DeepBlueAI at SemEval-2021 Task 1: Lexical Complexity Prediction with A Deep Ensemble Approach

Chunguang Pan Bingyan Song Shengguang Wang Zhipeng Luo
DeepBlue Technology (Shanghai) Co., Ltd
{songby, panchg, wangshg, luozp}@deepblueai.com

Abstract

Lexical complexity plays an important role in reading comprehension. lexical complexity prediction (LCP) can not only be used as a part of Lexical Simplification systems, but also as a stand-alone application to help people better reading. This paper presents the winning system we submitted to the LCP Shared Task of SemEval 2021 that capable of dealing with both two subtasks. We first perform fine-tuning on numbers of pre-trained language models (PLMs) with various hyperparameters and different training strategies such as pseudo-labelling and data augmentation. Then an effective stacking mechanism is applied on top of the fine-tuned PLMs to obtain the final prediction. Experimental results on the Complex dataset show the validity of our method and we rank first and second for subtask 2 and 1.

1 Introduction

Lexical complexity is one of the main reasons leading to overall text complexity and thus result in poor reading comprehension for readers (DuBay, 2004). Different from the Complex Word Identification (CWI) (Shardlow, 2014) task, which aims to predict whether a given word is complex or not, the goal of **lexical complexity prediction** (LCP) is to predict the complexity value of the given parts from contexts as shown in Figure 1. The underlined parts of the sentence are the words that need to be predicted and the same words in different contexts may have different complexity scores. LCP plays an important role in the usual Lexical Simplification (LS) (Bott et al., 2012) pipeline since it can help simplifiers find the challenging words and replace them with appropriate alternatives that easy to understand. Either LCP or CWI can not only be used as a component of LS systems but also as a stand-alone application within intelligent tutoring systems for second language learners or in reading devices for people with low literacy skills (Gooding and Kochmar, 2018).

Single word	Multi-words
Context1: They shall be to you for a **refuge** from the avenger of blood. *Complexity score:* 0.3475	*Context1:* SEM confirmed many of the observations made by **confocal microscopy**. *Complexity score:* 0.64473
Context2: There will be a pavilion for a shade in the daytime from the heat, and for a **refuge** and for a shelter from storm and from rain. *Complexity score:* 0.075	*Context2:* SJ and SVJ carried out **confocal microscopy** on whole-mounts of stria vascularis. *Complexity score:* 0.7750

Figure 1: Examples of LCP including single words and multi-words. The complexity score is the score for the underlined words.

In this paper, we introduce our system for the lexical complexity prediction task of the SemEval-2021 (Matthew et al., 2021). We fulfill this task by leveraging multiple pre-trained language models (PLM) with different training strategies. There are two main steps for our system: **(i)** fine-tuning numbers of heterogeneous PLMs, including BERT (Devlin et al., 2019), ALBERT (Lan et al., 2019), RoBERTa (Liu et al., 2019) and ERNIE (Zhang et al., 2019), with various hyperparameters and training strategies, obtaining diverse models; **(ii)** applying an effective stacking mechanism on top of these PLMs to predict the final complexity scores.

Our experiments, merging PLMs in total, indicate that our method successfully utilizes weaker PLMs as well as high-performing PLMs. As a result, our system ranks second and first for Subtask 1 and 2 of LCP 2021, SemEval-2021.

2 Related Work

2.1 Lexical Complexity Prediction

There has been some work for the creation and evaluation of automatically graded vocabulary lists

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 578–584
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

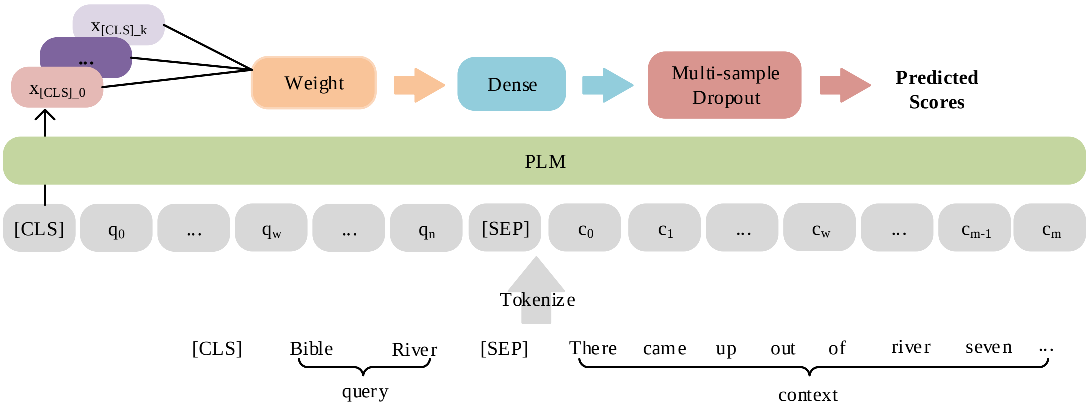

Figure 2: The overall architecture for predicting complexity scores.

