Letter2vec+Freq2vec: A Comprehensive Approach to Text Vectorization and Predicting Word Difficulty

Songru Li[1], Yuan Cheng[1], Hanbin Lu[2, *]

[1]School of Electrical Engineering and Artificial Intelligence, Xiamen University Malaysia, Selangor, Malaysia

[2]School of Mathmatics and Physics, Xiamen University Malaysia, Selangor, Malaysia

*Corresponding author: 13705720232@163.com

Abstract: This paper introduces the self-developed word embedded algorithm Letter2vec+Freq2vec. It combines with biKmeans and Stacking algorithm to judge word difficulty in Wordle. The algorithm combines Letter2vec, which models letter interactions and context, and Freq2vec, which calculates letter frequencies at different positions of words to produce a more comprehensive text vectorization method. The results show that this algorithm can get better results than BERT algorithm in small sample size when vectorizing a single word. Overall, this paper provides insights into the development and application of new algorithms for text vectorization and the prediction of word difficulty.

Keywords: Freq2vec+Letter2vec, BERT, biKmeans, Natural Language Processing.

1. Introduction

Wordle is a popular puzzle game provided by the New York Times, in which players try to solve a puzzle through six or fewer guesses that elicit feedback, leading to a five-letter word. In this study, we hope to complete the classification prediction of word difficulty after the release of the word through a reasonable mathematical model. If you want to complete the analysis and prediction through the mathematical model, you must face a complex nonlinear problem, how to convert text data (words) into data.

In the field of natural language processing, BERT, ERNIE are popular pre-trained language models in natural language processing. In 2018, the BERT model proposed by Google is a milestone in the field of natural language processing [1]. Text vectorization research has gradually shifted to the direction of deep learning, using pre-training technology and self-attention mechanism to solve some problems in text vectorization to a certain extent. In China, the ERNIE model proposed by Baidu uses multi-task learning and knowledge distillation technology to vectorize text and achieves good results [2]. However, they have limitations such as large model size and long training time. Research focuses on improving efficiency and adapting to more tasks/languages.

In this paper, a new algorithm called Letter2vec+Freq2vec is proposed. It combines Letter2vec, which borrows from word2vec to model letter interactions, and Freq2vec, which considers letter frequency in different positions of words. This produces a more comprehensive approach to text vectorization with limited sample size.

2. Construction of word difficulty predict model

Figure 1 shows the thought flow of the whole paper. In this part of the report, the clustering algorithm is firstly used to conduct unsupervised clustering on the features related to word difficulty as shown in Table 1. The output clustering results are evaluated and used as labels. The vector-quantized data of word is used as features for multiple regression prediction.

