. The shared task included two subtasks: The first task was a binary classification of a target word in a context sentence as being complex or not complex. The second task was a probabilistic classification in which the participants were asked to assign the probability of a target word being considered complex (Yimam et al., 2018).

There were two subtasks of complexity prediction of single words and multi-word expressions as a regression task in the *SemEval* 2021 lexical complexity prediction task (Shardlow et al., 2021). The data includes around 10,000 instances for lexical complexity in which the target words were annotated on a five point Likert scale.

Russian simple sentence evaluation in 2021 is another related activity in which the task was developing systems to generate a simplified version of a given input complex sentence in Russian (Sakhovskiy et al., 2021). The proposed data set for the task includes around 3,000 complex sentences, each have 2.2 corresponding simplified sentences on an average.

As another related effort, Stajner et al. organized a shared task for the assessment of text simplification in which systems should automatically assign a label (e.g., good, OK, and bad) to four aspect of the pairs of original and simplified sentences (Stajner et al., 2016). The four aspects of interests include the quality of the generated sentences from grammar, meaning preservation, simplicity, and overall quality point of views.

2.2 Approaches

In this section we overview some of the recent approaches and models for automatic text complexity and readability assessment. We review the state-of-the-art models for English and German texts.

As one of the recent models for English text readability assessment, (Lee et al., 2021) developed different hybrid models using traditional machine learning approaches based on handcrafted features, and also transformer-based models. Based on their experiments, the combination of RoBERTA and Random Forrest models could outperforms the other models and achieved almost perfect classification accuracy (Lee et al., 2021). Hybrid models show promising results for the task in different languages and were the main trend among the submitted models for *GermEval* 2022.

Naderi et al. proposed a model for German text readability assessment based on linguistic features (Naderi et al., 2019b). They extracted traditional, lexical and morphological linguistic features (73 features in total). Their experiments show that again the Random Forest Regressor outperforms the other supervised models including SVM, Linear Regression, and Polynomial Regression models for the task (Naderi et al., 2019b).

In another study Weiss and Meurers proposed a model for sentence-wise German readability assessment for L2 readers (Weiss and Meurers, 2022). They compared different machine learning models in two different tasks for readability assessment; predictive regression and sentence pair ranking. The obtained results in their experiments show that a Bayesian Ridge Regression model achieved the best performance against the other models including the proposed model in (Naderi et al., 2019b) and also against the widely used readability formulae for the task of predictive regression. Moreover, regarding the document level text complexity assessment, their findings show that the readability of texts is driven by the maximum rather than the overall readability scores on the sentence level.

3 Task Description

In this section we describe the proposed task in detail. The data set and the evaluation metrics are presented in the next section.

The mapping of a body of text to a mathematical unit quantifying the degree of readability is the basis of readability assessment. This quantified unit is significant in informing the reader about how difficult the text content is to read. We defined the task of German text complexity assessment as a text regression task in which the participants were asked to develop systems to automatically assign a variable in the range from 1 to 7 to given German texts. We considered German learners at the B level as the target group. This means the system should predict the complexity/difficulty of a piece of text for a person who learns German at a B level.

The shared task is organized on the *Codalab* platform (Pavao et al., 2022), where the participants could access the data and submit their prediction on the provided data sets and get informed about the obtained results via the platform. More information about the competition is accessible via the corresponding web-page on the Codalab website [1].

Although the task is defined as a text regression task, there is no restriction on re-formulation of the task. Moreover, there was no restriction about using additional data sets for training purposes.

The shared task is organized in two phases; the development and the test phases. During the development phase the teams could develop their systems and test it against a validation data set. There was no restriction on the number of submissions during the development phase. The obtained results on the validation set were accessible for the teams immediately after submitting the predictions.

The test phase was a one week time period in which the participants could submit their results on the provided test data set. The test data was shared with the participants one week before the start of the test phase. During the test phase each team could submit a maximum number of two submissions per day on the test data set. The participants could only know about the achieved results on the test data (i.e., the leaderboard) when the competition ended. The detailed information about the provided data set and the evaluation metrics are presented in Section 4.

4 Data Set and Evaluation

In this section we discuss briefly the compiled data set for the competition and also overview the evaluation metrics that have been used to assess and ranked the submitted results.

4.1 Data Set

Three different data sets were available to the participants during the competition. We provided a training data set with complexity scores that could be used to train and tune the models and the systems. Moreover, two collections of sentences without the complexity score were shared as the validation and the test sets. The participants could evaluate their models using this data set during the development phase.

4.1.1 Train set

The training data set consisting of 1,000 German sentences taken from 23 Wikipedia articles. The data set includes subjective assessment of different text-complexity aspects provided by German learners at level A and B (Naderi et al., 2019a).

An online survey system was created to collect the subjective assessment of the 1,000 sentences using three items each rated on a 7-point Likert scale. A survey session consisted of training and rating sections. The training section was containing three sentences which participants needed to rate on the same scale as the main section. The sentences in the training section were constant and represent very easy, average and very complex sentences. Afterward, participants rated *complexity*, *understandability* and *lexical difficulty* of ten sentences. For each sentence in the data set the Mean Opinion Score (MOS) is calculated. The MOS score is the arithmetic mean over the all ratings of a particular aspect (complexity, understandability or lexical difficulty) provided for that sentence. The data set is published as *TextComplexityDE* in (Naderi et al., 2019a). For this shared task we only used the *complexity* scores of the sentences.

Figure 1 shows a few sample sentences from the training set. The training data set is freely available in a GitHub repository[2]. Moreover, a more detailed description of the *TextComplexityDE* data

[1] https://codalab.lisn.upsaclay.fr/competitions/4964

[2] https://github.com/babaknaderi/TextComplexityDE

Als Nebenprodukt entstand damals natürlich auch die erste Seifenblase.

MOS complexity score: 1.60

Translation: As a by-product, of course, the first soap bubble was created at that time.

In Abgrenzung zum klassischen Rasiermesser wird ein Rasiermesser mit Wechselklinge als Shavette bezeichnet.

MOS complexity score: 3.25

Translation: In distinction from the classic razor, a razor with interchangeable blade is called a shavette.

In Pompeji gefundene Exemplare von frühen Klapp-Rasiermessern mit 12 Zentimeter langen trapezförmigen Klingen und Griffen aus Elfenbein gehörten als Luxusobjekte zum Hausstand höherer Schichten.

MOS complexity score: 4.36

Translation: Specimens of early folding razors with 12-centimeter-long trapezoidal blades and ivory handles found in Pompeii belonged to the household of higher classes as luxury objects.

Figure 1: Sample sentences from the training set

set including the conducted pilot study to determine relevant dimensions of text complexity and the manually simplified sentences are presented in (Naderi et al., 2019a).

4.1.2 Test set

The ratings for the validation and test data sets are collected in four different experiments. For each experiment, 100 sentences were complied in which 80 sentences were from 18 different Wikipedia articles, and 20 sentences were shared between all experiments and taken from the *TextComplexityDE* data set. Participants are recruited through online German learner groups in social media and also language schools. For online participants, there was a short mandatory listening and comprehensive language test to make sure they have basic to intermediate knowledge of German. We used a same 7-point Likert Scale as it was used in the training data set (*TextComplexityDE*). In the data cleansing step, all submissions from users with one of the following conditions were removed from the data.

- Users with wrong answer to the gold standard question[3]
- Users who failed in the language test
- Users with specific click patterns (i.e. small variance) or those who were too fast in finishing a session

Like the *TextComplexityDE* data set, a MOS score for complexity is calculated for each sentence. Using the 20 shared sentences in each experiment, a first-order mapping function for MOS values from each experiment to the MOS values of the *TextComplexityDE* data set are fitted. This was done to remove the well-known bias and gradient between different subjective tests.

The final data set includes 310 new sentences from 18 Wikipedia articles which were rated by a minimum of 16 participants. 100 sentences of this data have been used as the validation set. The participating teams used the validation set to tune their models and parameters during the development phase. The reminding 210 sentences were used as test data set to assess the performance of the submission in the test phase. All the reported results in this paper are the achieved results on the test data set with 210 instances.

Table 1 provides a summary of statistics and frequency distribution of the training and test data sets. Moreover, the histogram of MOS values in the training and test data sets are presented in Figure 2. As it is highlighted in the figure, the sentences in the training set tend to be more balanced. In other words, more complex and difficult sentences are presented in the training data set, compared to the test data set.

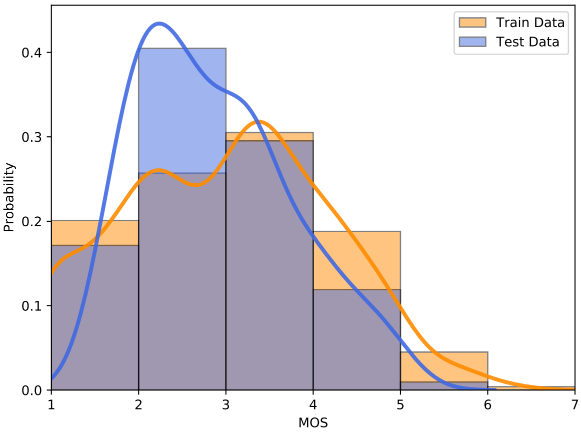

Figure 2: The distribution of MOS values in the training and test data sets

[3] gold standard question contains a text that its complexity is known to organizers (i.e. very simple or very complex ones) and used to filter participants who not following the instructions.

	Training data	**Test data**
Number of records (i.e., sentences)	1,000	210
Max length of sentences (in character)	487	486
Min length of sentences (in character)	19	38
Average length of sentences (in character)	147.3	160.03
Number of terms	20077	4400
Number of unique terms	7539	2249
Average of the complexity score	3.01	2.87
Standard Deviation	1.18	0.87

Table 1: Summary of statistics and frequency distribution of the training and test data sets

4.2 Evaluation Metrics

We used the Root Mean Square Error (RMSE) MAPPED metric to evaluate and rank the submitted results. Moreover, the normal RMSE scores were evaluated and reported.

RMSE shows the root of average squared difference between the estimated values $\widehat{y}_i$ (complexity scores) and the actual value y for the sentence i, as presented in the following equation.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \widehat{y}_i)^2}{N}} \quad (1)$$

Since slightly different ratings and consequently different MOS values could be obtained by repeating a subjective test and adding bias to the data, the RMSE MAPPED score has been used to assess the submitted runs. We used a mapping function to get ride of this offset/bias. The RMSE MAPPED is calculated based by the following steps:

1. A team submits its predictions (mos_pre).

2. A f(mos_pre) function is created by minimizing the absolute value between (true_mos) and f(mos_pre).

3. We call the outcome of the function f to be mapped_mos_pre:
mappend_mos_pre = f(mos_pre)

4. We calculate the RMSE between the mappend_mos_pre and the true_mos.

The *f* function is created for each model, and is a linear function.

5 Results

In this section we present the baseline model and also survey the submitted models for the shared task.

5.1 Baseline Model

For the baseline model we fine-tuned a GBERT pre-trained model (Chan et al., 2020) on the training set. After feeding the input text into the model the last hidden state is passed through a dense linear layer by applying a *Tanh* activation. A dropout layer is also put on top before the output layer.

Regarding the hyper parameters, the *AdamW* optimizer (Loshchilov and Hutter, 2019) was used with a learning rate of $5e-5$. The model was fine-tuned in 3 epochs.

5.2 Proposed Models

In this section we highlight the main contributions of the proposed models in the shared task. The overall performance of the submitted results is presented in Table 2.

Among the submitted models, hybrid approaches in which the traditional machine learning models based on linguistic feature extraction are combined with state-of-the-art pre-trained language models show promising results for the task.

The top ranked team (Mosquera, 2022), HHUplexity team (Arps et al., 2022) and HIIG team (Asghari and Hewett, 2022) proposed hybrid models that combine a feature engineering approach and transfer learning via pre-trained transformers. Although the approaches are similar in general, different features and models have been used by different teams. For instance while Bert (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) are fine-tuned in (Mosquera, 2022), the HHUplexity team extracted features from Bert and DistilBERT (Sanh et al., 2019) and the HIIG team fine-tuned XLM-R (Conneau et al., 2020). Moreover, different approaches have been used by different teams to combine the outcome of the feature engineering models and the pre-trained models. However, the hybrid models couldn't always outperform the simple models. For instance, the obtained results from the HHUplexity team show that fine-tuning DistilBERT can outperform the other models including the hybrid model based on linguistic features. Also, the experiments from the HIIG team show that data augmentation could not increase the overall performance of the proposed model.

The AComplexity team (Blaneck et al., 2022), TUMuch Complexity team (Vladika et al., 2022) and TUM Social Computing team (Anschütz and Groh, 2022) used a similar approach of hybrid models. The AComplexity team extracted 154 features for each sentence and fine-tuned GBERT and GPT-2-Wechsel (Minixhofer et al., 2022) models. They combined the output of the pretrained model with the readability features calculated for each sentence using a multi-layer perceptron with two layers (Blaneck et al., 2022). On the other side, the TUMuch Complexity team stacked RoBERTa and Gaussian process models as the proposed hybrid approach. As the stacking approach, they averaged the output predictions of the Gaussian process model and the fine-tuned XLM-RoBERTa (Conneau et al., 2020). The TUM Social Computing team (Anschütz and Groh, 2022) computed 6 different readability formulae based on some statistics and combined them with the fine-tuned DistilBERT model. Their analysis on the relevance of different features on the predictions highlight the importance of pre-trained models and also some statistics from text like the average sentence length (Anschütz and Groh, 2022).

The BBAW Zentrum Sprache team (Hamster, 2022) trained a random forest model on the set of extracted features like statistical, lexical, and grammatical ones. They also extracted a set of features from pre-trained NLP models like Sentence-BERT (Reimers and Gurevych, 2019). Their experiments show the linear relationship between the complexity score and the logarithm of the number of characters per sentence. Moreover, their results reveal that Sentence-BERT features also impact the complexity scores.

Due to the fact that the provided training data set was small and included only 1,000 sentences, different teams applied different strategies to increase the training data. The Deepset team used more than 220,000 pseudo-labels to train Transformer-based models in order to refrain from feature engineering step (Kostic´ et al., 2022). They used 12,562,164 distinct sentences from German Wikipedia and other corpora like news articles from Zeit Online for their semi-supervised learning approach. The proposed approach includes training a base model on the training set and pseudo-labeling the collected corpus with the base model. Finally, the pre-trained language models Fine-tuned on the pseudo-labels and the training sets and trained a linear regression model on the out-of-fold predictions from the cross-validations (Kostic´ et al., 2022).

As another approach to increase the data set size, the LGirrbach team turned the text regression task into a pairwise regression for complexity prediction (Girrbach, 2022). In this setting, instead of the direct prediction of the complexity score for the sentences, the model receive two sentences and predicts the relative difference in complexity of two sentences. However, the obtained results on the training set during the development phase show that ”pairwise regression does not perform better than standard regression” (Girrbach, 2022). Unfortunately, the team could not test the proposed model on the test data set due to an error in the submission.

6 Conclusion

In this paper we described the *GermEval* 2022 task on ”Complexity Assessment of German Text”. The shared task is co-located with the Conference on Natural Language Processing (KONVENS) 2022. We presented the compiled data sets for the training and the test phases and the models proposed by the participants. The training and the test sets included 1,000 and 210 German sentences from Wikipedia articles, respectively, with a readability/complexity score from 1 to 7. Regarding the models, combining the traditional feature extraction models with state-of-the-art pretrained language models was the main trend in the submitted systems. Although different teams used different feature set, pre-trained models and also different strategies to combine the outcomes of the models, there were similarities between the overall procedure from different participants. Almost all of the submissions could outperform the transfer learning based model as the competition's baseline.

For the next round of the shared task, the interpretability of the models (i.e., explainability) can be taken into account to make the predictions more

Team name	**RMSE MAPPED**	**RMSE**
Alejandro Mosquera (Mosquera, 2022)	**0.430**	0.449
AComplexity (Blaneck et al., 2022)	0.435	**0.442**
HIIG (Asghari and Hewett, 2022)	0.446	0.462
TUM Social Computing (Anschütz and Groh, 2022)	0.449	0.466
Deepset (Kostic ´ et al., 2022)	0.454	0.484
TUMuch Complexity (Vladika et al., 2022)	0.457	0.489
HHUplexity (Arps et al., 2022)	0.473	0.486
Baseline	0.477	0.489
CCL	0.516	0.586
BBAW Zentrum Sprache (Hamster, 2022)	0.553	0.583
LGirrbach (Girrbach, 2022)	-	-

Table 2: The results on the test data set

understandable. Moreover, the training and the test sets can be enriched by more samples from more diverse resources.

Acknowledgments

We are grateful to the *KONVENS* 2022 conference organizers for their support on the shared task. We would also like to show our gratitude to the participants of *GermEval* 2022 whose participation and effort made *GermEval* 2022 a great success.

Finally we thank Kaspar Ensikat for his support for data preparation and Faraz Maschhur, Chuyang Wu, and Max Reinhard for developing the baseline model.

References

Sandra Aluisio, Lucia Specia, Caroline Gasperin, and Carolina Scarton. 2010. Readability assessment for text simplification. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9.

Miriam Anschütz and Georg Groh. 2022. TUM Social Computing at GermEval 2022: Towards the Significance of Text Statistics and Neural Embeddings in Text Complexity Prediction. In *Proceedings of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text*, Potsdam, Germany, September. Association for Computational Linguistics.

David Arps, Jan Kels, Florian Krämer, Yunus Renz, Regina Stodden, and Wiebke Petersen. 2022. HHUplexity at Text Complexity DE Challenge 2022. In *Proceedings of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text*, Potsdam, Germany, September. Association for Computational Linguistics.

Hadi Asghari and Freya Hewett. 2022. HIIG at GermEval 2022: Best of Both Worlds Ensemble for Automatic Text Complexity Assessment. In *Proceedings of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text*, Potsdam, Germany, September. Association for Computational Linguistics.

Darina Benikova, Chris Biemann, Max Kisselew, and Sebastian Pado. 2014. Germeval 2014 named entity recognition shared task: Companion paper.

Patrick Gustav Blaneck, Tobias Bornheim, Niklas Grieger, and Stephan Bialonski. 2022. Automatic Readability Assessment of German Sentences with Transformer Ensembles. In *Proceedings of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text*, Potsdam, Germany, September. Association for Computational Linguistics.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In Donia Scott, Núria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6788–6796. International Committee on Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Scott A. Crossley, Stephen Skalicky, Mihai Dascalu, Danielle S. McNamara, and Kristopher Kyle. 2017. Predicting text comprehension, processing, and familiarity in adult readers: New approaches to readability formulas. *Discourse Processes*, 54(5-6):340–359.

Edgar Dale and Jeanne S Chall. 1949. The concept of readability. *Elementary English*, 26(1):19–26.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Leander Girrbach. 2022. Text Complexity DE Challenge 2022 Submission Description: Pairwise Regression for Complexity Prediction. In *Proceedings of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text*, pages 45–50, Potsdam, Germany, September. Association for Computational Linguistics.

Ulf A. Hamster. 2022. Everybody likes short sentences - A Data Analysis for the Text Complexity DE Challenge 2022. In *Proceedings of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text*, Potsdam, Germany, September. Association for Computational Linguistics.

Bogdan Kostic´, Mathis Lucka, and Julian Risch. 2022. Pseudo-Labels Are All You Need. In *Proceedings of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text*, Potsdam, Germany, September. Association for Computational Linguistics.

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, November. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Matej Martinc, Senja Pollak, and Marko Robnik-Sikonja. 2021. Supervised and unsupervised neural approaches to text readability. *Comput. Linguistics*, 47(1):141–179.

Tristan Miller, Darina Benikova, and Sallam Abualhaija. 2015. Germeval 2015: Lexsub–a shared task for german-language lexical substitution. *Proceedings of GermEval*, pages 1–9.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2022. WECHSEL: effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. In Marine Carpuat, Marie-Catherine de Marneffe, and Iván Vladimir Meza Ruíz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 3992–4006. Association for Computational Linguistics.

Alejandro Mosquera. 2022. Tackling Data Drift with Adversarial Validation: An Application for German Text Complexity Estimation. In *Proceedings of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text*, Potsdam, Germany, September. Association for Computational Linguistics.

Babak Naderi, Salar Mohtaj, Kaspar Ensikat, and Sebastian Möller. 2019a. Subjective assessment of text complexity: A dataset for german language. *CoRR*, abs/1904.07733.

Babak Naderi, Salar Mohtaj, Karan Karan, and Sebastian Möller. 2019b. Automated text readability assessment for german language: A quality of experience approach. In *11th International Conference on Quality of Multimedia Experience QoMEX 2019, Berlin, Germany, June 5-7, 2019*, pages 1–3. IEEE.

Gustavo Paetzold and Lucia Specia. 2016. Semeval 2016 task 11: Complex word identification. In Steven Bethard, Daniel M. Cer, Marine Carpuat, David Jurgens, Preslav Nakov, and Torsten Zesch, editors, *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pages 560–569. The Association for Computer Linguistics.

Adrien Pavao, Isabelle Guyon, Anne-Catherine Letournel, Xavier Baró, Hugo Escalante, Sergio Escalera, Tyler Thomas, and Zhen Xu. 2022. CodaLab Competitions: An open source platform to organize scientific challenges. Technical report, Université Paris-Saclay, FRA., April.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Julian Risch, Anke Stoll, Lena Wilms, and Michael Wiegand. 2021. Overview of the GermEval 2021 shared task on the identification of toxic, engaging, and fact-claiming comments. In *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments*, pages 1–12, Duesseldorf, Germany, September. Association for Computational Linguistics.

Andrey Sergeyevich Sakhovskiy, Alexandra Izhevskaya, Alena Pestova, E. Tutubalina, Valentin Malykh, Ivan Smurov, and E. Artemova. 2021. Rusimplesenteval-2021 shared task: Evaluating sentence simplification for russian.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. Semeval-2021 task 1: Lexical complexity prediction. In Alexis Palmer, Nathan Schneider, Natalie Schluter, Guy Emerson, Aurélie Herbelot, and Xiaodan Zhu, editors, *Proceedings of the 15th International Workshop on Semantic Evaluation, SemEval@ACL/IJCNLP 2021, Virtual Event / Bangkok, Thailand, August 5-6, 2021*, pages 1–16. Association for Computational Linguistics.

Sanja Stajner, Maja Popovic, Horacio Saggion, Lucia Specia, and Mark Fishel. 2016. Shared task on quality assessment for text simplification.

Juraj Vladika, Stephen Meisenbacher, and Florian Matthes. 2022. TUM sebis at GermEval 2022: A Hybrid Model Leveraging Gaussian Processes and Fine-Tuned XLM-RoBERTa for German Text Complexity Analysis. In *Proceedings of the GermEval 2022 Shared Task on Text Complexity Assessment of German Text*, Potsdam, Germany, September. Association for Computational Linguistics.