for analyzing lexical complexity. François et al. (2014) present the first graded lexicon for French as a foreign language that reports word frequencies by difficulty level and Gala et al. (2014) train two SVM classifiers with 49 features, one for L1 learners and one for learners of French as a foreign language. Alfter and Volodina (2018) map the use of previously created word lists to a single CEFR scale (Common European Framework of Reference for Languages) (De l'Europe, 2003), then they add topics as additional features to predict the complexity level for learners of Swedish as a second language. Shardlow et al. (2020) point out the limitation of treating lexical complexity as a binary classification task. Therefore, they present the first English dataset for continuous lexical complexity prediction and develop a linear regression-based method with various features.

2.2 Complex Word Identification

A related area of LCP is CWI. Early studies on CWI either attempt to simplify all words (Thomas and Anderson, 2012) or set a frequency-based threshold (Biran et al., 2011). Shardlow (2013) indicates that a classification-based method to CWI is the most promising one. Most of the teams participating in two CWI shared tasks also use classification approaches with extensive feature engineering. In CWI 2016 (Paetzold and Specia, 2016a), complexity was defined as whether or not a word is difficult to understand for non-native English speakers and the words in the dataset are tagged as complex or non-complex by 400 non-native English speakers. The results highlight the effectiveness of Decision Trees (Quijada and Medero, 2016; Mukherjee et al., 2016) and Ensemble methods (Paetzold and Specia, 2016b; Malmasi et al., 2016) for the task. In CWI 2018 (Yimam et al., 2018), a multilingual dataset was provided containing English, German, Spanish and French and there were two subtasks: binary classification and probabilistic classification. The submitted systems mainly use traditional machine learning classifiers(e.g. SVM, Random Forests) with features (Butnaru and Ionescu, 2018; Kajiwara and Komachi, 2018), deep learning methods (Hartmann and Dos Santos, 2018; De Hertog and Tack, 2018) and ensemble methods (Gooding and Kochmar, 2018; Aroyehun et al., 2018). More recently, (Gooding and Kochmar, 2019) propose a new perspective by treating CWI as a sequence labeling task that can detect both complex words and phrases. All these methods are different from ours which utilizes heterogeneous PLMs with various training strategies.

3 Background

Task Definition There are two subtasks in the LCP task. For subtask 1, the goal is to predict the complexity score for a single word from the given context. As an example shown in Figure 1, the 'refuge' is the word that needs to be predicted and since the meaning of it is harder to get in the first context, its complexity score in the first context is much higher. For subtask 2, the goal is to predict the complexity score for a multi-word expression from the given context. An example is also shown in the right part of Figure 1.

Dataset Shardlow et al. (2020) introduce a new English corpus, Complex, as the dataset for the LCP task of SemEval-2021. Instead of assigning binary scores for lexical complexity, they use crowdsourcing to annotate 8979 instances covering three genres with lexical complexity scores using

Parameters	**BERTLARGE**	**ALBERTXXLARGE**	**RoBERTLARGE**	**ERNIELARGE**
batch_size	16	16	16	16
learning rate	5e-6	5e-6	5e-6	5e-6
hidden_layer	3	3	1, 3, 5	3, 5
dropout	0.2, 0.3	0.2, 0.3	0.2, 0.1; 0.2, 0.5;0.2, 0.3	0.2, 0.3; 0.2, 0.5
loss function	MSE	MSE	RMSE, MSE, MAE	MSE, MAE

Table 1: Parameter settings for different base models

a 5-point Likert scale: one for very easy, two for easy, three for neutral, four for difficult, and five for very difficult. The numerical labels were transformed to a 0-1 range as shown in Figure 1. To add further variation to the data, three corpora were selected including Bible, Europarl (Koehn, 2005) and Biomedical (Bada et al., 2012). Each corpus has its own unique language features and styles. In addition to single words, multi-word expressions were also selected for annotating. In the end, there were 9476 annotated contexts with 5166 unique words.

4 System

4.1 PLMs-based Method

PLMs such as BERT (Bidirectional Encoder Representations from Transformers) use the encoder structure of the Transformer (Vaswani et al., 2017) for deep self-supervised learning, which requires task-specific fine-tuning. In this paper, the downstream task is to predict the complexity scores, a real-value in the range of [0,1], of given words. Our method is capable of dealing with both subtask 1 and 2. Figure 2 shows the main architecture of our BERT-based model for predicting complexity scores.

