The 35th Conference on Computational Linguistics and Speech Processing (ROCLING 2023)
Taipei City, Taiwan, October 20-21, 2023. The Association for Computational Linguistics and Chinese Language Processing

Lexical Complexity Prediction using Word Embeddings

Cheng-Zen Yang[1], Jin-Jian Li[1], and Shu-Chang Lin[2]
[1] Department of Computer Science and Engineering
[2] Department of Foreign Languages and Applied Linguistics
Yuan Ze University
Taoyuan, Taiwan
{czyang,shuchang}@saturn.yzu.edu.tw, jjl18@syslab.cse.yzu.edu.tw

Abstract

Lexical complexity is crucial for reading comprehension. In the past, research work of lexical complexity prediction mainly focuses on differentiating the complexity difference between two words. Moreover, most of the previous lexical complexity prediction approaches only consider traditional lexically relevant features. In this paper, we propose a novel supervised approach using word embeddings features to tackle the lexical complexity prediction problem as a single-label multi-classification problem. We discuss four word embeddings techniques including Word2Vec, fastText, GloVe, and BERT. We also discuss five classification models including *k*-Nearest Neighbors, Support Vector Machines, Multilayer Perception, Random Forest, and XGBoost. The prediction models are evaluated with three datasets in English, Traditional Chinese, and Japanese. The results show that SVM with fastText can achieve the highest accuracy of 66.23% for the English dataset. SVM with GloVe can achieve the highest accuracy of 53.84% for the Traditional Chinese dataset. SVM with Word2Vec can achieve the highest accuracy of 49.96% for the Japanese dataset.

Keywords: Lexical Complexity, Word Embeddings, Classification Models

1 Introduction

In many human reading activities, lexical complexity plays an important role for reading comprehension (North et al., 2023). Lexical complexity prediction, therefore, becomes a crucial sub-task for various natural language understanding (NLU) tasks. For example, in the lexical simplification task (Specia et al., 2012) aiming to replace words that are difficult for readers to completely understand with alternative words that can be more easily understood, identifying complex words by estimating their lexical complexity degree enables the simplification systems to find complex words for the following replacements (Shardlow, 2013, 2014).

To predict lexical complexity levels of words, various approaches (Elhadad & Sutaria, 2007; Keskisärkkä, 2012; Maddela & Xu, 2018; North et al., 2023; Paetzold & Specia, 2010; Shardlow, 2013; Zeng et al., 2005) have been proposed to identify complex words. For example, word frequency thresholds are used to identify complex words in the work of Shardlow (Shardlow, 2013) and the work of Keskisärkkä (Keskisärkkä, 2012). Elhadad et al. proposed a lexicon-based approach to recognize complex words (Elhadad & Sutaria, 2007). The identification performance of approaches based on Support Vector Machines (SVM) (Vapnik, 1996) are evaluated in the work of Zeng et al. (Zeng et al., 2005) and the work of Shardlow (Shardlow, 2013). However, these lexical complexity prediction approaches only consider traditional lexically relevant features, such as word frequencies and Part-of-Speech attributes, to construct the prediction models. Since past studies show that word embeddings features can effectively represent the semantic space to capture semantic relations (Incitti et al., 2023), lexical complexity of the context is latently embedded in the lexical co-occurrence space. As shown in (Shardlow et al., 2022), collocations are an important feature to assess lexical complexity. In this paper, we propose a novel supervised approach using word embeddings features to facilitate the construction of lexical complexity prediction models.

In this paper, four word embeddings techniques are investigated to study their effectiveness:

Word2Vec (Mikolov, Sutskever, et al., 2013), fastText (Bojanowski et al., 2017; Joulin et al., 2017), GloVe (Pennington et al., 2014), and BERT (Devlin et al., 2018). Five machine learning models are studied to explore their prediction performance: *k*-Nearest Neighbors (Fix & Hodges, 1989), Support Vector Machines (SVM) (Vapnik, 1996), Multilayer Perception (MLP) (Rumelhart et al., 1986), Random Forest (RF) (Breiman, 2001), and XGBoost (Chen & Guestrin, 2016). Different with the previous lexical simplification studies that consider the lexical complexity prediction problem as a binary classification problem, the proposed approach tackles this prediction problem as a single-label multi-classification problem.