Zarah Weiss and Detmar Meurers. 2022. Assessing sentence readability for German language learners with broad linguistic modeling or readability formulas: When do linguistic insights make a difference? In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 141–153, Seattle, Washington, July. Association for Computational Linguistics.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Stajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In Joel R. Tetreault, Jill Burstein, Ekaterina Kochmar, Claudia Leacock, and Helen Yannakoudakis, editors, *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications@NAACL-HLT 2018, New Orleans, LA, USA, June 5, 2018*, pages 66–78. Association for Computational Linguistics.

Albin Zehe, Leonard Konle, Svenja Guhr, Lea Katharina Dümpelmann, Evelyn Gius, Andreas Hotho, Fotis Jannidis, Lucas Kaufmann, Markus Krug, Frank Puppe, Nils Reiter, and Annekea Schreiber. 2021. Shared task on scene segmentation @ KONVENS 2021. In *Proceedings of the Shared Task on Scene Segmentation co-located with the 17th Conference on Natural Language Processing (KONVENS 2021), Düsseldorf, Germany, September 6th, 2021*, volume 3001 of *CEUR Workshop Proceedings*, pages 1–21. CEUR-WS.org.

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 $< 10^{-202}$.

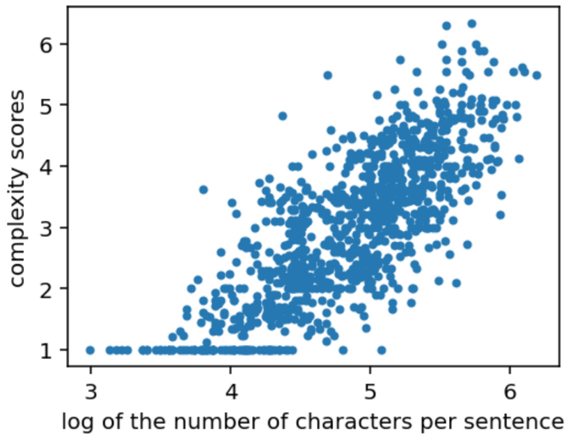

Figure 1: Complexity score versus the log of the number of characters per sentence, or text length (length_1).

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)

HIIG at GermEval 2022: Best of Both Worlds Ensemble for Automatic Text Complexity Assessment

Hadi Asghari *[1]
[1]AI & Society Lab
Humboldt Institute for Internet and Society
Berlin, Germany
`firstname.lastname@hiig.de`

Freya Hewett *[1,2]
[2]Applied Computational Linguistics
University of Potsdam
Potsdam, Germany
`firstname.lastname@hiig.de`

Abstract

In this paper we explain HIIG's contribution to the shared task Text Complexity DE Challenge 2022. Our best-performing model for the task of automatically determining the complexity level of a German-language sentence is a combination of a transformer model and a classic feature-based model, which achieves a mapped root square mean error of 0.446 on the test data.

1 Introduction

Text complexity is not only a highly interesting topic from a linguistic perspective; it also has several implications on a societal level. A text that has the appropriate complexity level for a specific reader not only ensures that the reader can fully understand the information presented in the text, but it also keeps the reader engaged and can help the reader to learn new structures and expand their vocabulary. This last point is particularly relevant for language learners and readers who are reading text in a language that is not their native language. The Text Complexity DE Challenge focuses on this specific target group as the task involves predicting the complexity of a sentence in German, which have been annotated on a scale of 1 to 7 by German learners whose language proficiency is at B level (on the CEFR scale). An overview of the shared task and the results from all the teams can be found in (Mohtaj et al., 2022).

In this paper we briefly report on related work in Section 2, before describing the dataset used in the shared task in Section 3. In Section 4 we outline our various approaches to the task, before reporting on the results and briefly discussing them in Section 5. In Section 6 we conclude the paper.

2 Related work

Previous work aimed at automatically assessing the text complexity level of sentences has focused mostly on the English language. Stajner et al. (2017) use the Newsela corpus (English-language newspaper articles, simplified at multiple levels for different aged school children) and calculate scores for unigrams, bigrams and trigrams by looking at what levels of the corpus they occur in. They experiment with different classifiers and achieve the best results with a Random Forest. Pitler and Nenkova (2008) conduct a small-scale analysis on 30 articles from the Wall Street Journal which have been manually annotated on a scale from 1-5 for the question of how well-written the article is. They investigate how various linguistic features correlate with these scores. Vocabulary and discourse relations are the strongest predictors of readability, followed by average number of verb phrases and length of the text. Lee et al. (2021) work with three English-language datasets and produce hybrid models which consist of a transformer based model combined with a feature-based model. They predict 3 and 5 classes (depending on the dataset) and achieve the state of the art, with a ROBERTA-based transformer model performing best.

Work on German-language text complexity assessment is fairly rare. Hancke et al. (2012) look at text-level binary readability classification using a corpus of 1627 articles in original form and a version aimed at children. Their classifier uses the Sequential Minimal Optimization algorithm with five groups of features (traditional readability formulas, lexical, syntactic, language model, and morphological), with a best accuracy score of 89.7%. Stodden and Kallmeyer (2020) work with various corpora from different languages from the text simplification domain and evaluate 104 different features using statistical tests, with the aim to determine differences between simplified and complex texts. They also work with a German-language corpus of 1888 texts (Klaper et al., 2013) and find that the feature lexical complexity, in particular, is relevant specifically for German texts. Battisti

* Equal contribution

et al. (2020) build on the same corpus and release a newer version with 6217 documents. Hewett and Stede (2021) create a corpus of 2655 texts from online lexica at three different levels (adults, children, children who are beginner readers) and use knowledge graph based features to estimate conceptual complexity. In a pairwise classification task they achieve an accuracy score of 91%.

3 Dataset

The dataset for the challenge consists of sentences that have been taken from 41 Wikipedia articles from different article genres. Groups of German learners, with language levels between A2 and B2, rated the sentences according to complexity, understandability and lexical difficulty on a scale from 1 to 7. For each aspect, the arithmetic mean (or Mean Opinion Score; MOS) was calculated and the task was to predict the MOS complexity score of the sentences. More information on the dataset can be found in (Naderi et al., 2019).

The training dataset consists of 1000 sentences, the validation set (for development phase) of 100 and the test set (for the evaluation phase) of 210 sentences. Figure 1 shows a histogram of the target variable (MOS) in the training set (mean=3.02, stdev=1.18). Some examples from the training set can be seen in Table 1.

It is also worth mentioning that 'complexity' can be subjective. For example as can be seen in Table 1, the second sentence '*Das Meerwasser ist leicht basisch*' has a score of 1; whilst the sentence is clearly short and has a very simple structure, arguably the words alkaline (*basisch*) and even seawater (*Meerwasser*) are not usually part of a language learner's vocabulary. The sentence structure may not be 'complex' but the lexical items do seem more advanced. These kinds of scores may be due to the fact that participants were also asked to rate the understandability of a sentence, a score which was not used in this shared task. The subjective nature of complexity is a limitation of the dataset which the shared task organisers try to compensate for by using a mapped root mean squared error as a metric, more information can be found in the task overview paper (Mohtaj et al., 2022).

4 Approaches

In this section we outline our different approaches. As a baseline, we take the simple approach of predicting the mean MOS value (3.02) for all samples. Using this baseline, the root mean squared error (RMSE) is 1.18.[1]

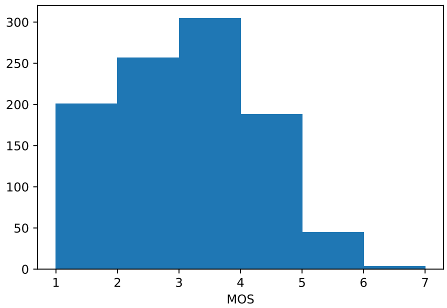

Figure 1: Histogram of mean opinion scores.

4.1 Additional Augmented Data

As the training dataset is not particularly large (1000 sentences), one of our approaches was to create additional training data. We used the extended lexica corpus from Hewett and Stede (2021) which consists of 86613 sentences at three different levels. We created artificial scores on a scale of 1 to 7 using a simple method. We took the original labels (1-3) and scaled them using the feature of sentence length, which we found to be a strong predictor for complexity. For example, a sentence with the original label of 1, with one of the longest sentence lengths for this class would have a transformed score of around 2.3, whereas a short sentence originally labelled with 1 would have a transformed score closer to 1.

We used this additional data together with the training data in several different models and the results were consistently worse than the basic baseline using only the training data. This is most likely due to the noise that is introduced when producing these artificial labels, and the fine-grained nature of the labels. Another reason could also be the different target groups; the shared task data has been labelled by non-native speakers whereas the lexica corpus has children as its target group. Children and non-native speakers are two different target groups of simplified language with different needs.

4.2 Neural Approach

The neural method we use is to fine-tune a pretrained transformer model for our given task. As

[1] Aside from the final results in Table 2, all reported scores refer to 20% of the training set and to the non-mapped RMSE.

Original Sentence	Literal Translation	MOS Complexity
Die Folgen dieser Versauerung betreffen zunächst kalkskelettbildende Lebewesen, deren Fähigkeit, sich Schutzhüllen bzw Innenskelette zu bilden, bei sinkendem pH-Wert nachlässt.	The consequences of this acidification have an effect on calcium-skeleton-forming organisms, whose ability to form protective shells or internal skeletons diminishes with decreasing pH.	5.25
Das Meerwasser ist leicht basisch.	Seawater is slightly alkaline.	1

Table 1: Example sentences from the dataset.

our base model, we chose XLM-R (also known as XLM-RoBERTa) by Conneau et al. (2019).

XLM-R is a self-supervised cross-lingual transformer model – trained on 2.5TB of filtered CommonCrawl data containing 100 languages – using a masked language modeling objective. It is mostly intended to be fine-tuned on downstream tasks (HuggingFace, 2022), and offers state-of-the-art performance for many language tasks. Specifically it outperforms multilingual BERT on a variety of metrics (Conneau et al., 2019).

The fact that XLM-R has great performance out of the box and is multilingual, make it a suitable choice for the challenge. We downloaded the pretrained model using the Hugging Face Python library.[2] We changed the model head to a (single) regressor layer plus a dropout, inspired by Kozodoi (2022). (As is typical, the weights for the new layers are randomly assigned, while the rest of the model is initialized to the pretrained weights.) We used a custom trainer to set RMSE as the loss function, and did not freeze any of the layers for higher accuracy. For preprocessing, was used the XLM-R Tokenizer with padding and truncation, which is how this model expects the data.

During the earlier phases of the Text Complexity DE challenge, we used a simple 80:20 data split for training and validation; and observed that our modified XLM-R model performed quite well after 10 training epochs with the default AdamW optimizer. For the final stage of the challenge, we adopted k-fold validation (with k=5) to ensure that all the available data was used during training. Thus we ended up with five models (with RMSEs between 0.55 and 0.70). For the actual predictions on the test dataset, we averaged the prediction of these five models.

4.3 Feature-based Approach

A further approach was to use the 43 'single features' which Stodden and Kallmeyer (2020) applied in their cross-lingual study on text complexity (see Section 2). These features are calculated using sentences as input; we therefore did not perform any additional pre-processing. We applied feature ranking using the recursive feature elimination implementation from scikit-learn (Pedregosa et al., 2011) and used the top 34 features; the full list of features can be found in Appendix A. The most important features were number of words per sentence, number of syllables per sentence and number of characters per word. We used these with a linear regression model, using the default parameters from scikit-learn. When applied to the training data in a 80/20 split, the RMSE was 0.7. We then retrained the model in the whole training set before using the official test data set as input (the results of which can be seen in Table 2). Approaches using sentence embeddings or lexical complexity values derived from our additional data did not beat our simple baseline on the training set and so were therefore not pursued any further.

5 Ensemble Results & Discussion

Our final approach is to combine our feature-based and neural approach by averaging the outputs of these two models.[3]

While both our transformer and feature-based models perform better than the baseline RMSE, among them, the transformer model does generally better on different data splits. Thus, it might seem paradoxical that our final model is a weighted average (ensemble) of the two. Using an ensemble method is, however, a theoretically sound practice, and quite common in machine learning competitions.

In the words of Page (2018), "To rely on a single model is hubris. It invites disaster. [...] Wisdom can be achieved by averaging models." Simply explained, due to both over-fitting and under-fitting, any one model will predict some samples (especially among the unobserved) quite wrongly. As long as the individual models in the collection do

[2] We chose the base model, not large, so that the training could be done efficiently on our laptop GPU.

[3] Our implementation can be found at https://github.com/hadiasghari/konvens22-shared-task

not share a common bias, then any diverse collection of the models will be more accurate than the average member—an implication of the so called diversity prediction theorem (Page, 2018).

To illustrate the point, we can compare the predictions from both models on the test dataset (Figure 2). The Pearson correlation coefficient between the two is 0.85. On average, the predictions are close, with the transformer model predicting slightly lower scores. In about ten percent of the samples, the difference between the predictions is bigger than 1, and crucially, in both directions.[4]

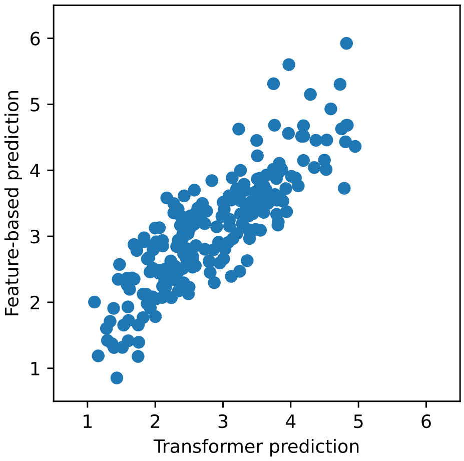

Figure 2: Histogram of mean opinion scores.

Closer to home and in the readability assessment (RA) literature, Lee et al. (2021) also propose using ensemble methods. In particular "[when] a transformer shows weak performance on small datasets, there must be some additional measures done to supply the final model (e.g. ensemble) with more linguistic information", adding that such studies are rare in RA.

The final results on the test dataset are presented in Table 2. We hypothesized that since our transformer model does slightly better than our feature based model, a weighted average favoring the former might yield better accuracy, which turned out to be the case.[5]

RMSE	Model Description
0.541	Linear regression (feature based)
0.484	Ensemble 70:30 (lr:xlmr)
0.479	XLM-R (without k-fold)
0.458	XLM-R (with k-fold)
0.457	Ensemble 50:50
0.450	Ensemble 40:60
0.446	**Ensemble 30:70** (lr:xlmr)

Table 2: Mapped RMSE results for different models (more information on the mapping can be found in the shared task overview paper (Mohtaj et al., 2022))

6 Conclusion

In this paper we explained our contribution to the shared task Text Complexity DE Challenge 2022. We experimented with both neural and feature-based approaches. Our best-performing model is a weighted average of a fine-tuned XLM-R transformer model and a classic feature-based model with linear regression. The ensemble achieves a mapped root square mean error of 0.446 on the test data which is better than either of the models alone.

Acknowledgements

Thank you for the organisers of the shared task for providing the data and relevant information.

References

Alessia Battisti, Dominik Pfütze, Andreas Säuberli, Marek Kostrzewa, and Sarah Ebling. 2020. A Corpus for Automatic Readability Assessment and Text Simplification of German. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3302–3311, Marseille, France. European Language Resources Association. ArXiv: 1909.09067.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability Classification for German using Lexical, Syntactic, and Morphological Features. In *Proceedings of COLING 2012*, pages 1063–1080, Mumbai, India. The COLING 2012 Organizing Committee.

Freya Hewett and Manfred Stede. 2021. Automatically evaluating the conceptual complexity of German texts. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*,

[4] Without the MOS scores for the test dataset, we can only speculate about this discrepancy between the two model predictions. See Appendix B for a few examples.

[5] In future work, the averaging weights could themselves be learnt from the data, and obviously, more models be added to the ensemble.

pages 228–234, Düsseldorf, Germany. KONVENS 2021 Organizers.

HuggingFace. Xlm-roberta (base-sized model) model card [online]. 2022.

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

Nikita Kozodoi. Estimating text readability with transformers [online]. 2022.

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.

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

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

Scott E. Page. 2018. *The Model Thinker: What You Need to Know to Make Data Work for You*, 1st edition edition. Basic Books, New York.

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.

Emily Pitler and Ani Nenkova. 2008. Revisiting Readability: A Unified Framework for Predicting Text Quality. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 186–195, Honolulu, Hawaii. Association for Computational Linguistics.

Sanja Stajner, Simone Paolo Ponzetto, and Heiner Stuckenschmidt. 2017. Automatic Assessment of Absolute Sentence Complexity. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 4096–4102, Melbourne, Australia. International Joint Conferences on Artificial Intelligence Organization.

Regina Stodden and Laura Kallmeyer. 2020. A multilingual and cross-domain analysis of features for text simplification. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 77–84. European Language Resources Association.

Appendix

A Features Used

The features we used in our feature-based model (discussed in Section 4.3) include[6]:
get type token ratio, get ratio of function words, get ratio of coordinating clauses, get ratio of subordinate clauses, get ratio prepositional phrases, get ratio relative phrases, get ratio clauses, get ratio named entities, check if head is noun, check if one child of root is subject, check passive voice, is non projective, get ratio of nouns, get ratio of verbs, get ratio of adjectives, get ratio of adpositions, get ratio of adverbs, get ratio of auxiliary verbs, get ratio of conjunctions, get ratio of determiners, get ratio of numerals, get ratio of particles, get ratio of pronouns, get ratio of punctuation, count words, count sentences, count syllables in sentence, count words per sentence, count syllables per sentence, count characters per word, count syllables per word, max pos in freq table, average pos in freq table, sentence fkgl.

B Discrepancy between Model Predictions

Without the MOS scores for the test dataset, we can only speculate about this discrepancy between the two model predictions. After manually inspecting some cases, we found that when the prediction of the feature-based model was higher (i.e. more complex) than the transformer model, these were long sentences which in fact were often just lists. When the prediction of the transformer model was higher, these were often shorter sentences with uncommon words (often compounds). See Table 3 for some examples.

[6] From the implementation from Stodden and Kallmeyer (2020): https://github.com/rstodden/text-simplification-evaluation

ID	Sentence	Translation	XLMR	LR
2115	"Die danach häufigsten Wohnungstypen waren Wohnungen in kleinen Apartmentkomplexen (2–9 Einheiten, 12,8 % der Bevölkerung), Wohnungen in mittleren Apartmentkomplexen (10–49 Einheiten, 7,9 %), Einfamilienreihenhäuser (5,9 %), Mobilheime (5,7 %), Wohnungen in großen Apartmentkomplexen (50+ Einheiten, 5,0 %) und Boote, Wohnmobile und Ähnliches (0,1 %)."	The next most common housing types were flats in small apartment complexes (2-9 units, 12.8 % of the population), flats in medium apartment complexes (10-49 units, 7.9 %), single-family terraced houses (5.9 %), mobile homes (5.7 %), flats in large apartment complexes (50+ units, 5.0 %), and boats, mobile homes, and the like (0.1 %).	3.233	4.624
2053	"Daneben gibt es auch konfessionelle (VkdL im CGB) und weitere Verbände (Waldorflehrkräfte, Lehrkräfte der Montessori-Schulen)."	There are also confessional (VkdL (Association of Catholic German Teachers) in the CGB (Christian Trade Union Federation of Germany)) and other associations (Waldorf teachers, Montessori school teachers).	3.123	2.393

Table 3: Example predictions on the test set with large discrepancy between the transformer (XLMR) and the feature based linear regression (LR) models.

TUM Social Computing at GermEval 2022: Towards the Significance of Text Statistics and Neural Embeddings in Text Complexity Prediction

Miriam Anschütz
Technical University of Munich
Department of Informatics
Germany
miriam.anschuetz@tum.de

Georg Groh
Technical University of Munich
Department of Informatics
Germany
grohg@in.tum.de

Abstract

In this paper, we describe our submission to the GermEval 2022 Shared Task on Text Complexity Assessment of German Text. It addresses the problem of predicting the complexity of German sentences on a continuous scale. While many related works still rely on handcrafted statistical features, neural networks have emerged as state-of-the-art in other natural language processing tasks. Therefore, we investigate how both can complement each other and which features are most relevant for text complexity prediction in German. We propose a fine-tuned German DistilBERT model enriched with statistical text features that achieved fourth place in the shared task with a RMSE of 0.481 on the competition's test data.

1 Introduction

Text readability describes how easy a given text is understood by a specific reader (Hancke et al., 2012). Factors that influence the readability are, for example, the number of technical terms in the text or the length and convolution of the sentences. Assessing a text's readability can be used to select the proper texts for a specific user group or provide authors feedback about their texts. Moreover, it can be integrated into an automatic text simplification system. On the one hand, it helps to decide whether and, if so, how much a text should be simplified. On the other hand, readability assessment is a measure to evaluate a simplification system by checking if the output has a higher readability (Garbacea et al., 2021; Martinc et al., 2021). Text complexity is inversely related to text readability; thus, in this work, the terms text complexity prediction and readability assessment are used interchangeably.

This paper is a contribution to the GermEval 2022 Shared Task on Text Complexity Assessment of German Text that aims to predict the complexity of a German text on a continuous scale (Mohtaj et al., 2022). We propose a model based on fine-tuned German DistilBERT (Sanh et al., 2019) combined with traditional readability formulas and statistical text features. This model achieved fourth place in the competition. Moreover, we used SHAP (Lundberg and Lee, 2017) to explain our model's predictions and discuss which features contribute to higher complexity. By knowing the feature relevance, authors and machine learning engineers can pay attention to them when generating new texts. Our code is released on Github for further research and development.[1]

This paper is structured as follows: Section 2 gives an overview of existing readability formulas and prediction models. In section 3, we present the organization of the shared task and introduce its dataset. Then, section 4 walks through our proposed approaches and entails their performance. Finally, in section 5, we apply explainability methods to discuss text features relevant to complexity prediction.

2 Related work

We investigated two approaches for readability assessment, traditional readability formulas, and deep learning. Therefore, this section gives an overview of existing formulas and models. Moreover, we analyze which text features yielded promising prediction results in previous work.

2.1 Traditional complexity measures

Multiple formulas exist to calculate the readability of a text based on statistical values such as word counts or average word length. Flesch (1948) proposed the Flesh reading ease (FRE) score that calculates a value between $0 - 100$, where a higher value indicates a lower complexity. Similarly, the readability index (LIX) (Björnsson, 1983) returns a readability estimate ranging from 20 to 60. However, with this score, an easier text gets a lower

[1] https://github.com/MiriUll/text_complexity

value. As German words tend to be longer than English words on average, Amstad (1978) adapted the FRE measure to the German language by adapting the weight of the average word length measure. Kincaid et al. (1975) used the FRE score as a basis for a new measure, the Flesch-Kincaid-Grade-Level (FKGL). In contrast to the previous scores, this returns the U.S. school grade in which the text can be understood. Other complexity scores returning the number of years in education needed to grasp the content of a text are SMOG (Laughlin, 1969) and Gunning fog index (Gunning et al., 1952). The Wiener Sachtext formulas are four slightly varying formulas returning the required grade adapted to the German school system and specificities of the German language (Bamberger and Vanacek, 1984).

