Project Report
CS: 421
Natural Language Processing

Automated Essay Grading with Neural Networks

Karan Raghani

Utsav Shah

December 2, 2019

1. Introduction

Essays are crucial testing tools for assessing academic achievement, integration of ideas and ability to recall, but are expensive and time consuming to grade manually. The purpose of this project is to implement and train neural network to automatically assess and grade essays. The grades from the automatic grading system should match the human grades consistently.

Implementing automated essay scoring (AES) helps reduce manual workload and speed up learning feedback. Recently, neural network models have been applied to the task of AES and demonstrates tremendous potential. [1]

AES Challenge: 'Automated Student Assessment Prize'

In 2012, the Hewlett Foundation sponsored a competition on Kaggle called the Automated Student Assessment Prize (ASAP). The competition used quadratic weighted kappa to measure the similarity between the human scores and the automated scores. The winning team got the kappa score of 0.81407. We are implementing a model that is built using predefined features using neural network. [2]

This project aims to build a neural network based model that can take in an essay and automatically output the grade of that essay. Using a feed forward and Long Short Term Memory neural network.

Problem Statement

Our aim is to build an implementation that can take in an essay and grade it automatically. We shall extract different features of the essay and use neural networks such as feed forward,

recurrent neural network to evaluate these essays with aim of achieving low loss values and higher kappa values.

2. Related Work

Research on AES began decades ago. In the field of application, the first AES system named Project Essay Grade (PEG) [3] for automating the educational assessment was seen in 1967. Intelligent Essay Assessor (IEA) [4] adopts a Latent Semantic Analysis (LSA) algorithm to produce semantic vectors for essays and computes the semantic similarity between the vectors. The E-rater system [5], which can extract various grammatical structure features of the essay, now plays a facilitating role in the Graduate Record Examination and Test of English as a Foreign Language.

In the above works features such as words, Part-of-Speech (POS) tagging, n-grams features, complex grammatical features, are extracted. This is assumed to be close to how the human grader grades the essays except the contextual references. Shristi Drolia et al. [6] proposed a regression-based approach for automatically scoring essays that are written in English; they use standard Natural Language Processing (NLP) techniques for extracting the features from the essays.

Since deep learning was introduced into natural language processing, more and more researchers have carried out related research. Cicero Nogueira dos Santos [7] proposed a deep convolutional neural network which focuses on different levels of analysis that from character-level to sentence-level information to perform sentiment analysis of short essays. Taghipour et al. [8] developed an approach based on recurrent neural networks to learn the relation between an essay and its assigned score, without any feature engineering. They combined convolutional neural networks and recurrent neural networks for AES and demonstrated that LSTM and CNN are capable of outperforming systems that extensively require handcrafted features. [9]

As reported by Huygen and Lucio in their paper [2], the Deep Learning techniques achieved an average Kappa score of 0.9447 using 2-layer neural network that trains word vectors together. Inspired by this we put forward this implementation that had derived its methodology from the aforementioned work.

3. Analysis

Data Set

We use the dataset provided for Hewlett Foundation's Automated Student Assessment Prize com-petition on Kaggle. Some characteristics [10] of the dataset:

1. There are 8 different sets of essays, each generated from a single prompt. To fasten computation.

2. Selected essays range from an average length of 150 to 550 words per response.
3. Each essay was hand-graded by two or three instructors.
4. There are around 13000 samples in total. For this implementation due to lack of processing power we used essays in SET1 to compute accuracy of the system. We used 10% of the essays for testing, 80% was used for training and 10% was used for validation.
5. Each set has a different grading scale.

Statistics of ASAP dataset.

Prompt	Number of Essays	Average Length	Scores
1	1788	350	2–12
2	1800	350	1–6
3	1726	150	0–3
4	1772	150	0–3
5	1805	150	0–4
6	1800	150	0–4
7	1569	250	0–30
8	723	650	0–60

Inputs

The data set contains following parameters:

- essay id: A unique identifier for each individual student essay
- essay set: The set of a given essay.
- essay: The ascii text of a student's response
- rater1 domain1: Rater 1's domain 1 score
- rater2 domain1: Rater 2's domain 1 score
- domain1 score: Resolved score between the raters.

4. Features

Total 8 features were generated using Natural Language Toolkit (NLTK). This mainly involved removing placeholders for proper nouns, preparing the text, tokenizing and stripping essays of punctuation. Intrinsic variables in an essay like the style and fluency cannot be directly measured they have to be approximated with measurable quantities like the sentence and word length, length of essay etc. The features extracted can be divided into the following broad categories:

1. **Bag of Words features:** Several heuristic features that are likely to contribute to a good essay were generated. Some of the heuristic features are: word count, average word length per essay, number of characters, sentence count and average length of sentences in terms of words.

