Lexical Complexity Prediction

Winnie Har
Purdue University Fort Wayne
Department of Computer Science

Abstract

Complex words present a challenge for English learner. In this paper, we present five models that we used to predict the complexity of a word. If a robust and reliable system can be built to make lexical complexity prediction, it could improve readability of English learners and also contribute to downstream Natural Language Processing related tasks.

1 Introduction

English is the official language of over 80 sovereign states and non-sovereign entities all over the world. It is also the *de facto* language used for communication in education, government and businesses in over 20 countries where it is not the official language. With English being one of the working or official language of international organizations such as the Organisation of Islamic Cooperation, the European Union, NAFTA, the United Nations and the sole official language of the Association of Southeast Asian Nations and the Commonwealth Nations, it is therefore even more important to be proficient in English.

Despite the widespread use of English, many English learners find it difficult to learn the language. English learners, when reading may face difficulty when they come across complex words which may then diminish their motivation to learn. If complex words can be identified automatically and be replaced with simpler synonyms, it will definitely improve readability and the learning experience of people with low literacy. In addition, with the increasing popularity of distant learning, there is also higher demand for lexical complexity prediction (LCP). An extension of complex word identification (CWI), lexical complexity prediction is also a building block for other natural language processing (NLP) tasks such as machine translation and text simplification.

In Section 2, we review related work of this task. In Section 3, we describe the task and dataset in more details. In Section 4 and 5, we introduce the methodology, our approaches and the models we trained. In Section 6, we present the results we obtained and comparison with SemEval 2021 official results. Finally, in Section 7 and 8, we discuss potential improvement and future work.

2 Related Work

This task is taken from SemEval 2021 Task 1. Prior to this, there were 2 similar versions in the SemEval 2016 and 2018. In the first version, CWI 2016, the goal of the task was to predict whether a word is complex or not. Due to the success of the task, a second version, CWI 2018 was held. Besides giving a binary prediction, the CWI 2018 required participants to also provide the probabilistic classification. There are also interest in lexical complexity of other languages such as German, Japanese and French. For example, Billami et al. (2018) created a French lexicon, ReSyf, as part of their work in the lexical complexity of French words.

3 Task and Dataset

3.1 Task

In the past Complex Word Identification tasks, the annotators had to make subjective judgment to classify the words to binary label. Their judgment may be affected by factors including their education level, first language, specialism or familiarity with the text at hand (Shardlow et al., 2021). These factors may cause differences in which an annotator will label a word as complex. One annotator may label words that are completely indecipherable, another may label every word they feel is above average difficulty, while another may label words that they could guess the meaning from context even though they are unfamiliar with the word. Another problem with binary classification is that a word which is close to the decision boundary is

assumed to be as complex as another word that is far away.

This LCP task seeks to address these differences by giving annotators a 5-point Likert Scale as a guidance in providing the annotations. By doing so, while the annotator must still use their subjective judgment, the complexity score given will be able to capture the meaning behind each judgment given by the annotators. Instead of a binary label, a system must now predict the word complexity on a continuous scale between 0 and 1.

The Likert Scale points is defined as such:
Very easy: Very familiar words
Easy: An annotator was aware of the meaning
Neutral: Neither difficult nor easy
Difficult: Words for which an annotator was unclear of the meaning, but may have been able to infer the meaning from the sentence
Very Difficult: Words that an annotator had never seen before, or were very unclear

The Likert Scales points are then averaged and converted to a continuous scale as follows: Very Easy $\rightarrow$ 0, Easy $\rightarrow$ 0.25, Neutral $\rightarrow$ 0.5, Difficult $\rightarrow$ 0.7 , Very Difficult $\rightarrow$ 1.0.

Hence, given a sentence and a token which is a word in the sentence, our task is to predict the complexity of the token in context (subtask 1 of LCP 2021). Subtask 2 involves predicting the complexity of multi-word expressions but we will only be focusing on subtask 1 in this paper.

3.2 Dataset

The dataset provided for this task is an augmented version of the CompLex dataset (Shardlow et al., 2020) and they are available publicly from the authors' GitHub repository[1]. From the original dataset, instances with less than 3 annotations are dropped. The authors then requested 10 more annotations on the same data for each instance using Amazon's Mechanical Turk platform. The data comes from 3 different domains: The Bible - the most massive parallel corpora (Christodoulopoulos and Steedman, 2014), Europarl - taken from the proceedings of the European Parliament (Koehn, 2005), and Biomedical articles from the CRAFT corpus (Bada et al., 2012). The distribution of data across these 3 domains are even in both the training set and testing set (Table 1). Trial data was also provided but the details are not included as we did not make use of them.