These formulas are based on an analysis of textual features. In the literature, different text properties are distinguished (Santucci et al., 2020; vor der Brück et al., 2008; Hancke et al., 2012): Statistical features analyze the number of sentences or the number of words in a sentence, while syntactic features investigate the sentence structure, e.g., the depth of the dependency tree. Other categories are lexical features, such as the number of unique words, or semantic features, i.e., the length of causal chains. As indicated by Solnyshkina et al. (2017), using the plain text properties as features can outperform the complexity estimation of readability formulas.

2.2 Learning complexity prediction models

Syntactic, semantic, or lexical text features have been exploited for readability prediction in different languages such as Italian (Santucci et al., 2020) or English (Štajner and Hulpus, 2020). Other approaches use neural language models like BERT for their predictions (Martinc et al., 2021). For the German language, Weiß and Meurers (2018) proposed a binary prediction model based on linguistic features, such as lexical or morphological complexity, and psycholinguistic features, i.e., cognitive complexity and language use. Their work was based on the binary prediction model by Hancke et al. (2012). In a very recent work (Anonymous, 2021), the neural approaches by Martinc et al. (2021) were transferred to German, yielding promising results in a language-level prediction task. These approaches focus on a classification task, while vor der Brück et al. (2008) worked on a seven-point Likert scale, similar to the Shared Task data. They used syntactic and semantic features together with a nearest neighbor model for their predictions.

2.3 Feature relevance analysis

To understand why a model deems a sentence complex, but also to use the complexity scores for further tasks such as text simplification (Garbacea et al., 2021), the features that contributed to the predictions are of interest. Santucci et al. (2020) used the Gini measure and permutation importance to inspect which text property was important for their predictions. They reported that the most relevant features were the syntactic and morphosyntactic ones. Similarly, Hancke et al. (2012) discovered the essential features for their classification were the average word length or the number of complex nominals in the sentences.

3 Shared task and Dataset

This paper explains our submission to the GermEval 2022 Shared Task on Text Complexity Assessment of German Text (Mohtaj et al., 2022). The shared task was split into two different phases, a development and a final phase. During development, participants were provided a labeled training and an unlabeled validation dataset. Predictions on this validation data could be uploaded to the competition page with immediate evaluation feedback. In contrast, during the final phase, the results on the final test dataset were only published at the end of the competition. The two evaluation datasets, the validation and the final test data, consist of 100 sentences each. The training dataset for this shared task originates in work by Naderi et al. (2019). It contains 1000 sentences from the German Wikipedia together with a complexity score ranging from 1 to 7. Naderi et al. (2019) used crowdsourcing to let non-native speakers of a B level annotate the respective sentences by their perceived readability and averaged the scores among the participants. The mean complexity value is 3.016 with a standard deviation of 1.181. There are 76 sentences with an observed complexity of 1.0, but only two samples with a complexity higher than six, making the dataset unbalanced towards the easier sentences. To counteract this imbalance, we replicated sentences with a complexity higher than 5.5 multiple times, yielding a dataset with 1054 samples.

The rooted mean squared error (RMSE) between predicted and correct complexity scores was used to evaluate a model's performance. In addition, a third-order polynomial function was fitted between the predicted and correct scores to counteract the bias by subjective annotation of text complexity. Then, the predicted scores were projected using this function, and the error was calculated on the mapped predictions as well (Mohtaj et al., 2022).

4 Approaches

In this section, we explain the three approaches we explored to predict the complexity score of a sentence. We did not apply any preprocessing to the data, i.e., fed the sentences into the model's tokenizer directly.

4.1 Learning from text statistics

We analyzed different textual features and readability scores calculated based on them. Table 1 shows which statistics were calculated. On the one hand, statistics on a sentence level were investigated, such as the average sentence length or the maximal depth of the dependency tree. We assumed that a more complex sentence holds subclauses or multi-word expressions that show in a high dependency tree depth. For our data, the average sentence length is similar to the number of words in a sentence, as our data samples contain only one sentence. On the other hand, we examined the characteristics of the words in a sentence, e.g., the average number of syllables among all words. Moreover, the percentage of words consisting of only one syllable was calculated. These are very short and easy-to-understand words, i.e., a high percentage can indicate a simple sentence.

Feature	Description
asl	Average sentence length
mtd	Maximal dependency tree depth
pw6	Percentage of words with at least six letters
asc	Average number of syllables
ps1	Percentage of words with only one syllable
ps3	Percentage of words with at least three syllables

Table 1: Statistical features calculated from sentences.

These statistics are part of different readability formulas. Equations 1 to 6 show the formulas for the scores used in this work. We propose calculating the Flesh reading easy (FRE) by Amstad (Amstad, 1978), the four Wiener Sachtext formulas (Bamberger and Vanacek, 1984) and the SMOG score (Laughlin, 1969). The FRE formula uses the average sentence length and the average number of syllables among all words and returns a value between 0 and 100, where a higher score indicates better readability. The Wiener Sachtext formulas are a collection of four formulas that slightly vary the statistics they use and their weights. The formulas calculate for which school grade between four and 15 the text is suited. Similarly, the SMOG score returns how many years of education the reader needs to understand the text. Thus, a lower value is desirable for the Wiener Sachtext formulas and the SMOG score. In contrast to the other formulas, the SMOG score only uses the number of words with at least three syllables (ns3) as a statistical measure.

$$\begin{aligned}
\textbf{fre\_amstad} &= 180 - \text{asl} - (58.5 \cdot \text{asc}) && (1)\\
\textbf{wstf1} &= 0.1935 \cdot \text{ps3} + 0.1672 \cdot \text{asl} && (2)\\
&\quad + 0.1297 \cdot \text{pw6} - 0.875 &&\\
&\quad - 0.0327 \cdot \text{ps1} &&\\
\textbf{wstf2} &= 0.2007 \cdot \text{ps3} + 0.1682 \cdot \text{asl} && (3)\\
&\quad + 0.1373 \cdot \text{pw6} - 2.779 &&\\
\textbf{wstf3} &= 0.2963 \cdot \text{ps3} + 0.1905 \cdot \text{asl} && (4)\\
&\quad - 1.1144 &&\\
\textbf{wstf4} &= 0.2744 \cdot \text{ps3} + 0.2656 \cdot \text{asl} && (5)\\
&\quad - 1.6930 &&\\
\textbf{SMOG} &= 1.0430 \cdot \sqrt{\text{ns3}} + 3.1291 && (6)
\end{aligned}$$

We computed the statistics in Table 1 and scores in Equations 1 to 6 for all samples in our data. Then, we fitted a support vector regression based on these statistical vectors as a prediction baseline. For this, we used the implementation by sklearn and its default hyperparameters parameters.[2] The model achieved a RMSE of 0.657 and mapped RMSE of 0.647 on the training data.

4.2 Fine-tuning a transformer model

To investigate the complexity prediction quality of neural networks, we fine-tuned a German DistilBERT model. We utilized Huggingface (Wolf et al., 2020) to load and fine-tune the distilbert-base-german-cased (von Platen, 2020) model. We

[2] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html

trained the model on the shared tasks' training data with the default setup of Huggingface's trainer API[3] for two epochs. Table 2 shows the promising results achieved by this model on the training, validation, and final test data. The model outperformed the statistics-only SVR baseline model by far.

Dataset	RMSE	RMSE_mapped
training	0.402	0.399
validation	0.405	0.404
final test	0.481	0.460

Table 2: Complexity prediction results by fine-tuned DistilBERT model.

4.3 Combining DistilBERT embedding with textual features

The pure text statistics model and the fine-tuned DistilBERT model yielded promising results. To take advantage of both their handcrafted features and deep textual understanding, we combined both models. We used the last hidden state of the DistilBERT model as an embedding of size 768. Then, we concatenated the embedding with the vector of statistical measures and readability scores. Finally, we trained a support vector regression model on these representations with the same setup as the statistical SVR. Table 3 highlights the performance on the three different datasets. With this model, we achieved fourth place in both the competition's development and final evaluation phase.

Dataset	RMSE	RMSE_mapped
training	0.404	0.403
validation	0.395	0.390
final test	0.466	0.449

Table 3: Complexity prediction results by SVR with DistilBERT embedding and statistical features.

5 Explaining the predictions

To evaluate which of the suggested statistics and formulas help to predict the complexity of German texts, we calculated the SHapley Additive exPlanations (SHAP) values (Lundberg and Lee, 2017) for each of our models. SHAP measures each feature's contribution by masking their different combinations and rerunning the predictions with these masks. Features for which the masked predictions deviate strongly from the initial prediction have a substantial impact and are, thus, the most relevant ones. The SHAP values are calculated per sample and averaged among them. Figure 1 shows the mean SHAP values for each feature in the statistical SVR model (Section 4.1). The most relevant statistic is the average sentence length, i.e., the longer a sentence is, the more likely it is complex. The FRE score uses this statistic; thus, it is reasonable that it has high importance. Even though the Wiener Sachtext formulas also include this statistic, their contribution to the predicted score is smaller. They incorporate more advanced measures like the percentage of words with more than three syllables. As indicated by the small SHAP values, these additional statistics are not helping our complexity prediction model. The third most relevant feature is the maximum tree depth, indicating how convoluted a sentence is.

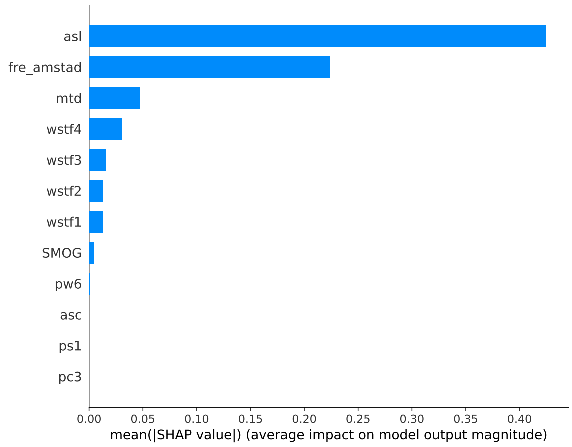

Figure 1: SHAP values for statistical text features in our support vector regression model, sorted in descending order.

For a neural network, it is unknown what functionality a specific neuron models. Therefore, a feature-relevance analysis is not beneficial for interpreting a neural network. Instead, we selected the example sentence "Dieser Vorgang wird Gletscherschwund oder Gletscherschmelze genannt." (*"This process is called glacier recession or glacier melt."*) and investigated which words have an impact on the prediction. The correct complexity for this sentence is 2.266667, and our model (Section 4.2) predicts a complexity of 2.373029. Figure 2

[3] https://huggingface.co/docs/transformers/main_classes/trainer#transformers.TrainingArguments

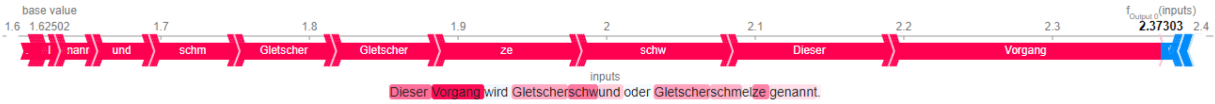

Figure 2: DistilBERT prediction on an example sentence (English translation: "This process is called glacier recession or glacier melt."): contribution of each word and word chunk to the prediction result.

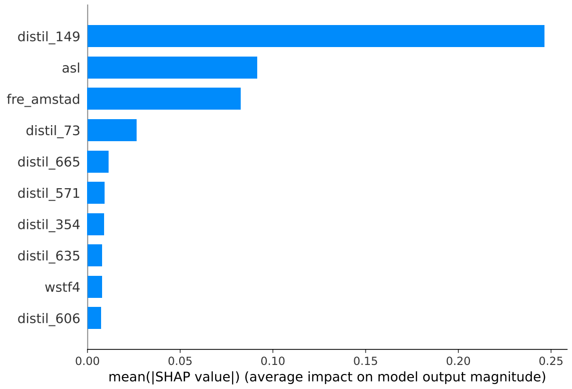

Figure 3: Feature relevance analysis for combined model: the ten features with highest SHAP values. "distil_i" indicates the ith index in the DistilBERT embedding.

shows which words and parts of words increase or decrease the predicted score compared to a base value. Words like "wird" (*"is"*) and "oder" (*"or"*) have a negative contribution, i.e., they indicate an easier sentence. Contrary, the word "Vorgang" (*"process"*) has the highest positive impact. The word itself is not very difficult, but it is often used to describe complex procedures and, thus, can be seen as a signal for a complex sentence. In German, compound nouns such as "Gletscherschwund" (*"glacier recession"*) are very common. However, the DistilBERT tokenizer splits them into multiple tokens. Therefore, different parts of these compound words have different contributions to the prediction, making it harder to identify their overall contribution.

Finally, Figure 3 depicts another feature relevance analysis, but for the SVR model that combined our neural embedding with statistical text features (Section 4.3). The scores were calculated on a subset of the data, and we only highlight the values for the ten highest ranking features. The strongest impact on the prediction comes from the embedding value at index 149, but text statistics like the average sentence length and Amstad's FRE score are also relevant. This implies that both learned neural features and traditional text statistics impact text complexity prediction. Moreover, they complement each other to yield the most accurate predictions. Therefore, we have shown that neural models have not yet outperformed handcrafted features regarding German text complexity prediction.

6 Discussion

Readability is a subjective measure that depends on the reader's background knowledge and reading ability (Crossley et al., 2017). Our work is based on the shared task's dataset labeled with a crowdsourcing approach among non-native speakers. Therefore, the findings in this paper should be tested for transferability to other datasets and groups of readers. In addition, the dataset is unbalanced with an overrepresentation of simple sentences and contains some noise. For example, the sentence "Martin Luther King Jr (* 15 Januar 1929 in Atlanta als Michael King Jr; † 4 April 1968 in Memphis) war ein US-amerikanischer Baptistenpastor und Bürgerrechtler." (*"Martin Luther King Jr (born January 15, 1929 in Atlanta as Michael King Jr; † April 4, 1968 in Memphis) was a U.S. Baptist pastor and civil rights activist."*) has a complexity of 1.0, indicating it was a very easy sentence. This shows that some samples have lower complexity than they would have when relabeling the dataset.

7 Conclusion

In this paper, we have demonstrated three approaches for text complexity prediction in German, one model that relies on handcrafted statistical features only, one fine-tuned transformer network, and a combination of both. In addition, we found that the feature most indicative of a complex sentence is the sentence length and that the FRE formula by Amstad (1978) gives a good indication of text complexity. Modern transformer architectures with deep textual understanding can build accurate complexity prediction pipelines. However, they can still be improved with handcrafted statistical features, showing that they have not yet superseded traditional approaches. In future work, these findings will be extended to a paragraph and full-text level instead of a sentence-wise prediction.

References

Toni Amstad. 1978. *Wie verständlich sind unsere Zeitungen?* Ph.D. thesis, Universität Zürich.

Anonymous. 2021. Language level classification on german texts using a neural approach. ACL ARR 2021 November Blind Submission.

Richard Bamberger and Erich Vanacek. 1984. *Lesen-Verstehen-Lernen-Schreiben. Die Schwierigkeitestufen von Texten in deutscher Sprache*. Diesterweg.

C. H. Björnsson. 1983. Readability of newspapers in 11 languages. *Reading Research Quarterly*, 18(4):480–497.

Scott A. Crossley, Stephen Skalicky, Mihai Dascalu, Danielle S. McNamara, and Kristopher Kyle. 2017. Predicting text comprehension, processing, and familiarity in adult readers: New approaches to readability formulas. *Discourse Processes*, 54(5-6):340–359.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Cristina Garbacea, Mengtian Guo, Samuel Carton, and Qiaozhu Mei. 2021. Explainable prediction of text complexity: The missing preliminaries for text simplification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1086–1097, Online. Association for Computational Linguistics.

Robert Gunning et al. 1952. Technique of clear writing.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability classification for German using lexical, syntactic, and morphological features. In *Proceedings of COLING 2012*, pages 1063–1080, Mumbai, India. The COLING 2012 Organizing Committee.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

G. Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of Reading*, 12(8):639–646.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4765–4774. Curran Associates, Inc.

Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. Supervised and Unsupervised Neural Approaches to Text Readability. *Computational Linguistics*, 47(1):141–179.

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

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

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Valentino Santucci, Filippo Santarelli, Luciana Forti, and Stefania Spina. 2020. Automatic classification of text complexity. *Applied Sciences*, 10(20).

Marina Solnyshkina, Radif Zamaletdinov, Ludmila Gorodetskaya, and Azat Gabitov. 2017. Evaluating text complexity and flesch-kincaid grade level. *Journal of Social Studies Education Research*, 8(3):238 – 248.

Sanja Štajner and Ioana Hulpus. 2020. When shallow is good enough: Automatic assessment of conceptual text complexity using shallow semantic features. In *LREC 2020 Marseille : Twelfth International Conference on Language Resources and Evaluation : May 11-16, 2020, Palais du Pharo, Marseille, France : conference proceedings*, pages 1414–1422, Paris. European Language Resources Association, ELRA-ELDA.

Patrick von Platen. 2020. distilbert-base-german-cased.

Tim vor der Brück, Sven Hartrumpf, and Hermann Helbig. 2008. A readability checker with supervised learning using deep indicators. *Informatica (Slovenia)*, 32(4):429–435.

Zarah Weiß and Detmar Meurers. 2018. Modeling the readability of German targeting adults and children: An empirically broad analysis and its cross-corpus validation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 303–317, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

HHUplexity at Text Complexity DE Challenge 2022

David Arps, Jan Kels, Florian Krämer, Yunus Renz, Regina Stodden, Wiebke Petersen
Heinrich Heine Universität
Düsseldorf, Germany
first.last@hhu.de
all authors contributed equally

Abstract

In this paper, we describe our submission to the 'Text Complexity DE Challenge 2022' shared task on predicting the complexity of German sentences. We compare performance of different feature-based regression architectures and transformer language models. Our best candidate is a fine-tuned German Distilbert model that ignores linguistic features of the sentences. Our model ranks 7th place in the shared task.[1]

1 Introduction

Texts are a basic form of human information exchange. Too high of a text complexity, however, can result in text comprehension failures (Bormuth, 1966) and therefore miscommunication. Text complexity and readability assessment are a long known problem and several computational approaches and metrics have been proposed (Dascalu, 2012; Hancke et al., 2012; Collins-Thompson, 2014), relying on different linguistic features and primarily aiming at English.

Among other application objectives, an adequate, quantificational metric for text complexity can be of high benefit to the educational domain (as a means of providing textual material according to student levels), writing support systems (as feedback) or for other natural language processing tasks like estimating the complexity of the output of text simplification systems or chatbots. Most related tasks focus either on the prediction of complex words (Paetzold and Specia, 2016; Shardlow et al., 2021) or the assessment of readability levels (Collins-Thompson, 2014). However, the goal of the 'Text Complexity DE 2022' shared task is the prediction of an empirically determined complexity score called 'Mean Opinion Score' (MOS) for German sentences. Overall, our best model is ranked on the 7th place out of 10. In the following, we present the approach and results of our team "HHUplexity" in more detail.

1.1 Shared Task Data

The training data (Naderi et al., 2019) for the shared task contains 1000 sentences from 25 Wikipedia texts. The development data and test data contain 100 and 210 sentences, respectively, for which the document distribution is not known. The sentences were rated by German language learners (between CEFR level A and B) on a 7 point Likert-scale regarding their complexity (1 – very easy to 7 – very complex). The arithmetic mean of these ratings is the target score – MOS score – of the shared task. 7.6% of the training samples are rated as very easy (score = 1), whereas 20.3% are rated as rather complex (score > 4) and 3.4% have a score higher than 5.

The root mean squared error (RMSE) after third order mapping as well as a more balanced RMSE score (RMSE$_{mapped}$) are used to evaluate the predicted MOS scores (Mohtaj et al., 2022).

2 Method

Our main approach is to combine hand-crafted features with text embeddings of language models. Therefore, we have calculated several features as described in subsection 2.1. To compare the effect of these features in combination with language models, we follow two baseline approaches: i) training different regression models with the features (see subsection 2.2), and ii) fine-tuning language models without features (see subsection 2.3). Afterwards, we combine the features with the language models in a multimodal model (see subsection 2.4).

2.1 Features

We calculate 349 features of seven main categories: features based on length, readability assessment features, features based on language proficiency, morphological features, syntactic features, morphosyn-

[1] Our code is available at https://github.com/Vipitis/HHUplexity

	Feature
Length-based	number of words [♠]
	number of types
	number of characters [♠]
	number of syllables [♠]
	avg. word length in characters
	max. word length in characters
	avg. word length in syllables
	number of sentences
Readability	Flesch Reading Ease Score [♥]
	Flesch-Kincaid Grade Level [♥]
	Dale-Chall Readability Score
	Linsear Write Formula
	Automated Readability Index
	difficult words
Morphological	ratio of negations & negated words
	ratio of compounded words & nouns
	number of nominalizations
	N-gram frequencies
	ratio of nouns in cases
Morphosyntactic	number of verbs, auxiliaries, nouns, pronouns
	ratio of coarse-grained POS-tags [◆] [♣]
	ratio of fine-grained POS-tags (STTS) [◆] [♣]
	noun-to-verb ratio
	number of stop words [◀]
	ratio of function words [◀]
	ratio of named entities
Syntactic	max. depth of the dependency parse tree [◀]
	max. & avg. distance between tokens in the parse tree
	max. & avg. distance between verbs and verb particles in the parse tree
	avg. length of NP & VP & PP [◆]
	± projective parse tree [▶]
	± head of the parse tree is a noun or verb [▶]
	± one child of the head of the parse tree is a subject [▶]
	± passive voice [◆]
	± subjunctive mood [◆]
	ratio of multi-word expressions [▶]
	number of clauses
	ratio of all tokens of coordinating & subordinating clauses [◆]
	ratio of tokens marking relative clauses [◆]
	ratio of tokens marking prepositional phrases [◆]
	ratio of tokens marking referential phrases [▶]
Lexical	ratio of words that are in the vocabulary lists for CEFR levels A1, A2, & B1
	type-token ratio [◀]
	avg. lemma frequency & rank (based on deCOW)
	lexical complexity based on ranks of German FastText embeddings [♥]
	max. and avg. rank in the German FastText embeddings [♥]
Other	perplexity score (based on GerPT2)
	label of target group and their softmax scores predicted by a fine-tuned model on this labeling task
	cosine similarity between original sentence and backtranslated sentences into German from English, Turkish, Hungarian, Chinese, and Georgian
	avg. imagebility and concreteness score [◀]

Table 1: Overview table of all features per category. The symbols stand for the papers in which the features were introduced: [♠] Scarton et al. (2018), [♥] Martin et al. (2018), [♣] Kauchak et al. (2014), [◆] Gasperin et al. (2009), [◀] Collins-Thompson (2014), [▶] Stodden and Kallmeyer (2020).

tactic features, and other features. An overview of all features is provided in Table 1. In general we find that 78% of the features have a significant Pearson correlation with the MOS target value (p-value > 0.05). Of those 66% have a weak correlation ($|r| < .4$), 21% a moderate correlation ($.4 \leq |r| < .6$) and 12% a strong correlation ($.6 \leq |r|$).