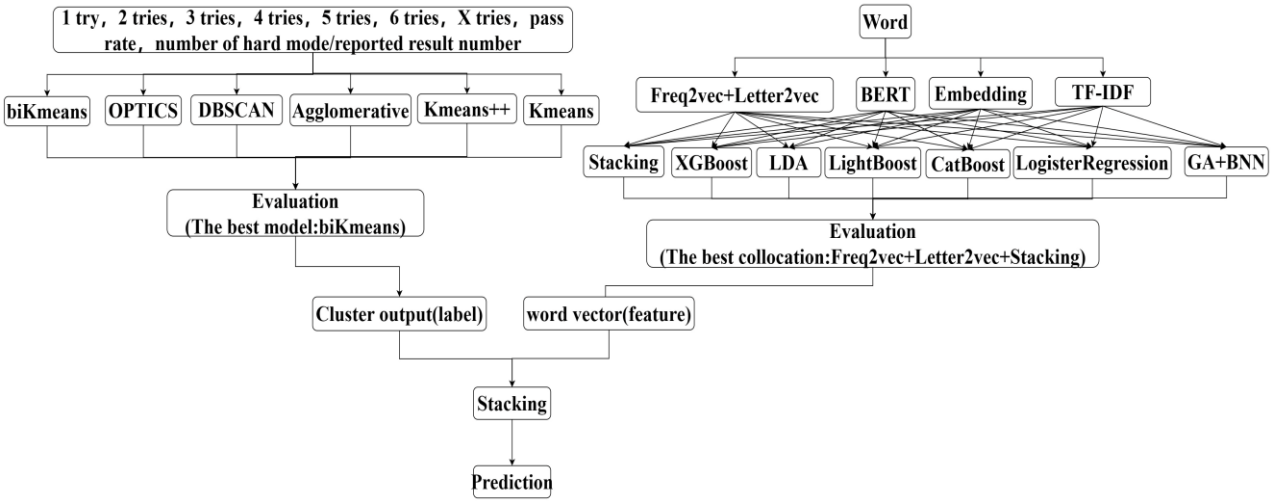

Figure 1. The model construction process

2.1. Feature Selection

The features that are utilized in clustering algorithm are shown in Table 1.

Table 1. The relevant features

ID	Feature	Comment
1	1 try	Proportion of pass in 1 try
2	2 tries	Proportion of pass in 2 tries
3	3 tries	Proportion of pass in 3 tries
4	4 tries	Proportion of pass in 4 tries
5	5 tries	Proportion of pass in 5 tries
6	6 tries	Proportion of pass in 6 tries
7	X tries	Proportion of try in 7 or more times
8	Percentage of hard mode	number of hard mode/numbers of reported result
9	Pass rate	sum (1 try,2 tries,3 tries,4 tries,5 tries,6 tries)

2.2. Unsupervised Model Selection

We have applied different clustering algorithm such as DBSCAN, Kmeans, Kmeans++, biKmeans [3-7]. To obtain the most suitable model, we select the average silhouette cofficient and Calinkski-Harabasz values of different clustering algorithms in different number of clusters.

According to the Figure 2, the biKmeans model has the best performance in the evaluation indicators of silhouette cofficient and Calinkski-Harabasz.Therefore, we choose biKmeans model as our clustering model.

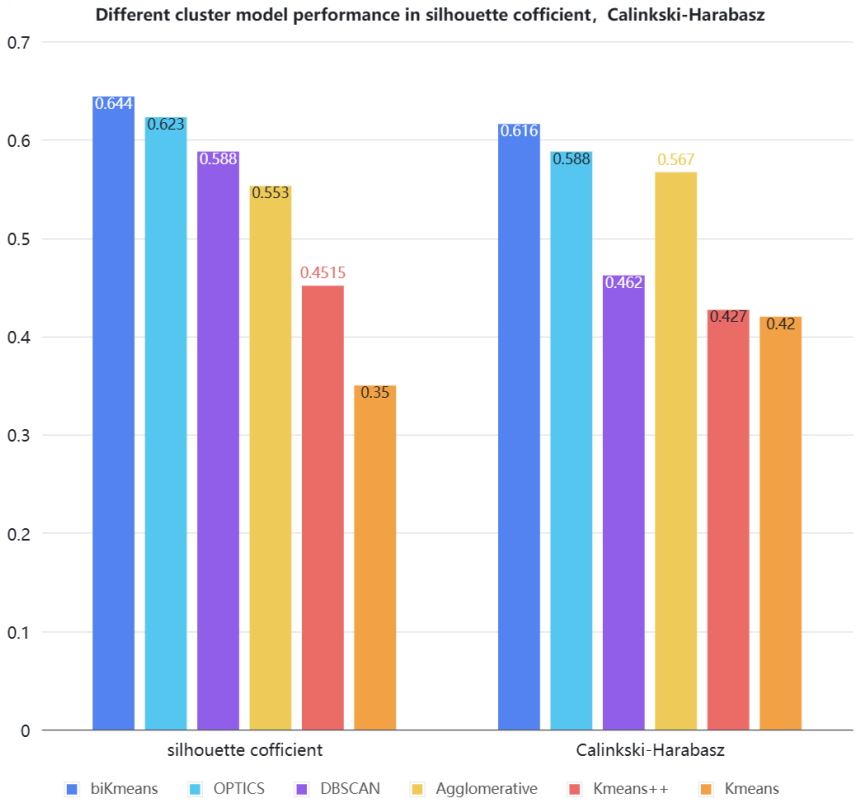

Figure 2. Different cluster model performance evaluation

2.3. Selection of the number of clusters

We used the elbow method to determine the optimal number of clusters for biKmeans. Figure 3 shows the SSE values of Kmeans, Kmeans++, and biKmeans for different numbers of clusters. The number of clusters that produces a turning point in the curve is chosen as the optimal value. In this case, the curve for biKmeans has a lower SSE value than Kmeans and Kmeans++, and the turning point occurs at four clusters, so we choose four as the optimal number of clusters.

