PolyU CBS-Comp at SemEval-2021 Task 1: Lexical Complexity Prediction (LCP)

Rong Xiang
Department of Computing
The Hong Kong Polytechnic University
csrxiang@comp.polyu.edu.hk

Jinghang Gu
Chinese and Bilingual Studies
The Hong Kong Polytechnic University
jinghang.gu@polyu.edu.hk

Emmanuele Chersoni
Chinese and Bilingual Studies
The Hong Kong Polytechnic University
emmanuelechersoni@gmail.com

Wenjie Li
Department of Computing
The Hong Kong Polytechnic University
cswjli@comp.polyu.edu.hk

Qin Lu
Department of Computing
The Hong Kong Polytechnic University
csluqin@comp.polyu.edu.hk

Chu-Ren Huang
Chinese and Bilingual Studies
The Hong Kong Polytechnic University
churen.huang@polyu.edu.hk

Abstract

In this contribution, we describe the system presented by the PolyU CBS-Comp Team at the Task 1 of SemEval 2021, where the goal was the estimation of the complexity of words in a given sentence context.

Our top system, based on a combination of lexical, syntactic, word embeddings and Transformers-derived features and on a Gradient Boosting Regressor, achieves a top correlation score of 0.754 on the subtask 1 for single words and 0.659 on the subtask 2 for multiword expressions.

1 Introduction

The notion of *complexity* has often been debated in linguistics and, depending from the disciplines, it might have different meanings.

In linguistic typology, for example, complexity is generally studied as a property of the language system as a whole, it is conceived as the number of (morphological, syntactic, semantic etc.) distinctions that a speaker has to master, and it is assessed by comparing different languages (McWhorter, 2001; Parkvall, 2008). On the other hand, in the perspective of psycholinguistics and cognitive science, the notion of complexity can be described as the *difficulty* encountered by language users while processing concrete linguistic realizations (sentences, utterances etc.) (Blache, 2011; Chersoni et al., 2016, 2017, 2021; Iavarone et al., 2021; Sarti et al., 2021). Finally, in the Computational Linguistics community, the assessment of complexity at the lexical level is often related to readability applications (Shardlow et al., 2020), with the goal of determining if a word in a given text will be difficult to understand for the language users. Such applications are extremely useful for second language learners, for speakers with relatively low literacy and for people with reading disabilities, helping to tailor the difficult level of the texts to the needs of the target users.

Task 1 of SemEval 2021 (Shardlow et al., 2021) aims at the development of systems for the estimation of lexical complexity in context, both for single words and for multiword expressions. The organizers provided two datasets with the target words in a sentence context, with annotations consisting of a mean of the complexity ratings assigned by humans. In our paper, we present the system developed by the PolyU CBS-Comp team for the competition. Our top system achieves a Pearson correlation of, respectively, 0.754 on the single words dataset and 0.659 on the multiword expressions one.

2 Related Work

In the earliest shared task on the lexical complexity problem, organized in 2016 (Paetzold and Specia, 2016), complexity was defined as a binary variable: given a word in context, the word will be judged as complex or not. Of course, this was a simplifying assumption, since there might be many situations where the boundary is not a clear-cut one, and annotators would rather indicate a value in a continuous scale. Moreover, the ”complex” words in the data only needed to be categorized as such by just one

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 565–570
Bangkok, Thailand (online), August 5–6, 2021. ©2021 Association for Computational Linguistics

of the annotators. A further study by Zampieri et al. (2017) analyzed the output of the participating systems, showing that modeling complexity as binary actually hindered their performance.

A second iteration of the shared task was organized in 2018 (Yimam et al., 2018), this time features two separate subtasks: the traditional binary classification task, where systems had to predict whether one word was complex or not, and a regression task, where systems had to estimate the probability that an annotator would have considered a given word as complex.

Recently, Shardlow et al. (2020) have introduced CompLex, a new gold standard for the estimation of lexical complexity in context for English: the corpus, including sentences from different textual genres, is annotated with the mean complexity ratings for the target words. As a preliminary evaluation, the authors presented the results of a linear regression model trained on sets of features including word and sentence embeddings and some hand-crafted features that are traditionally associated to complexity, such as frequency, word length and syllable count. The best scores, in terms of mean absolute error, were obtained when using only the latter set of features, while models based on the dimensions of the embeddings were lagging behind.

3 Datasets

The datasets for the shared task are part of the CompLex corpus, which has been published and described by Shardlow et al. (2020). The annotated sentences were collected using three different corpora: the Europarl corpus (Koehn, 2005), which includes the proceedings of the European Parliament; the CRAFT biomedical corpus (Bada et al., 2012); and the Bible, in the modern version of the World English Bible translation (Christodouloupoulos and Steedman, 2015).

