hub at SemEval-2021 Task 1: Fusion of Sentence and Word Frequency to Predict Lexical Complexity

Bo Huang, Yang Bai, Xiaobing Zhou*
School of Information Science and Engineering
Yunnan University, Yunnan, P.R. China
⋆Corresponding author:zhouxb@ynu.edu.com

Abstract

In this paper, we propose a method of fusing sentence information and word frequency information for the SemEval 2021 Task 1-Lexical Complexity Prediction (LCP) shared task. In our system, the sentence information comes from the RoBERTa model, and the word frequency information comes from the Tf-Idf algorithm. Use Inception block as a shared layer to learn sentence and word frequency information. We described the implementation of our best system and discussed our methods and experiments in the task. The shared task is divided into two subtasks. The goal of the two subtasks is to predict the complexity of a predetermined word. The evaluation index of the task is the Pearson correlation coefficient. Our best performance system has Pearson correlation coefficients of 0.7434 and 0.8000 in the single-token subtask test set and the multi-token subtask test set, respectively.

1 Introduction and Background

Language and writing are the main ways we transmit knowledge and information. An accurate and efficient understanding of the meaning expressed in the text is of great significance to our learning and production. Vocabulary complexity and reading comprehension are inextricably linked, and overly complex terms may bring bad results (DuBay, 2004). The research of Leroy et al. showed that the use of vocabulary simplification technology is one of the ways to effectively improve readers' reading comprehension ability (Leroy et al., 2013). Accurately predicting lexical complexity can make the system better guide users to use appropriate text, or customize text according to their needs. Especially when some ordinary readers are reading technical text content (Wei et al., 2009). Lexical complexity detection and complex lexical simplification have attracted the attention of the NLP community, and systems have been developed to simplify the text of second language learners (Shardlow, 2014), native speakers with low literacy levels (Specia, 2010), and people with dyslexia (Rello et al., 2013).

The topic of the shared task of SemEval 2021 Task 1 is "Lexical Complexity Prediction (LCP)". The task data set uses English text data in a single language (Shardlow et al., 2020). There are two subtasks in the task, which are the subtasks for predicting the complexity of a single token and multiple tokens (Shardlow et al., 2021). In this article, we give the method and task result of predicting word complexity. Our system uses a method that combines sentence, word frequency, and context information. The acquisition of sentences and context information uses the RoBERTa model (Liu et al., 2019b). The word frequency information comes from the Tf-Idf algorithm (Ramos et al., 2003). The complexity is a continuous value, so the whole task can be regarded as a regression task. We provide the model code used in this task [1].

2 Related Work

Previously held similar to this shared task are SemEval 2016 task 11: Complex Word Identification (CWI2016) (Paetzold and Specia, 2016a), Complex Word Identification Shared Task 2018 (CWI2018) (Yimam et al., 2018).

In CWI2016, the system voting method used by Paetzold et al. has achieved excellent results in sharing tasks (Paetzold and Specia, 2016b). In CWI2018, Butnaru uses a kernel-based learning method for complex word identification (Butnaru and Ionescu, 2018). Sian and other methods using integrated voting have also achieved good scores (Gooding and Kochmar, 2018). In addition to the above methods, some common methods are applied to these tasks. For example, SVM, random forest,

[1] https://github.com/Hub-Lucas/task1

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 598–602
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

Yahweh
report
one
God
king
house
Committee
will
may
gene
Commission
people
cell
son
two
mice
Council
time
day
said
protein
behalf
now
made
expression
European Parliament
Israel
Mr President
et al
European
gene
will
mice
one
Yahweh
cell
may
protein
expression
Committee
house
report
role
two
God
people
Council
made
time
Mr
work
development
Commission
day
many
specific
Parliament
level
year

(a) The training set data of a single token

(b) The training set data of two tokens

Figure 1: The word cloud diagrams of the text data of the training set of a single token and two tokens provided by the task organizer team. The result shown in the figure is the data after removing the stop words.

artificial neural network system, naive Bayes, decision tree, etc. (Paetzold and Specia, 2016a; Yimam et al., 2018).

3 Data and Methods

3.1 Data Description

We obtain data sets related to this task from the task organizer team. The data set includes training data set and test data set. We analyze the structure and characteristics of the data set. The training data set includes ID, Corpus, Sentence, Token, Complexity. The texts in the data set come from different fields, and Corpus represents which corpus the data set belongs to. Token is the target word we need to predict the complexity of the task. Complexity is a continuous value between 0-1. It represents the complexity score of the token in the sentence. Compared with the training data set, the test set only does not contain the aforementioned Complexity part. We need to use our method to predict the complexity of the Token specified in the test set in Sentence. Table 1 shows the examples of the data we used in the task.

In subtask 1, 7662 and 917 different sample data constitute the training set and the validation set. In subtask 2, 1517 and 184 different sample data constitute the training set and the validation set. In our system, we use Tf-Idf encoding information as an externally introduced word embedding. We use word cloud graphs to visualize the text data in the two subtasks. The word cloud image clearly shows us the characteristics of word frequency distribution in the text data set. We can easily see those words that appear frequently. Figure 1 show the word cloud diagrams of the text data of two different subtasks.

