Feedback Prize : English Language Learning
Easy Auto Grading

Somil Swapnil Chandra (22m0809)   Sandeep Kumar (22m0790)   Shashank Kate (22m0756)

November 26, 2022

1 Introduction

Writing is a fundamental skill required by any student. But as the population of students is growing, grading them is a time consuming task for Teachers. This task specifically focuses on the students who learns English as a second language. The main task of this project assign marks on different parameters, instead of providing single score for overall essay. The model assigns marks for syntax, cohesion, vocabulary, phraseology, grammar, convention for the given essay. As the grading essays is very tedious task for any instructor, essay auto graders are very helpful for them. But as they not provide any feedback to the student, they are simply black box for them. This problem tries to provide feedback on different parameters, so students can focus on specific parts, where there is a need of improvement. The objective of this project is to develop the understanding of different word embedding and trying to predict the different required parameters using different embedding techniques and machine learning models.

Statement of Contribution

All authors contributed equally to all parts of the solution and have trained the models on different set of embeddings. Somil Swapnil Chandra trained the models on Glove word embedding. Shashank Kate conceived the idea of using tf-idf and BERT for training the language models, Sandeep Kumar used the extracted features to train the model.

2 Feedback Prize - English Language Learning

Our project is in association with the "Feedback Prize - English Language Learning" competition on Kaggle and seeks to produce output with maximum efficiency.This competition's objective is to evaluate the language skills of English language learners in grades 8 through 12. (ELLs). It will be possible to create proficiency models that better support all students by using a dataset of essays authored by ELLs.

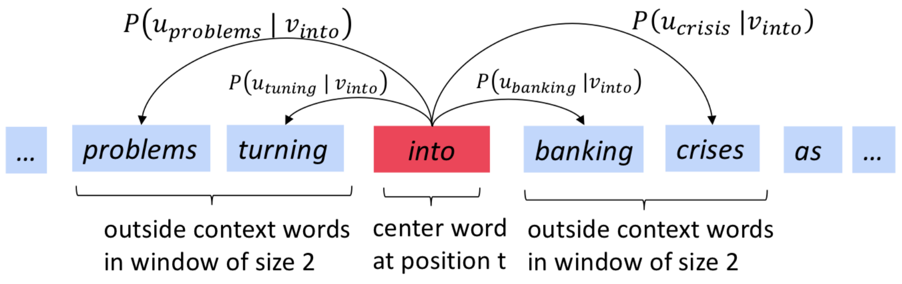
The goal of this project is to create a machine learning model that can accept an essay and output the essay's performance on different features automatically.

3 Problem Statement

With the goal of reaching low loss values especially MCRMSE (Mean Column RMSE), we will extract various essay elements and evaluate these essays using ML models such as feed forward neural network, linear regression models, SVR(Support Vector Regressor), Random Forest and LSTM.

4 Word Embedding

A vector is used to represent each essay. The first step in processing the data is to eliminate all punctuation and stop words. Each essay is represented as a vector by using learning algorithms such as GloVe, TF-IDF (Term Frequency Inverse Document Frequency) and BERT.


$P(u_{problems} \mid v_{into})$
$P(u_{tuning} \mid v_{into})$
$P(u_{banking} \mid v_{into})$
$P(u_{crisis} \mid v_{into})$
...
problems
turning
into
banking
crises
as
...
outside context words in window of size 2
center word at position t
outside context words in window of size 2

I `GloVe`: A weighted least-squares objective log-bilinear model is essentially what GloVe is. The basic insight behind GloVe is the possibility of storing meaning through the

ratios of word-to-word co-occurrence probabilities. GloVe's training goal is to acquire word vectors whose dot product equals the logarithm of the probability of the words occurring together.
To obtain the essay vector, we average the GLoVe word vectors of every word in the essay. A 300-dimensional GLoVe word vector was used for our experiments.

II `TF-IDF`:Term Frequency is referred to as an IDF. records with Inverse Document Frequency. It can be summed up as determining how pertinent a word is to a corpus or series of words in a text. The frequency of a term in the corpus offsets the meaning increase that occurs when a word appears more frequently in the text (data-set).

III `BERT`: Using a sizable corpus of words, BERT (Bidirectional Encoder Representations from Transformers) models were pre-trained. Briefly stated, the model is trained by masking a small number of words—roughly 15% of the words in a sentence and asking it to predict the remaining words. Additionally, as the model becomes more adept at making predictions, it develops a potent internal representation of words known as word embedding.