The organizers selected targets as either single words (Sub-Task 1) or multiword expressions (Sub-Task 2), and the datasets include also multiple examples with the same target, as different contexts can determine different complexity values. As for the multiword expressions, they were identified via syntactic patterns, being either adjective-noun or noun-noun phrases.

20 annotations per data instance were collected, with annotators coming from different English-speaking countries (US, UK and Australia): the possible ratings ranged from 1 → Very Easy to 5 → Very Difficult. Mean scores were then normalized in the 0-1 range.

Dataset Instance	Corpus	Score
This was the **length** of Sarah's life.	Bible	0.125
... **dissenters** by definition excluded.	Europarl	0.688
...due to reduction in **adipose** tissue...	CRAFT	0.813

Table 1: Examples of the instances from the different corpora, together with the mean complexity scores for the target words in bold.

In a first phase, the organizers released a training data of 7661 samples for the single words track and 1517 samples for the multiword expressions track, together with a trial/validation dataset of 420 and 99 samples, respectively. Later, they released a test set of 917 samples for the single words track and 184 samples for the multiword expressions track.

Examples of the instances are shown in Table 1.

4 Evaluation

For both the single words and the multiword expressions track, we used the same set of features as input for a regression algorithm. In the multiword expressions track, we computed the value of the features for each of the two words in the target expression and then we took the average.

4.1 Features

As hand-crafted features, we adopted the same ones used by Shardlow et al. (2020) in the original evaluation of their dataset: **Logarithmic Frequency**, **Word Length** and **Syllable Length**. The latter two have been extracted using the Python `textstat` for each target word. As for the frequency feature, we extracted a general, out-of-domain frequency for each target word using the SUBTLEX database (Brysbaert and New, 2009) and the `wordfreq` Python package (Speer et al., 2018), and then we extracted the frequency of the word in each one of the three corpora composing CompLex. In total, we obtained 6 features (4 frequency + 2 length features) for each instance. We also added two Boolean features for **Capitalization**: the first was equal to 1 if the first letter of the target word was upper case and 0 otherwise; the second one was equal to 1 if all the letters of the target word were upper case and 0 otherwise. The latter feature was added because we noticed that some of the target words in the dataset are acronyms.

Apart from the lexical information, **Syntactic Features** were explored for both single words and multiword expressions. The StanfordNLP tools (Manning et al., 2014) were first used to acquire both the part-of-speech (POS) tags and dependency trees. POS tags of target words were manipulated using one-hot encoding, for a total of 20 POS-based features. On the other hand, directed and path from the target word to the root were extracted as dependency features. We concatenated all dependency tags to the root, using one-hot encoding once again to encode every distinct path as a single feature. In total, we generated 267 dependency paths features with this mechanism.

Another feature was based on **Word Embedding similarity**: first, we computed the sum of the embeddings for all the words preceding the target, as a sort of general representation of the sentence context [1], and then we measured the cosine similarity with the embedding of the target word. If the target was a multiword expression, we summed the embeddings of the words composing it. As word embeddings, we used the publicly available FastText vectors, pre-trained on the Wikipedia corpus (Bojanowski et al., 2017). [2]

We added one feature based on the **BERT Transformer Model** (Devlin et al., 2019) [3] by masking the target word in the original sentence and taking the probability value provided in output by the Softmax. For multiword expressions, we sequentially masked the words composing the target and took the average value.

Similarly, we used the **GPT-2 Transformer Model** (Radford et al., 2019) [4] to obtain a probability score for the full sentence, computed as the product of the probabilities of the single tokens.

The total number of extracted features is 300. Finally, we decided to generate polynomial features from our set, in order to exploit potential interactions. We used the `PolynomialFeatures` functionality of the `scikit-learn` Python package to generate interaction features of order 2, so that the final number of features that was fed to the regressors was 45151.

4.2 Regressors

We tested several regression algorithms, using the implementations in the `scikit-learn` Python package. The adopted scikit-learn API and the main hyper-parameters are listed below:

- RR `Ridge`: Ridge Regression solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm. `alpha`=1.0, `normalize`=True.
- MLP `MLPRegressor`: Multi-layer Perceptron regressor optimizes the squared-loss using LBFGS or stochastic gradient descent. `hidden layer size`=5, `activation`=identity, `solver`=adam.
- PLSR `PLSRegression`: PLS Regression implements the PLS2 blocks regression in case of one dimensional response. `components`=5.
- BRR `BayesianRidge`: a Bayesian Ridge model implements the optimization of the regularization parameters lambda and alpha. `alpha_1,alpha_2`==1.0e-6, `lambda_1,lambda_2`=1.0e-6.
- LR `LinearRegression`: Linear Regression is trained based on ordinary least squares function. `normalize`=True.
- RF `RandomForestRegressor`: a Random Forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. `min_samples_split`=2, `min_samples_leaf`=1.
- GBR `GradientBoostingRegressor`: Gradient Boosting builds an additive model in a forward stage-wise fashion which allows for the optimization of arbitrary differentiable loss functions. `learning rate`=0.1,`min_samples_split`=2, `min_samples_leaf`=1.