Since PLMs can process multiple input sentences, we add a query sentence before the context to emphasize the words (e.g. river) that need to be predicted and the corpus (e.g. Bible) they come from. We add special tokens `[CLS]` and `[SEP]` to separate the query and the context as shown in Figure 2. BERT first tokenizes the input contents and then generates contextualized vector representations for each token in multiple hidden layers. We focus on the output of only the first position that we passed the special `[CLS]` token to. The last k hidden layers are selected to get the final representation of token `[CLS]` through a weighted calculation function as below,

$$x_{[CLS]} = \sum_{i=1}^{k} W_i x_{[CLS]i}$$

where W_i is the learning weight for each hidden layer. The calculated representation is then fed into a dense layer, and the technique of multi-sample dropout (Inoue, 2019) is utilized to accelerate training and finally obtain the predicted complexity scores. The loss function can be chosen among several options including Mean Square Error (MSE), Root Mean Square Error (RMSE), and Mean Absolute Error (MAE).

4.2 Training strategies

In order to further improve the diversity of trained models, we incorporate two training strategies as depicted below.

Pseudo-Labelling Pseudo-labelling is the process of using a labeled data model to predict labels for unlabeled data. We predict the unlabeled test dataset and mix these pseudo labels with the training set together to train the new model.

Data augmentation Data augmentation is the technique used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. It acts as a regularizer and helps reduce overfitting when training a machine learning model. In this paper, data augmentation consists of two parts. We first add the dataset released by CWI 2018 into the training set. Besides, for subtask 2, since its training dataset is small which only contains one thousand samples, we add the dataset of subtask 1 to train the model for subtask 2. Then, for a given sentence in the training set, we perform the operations containing synonym replacement, random insertion, random swap, and random deletion introduced by Wei and Zou (2019).

4.3 Stacking Trained Models

Model stacking is an efficient ensemble method to improve model accuracy. The main procedure of stacking trained models in our method including five steps. First, we use heterogeneous PLMs including BERT, RoBERTa, ALBERT, and ERNIE as base models. Second, we generate multiple

Scheme	Model	R	Rho	MAE	MSE	R2
Baseline	Complexity average	-	-	0.1049	0.0189	0.0007
	Log Frequency and Length	0.5376	0.5251	0.0867	0.0135	0.2864
BERT	ERNIE$_{\text{LARGE}}$	0.7838	0.7321	0.0647	0.0069	0.6120
	ALBERT$_{\text{XXLARGE}}$	0.7850	0.7332	0.0644	0.0069	0.6115
	BERT$_{\text{LARGE}}$	0.7862	0.7296	0.0672	0.0073	0.5849
	RoBERTa$_{\text{LARGE+PL}}$	0.7770	0.7279	0.0656	0.0070	0.6023
	RoBERTa$_{\text{LARGE+DA}}$	0.7870	0.7432	0.0670	0.0078	0.5598
	RoBERTa$_{\text{LARGE}}$	**0.7903**	**0.7356**	**0.0648**	**0.0068**	**0.6170**

Table 2: Comparison of different pre-trained language models with training strategies of subtask 1

Scheme	Model	R	Rho	MAE	MSE	R2
Baseline	Complexity average	-	-	0.1164	0.0219	0.0000
	Log Frequency and Length	0.6249	0.6162	0.0900	0.0136	0.3807
BERT	RoBERTa$_{\text{LARGE}}$	0.7900	0.8002	0.0753	0.0092	0.6178
	ALBERT$_{\text{XXLARGE+sub1}}$	0.7901	0.7952	0.0755	0.00929	0.6157
	RoBERTa$_{\text{LARGE+sub1}}$	**0.8101**	**0.8236**	**0.0715**	**0.0085**	**0.6498**
Ensemble	mean	0.8252	0.8343	0.0690	0.0079	0.6739
	LR	**0.8330**	**0.8348**	**0.0678**	**0.0074**	**0.6892**

Table 3: Comparison of different pre-trained language models of subtask 2

hyperparameter sets by setting different values of dropout, selecting different numbers of last hidden layers, and using different loss functions. Since our purpose here is not only to find the best hyperparameter sets but also to collect diverse sets with reasonable performances, we keep all the training results from different sets. Third, we perform 7-fold cross-validation during the whole training process to avoid overfitting or selection bias. Fourth, we adopt several training strategies including using pseudo-labelling (Iscen et al., 2019) and data augmentation to further improve the diversity of trained models.