To investigate the effectiveness of the proposed approach, experiments use three datasets in English, Traditional Chinese, and Japanese. The experimental results show that SVM with fastText achieves the highest accuracy of 66.23% for the English dataset, SVM with GloVe achieves the highest accuracy of 53.84% for the Traditional Chinese dataset, and SVM with Word2Vec achieves the highest accuracy of 49.96% for the Japanese dataset. Overall, SVM has the best accuracy performance among all studied classification models and fastText performs well on average among all studied word embeddings techniques.

The remainder of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 describes the proposed lexical complexity prediction approach and the studied machine learning models. Section 4 presents the datasets. Section 5 describes the experiments and discusses the results. Finally, Section 6 concludes the paper.

2 Related Work

In 2005, Zeng et al. (2005) proposed a text corpora-based approach using Support Vector Machines to predict term feasibility. In 2007, Elhadad and Sutaria (2007) proposed an unsupervised corpus-driven method to construct a lexicon in which medical terms are paired with semantically equivalent lay terms. They identified words using Part-of-Speech tagging and designed a semantic filter method by considering alternative association measures and UMLS (Unified Medical Language System) features.

In 2012, Keskisärkkä (2012) proposed an automatic lexical simplification approach to substitute words with other alternatives according to word frequencies, word lengths, and levels of synonyms. In the work of Shardlow (2013), a word frequency thresholding approach is evaluated against two other lexical simplification methods (i.e., the all-simplifying method and the SVM-based method). The experimental results show that the SVM-based method can effectively identify complex words to achieve the highest precision performance. However, it achieves a slightly lower recall performance because more complex words are misclassified as simple words.

In 2018, Maddela et al. (2018) proposed a Neural Readability Ranker (NRR) model with a Gaussian-based feature vectorization layer trained by a human-labeled lexical complexity lexicon for lexical simplification. To train the NRR model, they first built a word complexity lexicon with 15,180 words. Therefore, NRR can determine the relative complexity between two words. However, NRR does not consider how to predict lexical complexity for each word.

3 Prediction Methodology

This section first describes the proposed prediction approach. It then describes the word embeddings techniques discussed in this paper. The studied machine learning models are then presented.

3.1 Proposed Prediction Approach

Word embeddings technology has been proven very effective in many natural language processing (NLP) tasks (Incitti et al., 2023). With word embeddings processing, words are represented as *n*-dimensional numerical vectors to capture the semantic meaning based on contextual information. Since the vectors are derived from their surrounding context information, lexical complexity information is implicitly embedded in the word embeddings representation. Therefore, this paper proposes a supervised approach to predict the lexical complexity levels of words based on their word embeddings.

In this paper, the lexical complexity prediction problem is defined as a single-label multi-classification problem. Given a set of M predefined lexical complexity levels $C = \{c_1, c_2, \dots, c_M\}$ and a labeled lexicon L, the lexical complexity level of a word w_i is predicted to a lexical complexity level $c_j \in C$ as follows:

$$c_j = h(w_i|L) \quad (1)$$

where h is a trained classification model.

Figure 1 illustrates the proposed prediction model. If there is not an available pre-trained word embeddings corpus to train a classification model, word corpus data are first collected to generate the corresponding word embeddings. Before the word embeddings processing, the word corpus data are preprocessed. The details will be elaborated in Section 4 that describes the experimental datasets.

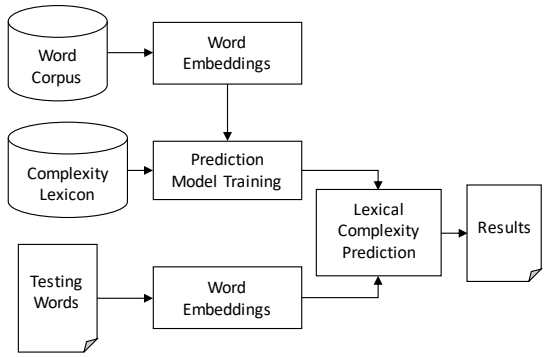

Figure 1: Lexical complexity prediction process.

The word embeddings and a complexity-labeled lexicon are the training data for the supervised classification models. In this paper, four well-known word embeddings models are studied: Word2Vec (Mikolov, Sutskever, et al., 2013), fastText (Bojanowski et al., 2017; Joulin et al., 2017), GloVe (Pennington et al., 2014), and BERT (Devlin et al., 2018). These word embeddings models are considered because they have shown the effectiveness in many NLP tasks (Incitti et al., 2023). In addition to the investigation of the word embeddings models, five supervised classification models are studied because they have been widely used in many machine learning tasks (Sen et al., 2020). The studied classification models include k-Nearest Neighbors (Fix & Hodges, 1989), Support Vector Machines (SVM) (Vapnik, 1996), Multilayer Perception (MLP) (Rumelhart et al., 1986), Random Forest (RF) (Breiman, 2001), and XGBoost (Chen & Guestrin, 2016).