4.3 Metrics

The performance of the participating systems was evaluated in terms of Pearson correlation (r) between the outputs and the human mean ratings. In the Results section, we also report the scores

[1] The use of vector sum as a compositional function has been used in Distributional Semantics since Mitchell and Lapata (2010).

[2] `https://fasttext.cc/docs/en/pretrained-vectors.html`.

[3] We used the BERT-large-cased model, in the implementation of the Happy Transformer library: `https://github.com/EricFillion/happy-transformer`.

[4] We used the GPT2-xl model, in the implementation of the lm-scorer package: `https://pypi.org/project/lm-scorer/`.

for Spearman correlation (ρ), Mean Absolute Error (MAE), Mean Squared Error (MSE) and R-Square ($R2$).

5 Results

We evaluated our system for two subtasks based on given trial datasets. For each regressor, we tuned hyper-parameters according to each subtask. Performance evaluation has been carried out in two aspects: the assessment of the overall correlation with human ratings and the analysis of the contribution of the features.

5.1 Complexity Prediction

Evaluation metrics are reported ranking by Pearson correlation in Table 2 and 3 for single words and multiword expressions, respectively.

Regressor	r	ρ	MAE	MSE	$R2$
RR	0.34	0.36	0.100	0.017	0.097
MLP	0.39	0.46	0.099	0.017	0.083
PLSR	0.46	0.53	0.093	0.015	0.206
BRR	0.47	0.51	0.094	0.015	0.181
LR	0.48	0.54	0.092	0.015	0.208
RF	0.49	0.65	0.078	0.010	0.472
GBR	0.75	0.72	0.070	0.008	0.561

Table 2: Performance on single words prediction (Sub-Task 1).

Regressor	r	ρ	MAE	MSE	$R2$
RR	0.26	0.28	0.128	0.021	0.083
MLP	0.28	0.37	0.117	0.019	0.091
BRR	0.40	0.42	0.112	0.017	0.151
PLSR	0.40	0.42	0.109	0.017	0.178
LR	0.41	0.42	0.110	0.016	0.183
RF	0.44	0.51	0.105	0.015	0.424
GBR	0.66	0.66	0.090	0.013	0.427

Table 3: Performance of multiword expressions prediction (Sub-Task 2).

Features	r	ρ	MAE	MSE	$R2$
Hand-crafted	0.73	0.69	0.072	0.009	0.527
+Synt.	0.73	0.69	0.073	0.009	0.528
+Embs.	0.73	0.69	0.073	0.009	0.530
+Trans.	0.74	0.71	0.071	0.009	0.545
+ALL	0.75	0.72	0.070	0.008	0.561

Table 4: Ablation study of feature groups.

It can be observed that predicting the complexity of single words is naturally less difficult than multiword expression. Concerning the regression algorithm, gradient boosting regression outperforms other investigated methods by a large gap, while PLS regression, Bayesian ridge regression, linear regression and random forest regression perform very similarly. Though PLSR has a worse Pearson correlation than BRR, its R2 and Spearman correlation are slightly better. Further studies about regressors brought some unexpected results for our feature based approaches: based on the features we selected, Ridge Regression performs worse than linear regression, suggesting that some features are not suitable for applying L2-norm.

5.2 Feature Study

As our proposed method heavily relies on feature selection, the acquired features are investigated in four groups: *Hand-crafted* (including Logarithmic Frequency, Word and Syllable Length and Capitalization), *Syntactic* (including the POS- and the Dependency-based features), *Embedding* and *Transformer* features. We adopted the features of the *Hand-crafted* group as baseline, and present a comparison between the performance of systems using the other features as add-on components. The scores in Table 4 refer to the performance on the single words dataset, by using GBR as a regressor.

According to Table 4, syntactic, embedding and transformer based features can all contribute to improve the prediction results. As expected, the combination of all feature type groups can achieve the best predicting capability.

Comparing with the baseline of hand-crafted features, syntactic and embedding features have very marginal contribution. Yet, it should not be neglected supplementing only transformer based features cannot achieve the maximum performance gain. This indicates that the interaction of the individual features can bring latent useful information to model, further revealing the complexity values of the target words.

6 Conclusion

In this paper, we presented the PolyU CBS-Comp system for lexical complexity prediction, which took part in the SemEval shared task 1. Our method, based on a combination of lexical, syntactic, embeddings and Transformers features, achieved a 0.754 correlation on single words and 0.659 on multiword expressions, when using Gradient Boosting as a regression algorithm.

