UPB at SemEval-2021 Task 1: Combining Deep Learning and Hand-Crafted Features for Lexical Complexity Prediction

George-Eduard Zaharia, Dumitru-Clementin Cercel, Mihai Dascalu
University Politehnica of Bucharest, Faculty of Automatic Control and Computers
george.zaharia0806@stud.acs.upb.ro
{dumitru.cercel, mihai.dascalu}@upb.ro

Abstract

Reading is a complex process which requires proper understanding of texts in order to create coherent mental representations. However, comprehension problems may arise due to hard-to-understand sections, which can prove troublesome for readers, while accounting for their specific language skills. As such, steps towards simplifying these sections can be performed, by accurately identifying and evaluating difficult structures. In this paper, we describe our approach for the SemEval-2021 Task 1: Lexical Complexity Prediction competition that consists of a mixture of advanced NLP techniques, namely Transformer-based language models, pre-trained word embeddings, Graph Convolutional Networks, Capsule Networks, as well as a series of hand-crafted textual complexity features. Our models are applicable on both subtasks and achieve good performance results, with a MAE below 0.07 and a Person correlation of .73 for single word identification, as well as a MAE below 0.08 and a Person correlation of .79 for multiple word targets. Our results are just 5.46% and 6.5% lower than the top scores obtained in the competition on the first and the second subtasks, respectively.

1 Introduction

Reading is a complex process due to the mental exercise readers are challenged to perform, since a coherent representation of the text needs to be projected into their mind in order to grasp the underlying content (Van den Broek, 2010). For non-native speakers, the lack of text understanding hinders knowledge assimilation, thus becoming the main obstacle that readers need to overcome. Complex words can impose serious difficulties, considering that their meaning is often dependant on their context and cannot be easily inferred. In order to facilitate text understanding or to perform text simplification, complex tokens first need to be detected. This can be performed by developing systems capable of identifying them by individual analysis, as well as contextual analysis.

There are two main approaches regarding the complexity task. Tokens can be binary classified as complex or non-complex, a procedure that helps users separate problematic words from the others. Words can be also labeled with a probabilistic complexity value, which in return can be used to simplify the text. Words with lower degrees of complexity can be easily explained, whereas more complex tokens can be replaced with simpler equivalent concepts.

The Lexical Complexity Prediction (LCP) shared task, organized as the SemEval-2021 Task 1 (Shardlow et al., 2021a), challenged the research community to develop robust systems that identify the complexity of a token, given its context. Systems were required to be easily adaptable, considering that the dataset entries originated from multiple domains. At the same time, the target structure evaluated in terms of complexity could contain a single word or multiple words, depending on the subtask.

The current work is structured as follows. The next section presents the state-of-the-art Natural Language Processing (NLP) approaches for LCP (probabilistic) and complex word identification (CWI). The third section outlines our approaches for this challenge, while the fourth section presents the results. Afterwards, the final section draws the conclusions and includes potential solutions that can be used to further improve performance.

2 Related Work

Probabilistic CWI. Kajiwara and Komachi (2018) adopted for the CWI task a system based on Random Forest regressors, alongside several features, such as the presence of the target word in certain

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 609–616
Bangkok, Thailand (online), August 5–6, 2021.

corpora. Moreover, they conducted experiments to determine the best parameters for their regression algorithms.

De Hertog and Tack (2018) introduced a deep learning architecture for probabilistic CWI. Apart from the features extracted by the first layers of the network, the authors also included a series of hand-crafted features, such as psychological measures or frequency counts. Their architecture included different Long Short-Term Memory (LSTM) modules (Hochreiter and Schmidhuber, 1997) for the input levels (i.e., word, sentence), as well as the previously mentioned psychological measures and corpus counts.

Sequence labeling CWI. Gooding and Kochmar (2019) introduced a technique based on LSTMs for CWI, which obtained better results on their sequence labeling task than previous approaches based only on feature engineering. The contexts detected by the LSTM offered valuable information, useful for identifying complex tokens placed in sequences.