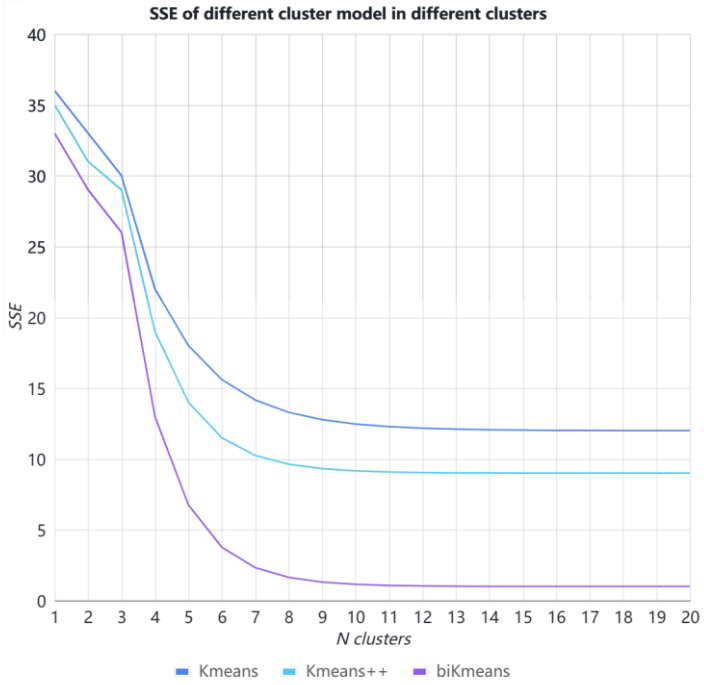

Figure 3. SSE of different cluster model in different clusters

2.4. Cluster Evaluations

Figure 4 shows the quantity distribution and pass rate of different difficulty categories. It can be found that there are four categories, C0 to C3, with the most number of C0 and the least number of C3, and the passing rate gradually decreases from C0 to C3, which indicates that the difficulty from C0 to C3 gradually increases.