Traditional hand-crafted features, followed by Transformer-based ones, seem to give the strongest contribution to the classification performance, which is further improved by adding feature interactions to the input for the regressor.

For future studies on lexical complexity, we plan to further exploit the text genre information, for example by adding domain-adapted language model features (Van Schijndel and Linzen, 2018) to the information available to our models.

Acknowledgments

We acknowledge GRF grant (CERG PolyU 15211/14E, PolyU 152006/16E, PolyU YW4H), The Hong Kong Polytechnic University Postdoctoral Fellowships Scheme Projects.

References

Michael Bada, Miriam Eckert, Donald Evans, Kristin Garcia, Krista Shipley, Dmitry Sitnikov, William A Baumgartner, K Bretonnel Cohen, Karin Verspoor, and Judith A Blake. 2012. Concept Annotation in the CRAFT Corpus. *BMC Bioinformatics*, 13(1):1–20.

Philippe Blache. 2011. Evaluating Language Complexity in Context: New Parameters for a Constraint-based Model. In *Proceedings of the International Workshop on Constraints and Language Processing*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Marc Brysbaert and Boris New. 2009. Moving Beyond Kučera and Francis: A Critical Evaluation of Current Word Frequency Norms and the Introduction of a New and Improved Word Frequency Measure for American English. *Behavior Research Methods*, 41(4):977–990.

Emmanuele Chersoni, Philippe Blache, and Alessandro Lenci. 2016. Towards a Distributional Model of Semantic Complexity. In *Proceedings of the COLING Workshop on Computational Linguistics for Linguistic Complexity*.

Emmanuele Chersoni, Alessandro Lenci, and Philippe Blache. 2017. Logical Metonymy in a Distributional Model of Sentence Comprehension. In *Proceedings of *SEM*.

Emmanuele Chersoni, Enrico Santus, Alessandro Lenci, Philippe Blache, and Chu-Ren Huang. 2021. Not All Arguments Are Processed Equally: A Distributional Model of Argument Complexity. *Language, Resources and Evaluation*, pages 1–28.

Christos Christodouloupoulos and Mark Steedman. 2015. A Massively Parallel Corpus: The Bible in 100 Languages. *Language, Resources and Evaluation*, 49(2):375–395.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*.

Benedetta Iavarone, Dominique Brunato, and Felice Dell'Orletta. 2021. Sentence Complexity in Context. In *Proceedings of the NAACL Workshop on Cognitive Modeling and Computational Linguistics*.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit*, volume 5, pages 79–86. Citeseer.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of ACL: System Demo*.

John H McWhorter. 2001. The World's Simplest Grammars Are Creole Grammars. *Linguistic Typology*, 5(2-3):125–166.

Jeff Mitchell and Mirella Lapata. 2010. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429.

Gustavo Paetzold and Lucia Specia. 2016. Semeval 2016 Task 11: Complex Word Identification. In *Proceedings of SemEval*.

Mikael Parkvall. 2008. The Simplicity of Creoles in a Cross-linguistic Perspective. *Language Complexity: Typology, Contact, Change*, pages 265–285.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models Are Unsupervised Multitask Learners. *OpenAI Blog*, 1(8):9.

Gabriele Sarti, Dominique Brunato, and Felice Dell'Orletta. 2021. That Looks Hard: Characterizing Linguistic Complexity in Humans and Language Models. In *Proceedings of the NAACL Workshop on Cognitive Modeling and Computational Linguistics*.

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex–A New Corpus for Lexical Complexity Predicition from Likert Scale Data. In *Proceedings of the LREC Workshop on Tools and Resources to Empower People with REAding DIfficulties*.

Matthew Shardlow, Richard Evans, Gustavo Paetzold, and Marcos Zampieri. 2021. SemEval-2021 Task 1: Lexical Complexity Prediction. In *Proceedings of SemEval*.

Robyn Speer, Joshua Chin, Andrew Lin, Sara Jewett, and Lance Nathan. 2018. Luminosoinsight/wordfreq: v2.2.

Marten Van Schijndel and Tal Linzen. 2018. A Neural Model of Adaptation in Reading. In *Proceedings of EMNLP*.

Seid Muhie Yimam, Chris Biemann, Shervin Malmasi, Gustavo H 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 NAACL Workshop on Innovative Use of NLP for Building Educational Applications*.

Marcos Zampieri, Shervin Malmasi, Gustavo Paetzold, and Lucia Specia. 2017. Complex Word Identification: Challenges in Data Annotation and System Performance. In *Proceedings of the IJCNLP Workshop on NLP Techniques for Educational Applications*.