[1]https://github.com/MMU-TDMLab/CompLex

	Single Word Instance	
Source	**Training**	**Test**
Biblical text	2,574	283
Biomedical articles	2,576	289
Europarl Proceedings	2,512	345
Total	**7,662**	**917**

Table 1: The dataset consists of texts from 3 highly specialized domains and are evenly distributed across each domain.

4 Methodology

The texts are preprocessed by removing stopwords based on the NLTK stopword list, removing non-alphabetical characters and converting the texts to lowercase. After preprocessing, we explored 2 ways of vectorizing the texts. The first way is Term Frequency–Inverse Document Frequency (TF-IDF), a static approach of word vectorization which depends on the frequency of a word in a document (one instance of data), and the inverse document frequency of the word across a set of documents (the entire dataset). Since TF-IDF resulted in a sparse matrix which is difficult to work with, we chose to use the pretrained Bidirectional Encoder Representations from Transformers (BERT) tokenizer and model for feature extraction. The maximum vector length is first set to 512. We extract the top 4 layers of the BERT model output as the finding in the original BERT paper (Devlin et al., 2018) suggested that information from the top 4 layers gave relatively good F1 score than fine-tuning the entire model.

The sentence is first tokenized using the PreTrainedTokenizerFast tokenizer and then fed into the BERT model (both using bert-base-uncased). Each layer in the BERT model outputs a vector of dimension [512 x 768]. The output of the top 4 layers are summed and the 768 feature values are averaged. Hence each sentence is represented by a 1-dimensional vector of length 512.

With these 2 embedding methods, we trained 5 models - SGD Classifier, fully connected Neural Network, Epsilon-Support Vector Regression, Random Forest Regressor and Voting Regressor.

5 Experiments

5.1 SGD Classifier

We started off by using a multi-class classifier and TF-IDF embeddings. Due to the number of in-

stances and vocabulary size, the TF-IDF resulted in a sparse matrix of dimension [7662 x 15148] with only 108,159 non-zero elements, less than 0.01% of the matrix. The output of the SGD Classifier is an ordinal categorical value between 0 and 4, with 0 being "Very Easy" and 4 being "Very Difficult" according to the Likert Scale points.

5.2 Fully connected Neural Network

From the BERT feature extraction technique describe earlier, after summing the top 4 layers, we extract the embedding for the token of which its complexity we are trying to predict and the average value is taken. The resulting 1-dimensional vector for the sentence and scalar for the token are concatenated, fed into a fully connected Neural Network (NN) with 3 hidden layers. The NN also outputs a categorical complexity score between 0 and 4. The model is trained for 30 epochs with RMSprop as the optimizer, sparse categorical cross entropy as the loss function and a learning rate of 0.005.

5.3 Epsilon-Support Vector Regressor

In contrast to the concatenation done after embedding the sentence and extracting the token when modelling using NN, we concatenate the raw token and sentence before feeding the string to the PreTrainedTokenizerFast tokenizer and BERT model. The remaining summing and averaging are as described in the Methodology section. The Epsilon-Support Vector Regression (SVR) is trained with RBF kernel with polynomial kernel function of 3 degrees. The model produces complexity score in continuous scale but we converted the score to a categorical value so that comparison can be made with the SGD Classifier and NN.

5.4 Random Forest Regressor

We used the same embedding technique as the SVR. The Random Forest Regressor is trained with 100 estimators. Likewise, the model produces complexity score in continuous scale and we converted the score to a categorical value as well.

5.5 Voting Regressor

With the same embedding technique as the SVR, we experimented with Voting Regressor. The Voting Regressor is made up of Gradient Boosting Regressor, Random Forest Regressor and Linear Regressor. We also converted the continuous complexity score from the Voting Regressor to a categorical value.