Changing the focus towards text analysis, Finnimore et al. (2019) extracted a series of relevant features that supports the detection of complex words. While considering their feature analysis process, the greatest influence on the overall system performance was achieved by the number of syllables and the number of punctuation marks accompanying the targeted tokens.

A different approach regarding CWI was adopted by Zampieri et al. (2017), who employed the usage of an ensemble created on the top systems from the SemEval CWI 2016 competition (Paetzold and Specia, 2016). Other experiments performed by the authors also included plurality voting (Polikar, 2006), or a technique named Oracle (Kuncheva et al., 2001), that forced label assignation only when at least one classifier detected the correct label.

Zaharia et al. (2020) tackled CWI through a cross-lingual approach. Resource scarcity is simulated by training on a small number of examples from a language and testing on different languages, through zero-shot, one-shot, and few-shot scenarios. Transformer-based models (Vaswani et al., 2017) achieved good performance on the target languages, even though the number of training entries was extremely reduced.

3 Method

3.1 Dataset

CompLex (Shardlow et al., 2020, 2021b) is the dataset used for the LCP shared task that was initially annotated on a 5-point Likert scale. Moreover, the authors performed a mapping between the annotations and values between 0 and 1 in order to ensure normalization. The dataset has two categories, one developed for single word complexity score prediction, while the other is centered on groups of words; each category has entries for training, trial, and testing. The single word dataset contains 7,662 entries for training, 421 trial entries, and 917 test entries. The multi-word dataset contains a considerably smaller number of entries for each category, namely 1,517 for training, 99 trial entries, and 184 for testing.

All entries from the LCP shared task are part of one of three different English corpora (i.e., Bible - biblical entries, Biomed - biomedical entries, and Europarl - political entries), evenly distributed, each one representing approximately 33% of its corresponding set. As such, the task is even more challenging when considering the vastly different domains of these entries.

3.2 Architecture

During our experiments, we combined features obtained from multiple modules described later on in detail, and then applied three regression layers, alongside a *Dropout* layer, to obtain the complexity score of the input (see Figure 1 for our modular architecture). The permanent components are represented by the target word embeddings and the Transformer features, which are concatenated and then fed into the final linear layers, designated for regression. The other components (i.e., character-level embeddings, GCN, and Capsule) are enabled in particular setups; similarly, the adversarial training component can also be disabled. At the same time, a series of hand-crafted features can be concatenated before the last layer with the aim to further improve the overall performance.

3.3 Pre-trained Word Embeddings

Pre-trained word embeddings were used as features for the final regression as an initial representation of the input. Throughout our experiments, three types of pre-trained word embeddings were consid-

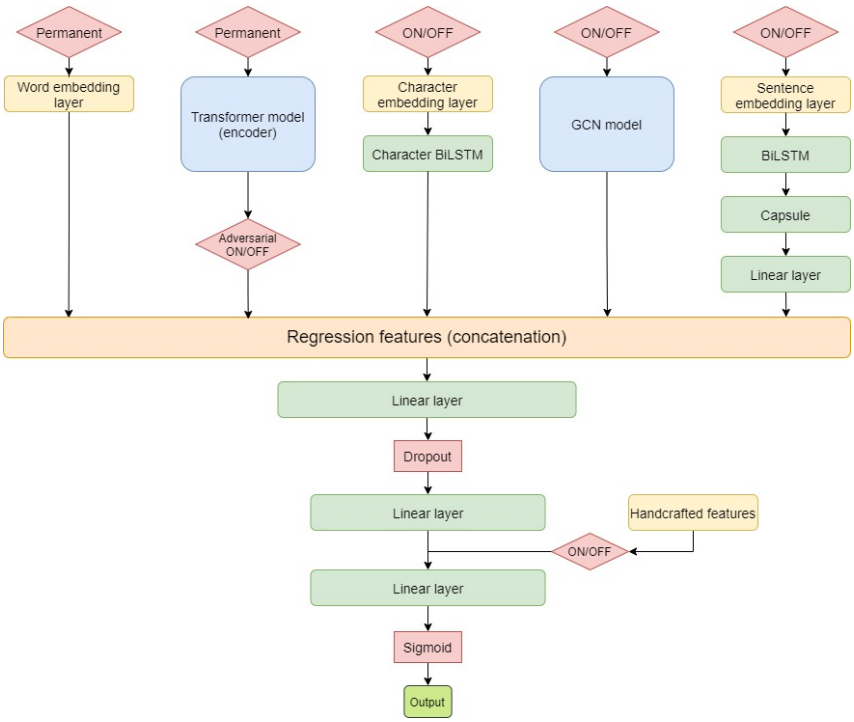

Figure 1: The overall model architecture used in our experiments.