3.2 Word Embeddings Models

Word embeddings can effectively capture the semantic meaning based on contextual information. Lexical complexity of the context is thus latently embedded in the lexical co-occurrence space. This paper investigates the following four word embedding techniques.

Word2Vec: Word2Vec (Mikolov, Chen, et al., 2013) is one of the most common word embeddings techniques. With Word2Vec, a word in a corpus is represented as an n-dimensional numerical vector in the semantic space. As shown in the work (Mikolov, Chen, et al., 2013), these word embeddings can capture semantic similarity relationship of words.

In the Word2Vec model, there are two approaches to derive word embeddings: the Continuous Bag-of-Words (CBOW) model and the Continuous Skip-gram model. The CBOW model derives the vector representation of the target word based on the context words surrounding it. In the Skip-gram model, the word embeddings of words surrounding a given word is derived from the given word. According to the study (Jang et al., 2019), the CBOW model can achieve higher accuracy performance and is more stable than the Skip-gram model.

fastText: To tackle the shortage of Word2Vec that does not consider subword information, Bojanowski et al. proposed fastText (Bojanowski et al., 2017). In fastText, syntactic relations of words are extracted to enhance word embeddings representation for morphologically rich languages.

GloVe: Pennington et al. (2014) proposed GloVe to derive word embeddings by considering global lexical co-occurrence statistics from the given corpus. Instead of using the entire sparse co-occurrence matrix of the corpus or the surrounding context, GloVe word embeddings are trained using only the non-zero elements of the co-occurrence matrix. Therefore, the vector space is constructed with substructures that are more meaningful.

BERT: In 2018, Devlin et al. proposed the BERT (Bidirectional Encoder Representations from Transformers) model (Devlin et al., 2018). BERT utilizes the bi-directional encoder representation of the transformer for unsupervised pre-training. In contrast to previous techniques, BERT takes into account the context of the target word to understand the meaning of the text.

3.3 Supervised Classification Models

Since words are represented as n-dimensional numerical vectors, the proposed approach employs supervised classification models to classify each word into a lexical complexity class. This paper investigates the following five supervised classification models.

***k*NN**: The *k*-Nearest Neighbors (*k*NN) (Fix & Hodges, 1989) model considers k nearest samples of a sample x to decide the class of x. If most of the k nearest samples belong to class Y, *k*NN classifies x to the class Y. Thus the intent of using *k*NN is that the lexical complexity of a word will be determined by the word embeddings in the neighborhood of the word in the semantic space.

SVM: Vapnik et al. proposed the Support Vector Machines (SVM) model to learn a hyperplane from the training data that distinguishes the training data. (Vapnik, 1996). According to its characteristics, using SVM is based on a hypothesis that word embeddings of the same lexical complexity are tentatively aggregated together in the semantic space. Therefore, SVM can find hyperplanes for lexical complexity prediction.

MLP: The Multilayer Perceptron (MLP) model is a feedforward artificial neural network using perceptron extension (Rumelhart et al., 1986). In the training process, the parameters of the MLP model are adjusted according to the numerical values of training word embeddings. The idea of using MLP is that different dimensions should be weighted differently for lexical complexity prediction.

RF: Breiman proposed the Random Forest (RF) model by considering multiple decision trees that are weak classification models (Breiman, 2001). RF combines the decision tree output to determine the final classification result. Since multiple weak classification models are aggregated, the classification performance can be improved. Therefore, the idea of using RF is to construct an ensemble learning model of many decision trees considering variations of word embeddings in different dimensions.

XGBoost: Chen et al. proposed the eXtreme Gradient Boosting (XGBoost) (Chen & Guestrin, 2016) by extending the Gradient Boosted Decision Tree (GBDT) model. Unlike RF, XGBoost is a boosting method and RF is a bagging method. In the training process, XGBoost optimizes the parameters to tackle the overfitting problem. Similar to the idea of using RF, the idea of using XGBoost is to consider word embedding variations in different dimensions, but applying the boosting approach.