5 Evaluation

Performance Evaluated using MCRMSE.RMSE is frequently used and is a great all-purpose error metric for numerical forecasts. RMSE amplifies and harshly penalises big errors in comparison to the analogous Mean Absolute Error. The RMSE calculation formula is as follows:

$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2}$

$MCRMSE = \frac{1}{m}\sum_{j=1}^{m}\sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(\frac{d_i - f_i}{\sigma_i}\right)^2}$ where:

m - number of predicted variables,

n - number of test samples,

yij - i-th actual value of j-th variable,

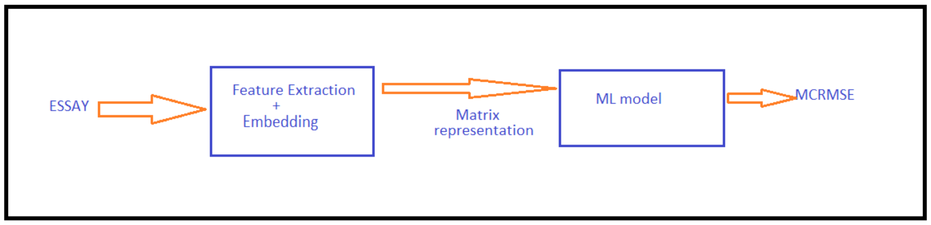
yij - i-th predicted value of j-th variable

We can still use a single-number evaluation metric even in the case of multiple outputs because the MCRMSE is simply an average across all RMSE values for each of our columns.

6 Methodology

We firstly extracted some features from the essay such as average word length, nouns used in the essay,adjective count of the essay, verb, adverb, conjunction count etc. which can indicate the strength of the essay on certain parameters.

Figure 1: Implementation Details, generalized for all models



The above diagram shows the method used to evaluate essays, we first get the matrix representation of the essay, using one of the embedding methods and extracted features. Then this matrix is passed to an ML model (Linear regression, SVR, NN, LSTM, Random Forest), which is then evaluated against the test data for error value.

Implementation

- **Linear Regression:** It is the simplest regeneration model, which tries to fit a hyperplane for the dataset to minimize the error. For our model we used linear regression without any regularization and with L2 regularization with lambda= 0.2 and 1.
- **SVR:** The SVR model is the regression counterpart of SVM. In SVR we fit a hyperplane and points which are closest to this hyperplane on either side, are called support vectors. For our model we set regularization parameter = 0-10 and epsilon = 0.01 (the range for which a prediction does not get any error)
- **Feed Forward Neural Network:**
  - For this project we used a 3 layer neural network, using tanh activation for the first two and relu for the output layer.
  - We ran NN for all 3 embeddings and also for extracted features.
  - Number of epochs 10-300
  - Dropout 0-0.4
  - Validation and testing percentage = 20
  - Loss function = mean_squared_error

- **LSTM:** LSTM model is used to remember the information form the past using its gated mechanism (input gate, output gate and forget gate) and cell state For our model parameters for LSTM network are :
  - 3 layers (1 LSTM, 2 Dense)
  - Activation function= tanh,relu
  - Number of epochs =10-100
  - Dropout =0.4
  - Loss function = mean_squared_error
  - optimizer= rmsprop.
- **Random Forest:** Random forest is an ensemble model, in which base learners are Decision trees. Where each decision tree is trained on a subset of the dataset. And the final result is given by averaging the output of all decision trees. As all decision trees are created on different data, the random forest reduces the chance of overfitting and also it is capable of handling large data with high dimension.

  For our model we used 100 learners(n_estimators) and random state= 30-50.

7 Result

After running all the models over different embeddings, we got our best results on the feed forward neural network, using TF-IDF embedding. However most of the models had nearly the same MCRMSE value and all methods were close to 0.56.

As TF-IDF uses word embedding, created using data from a given dataset, it seems to give high weightage to certain words for certain prediction variables. On the basis of our understanding of different models and embedding techniques, we can say that our model should perform better provided more training examples.

Also due to restriction of time, we were not able to perform different experimentation on models, mainly tuning the model, and also feature extraction could have been improved.

The best obtained result on our training are as follows:

Embedding/Extracted Feature	**MCRMSE**
Glove	0.29
TF-IDF	0.37
Extracted Features	0.35

Table 1: Linear Regression.

Embedding/Extracted Feature	**MCRMSE**
Glove	0.30
TF-IDF	0.29
Extracted Features	0.41

Table 2: SVR.

Embedding/Extracted Feature	**MCRMSE**
Glove	0.32
TF-IDF	.047
Extracted Features	1.48

Table 3: Neural Network.

Embedding/Extracted Feature	**MCRMSE**
Glove	0.41
TF-IDF	0.39
Extracted Features	0.38

Table 4: LSTM

Embedding/Extracted Feature	**MCRMSE**
Glove	0.30
TF-IDF	0.29
Extracted Features	0.36

Table 5: Random Forest.

8 Conclusion

From our experiments over different model the predictions are closer to actual values. The neural network and LSTM model perform good for data set, but we need to fine tune the models for more accurate predictions. Also to improve performance on content specific and richer essays we can use advanced NLP libraries and some more complex language modelling techniques.

9 GitHub repository

`https://github.com/Alankriss07/Essay-Auto-Grading`

10 References

- https://nlp.stanford.edu/projects/glove/
- https://github.com/Turanga1/Automated-Essay-Scoring
- https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html
- https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM
- https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
- https://cs224d.stanford.edu/reports/huyenn.pdf