Model	Performance Metrics	
	Accuracy	**R^2 Score**
SGD Classifier	0.42	-1.579
Neural Network	**0.61**	-2.647
SVR	0.59	-4.962
Random Forest	0.60	-7.900
Voting Regressor	**0.61**	-16.046

Table 2: Accuracy and R^2 Score for categorical output

Model	Performance Metrics		
	MAE	**MSE**	**Pearson's R**
SVR	0.106	0.018	0.110
Random Forest	0.101	0.017	0.115
Voting Regressor	**0.098**	**0.016**	**0.160**

Table 3: Mean absolute error, mean squared error and Pearson's R for continuous output

6 Results

We present the results obtained based on the types of output, whether categorical or continuous. Since the official results of SemEval 2021 are available, it serves as a good benchmark to evaluate the performance of our model. However, it is meaningless to compare the categorical results here with the official results since they are different in nature.

The NN and Voting Regressor both achieved an accuracy of 0.61. It is worth noting the negative R^2 scores in Table 2. In the case where the true prediction is a non-constant, a constant model which does not regard the inputs and only give a constant prediction will have an R^2 score of 0. What this means is, the models are worse than using a straight line when making prediction.

When predicting in continuous scale, the Voting Regressor achieved the best performance across all 3 metrics as compared to SVR and Random Forest. The final ranking of SemEval 2021 was based

SemEval Ranking	Performance Metrics		
	MAE	**MSE**	**Pearson's R**
Best	0.0609	0.0061	0.7886
Median	0.0652	0.0071	0.7533
Frequency Baseline	0.0804	0.0110	0.5834
Worst	0.2777	0.1270	-0.0272

Table 4: Selected results from SemEval 2021

on Pearson's R or Pearson correlation coefficient which measures the ***linear association*** between two interval- or ratio-level variables (Boslaugh, 2013). However, we are not sure if the linear association assumption is valid or not. Nonetheless, when comparing MAE and MSE against the SemEval results, the Voting Regressor fared very badly too, all of which fell below the frequency baseline.

7 Discussion

Our results suggested that using BERT to extract features and embed the texts improved the model performance in terms of accuracy. Furthermore, classical models appear to achieve relatively good results as well. When predicting categorical labels, the Voting Regressor obtained similar accuracy as compared to Neural Network. Furthermore, even though the first (Bani Yaseen et al., 2021) and second (Pan et al., 2021) ranked systems in SemEval 2021 made use of the SOTA BERT models, the system that came in third modeled their system using a Gradient Tree Boosting with extensive feature engineering (Mosquera, 2021). This shows that there is huge room for improvement for our models by fine-tuning and possibly other embedding techniques could also be experimented.

8 Conclusion

Lexical Complexity Prediction remains as a huge area of interest because of its many potential use cases and applications. If a reliable system can be built, it could help in various assistive technologies for learning and improving readability of texts. LCP can also be integrated with other downstream NLP tasks like machine translation and lexical simplification. Ongoing researches also involve multilingual LCP systems (training a system to predict word complexity in multiple languages) and cross-lingual LCP systems (training a system in one or more languages and predicting in another language that is not seen during training) that would benefit people who want to learn a second or third language and the NLP industry as a whole.

References

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William Baumgartner Jr, Kevin Cohen, Karin Verspoor, Judith Blake, and Lawrence Hunter. 2012. Concept annotation in the craft corpus. *BMC bioinformatics*, 13:161.

Tuqa Bani Yaseen, Qusai Ismail, Sarah Al-Omari, Eslam Al-Sobh, and Malak Abdullah. 2021. JUST-BLUE at SemEval-2021 task 1: Predicting lexical complexity using BERT and RoBERTa pre-trained language models. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 661–666, Online. Association for Computational Linguistics.

Mokhtar B. Billami, Thomas François, and Núria Gala. 2018. ReSyf: a French lexicon with ranked synonyms. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2570–2581, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Sarah Boslaugh. 2013. *Chapter 7. The Pearson Correlation Coefficient*. O'Reilly.

Christos Christodoulopoulos and Mark Steedman. 2014. A massively parallel corpus: the bible in 100 languages. *Language Resources and Evaluation*, 49:1–21.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

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.

Chunguang Pan, Bingyan Song, Shengguang Wang, and Zhipeng Luo. 2021. DeepBlueAI at SemEval-2021 task 1: Lexical complexity prediction with a deep ensemble approach. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 578–584, Online. Association for Computational Linguistics.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex — a new corpus for lexical complexity prediction from Likert Scale data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62, Marseille, France. European Language Resources Association.

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.