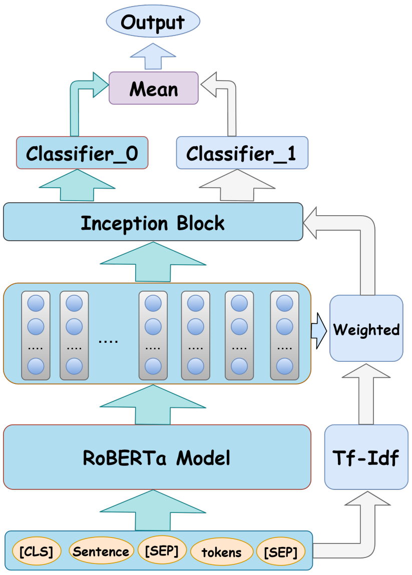

Figure 2: The model structure and data flow we used in the task.

3.2 Methods

The best result score we submitted is based on the system developed by RoBERTa, Tf-Idf, and Inception. Besides, we also use a BERT-based system to compare the result scores of different systems on the same verification set. Both BERT (Devlin et al., 2018) and RoBERTa's models (Liu et al., 2019b) are based on improvements in transformer architecture (Vaswani et al., 2017). RoBERTa has made some improvements to BERT and achieved better results than BERT. RoBERTa removed the task of predicting the next sentence in the pre-training phase and also used a new dynamic Masking mechanism. At the same time, RoBERTa has longer training time, larger batches, and more training data. Based on the working principle of LSTM (Olah, 2015) and considering the issue of training time, we chose Inception based on the CNN struc-

ID	corpus	sentence	token	complexity
subtask1-01	bible	He raises his hands against his friends.	hands	0.0278
subtask1-02	europarl	The first is Johann Wolfgang von Goethe.	Goethe	0.5
subtask2-01	bible	You shall tread on their high places..	high places	0.1625
subtask2-02	europarl	The first is legislative efficacy.	legislative efficacy	0.3833

Table 1: The training set sample data we use in the task. Subtask 1 has only one token, and subtask 2 has two tokens.

ture. The structure of the Inception Block used in our system is an improvement based on the solution implemented by Szegedy et al. (Szegedy et al., 2015). In the Inception block, we use the Conv1d convolution provided by Pytorch to adapt to our needs in the task. Inception Block has convolution kernels of different sizes, and can use windows of different sizes to extract non-continuous semantic features. At the same time, the parallel structure of Inception Block can save training time. The structure of the transformer allows interaction between the input sentence and the input token. We use the output of RoBERTa and the output of Tf-Idf weighted RoBERTa as different inputs of Inception Block, so that Inception Block can capture different information. Different classifiers are used to process the output results from different inputs of the Inception Block.

In step 1, we spliced the text data (Sentence) and the target word (Token) in the data with (SEP). Then the spliced result is used as the input of RoBERTa and Tf-Idf. In step 2, we use the output of Tf-Idf to weight the output of RoBERTa. In step 3, we use the weighted result of the previous step and the output result of RoBERTa as the input of the Inception Block. Here, the Inception Block is used as a shared layer to learn the output results of RoBERTa and the output results of RoBERTa weighted by Tf-Idf. In step 4, two linear classifiers are used to process the output from the Inception Block. In step 5, the output results of the two linear classifiers are averaged. In step 6, the average value is output as the final prediction result of the system.

4 Experiment and Results

In this section, we will introduce the data preprocessing methods and experimental settings we used in the task and the final results.

4.1 Data Preprocessing

In the part of data processing, we deleted the stop words in the text data. For the stop word list, we use the stop word package provided by NLTK. To use the Tf-Idf algorithm to obtain a weighted output, and to ensure that the shape of the text code processed by the Tf-Idf algorithm is consistent with the shape of the RoBERTa output, we removed the part of the text code that exceeded the maximum sentence. For those text encodings that are less than the maximum sentence length, we perform zero padding. The encoding of Tf-Idf is obtained using the toolkit provided by gsim (Řehůřek and Sojka, 2010) [2]. For the validation set, we randomly select 20% from the pre-processed training set as our validation set during the training process. The remaining 80% of the training set is used as our training set during the training process.

4.2 Experiment setting

During our training model, we designed 4 different models and observed the result scores of different models on the validation set. We adjust the parameters as much as possible to obtain the best results for each different model, so different systems may have different parameter combinations. The overall design and data flow of the BERT+Tf-Idf+Inception system is the same as the system we introduced in Figure 2. The difference is that we replace the RoBERTa model in Figure 2 with the BERT model. In all experiments, we use Radam (Liu et al., 2019a) as the optimizer and MSELoss as the loss function.