ered, namely: GloVe[1], FastText[2], and skip-gram[3]. Out of the three previous options, GloVe performed best in our experiments. As such, the results section exclusively reports the performance obtained by our configurations alongside GloVe embeddings for the target word.

3.4 Transformer-based Language Models

Considering that Transformers achieve state-of-the-art performance for most NLP tasks (Wolf et al., 2020), all our setups include a Transformer-based component. However, they are pre-trained in different manners; thus, we experimented with several variants, as follows:

- **BERT** (Devlin et al., 2019) - Extensively pre-trained on English, BERT-base represents the baseline of Transformer-based models;
- **BioBERT** (Lee et al., 2020) - Considering that some of the most difficult to understand entries are part of the Biomed corpus, we also experimented with a model pre-trained on biomedical data;
- **SciBERT** (Beltagy et al., 2019) - Similarly to BioBERT, SciBERT is pre-trained on scientific data and becomes a good candidate for fine-tuning on the scientific entries from the dataset;
- **RoBERTa** (Liu et al., 2019) - RoBERTa improves upon BERT by modifying key hyper-parameters, and by being trained with larger mini-batches and learning rates; RoBERTa usually has better performance on downstream tasks.

3.5 Adversarial Training

We also aimed to improve the robustness of the main element of our architecture, the Transformer-based component. Therefore, we adopted an adversarial training technique, similar to Karimi et al. (2020). The adversarial examples generated during training work on the embeddings level, and are based on a technique that uses the gradient of the loss function.

3.6 Character-level Embeddings

Alongside the previously mentioned word embeddings for the target word, we also employ character-level embeddings for the same word, such that its internal structure, as well as its universal context, can be properly captured as features in our architecture.

[1] https://nlp.stanford.edu/projects/glove/
[2] https://fasttext.cc/
[3] http://vectors.nlpl.eu/repository/

3.7 Graph Convolutional Networks

Besides the previous Transformer-based models, we also explored the relations between the dataset entries, as well as the vocabulary words. Therefore, Graph Convolutional Networks (GCN) (Kipf and Welling, 2016) were also considered for determining node embedding vectors, by taking into account the properties of the neighboring nodes. By stacking multiple GCN layers, the information embedded into a node can become broader, inasmuch as it incorporates considerably larger neighborhoods. Similar to Yao et al. (2019), we consider the graph to have several nodes equal to the number of entries (documents) in the corpus plus the vocabulary size of the corpus.

3.8 Capsule Networks

Alongside the relational approach derived from GCN and Transformer embeddings, we intended to further analyze our inputs by passing them through a Capsule Network (Sabour et al., 2017). This approach enables us to obtain features that reflect aspects specific to different levels of the inputs, as Capsule Networks increase the specificity of features, while the capsule layers go deeper.

3.9 Hand-crafted Features

Similar to Gooding and Kochmar (2018), we integrated a series of hand-crafted features for the target word: **Syllables**, **Synset length**, **Hypernyms length**, **Hyponyms length**, **Number of dependencies** obtained using NLTK[4] alongside CoreNLP[5], **SubIMDB presence**[6], **SimpWiki presence** (Coster and Kauchak, 2011), **CEFR level** obtained from the Cambridge English dictionary[7], **MRC features** (Wilson, 1988) (*Age of acquisition*, *Concreteness rating*, *Imageability rating*, *Word familiarity rating* , *Number of phonemes*), **Semantic Diversity** (Hoffman et al., 2013), **Sensorimotor Norms** (Lynott et al., 2019).