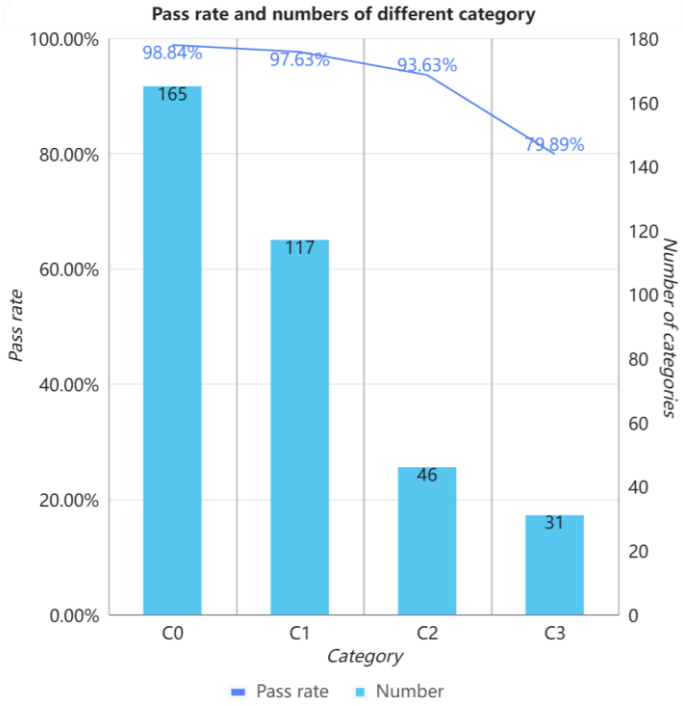

Figure 4. Pass rate and numbers of different category

2.5. Text vectorization

Text vectorization is crucial for successful data training, particularly when considering the complexity of individual words based on their letter composition and order. While traditional methods like TF-IDF only consider letter frequency, a more comprehensive approach is needed. Deep learning algorithms like BERT can overfit with limited data. As an alternative, we propose our self-created algorithm Letter2vec+Freq2vec, which leverages the word2vec neural network to model letter interactions and context [8]. This algorithm combines Letter2vec for letter vectorization and Freq2vec for letter frequency, resulting in a more holistic approach to text vectorization that accounts for the composition, order, and mutual influence of letters in words. Overall, Letter2vec+Freq2vec offers a valuable tool for enhancing model training.

Letter2vec draws on the idea of word2vec by building a neural network with hidden layers, taking the word as input, predicting the words around it according to the context, and updating the word vector according to the prediction result, so that similar words are closer in the vector space.

$$\sum_{c=1}^{C} \log P\left(w_{t-c} \mid w_t\right) = \sum_{c=1}^{C} \boldsymbol{v}_{\boldsymbol{w}_{t-c}} \boldsymbol{v}_{\boldsymbol{w}_t}^{\top} - \log\left(\sum_{i=1}^{V} \exp\left(\boldsymbol{v}_{\boldsymbol{i}} \boldsymbol{v}_{\boldsymbol{w}_t}^{\top}\right)\right) \tag{1}$$

Where w_t denotes the current letter, c denotes the size of the context window, w_{t-c} denotes the cth letter within the context window, v_{w_t} denotes the letter vector for the current letter, $v_{w_{t-c}}$ denotes the letter vector for the cth letter within the context window, and V denotes the size of the vocabulary.

Freq2vec takes the frequency of occurrence of different letters in different positions into account and converts 5 letters of word into the frequency of occurrence of 5 letters in corresponding positions.

Vectorization formula of Freq2vec model:

$$\boldsymbol{v}_w = \begin{bmatrix} f_{w1,1} & f_{w1,2} & f_{w1,3} & f_{w1,4} & f_{w1,5} \\ f_{w2,1} & f_{w2,2} & f_{w2,3} & f_{w2,4} & f_{w2,5} \\ f_{w3,1} & f_{w3,2} & f_{w3,3} & f_{w3,4} & f_{w3,5} \\ f_{w4,1} & f_{w4,2} & f_{w4,3} & f_{w4,4} & f_{w4,5} \\ f_{w5,1} & f_{w5,2} & f_{w5,3} & f_{w5,4} & f_{w5,5} \end{bmatrix} \quad (2)$$

Where $f_{w_{i,j}}$ denotes the frequency with which the ith letter of word w appears in the sth position in the word and v_w denotes the vector of words w.

2.6. Data Supplement

Figure 4 shows that C2 and C3 datasets have limited data. To avoid underfitting caused by dissimilar categories with inadequate data, we typically use up-sampling or down-sampling techniques to balance the data. Downsampling may cause significant information loss and is not recommended for small datasets like ours. Instead, we use the SMOTE method for up-sampling C2 and C3 datasets to increase negative samples.

$$X_{\text{new}} = X_{\text{old}} + rand(0,1) * (X_{\text{near}} - X_{\text{old}}) \quad (3)$$

X_{old} is the coordinate of the sample point corresponding to the category to be supplemented, and X_{near} is the coordinate of a sample point of another category around X_{old}

2.7. The Selection of Multiple Regression Model

Text vectorization is a complex, nonlinear process, making traditional linear models like LogisticRegression and LinearRegression unsuitable. Deep learning models like LSTM tend to overfit when data is insufficient. Instead, we use the Stacking approach9, integrating XGBoost [10], CatBoost [11], and LightBoost [12], to learn any form of function through sample data and handle nonlinear problems [9-12]. This approach effectively reduces errors and improves model robustness in small sample data sets, making it ideal for our problem.