However, several features are absolute count features such as e.g. syllables or character count that depend on sentence length. If one transforms these features into proportional features the rate of features having a significant Pearson correlation with the MOS target value drops to 57%.

Features based on Length. As basic features to estimate the complexity of a sentence, we consider the length of the sentence (in words, syllables and characters) and the length of the words (in syllables and characters).

Readability Assessment Features. The length of words and sentences can also be jointly used to estimate text complexity within traditional readability formulas for texts, e.g., Flesch Reading Ease score or Flesch-Kincaid Grade Level.[2] The established readability metrics have been calculated for the original German sentences as well as for automatically translated English sentences (altogether 24 features). It turns out that the German scores correlate better than or equally well as the English scores with the exception of the Dale-Chall Readability Score. While Dale-Chall shows no significant correlation with the MOS-values for the German sentences it correlates with $r = 0.392$ for the English sentences.

It turns out that these quite simple formulas lead to the features with the strongest MOS-correlations. Only four significant features have a Pearson correlation r-value above 0.7 of which three are established readability scores: Linsear Write Formula with $r = 0.745$, difficult words with $r = 0.741$, Automated Readability Index (ARI) with $r = 0.706$, and number of words $r = 0.701$.

Lexical Features and Features based on Language Proficiency. Even if a word is short, it can be still unknown to a user and, therefore, difficult to understand. In our work, we include some

[2] We use several readability metrics of the textstat package (https://pypi.org/project/textstat/).

lexical and language proficiency-based features to estimate the complexity of a sentence based on the choice of words. Simple words are often frequent and complex words more infrequent, so word frequency might help to estimate the complexity of a sentence (Martin et al., 2018; Collins-Thompson, 2014). We follow two approaches, first, we obtain the frequency and rank per lemma based on the deCOW-corpus (Bildhauer and Schäfer, 2014) and build the average of them per sentence. Second, we measure the lexical complexity based on the word ranks in the German FastText Embeddings as well as obtaining the highest and average position of the tokens in the sentence.

Additionally, we select vocabulary lists per CEFR level A1, A2, B1 by the Goethe institute[3] and measure the ratio of words in the input sentence that can be found in the CEFR vocabulary lists. Vocabulary lists for other CEFR levels have not been available. The correlations with the empirical MOS-values indicate that the study participants judging the complexity are familiar with the vocabulary up to the B1 level. All three correlations (ratio of A1 / A2 / B1 vocabulary words) are negative and lie in the range $-0.35 \leq r \leq -0.4$. That is the higher the proportion of A1/A2/B1 vocabulary words, the less complex the participants judged the sentence.

Morphological Features. Besides the length and the choice of the words, a morphological analysis of words can be helpful to assess the complexity of the sentences. For example, some morphemes can drastically change the meaning of a word, e.g., negation prefixes ("irr-" or "un-"), nominalization suffixes ("-heit" or "-keit"), or one-token compound nouns ("Staubecken", "Dampfschiff"). Therefore, we calculate the number of nominalizations, negations based on a fix list of affixes, count the number of n-grams[4], as well as the ratio of compounded words[5]. Furthermore, we include the ratio of nouns per case, as the genitive is often difficult to understand. The non-ngram morphological features exhibit a significant but weak correlation with the MOS-values.

Syntactic Features. Besides an analysis of the words, an analysis of the structure of a sentence can give additional insights into its complexity because some syntactic structures take longer to process and comprehend (Gibson, 1998). To reflect syntactic complexity in our features, we measure the maximum depth of the dependency parse tree, maximum and average distances between words and number of clauses.[6] Based on Gasperin et al. (2009), we add the average length of noun, verb, and prepositional phrases, and whether the sentence is written in active or passive voice and indicative or subjunctive mood. Furthermore, we check some regularities in the parse tree based on Stodden and Kallmeyer (2020) (see Table 1). Based on the parse tree, we also count the ratio of multi-word expressions and ratio of all tokens of some clauses (see Table 1). Maximum tree depth has the strongest correlation ($r = 0.583$) with the MOS-values. Tree width features like average NP or VP length show a moderate positive correlation as well ($r \approx 0.4$). A negative correlation is found for the number of clauses normalized by sentence length ($r = -0.46$).

Morphosyntactic Features. Part-of-speech (POS) tags combine some morphological information with syntactic information, therefore we use the number and ratio of coarse-grained / fine-grained POS tags and noun-to-verb ratio to estimate the sentence complexity as similar as in Gasperin et al. (2009) and Kauchak et al. (2014). Following Collins-Thompson (2014), we also include the ratio of function words and stop words to all tokens as a feature. However, none of these features has a moderate or strong correlation with the MOS-score.

Psycholinguistic Features. In readability literature, psycholinguistic-based features are often named as relevant features (Collins-Thompson, 2014; Davoodi and Kosseim, 2016). In our work, we obtain the imageability and concreteness of each word per sentence based on the Concreteness and imageability lexicon MEGA.HR-Crossling (Ljubešić, 2018) and measure the average per sentence as another feature. Both of these features do not show a moderate or strong correlation which might be due to the absence of the words of the sentences in the chosen resources.

Perplexity Feature. We calculate the perplexity of the sentence with "GerPT-2"[7]. The higher the perplexity score, the harder to predict the seen sen-

[3] https://www.goethe.de/de/spr/kup/prf/prf.html

[4] We consider all uni-grams in the training data (char vocab), and top k n-grams for k=20, n=2, 3, 4, 5.

[5] The compounded words are obtained by https://github.com/repodiac/german_compound_splitter.

[6] The number of clauses is heuristically derived by the clause-splitting method described in Dönicke (2020).

[7] https://huggingface.co/benjamin/gerpt2

tence and the more unlikely is the input sentence for the model. Hence, we hypothesize, the higher the perplexity score, the more uncommon/complex is the sentence. For the training data the hypothesis can be confirmed but only by a weak correlation ($r = 0.214$).

Translation-based Features. The idea is to test whether translation difficulties indicate higher MOS-values. Therefore, with GoogleTranslator the sentences have been translated into English, Turkish, Hungarian, Chinese, and Georgian and backtranslated into German. These languages vary in their morphological and syntactic similarity and in their degree of genetic relationship to German. For the original and the backtranslated sentences contextualized embedding vectors have been determined with a transformer language model[8]. Finally, the cosine similarity for the sentence pairs has been calculated and added as a feature. It turns out that only Georgian leads to a non significant feature (p-value = 0.08), all others are significantly correlating with the MOS-values indeed only weakly. The highest correlation is found for Chinese Simplified with $r = 0.146$ and $p = 0.00$.

Text Level. We fine-tune a 3-class text level classifier on the Lexica corpus (Hewett and Stede, 2021), a dataset with German Wikipedia texts for three different target groups: younger children, children and adults. From this dataset we sample roughly 38k sentences (taken from roughly 1650 different texts), and fine-tune a German BERT model[9] to predict one of the three labels: child, youth, adult. The fine-tuned language model is applied to the shared task dataset and the softmax scores for the three labels, as well as the predicted labels are used as additional text level features. All four text level features have a rather high moderate correlation with the MOS values: softmax adult $r = 0.589$, softmax youth $r = -0.447$, softmax child $r = -0.519$, and predicted label $r = 0.565$. The correlations show that the study participants judge sentences as less complex if they have a higher probability of being labeled as ‘child’ or ‘youth’ and as more complex the higher the probability of the label ‘adult’ is. This indicates that the German language proficiency level of the participants is in between the youth and the adult level.

2.2 Predicting MOS from features

We have compared different methods to predict MOS based on the features from the previous section. To choose an appropriate model architecture and hyperparameters, we train and test models on a 5-fold crossvalidation split of the shared task training data for which MOS scores are available. We compare linear regression models with different regularization (Ridge, ElasticNet), and XGBoost (Chen and Guestrin, 2016). Because XGBoost achieved the best crossvalidation performance by a margin of $> .05$ RMSE compared to the other models, we only report results for this model. Using the same 5-fold crossvalidation split, the best hyperparameters are determined. The best model is an XGBoostRegressor[10] with `n_estimators`=2500, a learning rate of `eta`=.005, and a `max_depth` of 5. This model achieves a RMSE of .545 (RMSE mapped .502) on the final test data.

2.3 Fine-tuning

We have explored fine-tuning a language model directly on the regression task using Huggingface’s *AutoModelForSequenceClassification* for various models available on Huggingface including English, German and multilingual versions of BERT (Devlin et al., 2019) and DistilBERT (Sanh et al., 2019). To select the best pre-trained model, we have trained on 900 sentences of the training data and evaluated RMSE on the remaining 100. With a learning rate of $2 * 10^{-5}$ and 5 epochs. Trading smaller batch sizes for more steps led to better results where 10 did better than 30 or 50.

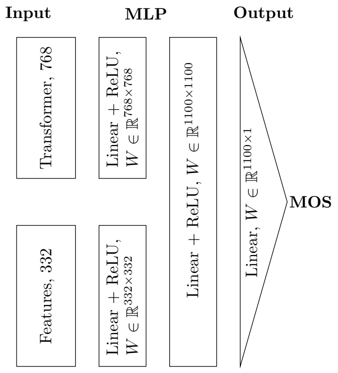
Input
MLP
Output
Transformer, 768
Features, 332
Linear + ReLU, $W \in \mathbb{R}^{768\times768}$
Linear + ReLU, $W \in \mathbb{R}^{332\times332}$
Linear + ReLU, $W \in \mathbb{R}^{1100\times1100}$
Linear, $W \in \mathbb{R}^{1100\times1}$
MOS

Figure 1: Architecture of the multimodal model that combines BERT embeddings with feature vectors

[8] https://huggingface.co/Sahajtomar/German-semantic

[9] https://huggingface.co/deepset/gbert-base

[10] https://docs.getml.com/1.1.0/api/getml.predictors.XGBoostRegressor.html

2.4 Multimodal model

To combine text embeddings and numerical features, we have written a custom version of Huggingface's *DistilbertForSequenceClassification* heavily inspired by Multimodal-Toolkit (Gu and Budhkar, 2021). BERT embeddings and text features are combined by a feedforward neural network, the architecture of which is displayed in Figure 1.

3 Results

	RMSE
XGBoost no ngrams	**.545**
XGBoost all feats	.639
ElasticNet no ngrams	.672
ElasticNet all feats	.659
Ridge no ngrams	.669
Ridge all feats	.713
bert-base-cased	.601
bert-base-german-cased	.552
bert-base-german-dbmdz-cased	.638
bert-base-multilingual-cased	.565
distilbert-base-cased	.600
distilbert-base-german-cased	**.486**
xlm-roberta-base	.511
distilbert-base-german-cased multimodal	**.622**

Table 2: Results for all models. Boldface results indicate performance on test data via a submission to the evaluation system. In all other cases, the performance is measured on a randomly selected held-out split of the training data. The first line separates regression models and fine-tuned language models, and the second line separates the multimodal model.

Results are presented in Table 2. For feature-based predictors, features and target MOS scores are transformed by removing the mean and scaling to unit variance. We find that gradient-boosting methods (XGBoost) work significantly better than linear models with ElasticNet or Ridge regularization. As shown in Table 2, the ablation of n-gram features clearly drops the RMSE score for XGBoost and Ridge regularization (> 0.04). XGBoost without n-gram features achieves a RMSE$_{mapped}$ score of .502 on the test data. For fine-tuned models, distilbert-base-german-cased was trained on 990 sentences with batch size 10 and 5 epochs. The submitted result reached .486 RMSE (.473 RMSE$_{mapped}$) on test data. When combining a transformer language model and features (subsection 2.4), we found that our implementation did not manage to improve results over the fine-tuning baseline. The best submission for this method reached .622 RMSE (.524 RMSE$_{mapped}$). A smaller feature set based on their importance might improve the results, similar as shown for the ablation of n-gram features with XGBoost and Ridge regularization (see Table 2).

3.1 Distribution of predictions

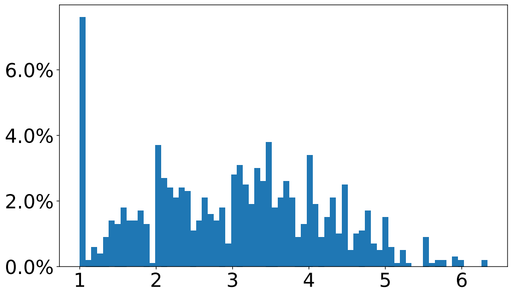

Figure 2: Distribution of MOS scores in the training data (70 bins).

On crossvalidated results, we find that many of our models do not predict the Gaussian distribution of MOS scores with an additional peak at the low end (Figure 2). All models correctly identify the mean scores of the general dataset, but generally tend to predict MOS scores of a lower standard deviation. Across feature-based models, the standard deviation of predicted scores on validation data is approximately 20% smaller than the standard deviation of the gold labels. We do not know the true labels of the validation and testing shared task data, but assume that this systematic error is also present in our submissions for these datasets.

4 Conclusion

In our contribution to the shared task, we have compared predictions based on linguistic features with an approach based on transfer learning, i.e., fine-tuning a language model. We find that even though linguistic features achieve relatively high correlation with the MOS scores, they are outperformed by a "simple" fine-tuned transformer language model.

References

Felix Bildhauer and Roland Schäfer. 2014. Decow14 lemma frequency list.

John R. Bormuth. 1966. Readability: A new approach. *Reading Research Quarterly*, 1(3):79–132.

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of* *the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA. ACM.

Kevyn Collins-Thompson. 2014. Computational assessment of text readability: A survey of current and future research. *ITL - International Journal of Applied Linguistics*, 165(2):97–135.

Mihai Dascalu. 2012. Analyzing discourse and text complexity for learning and collaborating: A cognitive approach based on natural language processing.

Elnaz Davoodi and Leila Kosseim. 2016. CLaC at SemEval-2016 task 11: Exploring linguistic and psycho-linguistic features for complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 982–985, San Diego, California. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tillmann Dönicke. 2020. Clause-level tense, mood, voice and modality tagging for German. In *Proceedings of the 19th International Workshop on Treebanks and Linguistic Theories*, pages 1–17, Düsseldorf, Germany. Association for Computational Linguistics.

Caroline Gasperin, Lucia Specia, Tiago F. Pereira, and Ra M. Aluisio. 2009. Learning when to simplify sentences for natural text simplification. In *In Proceedings of ENIA*, pages 809–818.

Edward Gibson. 1998. Linguistic complexity: locality of syntactic dependencies. *Cognition*, 68(1):1–76.

Ken Gu and Akshay Budhkar. 2021. A package for learning on tabular and text data with transformers. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*, pages 69–73, Mexico City, Mexico. Association for Computational Linguistics.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability classification for German using lexical, syntactic, and morphological features. In *Proceedings of COLING 2012*, pages 1063–1080, Mumbai, India. The COLING 2012 Organizing Committee.

Freya Hewett and Manfred Stede. 2021. Automatically evaluating the conceptual complexity of German texts. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*, pages 228–234, Düsseldorf, Germany. KONVENS 2021 Organizers.

David Kauchak, Obay Mouradi, Christopher Pentoney, and Gondy Leroy. 2014. Text simplification tools: Using machine learning to discover features that identify difficult text. *2014 47th Hawaii International Conference on System Sciences*, pages 2616–2625.

Nikola Ljubešić. 2018. Concreteness and imageability lexicon MEGA.HR-crossling. Slovenian language resource repository CLARIN.SI.

Louis Martin, Samuel Humeau, Pierre-Emmanuel Mazaré, Éric de La Clergerie, Antoine Bordes, and Benoît Sagot. 2018. Reference-less quality estimation of text simplification systems. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, pages 29–38, Tilburg, the Netherlands. Association for Computational Linguistics.

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

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

Gustavo Paetzold and Lucia Specia. 2016. SemEval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Carolina Scarton, Gustavo Paetzold, and Lucia Specia. 2018. Text simplification from professionally produced corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Matthew Shardlow, Richard Evans, Gustavo Henrique Paetzold, and Marcos Zampieri. 2021. SemEval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.

Regina Stodden and Laura Kallmeyer. 2020. A multilingual and cross-domain analysis of features for text simplification. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with READing DIfficulties (READI)*, pages 77–84, Marseille, France. European Language Resources Association.

Pseudo-Labels Are All You Need

Bogdan Kostić and **Mathis Lucka** and **Julian Risch**
deepset
{bogdan.kostic, mathis.lucka, julian.risch}@deepset.ai

Abstract

Automatically estimating the complexity of texts for readers has a variety of applications, such as recommending texts with an appropriate complexity level to language learners or supporting the evaluation of text simplification approaches. In this paper, we present our submission to the Text Complexity DE Challenge 2022, a regression task where the goal is to predict the complexity of a German sentence for German learners at level B. Our approach relies on more than 220,000 pseudo-labels created from the German Wikipedia and other corpora to train Transformer-based models, and refrains from any feature engineering or any additional, labeled data. We find that the pseudo-label-based approach gives impressive results yet requires little to no adjustment to the specific task and therefore could be easily adapted to other domains and tasks.

1 Introduction

What makes some texts more difficult to read for learners of a foreign language than others? How does a complicated sentence construction or the use of rare vocabulary increase complexity? The prediction of text complexity with machine learning methods addresses these questions. In contrast to last years' shared tasks at KONVENS, which focused on the disambiguation of German verbal idioms (Ehren et al., 2021), the identification of toxic, engaging, and fact-claiming comments (Risch et al., 2021), and scene segmentation in narrative texts (Zehe et al., 2021), the task of 2022 is about text complexity. In this paper, we present our submission to this Text Complexity DE Challenge 2022. It is a shared task addressing the automatic estimation of the complexity of German sentences for readers, in particular, German learners at level B. The provided training dataset contains about 1000 sentences and the test dataset about 300 sentences in German. Figure 1 shows an exemplary sentence from the shared task dataset in German, an English translation, and the arithmetic mean of ratings from all annotators. With a seven-level Likert-scale with values ranging from very easy (1) to very complex (7), this task is a regression task and it is evaluated using the Root Mean Squared Error (RMSE). A third-order mapping is applied before the error is measured so that the impact of any systematic bias in the predictions on the metrics is reduced. Thereby, the focus of the evaluation is shifted towards ranking sentences correctly with regards to their complexity rather then assigning the correct absolute complexity score. We refer to the overview paper of the shared task for more details about the dataset and the overall results (Mohtaj et al., 2022).

The remainder of this paper is structured as follows. Section 2 summarizes related work on text complexity estimation and on pseudo-labeling techniques for machine learning. We describe our approach in Section 3 and its evaluation in Section 4, with experiments on the validation dataset provided by the shared task organizers. We conclude in Section 5 and provide an outlook on future work.

2 Related Work

Research on reading complexity of German texts is so far relatively scarce with several papers introducing datasets of annotated German sentences or longer texts and mostly feature-based approaches for text complexity prediction. First of all, there is a dataset with sentence-level annotations, which is the basis of this shared task (Naderi et al., 2019). Rios et al. (2021) introduce a dataset for document-level text complexity with the application focus of text simplification and there are two other document-level text complexity datasets by Battisti et al. (2020) and by Hewett and Stede

German Sentence: Als Versauerung der Meere wird die Abnahme des pH-Wertes des Meerwassers bezeichnet.
English Translation: Ocean acidification is the term used to describe the decrease in the pH of seawater.
Compexity Score: 2.13

German Sentence: Nach chemischer Härtung des Rußes war er in der Lage, auf galvanoplastischem Wege ein Zink-Positiv und von diesem ein Negativ der Platte anzufertigen, das als Stempel zur Pressung beliebig vieler Positive genutzt werden konnte – die Schallplatte war erfunden.
English Translation: After chemical hardening of the carbon black, he was able to produce a zinc positive by galvanoplastic means and from this a negative of the record, which could be used as a stamp for pressing any number of positives - the record was invented.
Complexity Score: 4.70

Figure 1: Two sentences from the training dataset.

(2021). The latter follows the format of a similar study (Hulpuș et al., 2019) based on a dataset of English newspaper articles (Xu et al., 2015). Another dataset is from a Kaggle challenge called CommonLit Readability Prize, where the task is to rate the complexity of literary passages for school grades 3-12.[1] Last but not least, there are unlabeled datasets of German texts with simple language, such as the Tagesschau/Logo corpus and the Geo/Geolino corpus (Weiß and Meurers, 2018) or Klexikon (Aumiller and Gertz, 2022). These datasets cannot be used directly for fine-tuning models on the text complexity prediction task due to the lack of annotations. However, we show in our approach that they can be used in combination with pseudo-labeling.

Similar to the pseudo-labeling approach that we use, there is a data augmentation technique where a slow but more accurate cross-encoder model is used to label a large set of otherwise unlabeled data samples (Thakur et al., 2021). This technique augments the training data for a faster, less complex bi-encoder model to address a pairwise sentence ranking task. Du et al. (2021) present a data augmentation method where given a small, labeled training dataset, they retrieve additional training samples from a large unlabeled dataset and then label these samples automatically with a model trained on the original, smaller training dataset. The resulting augmented, synthetic dataset can then be used to train another model that generalizes better to unseen data. Of the related work presented, this approach, also referred to as self-training, is the most similar to the approach we present in this paper. The main difference is that Du et al. (2021) tailor their approach mainly to domain-specific pretraining, whereas we focus on task-specific fine-tuning. Xie et al. (2019) extend the self-training method by intentionally adding noise to the training process to foster better generalization of the trained models. Further, they repeat the self-training process several times, so that the model trained on pseudo-labels is again used to create another set of pseudo-labels, which are in turn used to train another model and so on. As this iterative approach is very resource-intensive in terms of training time, we limit our approach to only one iteration. However, no inherent limitation prevents our approach from more training iterations. To the best of our knowledge, there are no published approaches that use neural language models for the particular task of complexity prediction of German texts, but only of English texts (Martinc et al., 2021).

3 A Semi-Supervised Learning Approach for Text Complexity Prediction

Our semi-supervised learning approach uses neural language models based on the Transformer architecture (Vaswani et al., 2017). As pre-trained models that are not fine-tuned to a specific natural language processing task yet, we use GBERT and GELECTRA models by Chan et al. (2020) and an XLM-RoBERTa model by Conneau et al. (2020). Given that the training dataset provided by the shared task organizers is relatively small for fine-tuning these pre-trained models, the core idea of our approach is to increase the number of training samples by automatically generating pseudo-labels. Figure 2 visualizes the different steps of the entire approach with its three main steps: pseudo-labeling, fine-tuning, and ensembling. For the implementation of these steps, we use the two open

[1] https://www.kaggle.com/competitions/commonlitreadabilityprize/

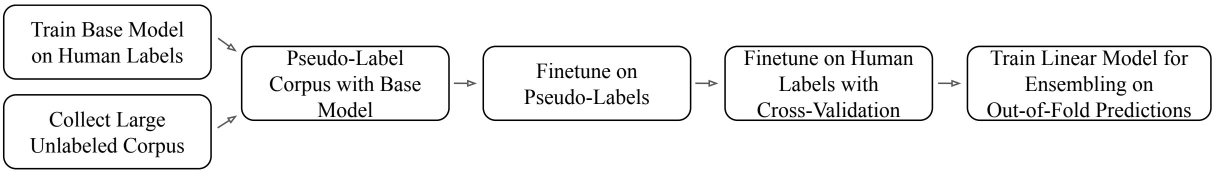
Train Base Model on Human Labels
Collect Large Unlabeled Corpus
Pseudo-Label Corpus with Base Model
Finetune on Pseudo-Labels
Finetune on Human Labels with Cross-Validation
Train Linear Model for Ensembling on Out-of-Fold Predictions

Figure 2: Overview of the different steps that comprise the pseudo-labeling, fine-tuning, and ensembling approach.

source frameworks FARM[2] and Haystack[3].