Character n-grams - The character n-gram approach consists of two steps: first, a vectorizer is applied on the inputs to select a maximum number of 5,000 most frequent n-grams; second, Tf-Idf scores for these elements are computed. The obtained values are then normalized in the $[0, 1]$ range and used as features.

ReaderBench indices - The ReaderBench framework (Dascalu et al., 2017) was used to extract additional textual complexity features reflective of writing style. Out of the 1311 features obtained by running ReaderBench on our inputs, we selected 278. The choice was made by considering only the features with a high linguistic coverage (i.e, were non-zero for at least 50% of the entries).

3.10 Traditional Machine Learning Baseline

Several machine learning algorithms, such as *Logistic regression*, *Random Forest Regressors*, *XGBoost regression*, or *Ridge regression* were experimented using the aforementioned handcrafted features.

We then switched to a ridge regression approach and trained it with a multitude of features, including Transformer-based embeddings (BERT, BioBERT, SciBERT, RoBERTa), pre-trained word embeddings (GloVe, fastText, Skip-gram), and handcrafted features.

3.11 Preprocessing and Experimental Setup

Text preprocessing is minimal and consists of removing unnecessary punctuation, such as quotes. The experimental hyperparameters for all modules are presented in Table 1.

4 Results

Table 2 introduces the results obtained using our deep learning architecture, while Table 3 focuses on the traditional machine learning baseline. The best results for the deep learning approaches applied on the single target word dataset are obtained using RoBERTa as Transformer model. The setup which maximizes performance considers RoBERTa, GCN, and Capsule features, obtaining a Pearson score of 0.7702 and a mean absolute error (MAE) of 0.0671 on the trial dataset. Moreover, the high performance is maintained on the test dataset, with a Pearson correlation coefficient of 0.7237 and a MAE of 0.0677. BERT, SciBERT, and BioBERT have similar results with marginal differences; GCN, Capsule, and adversarial training improve performance for all models, while character-level embeddings do not provide a boost in performance.

Table 3 presents the results obtained using the features described in Section 3, namely Transformer-based contextualized embeddings (BERT, RoBERTa, BioBERT, SciBERT), pre-

[4] https://www.nltk.org/

[5] https://stanfordnlp.github.io/CoreNLP/

[6] http://ghpaetzold.github.io/subimdb/

[7] https://dictionary.cambridge.org/dictionary/learner-english/

Transformers	GCN	Capsule	BiLSTM	Embedding	Full Model
Size: 768	GCN Size 1: 512 GCN Size 2: 256	Routings: 5 Number of capsules: 10 Capsule dimension: 16	Dimension: 128	Dimension: 300	Optimizer: AdamW Loss Function: MSELoss Learning Rate: *2e-5*

Table 1: Experimental hyperparameters for the probabilistic CWI.

Configuration	Single-Word Target				Multi-Word Target			
	Trial		Test		Trial		Test	
	Pearson	MAE	Pearson	MAE	Pearson	MAE	Pearson	MAE
BERT	0.7575	0.0689	0.7170	0.0682	0.7019	0.0969	0.7853	0.0781
BERT + Capsule	0.7641	0.0685	0.7113	0.0689	0.7170	0.0958	0.7774	0.0791
BERT + GCN + Capsule	0.7548	0.0693	0.7178	0.0682	0.6978	0.0905	0.7773	0.0812
BERT + GCN + Capsule + Adversarial Training	0.7608	0.0695	0.7171	0.0684	0.7077	0.0933	0.8008	0.0779
BERT + Char Embeddings	0.7505	0.0701	0.6925	0.0717	0.7091	0.0904	0.7800	0.0821
RoBERTa	0.7676	0.0685	0.7222	0.0681	0.7177	0.0925	0.7921	0.0764
RoBERTa + GCN + Capsule*	**0.7702**	**0.0671**	0.7237	**0.0677**	0.7160	0.0910	**0.7962**	0.0788
RoBERTa + GCN + Capsule + Adversarial Training*	0.7699	0.0682	**0.7324**	0.0703	**0.7227**	0.0893	0.7851	0.0808
RoBERTa + Hand-crafted Features	0.7476	0.0704	0.7028	0.0735	0.7165	0.0974	0.7932	**0.0754**
RoBERTa + Char Embeddings + Capsule + GCN + Adversarial	0.7663	0.0696	0.7264	0.0692	0.7221	0.0954	0.7958	0.0791
RoBERTa + Char Embeddings	0.7658	0.0695	0.7259	0.0682	0.7167	0.0958	0.7916	0.0772
SciBERT + GCN	0.7626	0.0714	0.7145	0.0715	0.6829	0.0876	0.7888	0.0762
SciBERT + GCN + Capsule + Adversarial Training	0.7617	0.0721	0.7086	0.0724	0.7164	**0.0863**	0.7882	0.0785
SciBERT + Char Embeddings	0.7512	0.0710	0.7079	0.0691	0.6855	0.1046	0.7729	0.0809
BioBERT	0.7658	0.0694	0.7151	0.0698	0.7014	0.0906	0.7814	0.0827
BioBERT + GCN + Capsule + Adversarial Training	0.7683	0.0677	0.7144	0.0690	0.7098	0.0968	0.7919	0.0795
BioBERT + Char Embeddings	0.7619	0.0689	0.7073	0.0697	0.7069	0.0995	0.7849	0.0810