- RoBERTa+Tf-Idf+CNN: The epoch, batch size, maximum sequence length, and learning rate for the model are 4, 32, 60, and 4e-5, respectively.
- BERT+Tf-Idf+CNN: The epoch, batch size, maximum sequence length, and learning rate for the model are 4, 32, 60, and 3e-5, respectively.
- RoBERTa: The epoch, batch size, maximum

[2] https://github.com/RaRe-Technologies/gensim

Team	subtask	Pearson	Spearman	MAE	MSE	R2
Top1	1	0.7886	0.7369	0.0609	0.0062	0.6172
Top2	1	0.7882	0.7425	0.0610	0.0061	0.6210
Top3	1	0.7790	0.7355	0.0619	0.0064	0.6062
Our	1	0.7434	0.6995	0.0658	0.0073	0.5486
Top1	2	0.8612	0.8526	0.0616	0.0063	0.7389
Top2	2	0.8575	0.8529	0.0672	0.0072	0.7035
Top3	2	0.8571	0.8548	0.0675	0.0072	0.7012
Our	2	0.8000	0.7797	0.0754	0.0089	0.6323

Table 2: The scores of the top three teams and our team on the test set announced by the task organizer. Mean absolute error (MAE), Mean squared error (MSE), R-squared (R2). 61 and 38 different teams submitted results for subtask 1 and subtask 2, respectively.

sequence length, and learning rate for the model are 4, 32, 60, and 3e-5, respectively.

- BERT: The epoch, batch size, maximum sequence length, and learning rate for the model are 4, 32, 60, and 3e-5, respectively.

5 Results and Analysis

According to the Pearson correlation coefficient, the results submitted by the teams participating in the two subtasks are ranked. In the published results, the task organizer team also announced some other evaluation indicators. These evaluation indicators are Spearman correlation (Rho), Mean absolute error (MAE), Mean squared error (MSE), R-squared (R2). We compare the scores of Pearson correlation coefficient results obtained by several different methods proposed in the experimental part. The results of these different methods can be found in Table 3.

Compare our result scores on the validation set of the two subtasks. First of all, our system can predict the word complexity required in the task. Secondly, under the same data and parameters, the score obtained by the RoBERTa model is higher than the score obtained by the BERT model. Then, the scores we get on the RoBERTa+Tf-ifd+Inception and BERT+Tf-ifd+Inception systems are higher than the single use of the RoBERTa model and the use of the BERT model. Finally, the above performance proves the feasibility of the improved method we used.

After comparing the result scores of different systems on the verification set, we used the RoBERTa+Tfifd+Inception system to predict the results of the test set and successfully submitted it to the task organizer team. Our test set prediction result scores are 38th and 22nd respectively in the ranking lists of the two subtasks. Table 2 shows the test set result scores of the top three teams and our team announced by the task organizer team.

Method	Pearson(1)	Pearson(2)
RoBERTa+Tf-ifd+Inception	0.7651	0.8072
BERT+Tf-Ifd+Inception	0.7426	0.7850
RoBERTa	0.7327	0.7846
BERT	0.7255	0.7644

Table 3: The scores of the Pearson correlation coefficient results obtained by our different systems on the validation set. The validation set comes from 20% of the training set provided by the task organizer. Pearson(1) is the result score of the Pearson correlation coefficient of subtask 1. Pearson(2) is the result score of the Pearson correlation coefficient of subtask 2.

6 Conclusion

In this article, we describe the system our team has developed for shared tasks in SemEval 2021 task1 LCP. The system combines lexical, syntactic, and contextual semantic features. We describe and analyze the tasks, data, experiments, and results. We compared the results of the RoBERTa model and the BERT model. In the final test set prediction result score ranking, our results in the competition ranked middle. In future work, we will study how the complexity of the phrase is affected by the context in the sentence. For our model and method, we can also try to introduce other types of word embedding, and use different models to fuse the output of RoBERTa and the output of word embedding.

References

Andrei M Butnaru and Radu Tudor Ionescu. 2018. Unibuckernel: A kernel-based learning method for complex word identification.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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

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, New Orleans, Louisiana. Association for Computational Linguistics.

Gondy Leroy, James E Endicott, David Kauchak, Obay Mouradi, and Melissa Just. 2013. User evaluation of the effects of a text simplification algorithm using term familiarity on perception, understanding, learning, and information retention. *Journal of medical Internet research*, 15(7):e144.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019a. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*.

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

Christopher Olah. 2015. Understanding lstm networks.

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, San Diego, California. Association for Computational Linguistics.

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)*.

Juan Ramos et al. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta. ELRA. `http://is.muni.cz/publication/884893/en`.

Luz Rello, Ricardo Baeza-Yates, Stefan Bott, and Horacio Saggion. 2013. Simplify or help? text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, pages 1–10.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

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, and Marcos Zampieri. 2021. Predicting lexical complexity in english texts. *arXiv preprint arXiv:2102.08773*.

Lucia Specia. 2010. Translating from complex to simplified sentences. In *International Conference on Computational Processing of the Portuguese Language*, pages 30–39. Springer.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Ruth Chung Wei, Linda Darling-Hammond, Alethea Andree, Nikole Richardson, and Stelios Orphanos. 2009. Professional learning in the learning profession: A status report on teacher development in the us and abroad. technical report. *National Staff Development Council*.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo Paetzold, Lucia Specia, Sanja Štajner, Anaïs Tack, and Marcos Zampieri. 2018. A report on the complex word identification shared task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.