4 Datasets

Two corpora are needed to perform lexical complexity prediction using word embeddings: a pre-trained word embeddings corpus and a labeled complexity corpus. These corpora are detailed in the following subsections.

4.1 Word Embeddings Datasets

In this study, we prioritize the use of available word embeddings corpora. If there is no appropriate corpus, we retrain a new one.

For English Word2Vec word embeddings, this study uses a pre-trained Google News corpus[1] of 3 million words. This corpus is trained using the CBOW model. The dimensionality is 300. For English GloVe word embeddings, this study uses a pre-trained GloVe corpus[2] having 6 billion 300-dimension word embeddings, `glove.6B.300d.txt`. For English fastText word embeddings, a pre-trained 300-dimention corpus[3], `wiki-news-300d-1M.vec`, is used. For English BERT word embeddings, this study uses a pre-trained 768-dimension corpus[4] that was released by MXNet using GluonNLP.

For Traditional Chinese word embeddings, we retrain the corpora using a Traditional Chinese Wikipedia dataset dumped on March 20, 2021 from the Traditional Chinese Wikipedia[5]. To retrain Chinese Word2Vec word embeddings, genism 3.7.3[6] is used with the CBOW model, the window size is 5, and the dimension is 300. These parameter settings are also used to retrain Traditional Chinese fastText word embeddings. To retrain Traditional Chinese GloVe word embeddings, we used GloVe[7] with a window size of 5 and 300 dimensions. To retrain Traditional Chinese BERT word embeddings, we use the traditional Chinese transformers model of CKIPLab[8] to extract the last hidden states. The dimensionality is 768.

For Japanese word embeddings, we use the same tools with the same parameters to retrain the Word2Vec, fastText and GloVe corpora using a

[1] https://drive.google.com/file/d/0B7XkCwpI5KDYNlNUTTlSS21pQmM/edit
[2] https://nlp.stanford.edu/projects/glove/
[3] https://fasttext.cc/docs/en/english-vectors.html
[4] https://pypi.org/project/bert-embedding/
[5] https://dumps.wikimedia.org/zhwiki/
[6] https://pypi.org/project/gensim
[7] https://github.com/stanfordnlp/GloVe
[8] https://huggingface.co/ckiplab/bert-base-chinese

Japanese Wikipedia dataset dumped on May 1, 2021. For Japanese BERT word embeddings, we use a pre-trained BERT model [9] trained with Japanese Wikipedia dumped on September 1, 2019. This model has the same model architecture as the original BERT and is trained using the same parameters. The model has 32000 words.

4.2 Lexical Complexity Datasets

For the English lexical complexity corpus, this study uses the WC15180 dataset (Maddela & Xu, 2018). The lexical complexity of each word in WC15180 is assessed by 11 non-native but fluent English speakers of different native languages. There are six complexity levels in WC15180: Very Simple (Level 1), Moderately Simple (Level 2), Simple (Level 3), Complex (Level 4), Moderately Complex (Level 5), and Very Complex (Level 6). Table 1 illustrates six WC15180 examples.

Word	Level
east	1
wet	2
producer	3
hypothetical	4
interdisciplinary	5
dehydrogenase	6

Table 1: Lexical complexity examples in WC15180.

For the Traditional Chinese lexical complexity corpus, we use a lexical corpus of 3 Grades and 7 Levels (三等七級詞語表)[10] released by National Academy for Educational Research (NAER). This corpus has 14,470 words. However, this corpus needs to be preprocessed. First, multiple words are regarded as a group in the corpus such as "一點/一點點/一點兒" (a little). For this word group, we divide it into three words of the same level. In this study, a total of 365 word groups are divided into 753 words. Second, a word may have different lexical complexity levels because of its different pronunciations. For example, the complexity level is Level 4 for "人家" (house/folk) when the word is pronounced as "rén jiā" (house/family status), but its level becomes Level 5 when the word is pronounced as "rén jia" (folk/people). In this study, the lowest level among multiple levels of the word is used as the representative level of the word. Therefore, the complexity level is Level 4 for "人家" in this study. The corpus has a total of 83 words of this condition. After these processing steps, there are 14,772 Chinese words. Table 2 illustrates seven examples in the NAER corpus.

Word	Level
一些 (some, a few)	1
西瓜 (watermelon)	2
雜誌 (magazine)	3
賺錢 (earn/make money)	4
出路 (outlet/way out)	5
興高采烈 (rejoicing)	6
無懈可擊 (invulnerability)	7

Table 2: Lexical complexity examples in the NAER corpus.