The first step is to create a large corpus of German sentences with varying text complexity to serve as a source for the pseudo-label sentences. This corpus comprises the following resources:

- eight random subparts of a German Wikipedia dump (about 8 percent of all 2.3 million German Wikipedia articles as of 2019),[4]
- 130,000 news articles from the German news platform Zeit Online,[5]
- three million sentences from German newspaper texts as part of the Leipzig Corpus Collection (Goldhahn et al., 2012),
- the Geo/Geolino/Tagesschau/Logo corpus (Weiß and Meurers, 2018),
- the Corpus Simple German (all subsets except for Klexikon),[6]
- the Klexikon (Aumiller and Gertz, 2022), and
- the Hurraki dictionary for plain language.[7]

Combining these datasets results in a total of 12,955,913 sentences. As some of them appear more than once, this corresponds to 12,562,164 distinct sentences. Each of them is embedded using a SentenceTransformers `msmarco-distilbert` model[8] (Reimers and Gurevych, 2019) and added to an OpenSearch index. Further, we fine-tune a `deepset/gbert-large` model on the task of sentence complexity on all of the provided training labels as a baseline. Subsequently, to get our set of pseudo-labels, we embed each of the sentences in the provided training set with the same SentenceTransformers `msmarco-distilbert` model and retrieve the 500 most similar sentences from our large corpus of German sentences as potential pseudo-labels. The baseline complexity scorer model produces a complexity score for each potential pseudo-label. To keep roughly the same distribution as in the original training dataset, we filter the generated pseudo-labels in the following way: we keep only those sentences whose predicted score does not deviate more than the standard deviation of the ratings of the original sentence used to retrieve the 500 potential pseudo-labels. This filtering results in a total of 228,796 pseudo-labels. Table 1 lists the number of pseudo-labels originating from the different data sources.

The pseudo-labels are used to fine-tune different Transformer-based models on the task of complexity scoring. We fine-tune `deepset/gelectra-large`, `deepset/gbert-large` and `xlm-roberta-large` using three different seeds for each model, resulting in a total of nine models. Subsequently, we fine-tune each of these models using five-fold cross-validation with the original training set, resulting in a total of 45 models. Finally, to combine these 45 models into an ensemble providing a single prediction score per data sample, we train linear regression models on the out-of-fold predictions from the previous cross-validations.

4 Experiments

We evaluate four different settings using five-fold cross-validation:

- a baseline `deepset/gbert-large` model fine-tuned on the provided training set,
- an ensemble of nine models fine-tuned only on pseudo-labels and with three different random seeds, scores aggregated by mean (`deepset/gbert-large`, `deepset/gelectra-large`, `xlm-roberta-large`

[2] https://github.com/deepset-ai/farm

[3] https://github.com/deepset-ai/haystack

[4] The most recent dump is available online: https://dumps.wikimedia.org/dewiki/20220720/

[5] https://www.zeit.de

[6] https://daniel-jach.github.io/simple-german/simple-german.html

[7] https://hurraki.de

[8] sentence-transformers/msmarco-distilbertmultilingual-en-de-v2-tmp-lng-aligned

Table 1: Number of pseudo-labeled sentences with the average length in characters, and the average mean opinion score per data source. The text complexity of the sources differs with Wikipedia and Hurraki being the most, respectively least difficult.

Data Source	**#Sentences**	**⌀Length**	**⌀MOS**
GERMAN WIKIPEDIA	137,228	133	3.0
ZEIT ONLINE	47,613	108	2.5
3 MILLION NEWS SENTENCES	25,928	110	2.6
GEO/GEOLINO/TAGESSCHAU/LOGO	7,971	98	2.5
CORPUS SIMPLE GERMAN	4,896	93	2.3
KLEXIKON	3,600	75	2.0
HURRAKI	1,559	43	1.5

- an ensemble of 45 models fine-tuned on pseudo-labels and the provided training set, with scores aggregated by mean, and
- an ensemble of 45 models fine-tuned on pseudo-labels and the provided training set, with scores aggregated by a linear model.

We submitted the predictions of each of the last three settings to the shared task competition.[9]

The first setting serves as our baseline with a `deepset/gbert-large` model fine-tuned on the provided training data. The model is trained on each of the cross-validation folds using early stopping for a maximum of four epochs. Each training run was tracked using MLflow and can be found here.

The second setting is an ensemble of the language models `deepset/gbert-large`, `deepset/gelectra-large` and `xlm-roberta-large`. Each of these models is fine-tuned for two epochs on the pseudo-labels described in Section 3 using three different random seeds. This results in an ensemble of nine models. One training run takes approximately three hours on an NVIDIA Tesla V100 GPU with 16 GB of RAM.

For the third and fourth setting, we fine-tune the resulting models of the previous step on the provided training dataset. To further increase the number of models in the ensemble, we perform five-fold cross-validation on each of the nine models, resulting in a total of 45 models. Again, each training run was tracked using MLflow and can be found here. To ensemble these 45 models, we use two different techniques. The third setting aggregates each individual score into a single score by simply taking the arithmetic mean of all scores. The fourth setting trains a linear ridge regression model on the out-of-fold predictions for each model that we trained, resulting in five linear regression models. Applying these linear regression models decreases the number of scores from 45 to 5. To get a single score out of these five scores, we calculate their arithmetic mean. Table 2 summarizes the hyperparameters that are used to train the models for the different described settings.

Table 3 lists the cross-validation RMSE on the provided training set. As expected, the approach of using pseudo-labels in combination with ensembling outperforms the simple baseline. We observe that fine-tuning the models only on the pseudo-labels already outperforms the baseline that uses only the original training data. Performance improves further if the models that were fine-tuned on the pseudo-labels are additionally fine-tuned on the original training data. Moreover, using a linear model to aggregate the individual scores instead of using the plain average does not further improve the final score. The best setting, consisting of an ensemble of 45 Transformer models fine-tuned on both the pseudo-labels and the provided training data, with results aggregated using a linear regression model, yields an RMSE of 0.433.

5 Conclusion

In this paper, we presented our submission to the Text Complexity DE Challenge 2022. We leveraged pseudo-labeled sentences from Wikipedia and several other publicly available, unlabeled corpora. Based on the labeled training dataset from the shared task and the additional pseudo-labeled data, we fine-tuned Transformer-based neural language models. Our best ensemble model achieved an

[9] `https://codalab.lisn.upsaclay.fr/competitions/4964`

Table 2: Hyperparameters for fine-tuning the language models on the pseudo-labels and the provided training data.

Hyperparameter	**Fine-Tuning on Pseudo-Labels**	**Fine-Tuning on Training Set**
Learning rate	1e-5	1e-6
LR schedule	linear	linear
Warm-up steps	10%	10%
Batch size	20 for `xlm-roberta-large`, 32 otherwise	20 for `xlm-roberta-large`, 32 otherwise
Early stopping	✖	✔
(Max.) epochs	2	4
Optimizer	Adam	Adam
Max sequence length	128	128

Table 3: Cross-validation RMSE.

Model	**1**	**2**	**3**	**4**	**5**	**∅**
Baseline	0.512	0.460	0.440	0.398	0.488	0.460
Ensemble pseudo-labels only	0.500	0.462	0.381	0.450	0.442	0.447
Ensemble simple mean aggregation	0.491	0.443	0.374	0.443	0.426	0.435
Ensemble linear model aggregation	0.445	0.455	0.405	0.443	0.418	**0.433**

RMSE of 0.433 in cross-validation on the public dataset without third-order mapping and an RMSE of 0.454 on the private test dataset with third-order mapping (0.484 without third-order mapping). For future work, our trained model could be used to create more pseudo-labels for another iteration of the entire approach, presumably resulting in a model that generalizes even better to unseen test data.

References

Dennis Aumiller and Michael Gertz. 2022. Klexikon: A German Dataset for Joint Summarization and Simplification. *arXiv preprint arXiv:2201.07198*.

Alessia Battisti, Dominik Pfütze, Andreas Säuberli, Marek Kostrzewa, and Sarah Ebling. 2020. A corpus for automatic readability assessment and text simplification of German. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 3302–3311, Marseille, France. European Language Resources Association.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Jingfei Du, Edouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. 2021. Self-training improves pre-training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5408–5418, Online. Association for Computational Linguistics.

Rafael Ehren, Timm Lichte, Jakub Waszczuk, and Laura Kallmeyer. 2021. Shared Task on the Disambiguation of German Verbal Idioms at KONVENS 2021. In *Proceedings of the Shared Task on the Disambiguation of German Verbal Idioms at KONVENS*.

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'12)*, pages 759–765, Istanbul, Turkey. European Language Resources Association (ELRA).

Freya Hewett and Manfred Stede. 2021. Automatically evaluating the conceptual complexity of German texts. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021)*, pages 228–234, Düsseldorf, Germany. KONVENS 2021 Organizers.

Ioana Hulpuș, Sanja Štajner, and Heiner Stuckenschmidt. 2019. A spreading activation framework for tracking conceptual complexity of texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3878–3887, Florence, Italy. Association for Computational Linguistics.

Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. Supervised and unsupervised neural approaches to text readability. *Computational Linguistics*, 47(1):141–179.

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

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

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.

Annette Rios, Nicolas Spring, Tannon Kew, Marek Kostrzewa, Andreas Säuberli, Mathias Müller, and Sarah Ebling. 2021. A new dataset and efficient baselines for document-level text simplification in German. In *Proceedings of the Third Workshop on New Frontiers in Summarization*, pages 152–161, Online and in Dominican Republic. Association for Computational Linguistics.

Julian Risch, Anke Stoll, Lena Wilms, and Michael Wiegand. 2021. Overview of the GermEval 2021 shared task on the identification of toxic, engaging, and fact-claiming comments. In *Proceedings of the GermEval 2021 Shared Task on the Identification of Toxic, Engaging, and Fact-Claiming Comments*, pages 1–12, Duesseldorf, Germany. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2021. Augmented SBERT: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 296–310, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Zarah Weiß and Detmar Meurers. 2018. Modeling the readability of German targeting adults and children: An empirically broad analysis and its cross-corpus validation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 303–317, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. 2019. Self-training with noisy student improves imagenet classification.

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.

Albin Zehe, Leonard Konle, Svenja Guhr, Lea Katharina Dümpelmann, Evelyn Gius, Andreas Hotho, Fotis Jannidis, Lucas Kaufmann, Markus Krug, Frank Puppe, et al. 2021. Shared task on scene segmentation@ konvens 2021. In *Proceedings of the Shared Task on Scene Segmentation at KONVENS*, pages 1–21.

Tackling Data Drift with Adversarial Validation: An Application for German Text Complexity Estimation

Alejandro Mosquera
Broadcom Corporation / 1320 Ridder Park Drive San Jose, 95131 California, USA
alejandro.mosquera@broadcom.com

Abstract

This paper describes the winning approach in the first automated German text complexity assessment shared task as part of KONVENS 2022. To solve this difficult problem, the evaluated system relies on an ensemble of regression models that successfully combines both traditional feature engineering and pre-trained resources. Moreover, the use of adversarial validation is proposed as a method for countering the data drift identified during the development phase, thus helping to select relevant models and features and avoid leaderboard overfitting. The best submission reached 0.43 mapped RMSE on the test set during the final phase of the competition.

1 Introduction

Automatically assessing how easy to read a text is has many applications, ranging from text simplification for language learners and people with disabilities to customizing content for a particular audience. For this reason, the Natural Language Processing (NLP) research community have been organizing shared tasks and compiled linguistic resources aiming to solve this problem, not only in English but also for other languages.

The Text Complexity DE Challenge 2022 (Mohtaj et al., 2022) proposes the evaluation of systems able to predict the complexity of German texts by rating each sentence using the Mean Opinion Score (MOS), derived from annotations from a 7 point Likert-scale. In order to solve this problem and addressing the unexpected data drift between training and testing sets, meta-modeling and adversarial validation techniques were applied by combining predictions from multiple estimators via stacked generalization (Wolpert, 1992) and leveraging both traditional feature engineering and pre-trained resources. The best submission generated by the described approach and selected through adversarial validation won the competition by achieving the lowest mapped Root Mean Squared Error (RMSE) score [1].

This paper is organized as follows: First, related work is reviewed in Section 2. Next, Section 3 contains an analysis of the individual models and feature engineering approaches used for this task. In Section 4, model selection and adversarial validation strategies are discussed. Further on, the performance of the system and its components are detailed in Section 5. Finally, in Section 6 the author draws the main conclusions and outlines future work.

2 Related Work

The application of NLP techniques for automatic textual complexity assessment has received attention in several languages other than English (Quispesaravia et al., 2016; Finnimore et al., 2019; Forti et al., 2019), although in an smaller scale. Despite the differences between languages, the use of lexical, morphological and word list-derived features are also common in research works focused on German (Weiss et al., 2019). Likewise, related NLP applications such as readability assessment (Hancke et al., 2012) or evaluation of text simplification pipelines (Suter et al., 2016) demonstrated that similar approaches used to estimate the complexity of English texts could be suitable for German as well, although with some known shortcomings.

3 Methodology

The TextComplexityDE (Naderi et al., 2019) dataset that consists of 1000 sentences in German language taken from 23 Wikipedia articles was the only resource provided by the organizers. In order to solve the challenge, this dataset was used

[1] https://qulab.github.io/text_complexity_challlenge/

as training data following two main approaches: feature engineering based on morphological and lexical information (Mosquera, 2021) and transfer learning via pre-trained transformers. The regression models trained using these two different strategies and the methodology applied to combine their predictions are described in detail below.

3.1 Feature Engineering Models

Several lexical features were calculated from word stats extracted from dlexDB (Heister et al., 2011), SUBTLEX-DE (Brysbaert et al., 2011) and averaged for each text. Likewise, sentence-level metrics from Textstat [2] and Readability [3] Python libraries were also used. A description of all the word and sentence features is as follows (entries ending with an asterisk denote a feature group):

dlexDB

- **typ_syls_cnt**: number of syllables.
- **typ_freq_***: absolute / normalized / log absolute / log normalized / rank / rank123 corpus frequency.
- **typ_fam_***: absolute / normalized / log absolute / log normalized / rank / rank123 familiarity (Kennedy et al., 2002) (cumulative frequency of all words of the same length sharing the same initial trigram).
- **typ_inf_***: absolute / normalized / log absolute / log normalized / rank / rank123 regularity (Kennedy et al., 2002) (the number of words of the same length sharing the same initial trigram).
- **typ_div_con_***: absolute / normalized / log absolute / log normalized / rank / rank123 document frequency.
- **typ_div_sen_***: absolute / normalized / log absolute / log normalized / rank / rank123 sentence count.
- **typ_uniq_orth_strict_pos**: length of the shortest prefix uniquely identifying the word.
- **typ_uniq_orth_strict_neg**: negative offset for the last character of the shortest prefix uniquely identifying the word.
- **typ_uniq_lemma_strict_pos**: length of the shortest prefix uniquely identifying the lemmatized word.
- **typ_uniq_lemma_strict_neg**: negative offset for the last character of the shortest prefix uniquely identifying the lemmatized word.
- **typ_pia_avgcondprob_big**: average conditional probability of a word, based on an evaluation of all bigrams having this word as their second component.
- **typ_pia_avginfcont_big**: average information content of a word, based on an evaluation of all bigrams having this word as second component (Piantadosi et al., 2011).
- **typ_pia_avgcondprob_trig**: average conditional probability of a word, based on an evaluation of all triigrams having this word as their third component.
- **typ_pia_avginfcont_trigr**: average information content of a word, based on an evaluation of all trigrams having this word as third component (Piantadosi et al., 2011).
- **typ_cts_cumfreq_token_***: absolute / normalized / log absolute / log normalized / rank / rank123 cumulative corpus frequency of all character trigrams contained in the word.
- **typ_cts_cumfreq_type_***: absolute / normalized / log absolute / log normalized / rank / rank123 cumulative lexicon frequency of all character trigrams contained in the word.
- **typ_init_trigr_***: absolute / normalized / log absolute / log normalized / rank / rank123 cumulative frequency of all words sharing the same initial character trigram (Lima and Inhoff, 1985).
- **typ_nei_col_all_cnt_abs**: absolute number of orthographic neighbors (Coltheart, 1977).
- **typ_syls_cumfreq_token_***: absolute / normalized / log absolute / log normalized / rank / rank123 cumulative corpus frequency of all syllables contained in the word.
- **typ_syls_cumfreq_type_***: absolute / normalized / log absolute / log normalized / rank / rank123 cumulative lexicon frequency of all syllables contained in the word.

[2] https://pypi.org/project/textstat/

[3] https://pypi.org/project/readability/

SUBTLEX-DE

- **WFfreqcount**: target word frequency in the German subtitle corpus.
- **spell-check OK (1/0)**: 1 if the word had no spelling errors, 0 otherwise.
- **CUMfreqcount**: case-independent word frequency in the German subtitle corpus.
- **SUBTLEX**: frequency per million based on CUMfreqcount.
- **lgSUBTLEX**: log10(CUMfreqcount+1).
- **Google00**: word frequency based on Google 2000-2009 Books corpus.
- **Google00cum**: case-independent word frequency based on Google 2000-2009 Books corpus.
- **Google00pm**: Google frequency per million words.
- **lgGoogle00**: log10(Google00cum+1).

Sentence Readability

- **Kincaid**: Kincaid grade level.
- **ARI**: Automated readability index (Senter and Smith, 1967).
- **Coleman-Liau**: Coleman-Liau readability score (Coleman and Liau, 1975).
- **Flesch reading ease**: Flesh reading ease score (Flesch, 1948).
- **Gunning-Fog index**: Gunning-Fog readability index (Gunning et al., 1952).
- **LIX**: LIX readability score (Anderson, 1983).
- **SMOG index**: SMOG readability index (Mc Laughlin, 1969).
- **RIX**: RIX readabiliy score (Anderson, 1983).
- **Dale-Chall index**: Dale-Chall readability index (Chall and Dale, 1995) of the whole sentence.
- **Wiener Sachtextformel**: grade level for German texts (Schulz et al., 1985)

Linear regression and gradient boosting models were trained with all the above features with default hyper-parameters. The 2 resulting estimators are referred across the paper as LR and LGB (Ke et al., 2017) respectively.

A list of the top 20 features in terms of minimum redundancy and maximum relevance (mRMR) (Ding and Peng, 2003) can be found in Table 1.

Table 1: Top 20 features (minimal-optimal set).

Feature
RIX
ARI
Kincaid
GunningFogIndex
LIX
SMOGIndex
typ_init_trigr_abs
wiener_sachtextformel
FleschReadingEase
typ_init_trigr_nor
Google00
Coleman-Liau
typ_uniq_orth_strict_pos
Google00pm
DaleChallIndex
typ_syls_cumfreq_type_rank123
Google00cum
typ_uniq_lemma_strict_pos
typ_cts_cumfreq_type_abslog
typ_fam_abs

3.2 Transformer Models

Regression models using neural network architectures based on the Transformer were trained via fine-tuning on the dataset provided by the task organizers. A selection of the estimators that were used in order to generate some of the best scoring submissions is as follows:

- **NN**: BERT (Devlin et al., 2019) fine-tuned for 1 epoch [4].
- **NNr**: BERT fine-tuned for 1 epoch (reverse word order) [5].
- **NN3**: RoBERTa (Liu et al., 2019) fine-tuned for 3 epochs [6].

[4] https://huggingface.co/dbmdz/bert-base-german-cased
[5] https://huggingface.co/dbmdz/bert-base-german-cased
[6] https://huggingface.co/xlm-roberta-base

- **NN5**: BERT fine-tuned for 2 epochs [7].

3.3 Ensemble

Meta-modeling techniques were applied in order to combine base models into single predictors by using stacking generalization. The second level algorithm used for this task was linear regression which used the following weights for Ensemble1 and Ensemble2 respectively:

$$Ensemble1 = 0.18 \times LR + 0.17 \times LGB + 0.21 \times NN + 0.39 \times NNr$$

$$Ensemble2 = 0.1 \times LR + 0.05 \times LGB + 0.02 \times NN + 0.05 \times NNr + 0.25 \times NN3 + 0.478 \times NN5$$

The out-of-fold cross validation scores of the base and meta models can be found in Table 2 .

Table 2: Train set errors calculated with 5-fold cross validation.

Model	RMSE	MAE
LR	0.726	0.585
LGB	0.707	0.561
NN	0.685	0.542
NNr	0.662	0.527
NN3	0.673	0.531
NN5	0.61	0.477
Ensemble1	0.625	0.5
Ensemble2	**0.588**	**0.464**

4 Model Selection and Adversarial Validation

In the final phase of the competition it became clear that validation and test data had relevant dissimilarities. Some potential reasons were identified by the participants such as the application of non-random splits or different pre-processing [8]. While this is not a totally uncommon phenomenon in NLP (Karpov, 2017; Mosquera, 2020) to the best of the authors' knowledge there have not been many efforts to address this problem in comparison with other domains.

The use of adversarial validation as a solution to identify concept drift has been explored recurrently in the literature (Pan et al., 2020). However, due the relatively small data sizes involved, the usual approach of training a binary classifier between train/dev/test sets and selecting the data points with the closest distribution (Qian et al., 2021) was deemed sub-optimal. Therefore, Principal Component Analysis (PCA) was used instead in order to calculate low-dimensional projections of the evaluation datasets and estimate their drift from the training data by analyzing the reconstruction errors. Taking that into account, the author hypothesized that models using features that would remain stable across different data splits based on the criteria define above would exhibit better correlation between the errors estimated during cross-validation and the final scores.

In Table 3, it can be observed that the Ensemble1 meta-model could be affected by the data drift and its estimated performance during the development phase would likely not translate to the final phase evaluation. These insights were particularly relevant since development phase models were also partially tuned using feedback from the public leaderboard and ignoring this valuable information would have resulted in a non-optimal model and feature selection.

Table 3: PCA reconstruction errors against train (3 components, 0.95 variance).

Model	Train	Development	Test
Ensemble1	0.0275	0.0311	0.029
Ensemble2	**0.0329**	**0.0334**	**0.0328**

5 Results

Aiming to compensate for the possible variance between several subjective ratings, the challenge organizers decided to use a custom evaluation metric by applying a 3rd order linear mapping function per each dataset before calculating the error, which meant that the RMSE score would always differ from the mapped RMSE. Considering that the mapping function was unknown to the participants, RMSE was used instead as the main metric for local validation and optimization purposes.