* The models marked with * are the ones used in our submissions.

Table 2: Results for the Deep Learning approaches.

trained word embeddings (GloVe, fastText, skip-gram), and hand-crafted features, all combined using various regression algorithms. Logistic regression, Random Forrest and XGBosst yield lower performance when compared to the previous deep learning approaches. However, we managed to increase the scores on the single target word dataset, with Pearson coefficients of 0.7738 and 0.7340 on the trial and test datasets, by combining the results obtained from training several instances of ridge regression. Nevertheless, the best results for the multiple target word task are still obtained by the deep learning approaches (RoBERTa, GCN, Capsule, adversarial training), which surpass the Ridge Regression + pre-trained word embeddings + Transformer embeddings + handcrafted features approach by a low margin of 0.0074 Pearson on the trial dataset and 0.0033 on the test dataset.

5 Discussion

The entries with the largest difference when compared to the gold standard are represented by the ones that are part of the Biomed category. This discrepancy is valid for both subtasks (i.e, single target word and multiple target words). The Biomed entries employ the usage of more complex terminology, quantities, or specific scientific names. Therefore, it becomes more difficult for standard pre-trained Transformer systems, such as BERT or RoBERTa, to adapt to the Biomed entries. In contrast, corpora with easier to understand language (i.e., Bible and Europarl) are not properly represented when using BioBERT or SciBERT, considering that the Transformers are mainly pre-trained for scientific or biomedical texts.

Moreover, a considerable part of the Biomed entries contains large amounts of abbreviations, while other entries from the same domain have

Method	Single-Word Target				Multi-Word Target			
	Trial		Test		Trial		Test	
	Pearson	MAE	Pearson	MAE	Pearson	MAE	Pearson	MAE
Logistic Regression	0.7158	0.0748	0.6868	0.0718	0.6533	0.0954	0.7558	0.0791
Random Forest Regressor	0.7390	0.0708	0.7011	**0.0691**	0.6714	0.0929	0.7651	**0.0785**
XGBoost Regressor	0.7488	0.0700	0.7033	0.0695	0.6503	0.0975	0.7544	0.0804
Ridge Regression*	**0.7738**	**0.0686**	**0.7340**	0.0699	**0.7153**	**0.0873**	**0.7929**	0.0787

* The solution marked with * is the one used in our submissions.

Table 3: Results for the Traditional Machine Learning solutions.

Entry	Target	Predicted complexity	True complexity
Genetic analyses of sitosterolemia pedigrees allowed the mapping of the STSL locus to human chromosome 2p21, between D2S2294 and D2S2298 [12,13].	pedigrees	0.4516	0.3125
Normally cells accumulate H3-2meK9 and H3-3meK9 marks and HP1B protein on the sex chromatin as they progress to diplonema, but we observed mutant diplotene cells lacking these features.	marks	0.2686	0.3409
p150CAF-1 knockdown in ES cells was quantified by Western blot analysis and IF.	ES	0.5587	0.6944

Table 4: Difficult Biomed entries.

specific terms or links, as seen in Table 4. The difference between our predictions and the correct labels are up to 0.14 for the complexity probability.