For the Japanese lexical complexity corpus, use the Japanese Language Education Vocabulary List (JEV) (日本語教育語彙表)[11](李在鎬, 2013). It contains 17,920 words divided into six levels: the first half of the elementary level (Level 1), the second half of the elementary level (Level 2), the first half of the intermediate level (Level 3), the second half of the intermediate level (Level 4), the first half of the advanced level (Level 5), and the second half of the advanced level (Level 6). However, a Japanese word may have different pronunciations and thus has different lexical complexity levels. For example, "今日" (today) can be pronounced as "キョウ" (kyo) or "コンニチ" (konnichi). The complexity level of the former is Level 1 but the level of the latter is Level 4. The situation becomes more complex when the word has different morphological variants. For example, "時" is pronounced as "ジ" (zi). When the type of "時" is "Suffix-Noun-Measure Word" (接尾辞-名詞的-助数詞) (e.g., 九時, ku zi), the level is 1. When its type is "Suffix" (接尾辞) (e.g., 使用時, shiyo zi), the complexity level becomes Level 2. As the process in Chinese, the lowest level among multiple levels of a Japanese word is used as the representative level. After these processing steps,

[9] https://huggingface.co/cl-tohoku/bert-base-japanese-whole-word-masking

[10] https://coct.naer.edu.tw/download/tech_report/

[11] http://jhlee.sakura.ne.jp/JEV/

there are 17,207 Japanese words. Table 3 illustrates six JEV examples.

Word	Level
午前 (morning)	1
言葉 (words)	2
最高 (the best)	3
予感 (premonition)	4
公共放送 (public broadcasting)	5
疾走 (dash)	6

Table 3: Lexical complexity examples in the JEV corpus.

5 Experiments

5.1 Experiment Settings

In the experiment of this research, we implemented each classification model with Python 3.8.5, Scikit-learn 0.23.1[12], and xgboost 1.3.1[13]. We used Jieba 0.42.1[14] as the Chinese segmentation tool, and MeCab 0.996.2[15] as the Japanese segmentation tool. The word embeddings retraining models are described in Section 4.1. The parameters used for the MLP model are shown in Table 4. The number of trees (`n_estimators`) in the RF model is 100. The default parameter settings are used for other classification model.

Parameter	Settings
Solver	lbfgs
Alpha	le-5
Hidden layer	sizes (5,5)
Random state	1

Table 4: Settings for the MLP model.

To measure the prediction performance, we only used words that appear in both the word embeddings dataset and the lexical complexity corpus for training and testing. Table 5 shows the numbers of words used in the experiments. Table 6 shows the complexity distributions of the datasets of three languages.

Language	Voc. Size
English	9,110
Traditional Chinese	14,538
Japanese	15,516

Table 5: Vocabulary sizes in the experiments.

	English	Trad. Chinese	Japanese
Level 1	606	435	370
Level 2	3646	435	687
Level 3	3730	472	1969
Level 4	1027	1463	5573
Level 5	88	2678	5606
Level 6	13	4159	1311
Level 7	–	4896	–
Total	9110	14538	15516

Table 6: Complexity distributions of the datasets of three languages.

In the experiments, we used the stratified 10-fold cross validation approach to measure the prediction accuracy of the test data. The measures of 10 folds are averaged for performance comparison.

5.2 Results and Discussions

Table 7 shows the experimental results for the English corpus. From the table we can find that the SVM model with fastText word embeddings can achieve the best accuracy performance of 66.23%. From the point of view of word embeddings, GloVe performs the best. However, if the kNN model is excluded in the performance comparison, fastText has the best performance.

	kNN	SVM	MLP	RF	XGB	Avg.
W2V	40.76%	64.60%	59.69%	57.37%	60.18%	56.52%
GloVe	55.59%	64.72%	61.04%	59.13%	61.60%	**60.42%**
fastText	48.91%	**66.23%**	63.70%	59.01%	62.38%	60.05%
BERT	55.79%	61.72%	63.13%	57.57%	61.19%	59.88%
Avg.	50.26%	**64.32%**	61.89%	58.27%	61.34%	

Table 7: Complexity prediction accuracy performance for the English corpus.