While in practice, the mapped scores seemed to correlate with the local validation, rankings based on RMSE scores differed substantially from rankings derived from the mapped version. This was particularly obvious during the development phase [9] where only 2 out of 14 participants (5 out of 14 during the final phase) had the same ranking when

[7] https://huggingface.co/amine/bert-base-5lang-cased

[8] https://codalab.lisn.upsaclay.fr/forums/4964/741/

[9] https://codalab.lisn.upsaclay.fr/competitions/4964#results

considering both mapped and original RMSE metrics which highlights the extra difficulty added by the chosen evaluation metric for this competition.

In Table 4 the results for the highest ranked submissions generated by the described approach during different phases of the competition are listed. As expected after the adversarial validation step, the meta-model Ensemble1, which produced a high score solution during the evaluation phase (lowest RMSE, second best mapped RMSE overall), underperformed in the final phase and would have ended up in the 9th position in absence of better submissions after being affected by the aforementioned data drift. On the other hand, the meta-model Ensemble2 had similar reconstruction errors in all the datasets and ended up generating the winning submission.

Table 4: Task results (mapped RMSE and RMSE) for selected submissions during different competition phases.

Model	Development	Test
Ensemble1	**0.326 - 0.361**	0.484 - 0.502
Ensemble2	n/a	**0.43 - 0.446**

Since the challenge organizers decided to not release the labels of the evaluation datasets and disabled post-competition submissions, additional ablation analysis can not be performed in this section.

6 Conclusions and Future Work

This paper introduces a meta-model for German text complexity estimation using both manual feature engineering and neural networks. The use of adversarial validation by comparing feature distribution changes between different datasets is proposed as a mechanism to detect data drift via PCA reconstruction errors. The described system has achieved the first ranking in mapped RMSE in the Text Complexity DE Challenge of KONVENS 2022. In a future work, this approach can be extended through AutoNLP techniques in order to build multi-lingual text complexity estimation solutions that could be integrated in other NLP pipelines.

References

Jonathan Anderson. 1983. Lix and rix: Variations on a little-known readability index. *Journal of Reading*, 26(6):490–496.

Marc Brysbaert, Matthias Buchmeier, Markus Conrad, Arthur Jacobs, Jens Bölte, and Andrea Böhl. 2011. The word frequency effect: A review of recent developments and implications for the choice of frequency estimates in german. *Experimental psychology*, 58:412–24.

Jeanne Sternlicht Chall and Edgar Dale. 1995. *Readability revisited: The new Dale-Chall readability formula*. Brookline Books.

Meri Coleman and Ta Lin Liau. 1975. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283.

Max Coltheart. 1977. Access to the internal lexicon.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Chris Ding and Hanchuan Peng. 2003. Minimum redundancy feature selection from microarray gene expression data. volume 3, pages 523– 528.

Pierre Finnimore, Elisabeth Fritzsch, Daniel King, Alison Sneyd, Aneeq Ur Rehman, Fernando Alva-Manchego, and Andreas Vlachos. 2019. Strong baselines for complex word identification across multiple languages. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 970–977, Minneapolis, Minnesota. Association for Computational Linguistics.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of applied psychology*, 32(3):221.

Luciana Forti, Alfredo Milani, Luisa Piersanti, Filippo Santarelli, Valentino Santucci, and Stefania Spina. 2019. Measuring text complexity for Italian as a second language learning purposes. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 360–368, Florence, Italy. Association for Computational Linguistics.

Robert Gunning et al. 1952. Technique of clear writing.

Julia Hancke, Sowmya V., and Detmar Meurers. 2012. Readability classification for german using lexical, syntactic, and morphological features. pages 1063–1080.

Julian Heister, Kay-Michael Würzner, Johannes Bubenzer, Edmund Pohl, Thomas Hanneforth, Alexander Geyken, and Reinhold Kliegl. 2011. dlexdb—eine lexikalische datenbank für die psychologische und linguistische forschung. *Psychologische Rundschau*, 62:10–20.

Nikolay Karpov. 2017. NRU-HSE at SemEval-2017 task 4: Tweet quantification using deep learning architecture. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 683–688, Vancouver, Canada. Association for Computational Linguistics.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3149–3157, Red Hook, NY, USA. Curran Associates Inc.

Alan Kennedy, Joël Pynte, and Stéphanie Ducrot. 2002. Parafoveal-on-foveal interactions in word recognition. *The Quarterly journal of experimental psychology. A, Human experimental psychology*, 55:1307–37.

Susan Lima and Albrecht Inhoff. 1985. Lexical access during eye fixations in reading. effects of word-initial letter sequence. *Journal of experimental psychology. Human perception and performance*, 11:272–85.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

G. Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

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

Alejandro Mosquera. 2020. Amsqr at SemEval-2020 task 12: Offensive language detection using neural networks and anti-adversarial features. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1898–1905, Barcelona (online). International Committee for Computational Linguistics.

Alejandro Mosquera. 2021. Alejandro mosquera at SemEval-2021 task 1: Exploring sentence and word features for lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 554–559, Online. Association for Computational Linguistics.

Babak Naderi, Salar Mohtaj, Kaspar Ensikat, and Sebastian Möller. 2019. Subjective assessment of text complexity: A dataset for german language. *arXiv preprint arXiv:1904.07733*.

Jing Pan, Vincent Pham, Mohan Dorairaj, Huigang Chen, and Jeong-Yoon Lee. 2020. Adversarial validation approach to concept drift problem in user targeting automation systems at uber.

Steven Piantadosi, Harry Tily, and Edward Gibson. 2011. Word lengths are optimized for efficient communication. *Proceedings of the National Academy of Sciences of the United States of America*, 108:3526–9.

Hongyi Qian, Baohui Wang, Ping Ma, Lei Peng, Songfeng Gao, and You Song. 2021. Managing dataset shift by adversarial validation for credit scoring.

Andre Quispesaravia, Walter Perez, Marco Sobrevilla Cabezudo, and Fernando Alva-Manchego. 2016. Coh-Metrix-Esp: A complexity analysis tool for documents written in Spanish. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4694–4698, Portorož, Slovenia. European Language Resources Association (ELRA).

Renate A. Schulz, Richard Bamberger, and Erich Vanecek. 1985. Lesen-verstehen-lernen-schreiben: Die schwierigkeitsstufen von texten in deutscher sprache. *Die Unterrichtspraxisteaching German*, 18:366.

R.J. Senter and Edgar A. Smith. 1967. Automated readability index. Technical report, CINCINNATI UNIV OH.

Julia Suter, Sarah Ebling, and Martin Volk. 2016. Rule-based automatic text simplification for german. In *13th Conference on Natural Language Processing (KONVENS 2016)*. s.n.

Zarah Weiss, Anja Riemenschneider, Pauline Schröter, and Detmar Meurers. 2019. Computationally modeling the impact of task-appropriate language complexity and accuracy on human grading of German essays. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 30–45, Florence, Italy. Association for Computational Linguistics.

David Wolpert. 1992. Stacked generalization. *Neural Networks*, 5:241–259.

Text Complexity DE Challenge 2022 Submission Description: Pairwise Regression for Complexity Prediction

Leander Girrbach
University of Tübingen
leander.girrbach@student.uni-tuebingen.de

Abstract

This paper describes our submission to the Text Complexity DE Challenge 2022 (Mohtaj et al., 2022). We evaluate a pairwise regression model that predicts the relative difference in complexity of two sentences, instead of predicting a complexity score from a single sentence. In consequence, the model returns samples of scores (as many as there are training sentences) instead of a point estimate. Due to an error in the submission, test set results are unavailable. However, we show by cross-validation that pairwise regression does not improve performance over standard regression models using sentence embeddings taken from pretrained language models as input. Furthermore, we do not find the distribution standard deviations to reflect differences in "uncertainty" of the model predictions in an useful way.

1 Introduction

This paper describes our submission to the Text Complexity DE Challenge 2022 (Mohtaj et al., 2022). The task is to predict the linguistic complexity of a given sentence. The task is defined as a regression task, where labels are $\in [1, 7]$. Labels are averaged human ratings, who rated the sentences for complexity, understandability, and lexical defficulty (see (Naderi et al., 2019a) for details). Only complexity labels are taken into account in this shared task. The train set consists of 1000 labelled sentences, the development set consists of 100 sentences, and the test set contains 210 sentences. Only the labels of training sentences where ever revealed to participants.

In this paper, we evaluate pairwise regression for complexity score prediction. Instead of predicting a single complexity score from a single sentence, we predict the relative difference in complexity of two sentences. In practise, this results in a distribution over complexity scores instead of a point estimate, because we predict the relative difference for each training sentence. However, further analysis reveals that pairwise regression neither performs better than standard regression nor does the standard deviation of score distributions contain useful information about model performance.

Furthermore, due to an erroneous submission we do not have test set score for this shared task. Therefore, all our analyses and observations are based on 10-fold cross-validation on the training data.

2 Related Work

Readability scoring of texts is has been researched for over a century. Research started by developing readability formulas based on surface features such as token counts or type-token ratios. Modern approaches use statistical methods, especially supervised learning, to learn readability models. Here, readability scoring can be defined both as a regression task (Naderi et al., 2019a; vor der Brück et al., 2008) and a classification task (Hancke et al., 2012; Weiss et al., 2021). Features usually rely on broad linguistic modelling (Weiß and Meurers, 2018; Naderi et al., 2019b).

Recently, deep neural networks have also been proposed for predicting readability labels (Martinc et al., 2021). Furthermore, the utility of linguistic features compared to deep representations was put into question by Deutsch et al. (2020). However, the main disadvantage of deep neural networks is their black-box nature. This is especially problematic, because practical applications of readability models generally require an especially high level of transparency, for example in an educational context (for giving feedback) or for essay scoring (where grades should be explainable and fair).

3 Method

In this section, we describe our approach at predicting the linguistic complexity of given sentences.

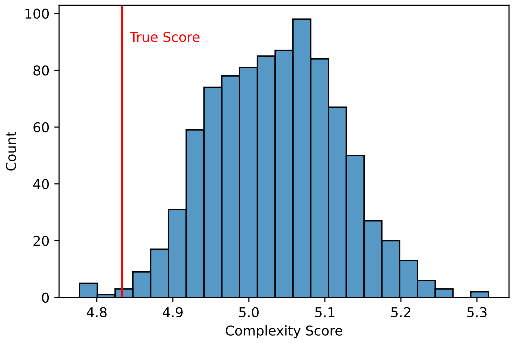

Figure 1: Distribution of scores predicted by a pairwise regression model for sentence "Infolge des gravitationsbedingten Auslaufens (Drainage) der zwischen den Seifenfilmoberflächen befindlichen Flüssigkeit dünnt eine Seifenblase in ihrem oberen Teil zunehmend aus."

We train deep learning models (described in Section 3.1) and also compare them to a traditional machine learning model based on linguistic features (see Section 3.2).

3.1 Pairwise Regression

Our main model is a deep learning model trained in a supervised fashion. Instead of directly predicting the complexity score from a single sentence, we use sentence pairs as inputs. Given a pair of sentences, we predict the difference in complexity of the sentences. At test time, after the model was trained, we predict the label of an unseen sentence by predicting the relative differences in difficulty to all sentences in the training set (in case of large training sets, taking a subset would also be possible). Because we know the true labels of train sentences, we use them to calculate an estimate of the complexity of the unseen sentence for every sentence in the training set. This gives us a sample of estimated complexity scores. We can arrive at a final estimate by taking the mean, or estimating the mode of the resulting distribution in a different way. An example of a predicted distribution produced by one of our models is in Figure 1.

Our main motivations for pairwise regression instead of single-sentence regression are: Given the data for this task is relatively small (1000 sentences in the train set), using sentence pairs is an easy way to increase the data set size. Furthermore, pairwise regression makes more use of the given data by treating sentences not only as isolated datapoints, but seeing them in relation to all other sentences in the dataset. Also, previous work (Lee and Vajjala, 2022; Weiss and Meurers, 2022) showed promising performance of pairwise readability ranking models. Therefore, we wanted to evaluate whether this also is true for a regression setting. In detail, or model is designed as follows:

Sentence Embedding First, we encode a sentence by 3 different openly available pretrained language models models:

- GottBERT (Scheible et al., 2020).[1] The sentence embeddings is calculated by averaging embeddings of all non-special tokens.
- dbmz's German BERT (cased) model.[2] The sentence embedding is simply the embedding of the "[CLS]" token.
- A multilingual sentence transformer model (Reimers and Gurevych, 2020).[3] We found German pretrained sentence transformers to not perform as well.

We concatenate all 3 embeddings to arrive at the final encoding of a sentence. Note that we do not fine-tune the pretrained models, but simply use them as feature extractors.

Prediction First, we transform each sentence separately (using the same model) by a MLP with 2 hidden layers and GELU activation. Then, we concatenate the transformed sentence embeddings and use a MLP with 1 hidden layer and GELU activation to predict the complexity difference. Optionally, we also predict the absolute complexity score of the input sentences. A visualisation of the model is shown in Figure 2.

Training Setup All models are implemented in PyTorch (Paszke et al., 2019). We train models for 6 epochs using batch size 32, dropout probability 0.3 (applied before every linear layer) and hidden sizes 300 and 600 (the first hidden layer of each MLP is twice the standard hidden size). In each of the 6 epochs, the model is trained on all combinations of sentences. Given the size of the present dataset, this is feasible, but in case of larger datasets sampling combinations is an option.

[1] https://huggingface.co/uklfr/gottbert-base

[2] https://huggingface.co/dbmdz/bert-base-german-cased

[3] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1

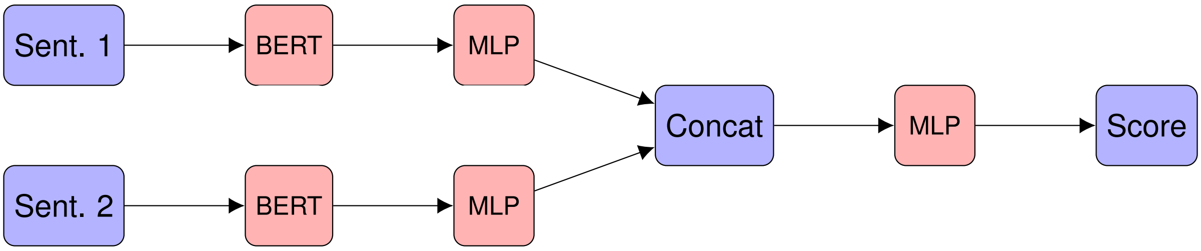

Figure 2: Flowchart showing how the pairwise regression model predicts relative complexity difference scores. Blue blocks are data and red blocks are neural networks.

The number of 6 epochs was found to work best by manual hyperparameter exploration. The optimizer is SGD with weight decay coefficient 1e-4. We set the learning rate according to a One-Cycle-Scheduler (Smith and Topin, 2019) with maximum learning rate 0.001. As regression loss, we use the smoothed L1 metric.

3.2 Baselines

In addition to the pairwise regression model described in Section 3.1, we evaluate 2 baselines:

One baseline is a random forest model trained on linguistic features extracted by CTAP (Chen and Meurers, 2016; Weiss et al., 2021).[4] We extract all features available for German. Then, we remove all features that resulted in NaN for at least 1 sentence, and we remove constant features. We train a random forest model using the scikit-learn implementation (Pedregosa et al., 2011) with the following hyperparameters: The number of trees is 450, the maximum percentage of features used for calculating splits is 85%, and both the minimum number of datapoints required for internal and leaf splits is 5.

Secondly, we train a simple (i.e. without pairwise regression) MLP regressor to predict complexity scores from single sentences. To be as comparable as possible to the pairwise regression model, we use the same hyperparameters. However, due to the different datasets, we need to change the number of epochs. We found 500 epochs to work best. Also, we evaluate all 3 pretrained language models as feature extractors and the combination of their sentence embeddings.

4 Results

Here, we present performance results of the pairwise regression model (see Section 3.1) and baselines (see Section 3.2). Unfortunately, we cannot present the shared task's test set scores due to an erroneous submission: Instead of submitting results on the real test set, we accidentally submitted results on a custom test set that we had created for internal evaluation. This error remained unnoticed until after the submission deadline. Therefore, we decide to report 10-fold cross-validation results on the training set, because we do also not have development set scores for all baselines and ablations.

Results for the pairwise regression model are in Table 1. Here, we can make 2 observations: Firstly, models perform similarly, but the best performing models only use GottBERT as sentence encoder. This suggests that the GottBERT model is, among the evaluated models, best at representing complexity-relevant features. Secondly, additionally predicting absolute complexity scores does not have a visible effect on the performance. Therefore, replacing absolute complexity score predictions by relative score predictions is possible.

In Figure 3, we show the loss curve for a pairwise regression model only predicting the relative difference and using all sentence embeddings. The curve shows that the loss starts to decrease quickly after about half an epoch. This may be an artifact of the initially very low learning rate due to the One-Cycle-Scheduler. After about 2 epochs, the loss only shows little improvements. This suggests that training for fewer epochs may already be sufficient. However, given that we did not observe better generalisation performance with shorter training, this may also suggest that the model is somewhat robust to longer training and still does not overfit the data.

Results for the baselines are in Table 2. The best performing model uses all 3 sentence embeddings and also yields the best overall results. This puts benefits of pairwise regression into question, since they apparently do not yield improvements in performance. However, neural models generally outperform the non-neural baseline, although the difference is not very large in absolute terms. Also,

[4] http://sifnos.sfs.uni-tuebingen.de/ctap/

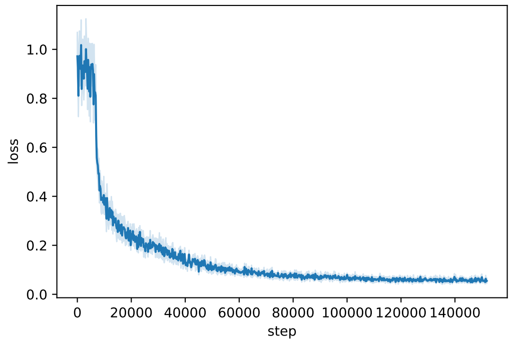

Figure 3: Loss curve for pairwise regression model (GottBERT + BERT + S-BERT + Only Δ). For each step, we display the loss mean and standard deviation (shaded area) calculated from the 10 cross-validation runs.

Only Δ	GottBERT	BERT	S-BERT	RMSE
✓	✓	✓	✓	0.6270
✓	✓	✓		0.6333
✓	✓		✓	0.6130
✓	✓			0.6178
✓		✓	✓	0.6596
✓		✓		0.6830
✓			✓	0.6725
	✓	✓	✓	0.6315
	✓	✓		0.6375
	✓		✓	0.6188
	✓			0.6170
		✓	✓	0.6593
		✓		0.6769
			✓	0.6706

Table 1: Ablation results of various pairwise regression configurations (10-fold cross-validation on training set). "Only Δ" mean whether we only predict the relative score differences or also predict abolute scores. "GottBERT", "BERT", "S-BERT" are the different sentence embedding models described in Section 3.1.

Model	RMSE
GottBERT	0.6123
BERT	0.6639
S-BERT	1.1612
Combined	0.6068
Random Forest	0.6946

Table 2: RMSE results (10-fold cross-validation on training set) for baselines. "Combined" means representing sentences by concatenating sentence embeddings calculated by all 3 pretrained models. Random Forest uses linguistic features extracted by CTAP.

the non-neural baseline has the advantage of being interpretable to some degree. We would also like to note that the neural models outperform the results reported by Naderi et al. (2019b) and Weiss and Meurers (2022), who use a similar setup. Finally, we note that the model based on sentence transformers did not converge and would need more epochs. For the sake of comparability, we decide to still keep the setup the same for all baseline models.

5 Analysis

In Section 4, we have established that pairwise regression does not achieve better performance than direct prediction of absolute complexity scores. However, we are still interested in whether having a distribution of scores instead of a single score can provide additional insights. For example, it would be of advantage if we could use the score sample standard deviation to detect uncertain predictions, i.e. sentences where the model is not confident about the complexity. To be able to do this, the sample standard deviation has to correlate with the prediction error. This is, however, not the case: Figure 4 shows that while most errors are small, sentence score distributions that result in larger prediction errors do not have larger standard deviation. In fact, Pearson correlation is -0.18, however the negative value could be an artifact of the small number of large errors. Therefore, we conclude that the score distribution predicted by pairwise regression models does not provide further insights into the model predictions.

Finally, we also evaluate whether we can find linguistic features that are informative about which sentences are hard to score by the deep learning models. To evaluate this, we conduct another 10-fold cross-validation experiment using a Lasso model (scikit-learn implementation) to predict the squared error from linguistic features. However, the resulting R^2-score is only 0.02, which is barely better than always predicting the average. Therefore we conclude that linguistic features in this case cannot help detect sentences that are difficult for the deep models to score and we refrain from further analysing the importance of individual features.

6 Discussion

We evaluated pairwise regression in comparison to standard regression (predicting a single complexity score from a single sentence). Our results are

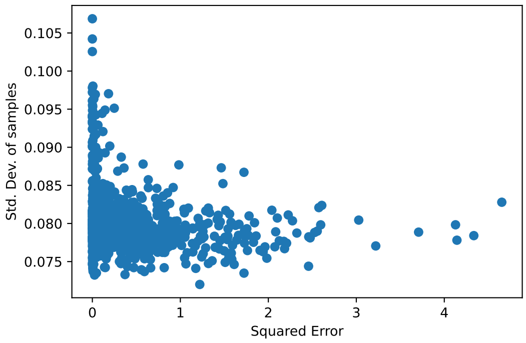

Figure 4: Scatter plot showing the relationship of standard deviation of score distributions predicted by a pairwise regression model (only Δ, all embeddings).

largely negative, showing that pairwise regression does not perform better than standard regression and the resulting score distribution does not seem to have additional use over the point estimates returned by standard regression. Furthermore, there seems to be no trend that can be captured by linguistic features about which sentences are more difficult to score by deep learning based models.

On the positive side, our evaluations show that pairwise regression and standard regression can be exchanged with only very little difference in prediction quality, and that deep learning based models perform somewhat better than models based on linguistic features.

Acknowledgements

We thank the organisers for organising this shared task.

References

Xiaobin Chen and Detmar Meurers. 2016. CTAP: A web-based tool supporting automatic complexity analysis. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 113–119, Osaka, Japan. The COLING 2016 Organizing Committee.

Tovly Deutsch, Masoud Jasbi, and Stuart M. Shieber. 2020. Linguistic features for readability assessment. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications, BEA@ACL 2020, Online, July 10, 2020*, pages 1–17. Association for Computational Linguistics.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability classification for German using lexical, syntactic, and morphological features. In *Proceedings of COLING 2012*, pages 1063–1080, Mumbai, India. The COLING 2012 Organizing Committee.

Justin Lee and Sowmya Vajjala. 2022. A neural pairwise ranking model for readability assessment. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3802–3813, Dublin, Ireland. Association for Computational Linguistics.