Suppose we have M base models, and the ith magical pre-output is in the form of $\widehat{y_i}$, corresponding to a real output of y. We use N samples for training, then for the jth sample we compose the output of the base model into an M-dimensional vector, denoted $\widehat{y_j}$. Then we can use a meta model to map $\widehat{y_j}$ to the final output value $\widehat{y_j}^{\text{ensemble}}$. Specifically, a linear regression model can be used to obtain the following output equation:

$$\widehat{y_J}^{\text{ensemble}} = \sum_{i=1}^{M} w_i \widehat{y_{iJ}} \quad (4)$$

Where is the weight of the linear regression model, indicating the importance of the ith base model in the integration. We can determine the value of the weights by methods such as cross-validation.

For the three models in this question, XGBoost, LightGBM and CatBoost, their output values are all real numbers and can be used directly in Stacking integration. Therefore, we can obtain the following integration output formula.

$$\hat{y}_j^{\text{ensemble}} = w_1 \hat{y}_j^{XGBoost} + w_2 \hat{y}_j^{\text{LightGBM}} + w_3 \hat{y}_j^{\text{CatBoost}} \quad (5)$$

Where $\hat{y}_j^{\text{XGBoost}}$, $\hat{y}_j^{\text{LightGBM}}$ and $\hat{y}_j^{\text{CatBoost}}$ denote the pre-regular values of the XGBoost, LightGBM and CatBoost models for the jth sample, respectively, and w_1,w_2 and w_3 denote the weights of the three models in the integration, respectively, which can be determined by cross-validation methods.

3. Result and Analysis

3.1. Model Selection Results and Evaluation

Figure 5 and Figure 6 below shows F1 scores corresponding to different multiple regression algorithm models under different text vectorization algorithms before&after data upsampling processing. It is found that F1 performance scores are the highest which is output by the Stacking integrated XGBoost, CatBoost, LightBoost models using the Freq2vec+Letter2vec text vectorization algorithm, and achieve a very good effect in many models. Meanwhile, by contrast with figures 6 and 7, it can be found that after supplementation with SMOTE, the model training effect has been significantly improved, F1-score up to 0.8262, F1 scores improved by about 3% compared to before data supplement, get a quite good effect.

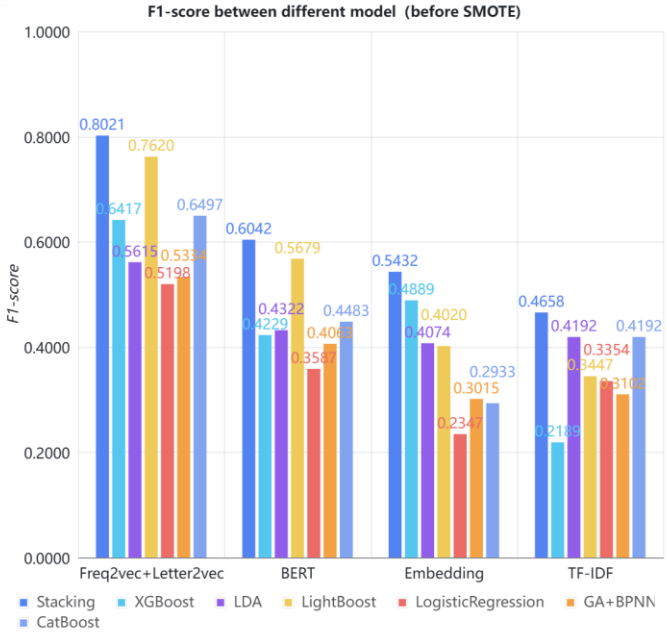

Figure 5. Model performance (before SMOTE)

