

Automated Essay-grading System

Project Overview

Our AES application provides a platform for augmented grading of essays using deep learning and machine learning algorithms that mainly revolve around the concepts of NLP and RNN. The application automates the process of grading the essays submitted through our GUI evincing a minimalistic design, prompting users with a diverse list of creative writing prompts, cutting down on the time, cost and effort while boosting accuracy and efficiency.

This system promises impressive yields in competitive exams like GRE, GMAT, TOEFL, IELTS alike. A calculated score is presented to the writer on submission, determined by the learning curve achieved through data processing and imposing machine learning models extended by the NLTK libraries and LSTM models.

Dataset

Our model is trained using the 'The Hewlett Foundation: Automated Essay Scoring Dataset' sourced by ASAP. The dataset includes the Essay_Set_Descriptions, encompassing the 8 prompts description and scoring criteria for each besides the training_set_rel3.tsv containing the essays for training the model.

Data Preprocessing

The datasets are processed for training and testing following some standard preprocessing steps starting out from handling the null values, resulting in the omission of 22 columns. Secondly, we filter out the valid features from the dataset and proceed with further processing steps like the exclusion of special symbols, stop words, punctuation to enhance our accuracy.

Splitting the Data

The data is split as x , input, consisting of the submitted essay, and y , output, which consists of the essay score. This is further processed before training and testing the model.

Model Training and testing

Our current dataset is comprised of fields containing textual inputs which are to be converted to numeric counterparts using CountVectorizer, which tokenizes the textual content and returns an encoded vector. Advancing to the next step, we split the data with a test size of 0.2 resulting in a train set of 1380 rows and a test set of 2596 rows.

Progressing to our model building stage, we establish an RNN with a Word2Vec Embedding layer to be fed into our LSTM layers. The Word2Vec layer passes all features to our first LSTM layer which then propagates 300 features into the second layer. This layer goes on to contribute 64 features as output transmitted to our Dropout layer valued 0.5.

The essay score is presented as an output of the Dense layer. The model is compiled with a loss function of Mean Square Error and optimized with 'rmsprop'. The model is trained for 100 epochs with a batch size of 64.

The model is tested with the test sets emerging with a 94% accuracy.

Flask App Integration

We deploy our model using the Flask framework, a prominent python API used for developing web applications. Our application elicits an essay submission from the writer, after choosing from the enlisted prompts. The essay is communicated to the app via a POST request. The application creates a navigation flow through the home page, submissions page, leading to the grade page where the writer receives the score graded by the trained model.

Conclusion

In this project, we developed an AES application that implements a deep neural network capable of assessing essays and imparting a calculated score based on the processing and the training instilled. This recurrent neural network model is interpretable and can be transformed to assist the writer with impactful feedback.