Matej Martinc, Senja Pollak, and Marko Robnik-Sikonja. 2021. Supervised and unsupervised neural approaches to text readability. *Comput. Linguistics*, 47(1):141–179.

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

Babak Naderi, Salar Mohtaj, Kaspar Ensikat, and Sebastian Möller. 2019a. Subjective assessment of text complexity: A dataset for german language. *CoRR*, abs/1904.07733.

Babak Naderi, Salar Mohtaj, Karan Karan, and Sebastian Möller. 2019b. Automated text readability assessment for german language: A quality of experience approach. In *11th International Conference on Quality of Multimedia Experience QoMEX 2019, Berlin, Germany, June 5-7, 2019*, pages 1–3. IEEE.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

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. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Raphael Scheible, Fabian Thomczyk, Patric Tippmann, Victor Jaravine, and Martin Boeker. 2020. Gottbert: a pure german language model. *CoRR*, abs/2012.02110.

Leslie N Smith and Nicholay Topin. 2019. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE.

Tim vor der Brück, Sven Hartrumpf, and Hermann Helbig. 2008. A readability checker with supervised learning using deep indicators. *Informatica*, 32(4).

Zarah Weiss, Xiaobin Chen, and Detmar Meurers. 2021. Using broad linguistic complexity modeling for cross-lingual readability assessment. In *Proceedings of the 10th Workshop on NLP for Computer Assisted Language Learning*, pages 38–54, Online. LiU Electronic Press.

Zarah Weiß and Detmar Meurers. 2018. Modeling the readability of German targeting adults and children: An empirically broad analysis and its cross-corpus validation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 303–317, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Zarah Weiss and Detmar Meurers. 2022. Assessing sentence readability for German language learners with broad linguistic modeling or readability formulas: When do linguistic insights make a difference? In *Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022)*, pages 141–153, Seattle, Washington. Association for Computational Linguistics.

TUM sebis at GermEval 2022: A Hybrid Model Leveraging Gaussian Processes and Fine-Tuned XLM-RoBERTa for German Text Complexity Analysis

Juraj Vladika*, **Stephen Meisenbacher***, **Florian Matthes**
Technical University of Munich
Department of Informatics
Garching, Germany
{juraj.vladika, stephen.meisenbacher, matthes}@tum.de

Abstract

The task of quantifying the complexity of written language presents an interesting endeavor, particularly in the opportunity that it presents for aiding language learners. In this pursuit, the question of what exactly about natural language contributes to its complexity (or lack thereof) is an interesting point of investigation. We propose a hybrid approach, utilizing shallow models to capture linguistic features, while leveraging a fine-tuned embedding model to encode the semantics of input text. By harmonizing these two methods, we achieve competitive scores in the given metric, and we demonstrate improvements over either singular method. In addition, we uncover the effectiveness of Gaussian processes in the training of shallow models for text complexity analysis.

1 Introduction

In this paper, we present a novel approach for the quantification of text complexity in the German language, as part of the Text Complexity DE Challenge 2022 (Mohtaj et al., 2022). Specifically, we emphasize a hybrid method for building a text complexity model, which combines a feature-based, shallow regression model with a fine-tuned XLM-RoBERTa model. In doing so, we hope to capture to the fullest both the linguistic aspects that contribute to text complexity, as well as the semantic factors. In the following Section 2, we briefly describe the task at hand, as well as the data used to train and test our models. Next, Section 3 introduces Gaussian Processes, which become central to our hybrid system. Likewise, fine-tuning RoBERTa for use in regression tasks is covered in Section 4. These concepts are brought together in 6, which describes our overall model architecture for the task. Before this, the feature set used to train both models is illustrated in Section 5. In Section 7, we present results from the training and validation phases, in which our model achieved the best score for this task's chosen metric. In the ensuing Section 8, we perform a qualitative analysis of our approach and lessons learned. Finally, Section 9 provides a few concluding remarks. The systems (and code used to create them) are publicly available under `https://github.com/sebischair/Text-Complexity-DE-2022`.

2 Dataset and Task

The dataset used in this task was first presented by Naderi et al. (2019). It consists of 1000 German language sentences sourced from 23 Wikipedia articles. These articles have been classified into three different genres. These sentences are annotated with ratings of 1-7 in the category of *Complexity*, *Understandability*, and *Lexical complexity*. These ratings are presented as a *Mean Opinion Score* (MOS), which is represented as the average scoring of the annotators. The sentences were scored by German language learners from A2 to B2 levels. For example, the sentence "*Eine Seifenblase entsteht, wenn sich ein dünner Wasserfilm mit Seifenmolekülen vermischt.*" is scored with 2.9.

For this challenge, one unified MOS score was given. How this score was derived from the original three scores is not explained. The original 1000 sentences are the same.

With this dataset, the task becomes to train a regression model that predicts the complexity of a sentence (i.e. MOS) from the sentence text. This score is intended to aid in the quantification of text complexity for language learners, as well as in the evaluation of the simplified text.

3 Gaussian Processes

A popular area in Machine Learning, particularly near the turn of the century, rested in the study of "kernel machines", which include the popular Support Vector Machines, but also the lesser known *Gaussian Process Models* (Rasmussen, 2003). At

[*] These authors contributed equally.

their core, Gaussian Processes (GPs) are powerful in the way they incorporate probabilistic thinking into kernel machines, making them a particularly suitable tool for supervised machine learning in small data settings (Urtasun and Darrell, 2007).

Literally, Gaussian processes are built upon multivariate Gaussian (normal) distributions, defined by a mean vector μ and covariance matrix Σ, i.e.:

$$\vec{X} = \begin{bmatrix} X_1 & X_2 & \dots & X_N \end{bmatrix} \sim \mathcal{N}(\mu, \Sigma)$$

This particular distribution has the useful property of being closed under marginalization (probability distribution of partitions) and conditioning (probability of one variable depending on another).

The main goal of Gaussian processes is to learn the underlying distribution of training data. Key to this process is the utilization of *Bayesian inference*, in which one assumes a prior and updates this hypothesis based upon new data. When modeling using Gaussian processes, a prior with dimensionality equal to that of the unseen points is chosen. A *kernel* is used to generate the covariance matrix Σ by evaluating it on all training points.

In order to form the posterior distribution (i.e. train the model), the model observes training data, and conditions the current distribution based upon these new points. As new data comes in, the set of functions that the model can take is constrained, as only those functions exactly containing the new points are valid. The change in distribution induced by observing new points is reflected in an adjustment of the mean and standard deviation (achieved through marginalization). In addition, uncertainty in the data is modeled by adding an error term to the training points, modeled by $\epsilon \sim \mathcal{N}(\mu, \psi^2)$.

Predictions from a trained Gaussian process model are made simply by sampling from the distribution of the model. In this way, Gaussian processes interestingly combine the ability to model (understand) the underlying distribution of the data at hand, as well as make accurate predictions from unseen instances. They, therefore, present a promising, powerful, and efficient method for tackling regression tasks (Williams and Rasmussen, 1995).

4 XLM-RoBERTa for Regression

Large pre-trained language models (PLMs) based on the transformer architecture (Vaswani et al., 2017) have achieved state-of-the-art performance on a wide array of common NLP tasks. Devlin et al. (2019) introduced BERT as a powerful language representation model that learns deep contextual representations of words. RoBERTa (Liu et al., 2019) is a robustly optimized extension of the BERT model. Both of these models work primarily on English text, so we decided to use the multilingual model XLM-RoBERTa (XLM-R), a variation of RoBERTa trained on data written in one hundred languages (Conneau et al., 2020). This model can recognize the language of an unseen textual input and achieves remarkable performance on a variety of non-English tasks, beating even the monolingual models optimized for specific languages.

The key benefit of using PLMs is the ability to load the already pre-learned contextual word embeddings and then to fine-tune them for the specific downstream task at hand. We tried out different pre-trained models for XLM-R and the best-performing ones for this task were the original *xlm-roberta-base*, *twitter-xlm-roberta-base*, and *xlm-roberta-base-wikiann-ner*. While PLMs like XLM-R are more commonly used for classification tasks, they can also be adapted for regression tasks. We achieved this by adding a new linear layer on top of the XLM-R. This linear layer had as its input the outputs of the final (12th) layer of XLM-R and learned what weights to assign to them.

Since our dataset contains only around 1000 examples, the process of fine-tuning had to be carried out carefully in order to prevent overfitting. Hyperparameters used were: number of folds 5, number of epochs 3, batch size 16, max. length of 100, no weight decay. The starting learning rate was 10^{-5}, after one third of all layers $5 \cdot 10^{-5}$, and after two thirds it was 10^{-4}. The idea behind this was that lower encoder layers can be understood as learning the lexical and syntactical features of the text, whereas higher layers model the semantic representation of it. For the task of text complexity, low-level features are more important so more emphasis was placed on them.

5 Feature Selection

The complete set of crafted features for model training is listed in Table 1. These features are separated into categories, followed a brief description, with supporting notes at the bottom. The character-, token-, and POS-based features were inspired by Falkenjack et al. (2013) and Chatzipanagiotidis et al. (2021). Before feature creation, preprocessing included stopword removal and lemmatization.

	Feature	Description
Character-based*		
	Avg_chars	Average number of characters per token in sentence
	Tokens_*N*	Number of tokens in sentence with length $> N$[1]
Token-based*		
	Type_token	Distinct number of token *types*
	Carroll_TTR	Carroll's Corrected TTR measure
Common Words*		
	Num_common	Number of tokens found in the top 500 most common German words[2]
	Common_score	Cumulative score based upon rank in top 500 list
Sentence Attributes*		
	Sentence_length	Length of sentence, i.e. number of tokens
	Longest_word	Length of the longest word in a sentence
	Commas	Number of commas in the sentence
	Parentheses	Number of (open) parentheses characters
	Digits	Number of numerical digits in the sentence
	Quotes	Number of quotation characters (ór ̈) in the sentence
	Avg_word_length	Average length of words in the sentence
	Wordrank_score	Overall score calculated from German Wiki frequency list[3]
POS Tags*		
	POS_ratio	Ratio of (spaCy.pos_) POS Tags in sentence [4]
	TAG_ratio	Ratio of (spaCy.tag_) detailed POS Tags in sentence[5]
spaCy Features*		
	Dep_length	Cumulative length (width) of dependencies in sentence
	Ne_length	Total length of all named entities in sentence
	Ne	Number of named entities in sentence
	L2_norm	L2 Norm of spaCy word vector representations
	Vec_exists	Number of sentence tokens for which a spaCy vector exists
Syntax Tree		
	Syn_height	Height of syntax tree
	Leaves	Number of leaves in syntax tree
	Subtrees	Number of subtrees in syntax tree
	Leaf_distance	Cumulative distance between the leave nodes in the sentence
Syllables		
	Tot_syl	Total number of syllables in sentence
	Avg_syl	Average number of syllables per word
	Single_syl	Number of single syllable words in sentence
Readability[6]		
	Flesch	Flesch reading ease score
	Flesch_mod	Modified Flesch score
	Easy_words	Number of words in sentence with ≤ 2 syllables
	Hard_words	Number of words in sentence with > 2 syllables
	Gunning_fog	Gunning Fog readability index
	Mod_smog	SMOG readability index
	Mod_forcast	Forcast readability formula
	Ari	Automated readability index
	Linsear	Linsear write readability metric
WordNet[7,†]		
	Synset_exists	Number of lemmas in sentence for which a synset exists
	Synset_depth	Cumulative maximum depth of all existing synsets in sentence
	Hyponyms	Cumulative number of hyponyms for existing synsets
	Senses	Cumulative number of word senses for existing synsets
	Syn_def	Total length of synset definitions for all existing synsets
	Avg_path	Average path length from one synset to the next, in sequential order
Experimental†		
	Scrabble_new	Scrabble score using the new German Scrabble point values
	Scrabble_old	Scrabble score using the old German Scrabble point values

[1] for $N \in \{2, 6, 7, 8, 10\}$

[2] https://www.thegermanprofessor.com/top-500-german-words/

[3] https://github.com/gambolputty/dewiki-wordrank

[4] for POS ∈ {'ADJ', 'ADP', 'ADV', 'AUX', 'NOUN', 'NUM', 'PRON','PROPN', 'VERB', 'X'}

[5] for TAG ∈{'ADJA', 'ADJD', 'ADV', 'APPR', 'ART', 'KON', 'KOUS', 'NN', 'PRELS', 'VAFIN', 'VVFIN', 'VVPP'}

[6] Where applicable, scores are modified for single sentences (denoted by *mod*)

[7] Using the *Open German WordNet*: https://github.com/hdaSprachtechnologie/odenet

[*] Features in these categories are calculated on the logarithmic scale, with either add-1 or add-0.1 smoothing, where necessary

[†] These features were not used in the final model (best test score)

Table 1: Feature Set

6 A Hybrid System

The development of the eventual final model took place in an iterative fashion. First, an array of popular shallow models were tested. In this process, the discovery of the effectiveness of Gaussian processes for this specific task led the authors to choose these particular models for tuning. The kernel used was the sum of Constant, Matern, and White kernels, optimized with 10 restarts. As the training of Gaussian process models seemed to hit a plateau, a deeper approach was pursued, namely using RoBERTa. This achieved good results (see Section 7), leading the authors to believe that some deep component was key to the task at hand.

Due to the documented success of stacking and ensemble methods (Pavlyshenko, 2018; Ganaie et al., 2021), the authors considered a third approach in which the best shallow and deep models (GPs and RoBERTa) were to be stacked. Concretely, the predictions of the two models could be harmonized in a way that combines the strengths of both. Traditionally, stacking is performed by training a "meta-model", which learns the optimal way to combine the outputs of the "level 0" models.

With this in mind, the authors took a simplified approach to stacking, in which the output predictions of the Gaussian process model and the fine-tuned XLM-RoBERTa were simply averaged. This resulted in the "meta" predictions, which were then used for submissions. In the development phase, this method proved to be the most effective, far outperforming both individual models. As such, a hybrid system was created, which was later utilized in the test phase. Results from the development phase are outlined in Section 7, where the performance of the individual and hybrid models are displayed.

7 Training and Results

In the following Table 2, we present the results from the development phase of the challenge. In particular, we include both the traditional Root Mean Squared Error (RMSE) for each model, as well as the *RMSE_Mapped* metric used for this specific task. Since the MOS of human annotators inherently includes subjective biases and offsets, some statistical uncertainty is always present in the scores (Yi et al., 2022). Therefore, a linear mapping function is applied to the RMSE in order to compensate for the possible variance between several subjective experiments. It should be noted that the specifics on how to calculate this mapped metric were not provided for this challenge.

Model	**RMSE**	**RMSE_mapped**
Lasso Regression	–	0.515
Ridge Regression	–	0.507
XGBoost Regression	0.520	0.490
Partial Least Sq. Regression	0.492	0.462
LightGBM Regression	0.465	0.434
Random Forest Regression	0.457	0.427
Gaussian Process Regression	0.453	0.401
w/ 50% train data	0.447	0.380
+ 20-dim PCA	0.442	0.377
+ noisy targets	0.427	**0.373**
XLM-RoBERTa (SQuAD 2.0)	0.443	0.442
XLM-RoBERTa (WikiAnn)	0.434	0.424
XLM-RoBERTa (Twitter)	0.434	0.420
w/ 70% train data	0.430	0.393
XLM-RoBERTa (Base)	0.426	0.415
w/ 150 features	0.438	0.403
w/ 20-dim PCA	0.440	0.399
w/ 70% train data	0.433	**0.384**
XLM-R (0.415) + GP (0.377)	0.395	0.349
XLM-R (0.399) + GP (0.377)	0.415	0.342
XLM-R (0.384) + GP (0.373)	0.397	0.331
XLM-R (0.384) + GP (0.377)	0.408	0.328
XLM-R (0.393) + GP (0.373)	0.394	0.328
XLM-R (0.393) + GP (0.377)	0.401	**0.324**

Table 2: Development Phase Results

In Table 2, bolded are the best-performing shallow and deep models, as well as the best-performing stacked model, which did not use either of the two best single models. The most effective shallow model was the Gaussian process regressor that used our handcrafted features described in Section 5 and Table 1 to learn the optimal distribution over the training data. Only 50% of training data was randomly selected and used to train the model since this provided the optimal performance, as measured by the mapped RMSE metric.

Model	**RMSE**	**RMSE_mapped**
XLM-R (Base, 70% train) + GP (20-dim PCA, 50% train)	0.514	0.489
XLM-R (WikiAnn, 70% train) + GP (20-dim PCA, 50% train)	0.518	0.488
XLM-R (Twitter, 70% train) + GP (20-dim PCA, 50% train)	0.518	0.465
XLM-R (Base + 20-dim PCA) + GP (20-dim PCA, 50% train)	0.473	0.459
XLM-R (Twitter, 60% train) + GP (20-dim PCA, 50% train)	0.485	**0.457**

Table 3: Final Phase Results

The best-performing deep model was the base model of XLM-RoBERTa. Although adding our handcrafted features to it improved the performance, the trick of using a reduced training data set again provided us with the best results (70% of the training data). For the final stacked model, various combinations of GP and XLM-R models were tried out. The optimal combination turned out to be the XLM-RoBERTa fine-tuned on a Twitter dataset and the Gaussian Process using a 20-dimensional PCA representation of handcrafted features. This hybrid model achieved the mapped RMSE score of 0.324, which was the winning score (1st place) of the development phase of the competition.

Table 3 shows the results of the models submitted for the final phase of the competition. The authors decided to submit the best-performing models from the previous phase. The best-scoring model was later revealed to be the stacked combination of the XLM-RoBERTa pre-trained on Twitter and the Gaussian process with 20-dimensional PCA features, both models trained on reduced data. This came as no surprise as this was also the best-performing model in the previous phase. The model achieves the mapped RMSE score of 0.457, which resulted in 6th place in the final phase.

8 Discussion

Here, the authors reflect on lessons learned, useful findings, and possible future directions.

A useful and somewhat surprising finding came with the excellent performance of Gaussian processes, particularly during the development phase. In pondering why this occurred, one can look to the nature of GPs in conjunction with the specifics of the text complexity task. At their core, Gaussian process models aim to capture the underlying distribution of data that is complex. Furthermore, GPs seem to shine when the amount of training data is relatively small, e.g. under 1000 instances. As such, GPs may have been a logical choice for this task, which comprised of a quite small dataset and whose goal represents a quite complex regression task. Indeed, the quantification of text complexity proves to be challenging to reason about. Nevertheless, the effectiveness shown by GPs merits their consideration in future, related tasks.

Regarding XLM-RoBERTa, it cannot be denied that its inclusion greatly strengthened the final model. It is interesting, though, that the model performance quite significantly varied based upon the particular pre-trained model that was chosen. In this light, a potential future direction would involve further investigation into better models, as well as *why* these might be superior. Similarly, a more focused tuning of the hyperparameters for the fine-tuning process could have been performed. This likewise remains as future work.

In the creation of the hybrid model which was eventually used in the test phase, an interesting lesson was learned regarding how to “stack” our two best models. A more systematic way of doing so would be to learn a meta-model, i.e. a simple linear regression model, to stack on top of our GP + RoBERTa hybrid. As it turns out, learning such a model actually performed worse than a simple average of the two models’ predictions. In this way, simplicity won the day in regards to the design of a well-performing hybrid regression model.

The analysis of our feature set also produces interesting insights. As a cursory analysis, heatmaps illustrating the correlation amongst features and to the target values are presented in Figures 1 and 2 in the Appendix. The question then becomes whether more features should have been included, particularly those with a firm linguistic foundation.

This notion is grounded in the authors’ initial intuition that was used to produce the feature set. Interesting findings were gained here, too, such as the relatively high correlation of complex linguistic concepts (e.g. passive voice, genitive case, depth of syntax tree) to the MOS. We pose that such linguistic thinking is important for further improvements.

As already alluded to, the complexity of the data itself, or rather the ability to describe it effectively via features, proved to be a central problem in tackling the task at hand. In the process of studying the datasets, the authors noticed particular characteristics that could be crucial to feature creation. Of these, notable observations include the presence of rare-occurring symbols (e.g. §), as well as sentences that are clearly “simplified” sentences, rather than sourced directly from Wikipedia. Possible discrepancies between the dev and test sets were also observed. Accounting for such factors in the features is likely key to model performance.

9 Conclusion

In this paper, we discuss our novel approach to the quantification of text complexity in the German language. In particular, we present a hybrid model using Gaussian Processes and a fine-tuned XLM-RoBERTa. We also provide our full feature set, which to be best of the authors’ knowledge includes features not previously presented in the literature. Finally, we discuss our results in the challenges and reflect upon their implications for future work.

References

Savvas Chatzipanagiotidis, Maria Giagkou, and Detmar Meurers. 2021. Broad linguistic complexity analysis for greek readability classification. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 48–58.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics 2019*, pages 4171–4186. Association for Computational Linguistics.

Johan Falkenjack, Katarina Heimann Mühlenbock, and Arne Jönsson. 2013. Features indicating readability in swedish text. In *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*, pages 27–40.

Mudasir A Ganaie, Minghui Hu, et al. 2021. Ensemble deep learning: A review. *arXiv preprint arXiv:2104.02395*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

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

Babak Naderi, Salar Mohtaj, Kaspar Ensikat, and Sebastian Möller. 2019. Subjective assessment of text complexity: A dataset for german language. *arXiv preprint arXiv:1904.07733*.

Bohdan Pavlyshenko. 2018. Using stacking approaches for machine learning models. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pages 255–258. IEEE.

Carl Edward Rasmussen. 2003. Gaussian processes in machine learning. In *Summer school on machine learning*, pages 63–71. Springer.

Raquel Urtasun and Trevor Darrell. 2007. Discriminative gaussian process latent variable model for classification. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, page 927–934, New York, NY, USA. Association for Computing Machinery.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Christopher Williams and Carl Rasmussen. 1995. Gaussian processes for regression. *Advances in neural information processing systems*, 8.

Gaoxiong Yi, Wei Xiao, Yiming Xiao, Babak Naderi, Sebastian Möller, Wafaa Wardah, Gabriel Mittag, Ross Cutler, Zhuohuang Zhang, Donald S. Williamson, Fei Chen, Fuzheng Yang, and Shidong Shang. 2022. Conferencingspeech 2022 challenge: Non-intrusive objective speech quality assessment (NISQA) challenge for online conferencing applications. *CoRR*, abs/2203.16032.

A Appendix

Figure 1 shows a correlation heatmap of (selected) features. Figure 2 shows the correlation of these selected features to the target MOS score.