Ultimately, we train a simple linear regression model as the final estimator. Suppose that the complexity score predicted by a based model with one hyperparameter set is $\hat{y_j}$, then the final complexity scores will be calculated as below,

$$\hat{y} = \sum_{j=1}^{N} W_j \hat{y_j}$$

where N is the total number of various fine-tuned PLMs with different hyperparameters sets and W_j is the weight for each predicted score from different PLMs learned by a linear regression model.

5 Experiments

5.1 Evaluation Metrics

As mentioned in the official evaluation procedure of LCP 2021, several evaluation metrics are chosen including Pearson correlation (R), Spearman correlation (Rho), Mean absolute error (MAE), Mean squared error (MSE), and R-squared (R2). The final results are ranked using Pearson correlation.

5.2 Parameter settings

All models are implemented based on the open-source transformers library of hugging face (Wolf et al., 2020), which provides thousands of pre-trained models that can be quickly downloaded and fine-tuned on specific tasks. Table 1 shows the four employed PLMs and different parameters we set for each PLM including different numbers of hidden layers, different dropout pairs, and different loss functions.

6 Results

6.1 Ablation Study

PLMs with Training Strategies For subtask 1, we use different PLMs including ERNIE$_{\text{LARGE}}$, ALBERT$_{\text{XXLARGE}}$, BERT$_{\text{LARGE}}$, RoBERTa$_{\text{LARGE}}$ as shown in Table 2. The results are the average scores of 7-fold cross-validation on the training dataset. Since RoBERTa$_{\text{LARGE}}$ performs best on this task, we further incorporate the training strategies including pseudo-labelling (PL) and data augmentation (DA) with it. However, for the training dataset, we find that by adding the training strategies, the results decrease a little bit.

For subtask2, we use two types of PLMs which are RoBERTa$_{\text{LARGE}}$ and ALBERT$_{\text{XXLARGE}}$. The results shown in Table 3 are also obtained by averaging the scores of 7-fold cross-validation on the training dataset. Since we have added the dataset of subtask 1 into subtask 2, we also show the results

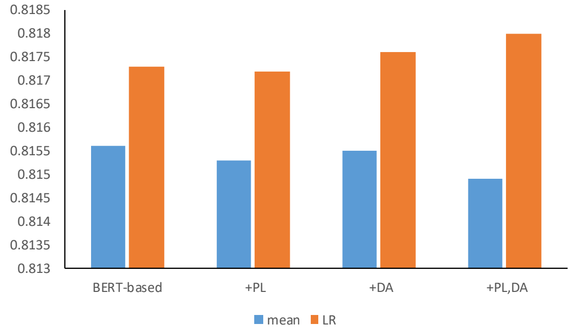

Figure 3: Comparison of Pearson Correlation values for stacking different models of subtask 1.

Subtask 1		Subtask 2	
System	**R**	**System**	**R**
qusaibanyismail	0.7886	**DeepBlueAI**	**0.8612**
DeepBlueAI	**0.7882**	rg_pa	0.8575
amsqr	0.7790	xiang_wen_tian	0.8571
armand.rotaru	0.7782	andi_gpu	0.8543
abdelkader	0.7779	ren_wo_xing	0.8541

Table 4: Leaderboard

of doing this in Table 3 and we can find that it is very effective by increasing 0.02 from base models.

Stacking trained models We use a linear regression (LR) model to stack different pre-trained models. We train the weights of each model in LR on the training set and then use the learning weights to predict the final scores of the test set.

Figure 3 shows the comparison of Pearson Correlation values for stacking different models of subtask 1. The columns in blue are the values computed by averaging predicted scores of different models while the columns in orange are the values through the LR function. We can clearly observe that the LR-based ensemble method outperforms those with the mean-based method, which verifies the validity of using the LR mechanism. Besides, although we find that adding training strategies to the base models would decrease performance according to Table 2, the performance will be improved when stacking them all. This indicates the positive effect of increasing model diversity.

6.2 Official Ranking

For both subtask 1 and subtask 2, among all the pre-submission experiments, we find that the scores obtained from stacking all the models performed best. The official ranking is presented in Table 4 and it demonstrates that our system is ranked first in subtask 2 and ranked second in subtask 1.

7 Conclusion

In this paper, we propose a top-performing model for the task of lexical complexity prediction. We fine-tune several pre-trained language models including BERT, ALBERT, RoBERTa, and ERNIE with different training strategies such as pseudo-labelling and data augmentation and stack them with a simple linear regression model. Experimental results show the effectiveness of this ensemble method and we win first place and second place for subtask 2 and 1.