6 Conclusions and Future Work

This work proposes a modular architecture, as well as different training techniques for the Lexical Complexity Prediction shared task. We experimented with different variations of the previously mentioned architecture, as well as textual features alongside machine learning algorithms. First, we used different word embeddings and Transformer-based models as the main feature extractors and, at the same time, we examined a different training technique based on adversarial examples. Second, other different models were added, such as character-level embeddings, Graph Convolutional Networks, and Capsule Networks. Third, several hand-crafted features were also considered to create a solid baseline covering both deep learning and traditional machine learning regressors.

For future work, we intend to experiment with altering the modular architecture such that the models are trained similar to a Generative Adversarial Network (Croce et al., 2020), thus further improving robustness and achieving higher scores in terms of both Pearson correlation coefficients and MAE.

References

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A Pretrained Language Model for Scientific Text. 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 3606–3611.

Paul Van den Broek. 2010. Using texts in science education: Cognitive processes and knowledge representation. *Science*, 328(5977):453–456.

William Coster and David Kauchak. 2011. Learning to simplify sentences using Wikipedia. In *Proceedings of the workshop on monolingual text-to-text generation*, pages 1–9.

Danilo Croce, Giuseppe Castellucci, and Roberto Basili. 2020. GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2119.

Mihai Dascalu, Gabriel Gutu, Stefan Ruseti, Ionut Cristian Paraschiv, Philippe Dessus, Danielle S McNamara, Scott A Crossley, and Stefan Trausan-Matu. 2017. ReaderBench: a multi-lingual framework for analyzing text complexity. In *European Conference on Technology Enhanced Learning*, pages 495–499. Springer.

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

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.

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.

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.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Paul Hoffman, Matthew A Lambon Ralph, and Timothy T Rogers. 2013. Semantic diversity: A measure of semantic ambiguity based on variability in the contextual usage of words. *Behavior research methods*, 45(3):718–730.

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.

Akbar Karimi, Leonardo Rossi, Andrea Prati, and Katharina Full. 2020. Adversarial training for aspect-based sentiment analysis with BERT. *arXiv preprint arXiv:2001.11316*.

Thomas N Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv preprint arXiv:1609.02907*.

Ludmila I Kuncheva, James C Bezdek, and Robert PW Duin. 2001. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern recognition*, 34(2):299–314.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

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*.

Dermot Lynott, Louise Connell, Marc Brysbaert, James Brand, and James Carney. 2019. The Lancaster Sensorimotor Norms: multidimensional measures of perceptual and action strength for 40,000 English words. *Behavior Research Methods*, pages 1–21.

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.

Robi Polikar. 2006. Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic Routing between Capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 3859–3869.

Matthew Shardlow, Richard Evans, Gustavo Paetzold, and Marcos Zampieri. 2021a. SemEval-2021 Task 1: Lexical Complexity Prediction. In *Proceedings of the 14th International Workshop on Semantic Evaluation (SemEval-2021)*.

Matthew Shardlow, Richard Evans, and Marcos Zampieri. 2021b. Predicting Lexical Complexity in English Texts. *arXiv preprint arXiv:2102.08773*.

Matthew Shardlow, Marcos Zampieri, and Michael Cooper. 2020. CompLex—A New Corpus for Lexical Complexity Predicition from LikertScale Data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62.

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.

Michael Wilson. 1988. MRC psycholinguistic database: Machine-usable dictionary, version 2.00. *Behavior research methods, instruments, & computers*, 20(1):6–10.

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.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph Convolutional Networks for Text Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7370–7377.

George-Eduard Zaharia, Dumitru-Clementin Cercel, and Mihai Dascalu. 2020. Cross-Lingual Transfer Learning for Complex Word Identification. In *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 384–390. IEEE.

Marcos Zampieri, Shervin Malmasi, Gustavo Paetzold, and Lucia Specia. 2017. Complex Word Identification: Challenges in Data Annotation and System Performance. In *Proceedings of the 4th Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA 2017)*, pages 59–63.