From the table we can also find that the kNN model has the worst performance. This reveals that considering semantically similar words to predict the lexical complexity is ineffective. From the following experimental results of Traditional Chinese and Japanese, we can also find that the kNN model has the poorest performance.

[12] https://scikit-learn.org
[13] https://pypi.org/project/xgboost/
[14] https://github.com/fxsjy/jieba
[15] https://taku910.github.io/mecab/

Table 8 shows the experimental results for the Traditional Chinese corpus. The results show that the SVM model with GloVe word embeddings can achieve the best accuracy performance of 53.84%. fastText has the best performance among all word embeddings techniques. The performance of BERT is very poor. A possible reason is that the Wikipedia document size for BERT word embeddings retraining is still small. Among all classification models, SVM has the best accuracy performance. Among all word embeddings approaches, fastText has the best prediction performance.

	kNN	**SVM**	**MLP**	**RF**	**XGB**	Avg.
W2V	41.90%	53.35%	51.22%	46.11%	48.93%	48.30%
GloVe	40.89%	**53.84%**	52.44%	44.54%	48.10%	47.96%
fastText	42.26%	53.10%	50.34%	46.72%	49.37%	**48.36%**
BERT	30.02%	37.14%	37.58%	36.26%	37.14%	35.63%
Avg.	38.77%	**49.36%**	47.90%	43.41%	45.89%	

Table 8: Complexity prediction accuracy performance for the Traditional Chinese corpus.

Table 9 shows the experimental results for the Japanese corpus. The results show that SVM with Word2Vec can achieve the best accuracy performance of 49.96% and SVM has the best average accuracy performance among all classification models. From the point of view of word embeddings, Word2Vec has the best performance for the Japanese corpus, followed by fastText. Similarly, the performance of BERT is still poor as the Traditional Chinese experiments. This may be due to the same reason of small Wikipedia data size.

	kNN	**SVM**	**MLP**	**RF**	**XGB**	Avg.
W2V	42.10%	**49.96%**	46.51%	45.17%	46.95%	**46.14%**
GloVe	39.93%	49.22%	47.42%	42.67%	45.30%	44.91%
fastText	41.99%	48.86%	46.22%	44.68%	47.08%	45.77%
BERT	36.07%	42.05%	40.96%	40.26%	40.58%	39.98%
Avg.	40.02%	**47.52%**	45.28%	43.20%	44.98%	

Table 9: Complexity prediction accuracy performance for the Japanese corpus.

From the above experimental results, we can find that fastText performs well on average in lexical complexity prediction. From the point of view of the classification model, SVM can be a good choice in lexical complexity prediction.

6 Conclusions

Lexical complexity is crucial for reading comprehension. In lexical simplification tasks, complex words are replaced with simpler words to ease human understanding. Effectively identifying complex words by predicting their lexical complexity levels becomes a research issue known as the lexical complexity prediction problem.

In the past, research work of lexical complexity prediction mainly focuses on differentiating the complexity difference between two words. The lexical complexity prediction problem is tackled as a binary classification problem. Moreover, most of the previous lexical complexity prediction approaches only consider traditional lexically relevant features. In this paper, we propose a novel supervised approach using word embeddings features to tackle the lexical complexity prediction problem as a single-label multi-classification problem. Since word embeddings features can effectively represent the semantic space to capture semantic relations, the proposed approach employs these features to predict lexical complexity.

We have conducted experiments to study the effectiveness of four word embeddings techniques and five classification models. The prediction models are evaluated with three datasets in English, Traditional Chinese, and Japanese. The experimental results show that SVM with fastText achieves the highest accuracy of 66.23% for the English dataset, SVM with GloVe achieves the highest accuracy of 53.84% for the Traditional Chinese dataset, and SVM with Word2Vec achieves the highest accuracy of 49.96% for the Japanese dataset. Overall, SVM has the best accuracy performance among all studied classification models and fastText performs well on average among all studied word embeddings techniques.

There are research issues that will be investigated in the future. First, a comprehensive study on other word embeddings techniques and classification models are planned to explore their effectiveness. For transformer-based word embeddings techniques like BERT, the training data size will be studied to observe its influences. For words that are not included in the labeled corpora, human evaluation will be the next step to verify the effectiveness of the proposed approach. It is expected that the proposed prediction approach can

be utilized to save valuable labor and time costs for lexical complexity assessment.

Acknowledgments

This work was supported in part by the National Science and Technology Council, Taiwan, ROC, under the Grant NSTC 112-2813-C-155-039-E.