References

David Alfter and Elena Volodina. 2018. Towards single word lexical complexity prediction. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 79–88.

Segun Taofeek Aroyehun, Jason Angel, Daniel Alejandro Pérez Alvarez, and Alexander Gelbukh. 2018. Complex word identification: Convolutional neural network vs. feature engineering. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 322–327.

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner, K Bretonnel Cohen, Karin Verspoor, Judith A Blake, et al. 2012. Concept annotation in the craft corpus. *BMC bioinformatics*, 13(1):1–20.

Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501.

Stefan Bott, Luz Rello, Biljana Drndarević, and Horacio Saggion. 2012. Can spanish be simpler? lexsis: Lexical simplification for spanish. In *Proceedings of COLING 2012*, pages 357–374.

Andrei Butnaru and Radu Tudor Ionescu. 2018. Unibuckernel: A kernel-based learning method for complex word identification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 175–183.

Dirk De Hertog and Anaïs Tack. 2018. Deep learning architecture for complexword identification. In *Thirteenth Workshop of Innovative Use of NLP for Building Educational Applications*, pages 328–334. Association for Computational Linguistics (ACL); New Orleans, Louisiana.

Conseil De l'Europe. 2003. *Cadre européen commun de référence pour les langues: apprendre, enseigner, évaluer*. Council of Europe.

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.

William H DuBay. 2004. The principles of readability. *Online Submission*.

Thomas François, Núria Gala, Patrick Watrin, and Cédrick Fairon. 2014. Flelex: a graded lexical resource for french foreign learners. In *International conference on Language Resources and Evaluation (LREC 2014)*.

Nuria Gala, Thomas François, Delphine Bernhard, and Cédrick Fairon. 2014. A model to predict lexical complexity and to grade words (un modèle pour prédire la complexité lexicale et graduer les mots)[in french]. In *Proceedings of TALN 2014 (Volume 1: Long Papers)*, pages 91–102.

Sian Gooding and Ekaterina Kochmar. 2018. Camb at cwi shared task 2018: Complex word identification with ensemble-based voting. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 184–194.

Sian Gooding and Ekaterina Kochmar. 2019. Complex word identification as a sequence labelling task. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1148–1153.

Nathan Hartmann and Leandro Borges Dos Santos. 2018. Nilc at cwi 2018: Exploring feature engineering and feature learning. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 335–340.

Hiroshi Inoue. 2019. Multi-sample dropout for accelerated training and better generalization. *arXiv preprint arXiv:1905.09788*.

Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. 2019. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5070–5079.

Tomoyuki Kajiwara and Mamoru Komachi. 2018. Complex word identification based on frequency in a learner corpus. In *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*, pages 195–199.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*.

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 preprint arXiv:1907.11692*.

Shervin Malmasi, Mark Dras, and Marcos Zampieri. 2016. Ltg at semeval-2016 task 11: Complex word identification with classifier ensembles. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 996–1000.

Shardlow Matthew, Evans Richard, Paetzold Gustavo, and Zampieri Marcos. 2021. Semeval2021 task 1 : Lexical complexity prediction.

Niloy Mukherjee, Braja Gopal Patra, Dipankar Das, and Sivaji Bandyopadhyay. 2016. Ju_nlp at semeval-2016 task 11: Identifying complex words in a sentence. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 986–990.

Gustavo Paetzold and Lucia Specia. 2016a. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569.

Gustavo Paetzold and Lucia Specia. 2016b. Sv000gg at semeval-2016 task 11: Heavy gauge complex word identification with system voting. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 969–974.

Maury Quijada and Julie Medero. 2016. Hmc at semeval-2016 task 11: Identifying complex words using depth-limited decision trees. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1034–1037.

Matthew Shardlow. 2013. A comparison of techniques to automatically identify complex words. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 103–109.

Matthew Shardlow. 2014. Out in the open: Finding and categorising errors in the lexical simplification pipeline. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 1583–1590.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. Complex: A new corpus for lexical complexity predicition from likert scale data. In *1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI) PROCEEDINGS Edited by Nuria Gala and Rodrigo Wilkens*, page 57.

S Rebecca Thomas and Sven Anderson. 2012. Wordnet-based lexical simplification of a document. In *KONVENS*, pages 80–88.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010.

Jason Wei and Kai Zou. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. 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 6383–6389.

Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. 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.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the 13th Workshop on Innovative Use of NLP for Building Educational Applications (NAACL 2018 Workshops)*, pages 66–78. Association for Computational Linguistics; Stroudsburg,.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.