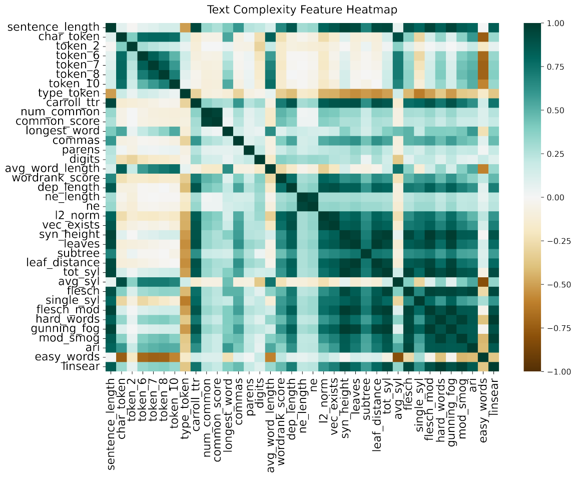

Figure 1: Feature - Feature Correlation Heatmap

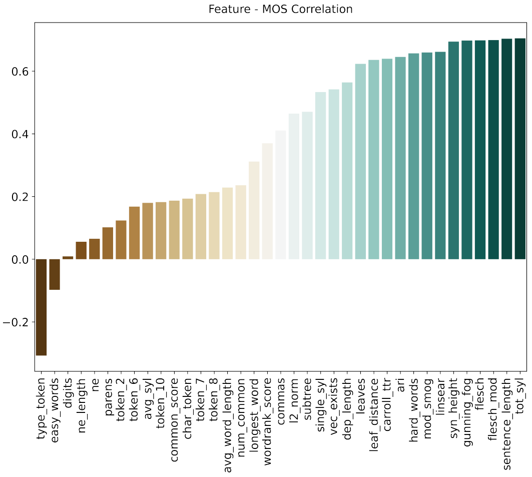

Figure 2: Feature - MOS Correlation Heatmap

Automatic Readability Assessment of German Sentences with Transformer Ensembles

Patrick Gustav Blaneck[1,†], Tobias Bornheim[1,2,†], Niklas Grieger[1,3,†] and Stephan Bialonski[1,3,*]

[1] *Department of Medical Engineering and Technomathematics*
FH Aachen University of Applied Sciences, Jülich, Germany

[2] *ORDIX AG – Team Data Science*

[3] *Institute for Data-Driven Technologies*
FH Aachen University of Applied Sciences, Jülich, Germany
[*] *bialonski@fh-aachen.de*, [†] *Equal contribution*

Abstract

Reliable methods for automatic readability assessment have the potential to impact a variety of fields, ranging from machine translation to self-informed learning. Recently, large language models for the German language (such as GBERT and GPT-2-Wechsel) have become available, allowing to develop Deep Learning based approaches that promise to further improve automatic readability assessment. In this contribution, we studied the ability of ensembles of fine-tuned GBERT and GPT-2-Wechsel models to reliably predict the readability of German sentences. We combined these models with linguistic features and investigated the dependence of prediction performance on ensemble size and composition. Mixed ensembles of GBERT and GPT-2-Wechsel performed better than ensembles of the same size consisting of only GBERT or GPT-2-Wechsel models. Our models were evaluated in the *GermEval 2022 Shared Task on Text Complexity Assessment* on data of German sentences. On out-of-sample data, our best ensemble achieved a root mean squared error of 0.435.

1 Introduction

Automatic Readability Assessment (ARA) is a well-known challenge in natural language processing (NLP) research (Martinc et al., 2021; Vajjala, 2021; Collins-Thompson, 2014). Systems for reliable readability assessment have the potential to support readers with learning disabilities, inform self-directed learning, or help control the reading level of automatically generated text translations (Vajjala, 2021).

The development of methods for text readability assessment may be described in three phases. (i) Traditional text readability formulas were based on statistical measures of lexical and syntactic features (such as word difficulty and length). Techniques from NLP further improved upon traditional formulas by incorporating high-level textual features such as semantic and discursive text characteristics (Martinc et al., 2021). (ii) In the early 21st century, engineered linguistic features were used to train shallow classifiers and regressors from machine learning (such as support vector machines and decision trees) which further improved prediction accuracy (Collins-Thompson, 2014). (iii) The latest phase has been characterized by the advent of large language models (LLMs) developed in the Deep Learning community. Such neural networks learn features (vector representations of text) automatically from large text corpora during self-supervised pretraining. Successful network architectures such as BERT (Devlin et al., 2019; Rogers et al., 2020) or GPT (Radford and Narasimhan, 2018; Radford et al., 2019; Brown et al., 2020) closely follow the influential transformer model (Vaswani et al., 2017) that allows for efficient modeling of long-range correlations in texts. By combining representations derived from BERT with linguistic features, recent studies observed increased accuracy in assessing the readability of English texts (Lee et al., 2021; Imperial, 2021).

Training large language models requires large text corpora, a prerequisite that is difficult to meet in languages with fewer resources (compared to English) such as German. Thus, most approaches to assess the readability of German texts have been based on linguistic features and traditional models from statistical learning such as polynomial regression, support vector machines, or random forests (Hancke et al., 2012; Weiß and Meurers, 2018; Naderi et al., 2019b; Weiß et al., 2021).

Only recently, large language models have become available for German, most notably GBERT (Chan et al., 2020), which is based on BERT, and GPT-2-Wechsel (Minixhofer et al., 2021) which was derived from the English GPT-2 model (Radford et al., 2019). It is largely unknown to which extent these German language models can improve the automatic readability assessment of German texts.

In this contribution, we investigate the ability of ensembles of GBERT and GPT-2-Wechsel models to assess the readability of German sentences. We combine these models with traditional linguistic features and evaluate our approach on a recently published dataset of German sentences (Naderi et al., 2019a). Inspired by previous work on ensembling large language models (Risch and Krestel, 2020; Bornheim et al., 2021), we studied the dependence of model accuracy on the number of ensemble members and ensemble composition. Finally, we describe the models that were evaluated in the *GermEval 2022 Shared Task on Text Complexity Assessment* (Mohtaj et al., 2022). The implementation details of our experiments (Team "AComplexity") are available online[1].

2 Data and tasks

The dataset consisted of 1000 labeled sentences (Naderi et al., 2019a) and was provided by the organizers of the *GermEval 2022 Shared Task on Text Complexity Assessment* (Mohtaj et al., 2022). The sentences were drawn from 23 Wikipedia articles. 250 of these sentences were manually simplified by native German speakers (Naderi et al., 2019a).

The scores (labels) were obtained via an online survey system. Participants were asked to rate the complexity, understandability, and lexical difficulty of the sentences on a 7-point Likert scale. On this scale, 1 denotes the lowest and 7 the highest possible value (Naderi et al., 2019a). In total, 10650 valid sentence ratings were collected, distributed among the 1000 sentences.

Following a data screening procedure, 5 to 18 ratings per sentence were deemed valid and then used to calculate the arithmetic mean, called the *Mean Opinion Score (MOS)*, of each metric (Naderi et al., 2019a).

The shared task was to predict the MOS of the text complexity of German sentences. Since the MOS was defined as a decimal value (see figure 2), we approached this task as a regression problem. The distribution of complexity scores (see figure 1) suggests that complex sentences are much less common within the dataset than simpler ones. Following previous work, we considered text complexity as a proxy of text readability (Wray and Janan, 2013).

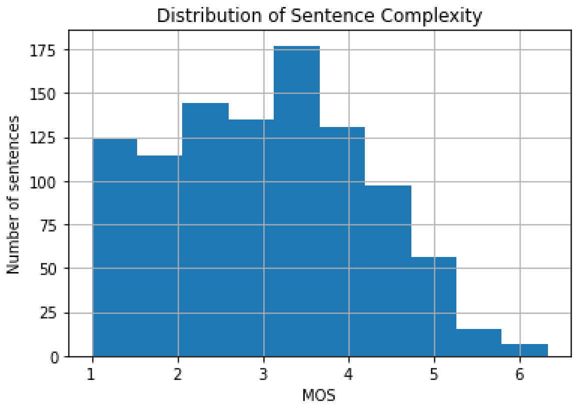

Figure 1: Histogram of Mean Opinion Scores (MOS) for the sentences in the dataset.

Bei der Tour de France liegt die höchste Durchschnittsgeschwindigkeit eines Fahrers bei 41 km/h. (MOS: 1.5)

Für die Union resultiert daraus sowohl ein Akzeptanzproblem bei den EU-Bürgern, denen "Brüssel" immer undurchsichtiger erscheint, als auch die mit dem Mitgliederwachstum verbundene Schwierigkeit, im bestehenden Institutionengefüge die Arbeits- und Handlungsfähigkeit der einzelnen Organe zu gewährleisten. (MOS: 6.33)

Figure 2: Samples (German sentences) from the dataset of the *GermEval 2022 Shared Task on Text Complexity Assessment*. Numbers in parentheses denote text complexity scores.

3 Methods

3.1 Preprocessing and data splits

Preprocessing. All datasets (training, validation, and test data) were preprocessed in the same way. First, we cleaned up all sentences by removing the leading and trailing quotation marks that were added by the CSV format to mask sentences containing comma separators. In the next step, all sentences were tokenized with model-specific tokenizers and padded to a uniform length of 128 tokens.

Data splits. During the model exploration phase,

[1] https://github.com/dslaborg/tcc2022

models were evaluated with a 5-fold cross validation scheme (each of the five folds contained 20% of the randomly shuffled training data). Additionally, we randomly selected 10% of the data in the training folds (i.e., 8% of the whole training data) as an *early stopping set* (see section 3.4). Thus, all models in the model exploration phase were trained on 72% of the training data.

To optimize model fitting, the final models that were submitted to the *GermEval 2022 Shared Task on Text Complexity Assessment* were retrained on all available training data, aside from a small dataset that was used for *early stopping*. The *early stopping set* consisted of 7.5% of the training data and consequently, all final models were trained on 92.5% of the training data.

3.2 Readability Features

We incorporated various traditional features in the training of our models that are commonly used in text readability and complexity assessment tasks. The features were generated using two publicly available libraries (van Cranenburgh, 2019; Proisl, 2022) and include simple sentence-based measures such as sentence length and punctuation as well as more complex measures such as word rarity. Furthermore, we included some customized features based on the number of words in a sentence that exceed a given amount of characters. To increase the amount and variety of the available features, we translated all sentences to English and calculated the features for the original German sentences as well as the English translations. In total, 154 features were created for each sentence.

3.3 Models

We studied two German language models. The GBERT model (Chan et al., 2020) is based on the BERT architecture (Devlin et al., 2019). We used model weights of the pretrained *gbert-large*[2] variant, which includes a tokenizer with a vocabulary size of 31000 case-sensitive tokens, has approximately 336 million parameters and a hidden state size of 1024. Each tokenized sentence was prepended with a classification token that was used for the *next sentence prediction* task during pretraining (Devlin et al., 2019).

The second model is a German GPT-2-Wechsel model (Minixhofer et al., 2021) based on the GPT-2 architecture (Radford et al., 2019). We used model weights of a pretrained *gpt2-xl-wechsel-german*[3] variant that was derived from the GPT-2-XL[4] model (Radford et al., 2019) using the WECHSEL method (Minixhofer et al., 2021). The tokenizer has a vocabulary size of 50000 case-sensitive tokens, while the model has roughly 1.5 billion parameters and a hidden state size of 1600. Since GPT-like models are usually not used for regression tasks, we needed to adjust the tokenizer as follows. First, we introduced a padding token that was used to pad all sentences to a uniform length of 128 tokens (see section 3.1). Second, we put a *beginning of sequence* token in front and added an *end of sequence* token to the end of every tokenized sentence.

For each transformer model, we employed two different multi-layer perceptron models (MLP) as regression heads. The first MLP was used to finetune the transformer models on the given training data and did not use the manually created readability features (see section 3.2). The second MLP was used after finetuning to incorporate the readability features and consisted of a fully connected layer, followed by ReLu activations and an output layer with one neuron and a linear activation for regression. The input vector for the second MLP consisted of the output of the last hidden state of the respective transformer model and 154 readability features calculated for each sentence.

3.4 Training

Evaluation score. To assess the prediction performance of each model, we calculated the *root mean squared error* (RMSE),

$$\text{RMSE} = \sqrt{\tfrac{1}{N} \textstyle\sum_{i=1}^{N} (\hat{y}_i - y_i)^2},$$

where y_i denotes the true readability score, $\hat{y}_i$ the predicted readability scores, and N the number of samples in the dataset. During model exploration, the RMSE was determined for each validation fold of the 5-fold cross validation scheme. We considered the average of these RMSE values as an indicator of model performance.

Training scheme. The training was carried out in two phases. In the first phase, we added a regression head to each model, used an AdamW optimizer (Loshchilov and Hutter, 2019) with a batch

[2] https://huggingface.co/deepset/gbert-large

[3] https://huggingface.co/malteos/gpt2-xl-wechsel-german

[4] https://huggingface.co/gpt2-xl

size of 16 and a learning rate of $\eta = 5 \cdot 10^{-5}$ with a linear warmup on the first 30% of the training steps from 0 to η. About every half training epoch (every 23 gradient updates during model exploration or every 28 gradient updates when training the submitted models), the models were evaluated on the *early stopping set*. If the training lasted for 100 epochs or the RMSE did not decrease for five consecutive evaluations, the training was stopped and the model with the lowest RMSE on the *early stopping set* was returned. This stopping mechanism was not used during the first 300 gradient updates of the training to prevent underfitting.

In the second phase of the training, the regression heads were discarded and the output of the last hidden state for each sentence of the dataset was extracted as follows. For GBERT, we used the output of the classification token. For GPT-2-Wechsel, we extracted the output of the *end of sequence* token. To create a feature vector for each sentence, we combined the output of the respective transformer model with the readability features calculated for each sentence. We trained a multi-layer perceptron (MLP) with two layers (see 3.3) with the RMSprop optimizer, a batch size of 16, and a constant learning rate of $\eta = 10^{-3}$. The MLPs were evaluated on the *early stopping set* after each training epoch. After 5000 epochs or if the RMSE did not decrease for 100 consecutive epochs, the training was stopped and the model with the lowest RMSE on the *early stopping set* was returned.

During inference, to predict a score for a given sentence, a feature vector was created by combining readability features with the output of the fine-tuned transformer model. The feature vector served as an input to the trained MLP which calculated the readability score.

Loss functions. We used the *mean squared error loss* for training all transformer models and MLPs.

3.5 Ensembling

To counteract the effects of overfitting that often occur when training large models on small datasets, we combined our trained models in ensembles (Risch and Krestel, 2020; Bornheim et al., 2021). Ensemble members differed in the initial model weights of the regression heads and the randomly selected *early stopping set*. We determined the predictions of an ensemble by averaging the predicted scores of the ensemble members.

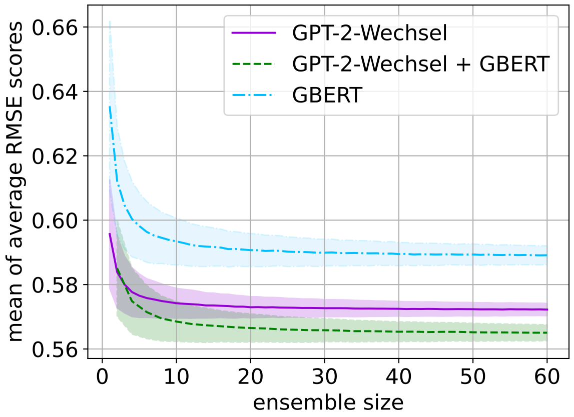

Figure 3: Dependence of the mean of the average root mean squared error (lines) on ensemble size for different ensemble compositions. Standard deviations are shown as shaded areas.

3.6 Postprocessing

When evaluating our ensembles on the provided test set during the final phase of the competition, we found that some trained models predicted readability scores smaller than 1.0 for a few sentences in the test set. Since the 7-point Likert scale used by the human annotators to score text readability started at a value of 1.0 (see section 2), we deemed all predicted values smaller than 1.0 as invalid and removed them in the ensembling process. Thus, the predictions of an ensemble were created by averaging only the predicted scores larger than 1.0. We hypothesize that the scores smaller than 1.0 on the test data were caused by a distribution shift in the generated readability features.

4 Results

Model exploration. During model exploration we investigated the performance (measured by the average RMSE) of ensembles with different ensemble sizes and compositions. The ensembles consisted of 1 to 60 models in three different compositions: (i) GBERT models only, (ii) GPT-2-Wechsel models only, (iii) a combination of GBERT and GPT-2-Wechsel models. In (iii), we combined both model types equally, so that an ensemble of 60 models consisted of 30 GBERT and 30 GPT-2-Wechsel models.

To investigate the dependence of prediction performance on ensemble size, we applied a bootstrapping scheme following (Risch and Krestel, 2020; Bornheim et al., 2021). In total, we trained 100 models each of GBERT and GPT-2-Wechsel on each cross-validation split. Given a specific ensemble size, we then randomly sampled with replacement 1000 ensembles from the set of trained models and measured the RMSE of each ensemble on each validation fold. The attained RMSE scores were then averaged over the 5 validation folds, so that we obtained 1000 averaged RMSE scores for each ensemble size.

Figure 3 shows the mean and standard deviation of the averaged RMSE scores for different ensemble sizes and compositions. Each ensemble composition benefited from increasing ensemble size, as the mean RMSE decreased considerably up to an ensemble size of 20 models, beyond which the RMSE decreased only slowly. Increasing the ensemble size also affected the stability of the ensembles' predictions, as can be observed from the decreasing standard deviation of all three ensemble compositions. Our findings are consistent with previous work (Risch and Krestel, 2020; Bornheim et al., 2021) which reported improvements in predictive performance when increasing ensemble sizes.

Furthermore, figure 3 shows large differences in the performance of the three ensemble compositions. The ensemble that consisted of only GBERT models performed the worst with a mean RMSE of 0.589 at ensemble size 60. Using GPT-2-Wechsel models instead of GBERT models reduced the mean RMSE to 0.572, and combining both model types in a mixed ensemble of 30 GPT-2-Wechsel and 30 GBERT models further improved the scores to 0.565.

Submitted models. Based on our results in the model exploration phase, we decided to submit two different ensembles in the final phase of the competition: (i) an ensemble of 340 GPT-2-Wechsel models and (ii) an ensemble of 100 GPT-2-Wechsel and 100 GBERT models. We chose not to submit an ensemble of only GBERT models due to the subpar performance observed during model exploration. All models were fine-tuned using all available training data, aside from a small dataset (7.5% of the training data) used for early stopping (see section 3).

On the test data of the shared task, ensembles (i) and (ii) achieved RMSE values of 0.461 (mapped RMSE: 0.454[5]) and 0.442 (mapped RMSE: 0.435[5]), respectively (Mohtaj et al., 2022). Ensemble (ii) ranked 2nd in the competition.

5 Conclusion

We studied the ability of ensembles of fine-tuned German language models to reliably predict the readability of German sentences. All proposed models also used traditional linguistic features that slightly increased prediction performance (data not shown), consistent with previous reports on text readability assessment of English texts (Imperial, 2021; Lee et al., 2021). We observed mixed ensembles of GBERT and GPT-2-Wechsel to better predict readability scores than ensembles of the same size consisting of only GBERT or GPT-2-Wechsel models. Furthermore, prediction accuracy as quantified by the *root mean squared error* decreased with increasing ensemble size, which resembled findings for hate speech classification reported previously (Risch and Krestel, 2020; Bornheim et al., 2021).

[5] A linear mapping function was used by the competition organizers; see section 7.3 of the recommendation ITU-T P.1401.

References

Tobias Bornheim, Niklas Grieger, and Stephan Bialonski. 2021. FHAC at GermEval 2021: Identifying german toxic, engaging, and fact-claiming comments with ensemble learning. *CoRR*, abs/2109.03094.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annu. Conf. on Neural Information Processing Systems 2020, NeurIPS 2020*, Virtual.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proc. 28th Int. Conf. on Computational Linguistics, COLING 2020*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kevyn Collins-Thompson. 2014. Computational assessment of text readability: A survey of current and future research. *Int. J. Appl. Linguistics*, 165(2):97–135.

Andreas van Cranenburgh. 2019. Readability. `https://github.com/andreasvc/readability/releases/tag/v0.3.1`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. 2019 Conf. North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, volume 1, pages 4171–4186, Minneapolis, MN, USA. Association for Computational Linguistics.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability classification for German using lexical, syntactic, and morphological features. In *COLING 2012, 24th Int. Conf. on Computational Linguistics, Proc. Conf.: Technical Papers*, pages 1063–1080, Mumbai, India. Indian Institute of Technology Bombay.

Joseph Marvin Imperial. 2021. BERT embeddings for automatic readability assessment. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021), Held Online, 1-3September, 2021*, pages 611–618. INCOMA Ltd.

Bruce W. Lee, Yoo Sung Jang, and Jason Hyung-Jong Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proc. 2021 Conf. on Empirical Methods in Natural Language Processing, EMNLP 2021*, pages 10669–10686, Punta Cana, Dominican Republic (Virtual). Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th Int. Conf. on Learning Representations, ICLR 2019*, New Orleans, LA, USA.

Matej Martinc, Senja Pollak, and Marko Robnik-Šikonja. 2021. Supervised and unsupervised neural approaches to text readability. *Comput. Linguist.*, 47(1):141–179.

Benjamin Minixhofer, Fabian Paischer, and Navid Rekabsaz. 2021. WECHSEL: Effective initialization of subword embeddings for cross-lingual transfer of monolingual language models. *CoRR*, abs/2112.06598.

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

Babak Naderi, Salar Mohtaj, Kaspar Ensikat, and Sebastian Möller. 2019a. Subjective assessment of text complexity: A dataset for German language. *CoRR*, abs/1904.07733.

Babak Naderi, Salar Mohtaj, Karan Karan, and Sebastian Möller. 2019b. Automated text readability assessment for German language: A quality of experience approach. In *11th Int. Conf. on Quality of Multimedia Experience QoMEX 2019*, pages 1–3, Berlin, Germany. IEEE.

Thomas Proisl. 2022. Linguistic and stylistic complexity. `https://github.com/tsproisl/textcomplexity/releases/tag/v0.11.0`.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pretraining.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Julian Risch and Ralf Krestel. 2020. Bagging BERT models for robust aggression identification. In *Proc. 2nd Workshop on Trolling, Aggression and Cyberbullying, TRAC@LREC 2020*, pages 55–61, Marseille, France. European Language Resources Association (ELRA).

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in BERTology: What we know about how BERT works. *Trans. Assoc. Comput. Linguistics*, 8:842–866.

Sowmya Vajjala. 2021. Trends, limitations and open challenges in automatic readability assessment research. *CoRR*, abs/2105.00973.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Annual Conf. Neural Information Processing Systems 2017*, pages 5998–6008, Long Beach, CA, USA.

Zarah Weiß, Xiaobin Chen, and Detmar Meurers. 2021. Using broad linguistic complexity modeling for cross-lingual readability assessment. In *Proc. 10th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2021)*, Linköping Electronic Conference Proceedings 177, pages 38–54.

Zarah Weiß and Detmar Meurers. 2018. Modeling the readability of german targeting adults and children: An empirically broad analysis and its cross-corpus validation. In *Proc. 27th Int. Conf on Computational Linguistics, COLING 2018*, pages 303–317, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

David Wray and Dahlia Janan. 2013. Readability revisited? The implications of text complexity. *The Curriculum Journal*, 24:553 – 562.