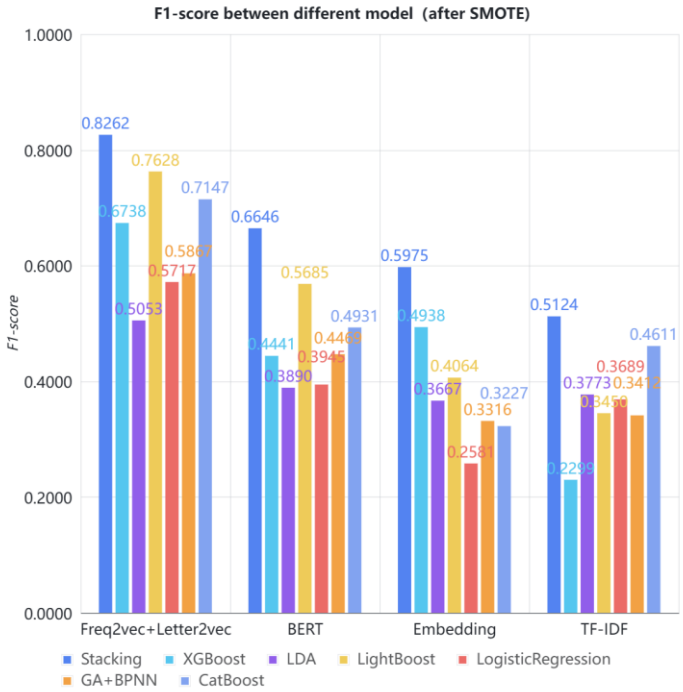

Figure 6. Model performance(after SMOTE)

3.2. Prediction Output-Take the word EERIE

Applying the preceding methods, the word 'EERIE' is first vectorized and standardized using the Letter2vec+Freq2vec algorithm. The standardized results are input into the Stacking model. The output EERIE is C1 and the difficulty is moderate.

4. Conclusion

The Letter2vec+Freq2vec algorithm is a reliable method for vectorizing text in natural language processing and information retrieval. This study explores unsupervised clustering and multiple regression techniques to evaluate word difficulty, using BiKmeans and the SMOTE method to improve training data. The Stacking model integrated with Letter2vec+Freq2vec effectively predicts word difficulty, exemplified by the categorization of "EERIE" as moderately difficult. Overall, this research advances text vectorization algorithms for predicting word difficulty.

Reference

[1] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. In Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 2227-2237).

[2] Baidu releases semantic understanding framework ERNIE 2.0 and pre-training model [J]. Smart City, 2019,5(14):26.

[3] LU Jianyun, SHAO Junming, ZHANG Wei. Parameter-free DBSCAN algorithm based on RAPIDS[J]. Data acquisition and processing,2023,38(02): 426-438.

[4] SHEN Qiuping, ZHANG Qinghua, GAO Man, DAI Yongyang. Three DBSCAN algorithms based on local radius[J/OL]. Computer Science:1-14[2023-04-30].

[5] ZHANG Ximei, XIE Bin, XU Tongtong, ZHANG Chunhao. Weighted Kmeans Clustering Intrusion Detection Algorithm Based on Inverse K-Nearest Neighbor and Density Peak Initialization[J]. Journal of Nanjing University of Science and Technology,2023,47(01): 56-65.

[6] CUI Peng, YANG Haifeng, CAI Jianghui, WANG Yupeng. Kmeans-DETR target detection method for multi-scale local clustering[J/OL]. Small microcomputer system:1-9[2023-04-30].

[7] WANG Yi, HUANG Ruiting, LU Jingyi. Application of Kmeans++ Model in TV Program Classification and Analysis[J]. Radio and Television Technology,2022,49(11): 11-16.

[8] XI Ningli, ZHU Lijia, WANG Lutong, CHEN Jun, WAN Xiaorong. An implementation method of Word vector model constructed by Word2vec[J]. Computer and Information Technology, 2023,31(01): 43-46.

[9] ZHANG Yan, BAO Shengli, WANG Xiaofei. Stacking algorithm based on dynamic clustering and its application[J]. Computer application,2022,42(S2):100-104.

[10] LUO Chunfang, ZHANG Guohua, LIU Dehua, ZHU Dinghuan. Research on XGBoost ensemble algorithm based on Kmeans clustering[J]. Computer age,2020(10): 12-14.

[11] WANG Li, WANG Tao, XIAO Wei, PAN Chao. Feature selection algorithm based on Catboost[J]. Journal of Changchun University of Technology,2021,42(01): 34-39.

[12] DAI Jinhui, ZHONG Xuan, *WANG Mengen. Used car valuation based on LightGBM and random forest algorithm[J]. Journal of Science and,2022,42(12):15-22.