References

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. https://doi.org/10.1162/tacl_a_00051

Leo Breiman. 2001. Random Forests. *Machine Learning*, *45*(1), 5-32. https://doi.org/https://doi.org/10.1023/A:1010933404324

Tianqi Chenand Carlos Guestrin. 2016. *XGBoost: A Scalable Tree Boosting System.* In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*, pp. 785-794. https://doi.org/10.1145/2939672.2939785

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, *arXiv:1810.04805*. http://arxiv.org/abs/1810.04805

Noemie Elhadadand Komal Sutaria. 2007. *Mining a Lexicon of Technical Terms and Lay Equivalents.* In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing (BioNLP '07)*, pp. 49–56.

Evelyn Fixand Joseph L. Hodges. 1989. Discriminatory Analysis - Nonparametric Discrimination: Consistency Properties. *International Statistical Review/Revue Internationale de Statistique*, *57*(3), 238-247. https://doi.org/https://doi.org/10.2307/1403797

Francesca Incitti, Federico Urli, and Lauro Snidaro. 2023. Beyond word embeddings: A survey. *Information Fusion*, *89*, 418-436. https://doi.org/10.1016/j.inffus.2022.08.024

Beakcheol Jang, Inhwan Kim, and Jong Wook Kim. 2019. Word2vec convolutional neural networks for classification of news articles and tweets. *PLOS ONE*, *14*(8), e0220976. https://doi.org/10.1371/journal.pone.0220976

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. *Bag of Tricks for Efficient Text Classification.* In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2017)*, pp. 427-431.

Robin Keskisärkkä. 2012. *Automatic Text Simplification via Synonym Replacement* Linköping University].

Mounica Maddelaand Wei Xu. 2018. *A Word-Complexity Lexicon and A Neural Readability Ranking Model for Lexical Simplification.* In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP 2018)*, pp. 3749-3760.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. *Efficient Estimation of Word Representations in Vector Space.* In *Proceedings of the 1st International Conference on Learning Representations (ICLR 2013)*, https://arxiv.org/abs/1301.3781

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. *Distributed Representations of Words and Phrases and their Compositionality.* In *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013)*, pp. 3111-3119. https://arxiv.org/abs/1310.4546

Kai North, Marcos Zampieri, and Matthew Shardlow. 2023. Lexical Complexity Prediction: An Overview. *ACM Comput. Surv.*, *55*(9), Article 179. https://doi.org/https://doi.org/10.1145/3557885

Gustavo Paetzoldand Lucia Specia. 2010. *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)*, pp. 969-974.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. *GloVe: Global Vectors for Word Representation.* In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pp. 1532-1543. https://www.aclweb.org/anthology/D14-1162

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning Internal Representations by Error Propagation. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*. pp. 318-362. MIT Press.

Pratap Chandra Sen, Mahimarnab Hajra, and Mitadru Ghosh. 2020. *Supervised Classification Algorithms in Machine Learning: A Survey and Review.* In *Emerging Technology in Modelling and Graphics*, pp. 99-111.

Matthew Shardlow. 2013. *A Comparison of Techniques to Automatically Identify Complex Words.* In *Proceedings of the Student Research Workshop at the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 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)*, pp. 1583-1590. http://www.lrec-conf.org/proceedings/lrec2014/index.html

Matthew Shardlow, Richard Evans, and Marcos Zampieri. 2022. Predicting lexical complexity in English texts: the Complex 2.0 dataset. *Language Resources and Evaluation*, *56*(4), 1153-1194. https://doi.org/10.1007/s10579-022-09588-2

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. 2012. *SemEval-2012 Task 1: English Lexical Simplification.* In *Proceedings of the 1st Joint Conference on Lexical and Computational Semantics (*SEM 2012)*, pp. 347–355. https://aclanthology.org/S12-1046

Vladimir Vapnik. (1996). *The Nature of Statistical Learning Theory* (1st ed.). Springer-Verlag.

Qing Zeng, Eunjung Kim, Jon Crowell, and Tony Tse. 2005. *A Text Corpora-Based Estimation of the Familiarity of Health Terminology.* In *Proceedings of the 6th International Symposium on Biological and Medical Data Analysis (ISBMDA 2005)*,

李在鎬. 2013. *大規模コーパスに基づく語彙リストの検証*. In *マレーシア研究集会*, pp. 10-19.