2. **Spelling and Grammatical features:** It is likely for a student to make grammatical and spelling errors, number of spelling mistakes were generated using PySpellCheck.

3. **Part of Speech (POS) tags:** A count for most regular POS tags that helps to identify a good sentence structure were used, for example, count of Nouns, Verb, Adjective and Adverb for a given essay using NLTK.

5. Word Embedding

Each essay is represented as vector. The data is first processed, and all the punctuation marks are removed from it. We use GloVe to represent each essay as a vector. GloVe is essentially a log-bilinear model with a weighted least-squares objective. [11] The main intuition of GloVe is the simple observation that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning. The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence. Owing to the fact that the logarithm of a ratio equals the difference of logarithms, this objective associate (the logarithm of) ratios of co-occurrence probabilities with vector differences in the word vector space. Because these ratios can encode some form of meaning, this information gets encoded as vector differences as well.

We take the GLoVe word vectors of all the word in an essay, average them to get the essay vector. We experimented with GLoVe word vectors with a dimension of 300.

6. Evaluation

Quadratic Weighted Kappa will be used as Evaluation Metric for this project. This is a robust metric as compared to the simple percent agreement calculation since it takes into account the possibility of agreement occurring by chance. The value of the metric ranges from 0 (only random agreement between the raters) to 1 (complete agreement). [10] In the event that there is less agreement between the raters the value may go below 0. The quadratic kappa is calculated between the automated generated scores for the essays against the gold standards (resolved scores for human raters) for each sets of the essays. The mean is taken after applying the Fisher Transformation to the kappa value. This mean is taken across all sets of essays.

7. Methodology

We extracted a set of features from each essay. We chose features that may serve as proxies for what a human grader might look for while grading the essay. We implemented two models for this problem. In the first approach we used a simple Feed Forward network. In the second

approach we used recurrent neural network. It was then used as our learning model to learn weights based on the features. Scores were predicted for a distinct set of test essays. These scores were compared against human graded scores to arrive at an error metric.

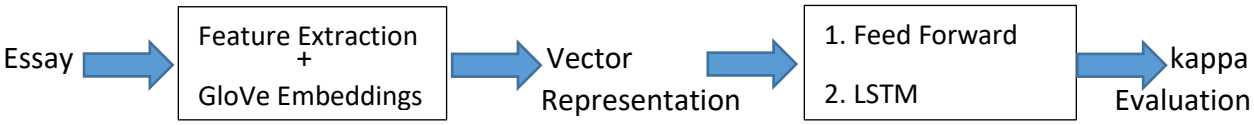
Implementations:

Following diagram shows the methodology used for the implementation of AES. The Different implementations we did for the project are:

- Feed Forward neural network on just the features extracted from the essay sets. An eight-dimensional vector is created in this case. This is a 3 layered neural network. As you will see in the result section, we got lower values of quadratic weighted kappa in this case.
- Feed Forward neural network on the GloVe vector representation of the essays. In this case the vector dimension was 300. This is a 4 layered neural network implementation
- Feed Forward neural network on a vector obtained by concatenating the GloVe representation of the essays with its feature vector. This is a 4 layered neural network implementation.

Similar approaches were implemented on the LSTM model:

- LSTM neural model on the GloVe vector representation of the essays. In this case the vector dimension was 300.



Feed Forward Neural

For our Neural Network we experimented with different forms of input. We constructed an Essay vector for each essay which was obtained by averaging all the word vectors of the words in the essay. This was done by using GloVe vectors for word representation. Using this approach of creating Essay vectors constituted our bag of words model.

The first layer is the input layer that ingests the averaged GloVe vector representation of the essays. Then there are multiple hidden Dense layers with sigmoid/tanh activation function. The number of neurons of each layer differ but constantly decrease. The final output layer has one neuron and it uses the Rectified Linear Unit activation function to produce final outcome. The hyper-parameters used were:

- The number of epoch: 300 - 800

- Validation and testing percentage: 0.2
- Function used for avoiding over-fitting: dropout
- Activation function: sigmoid function / tanh
- Loss function: mean squared error
- Optimize function: rmsprop
- Result measurement: quadratic weighted kappa

LSTM

Recurrent neural networks are one of the most successful machine learning models and have attracted the attention of researchers from various fields. Compared to feed-forward neural networks, recurrent neural networks are theoretically more powerful and are capable of learning more complex patterns from data. [12] Therefore, we have implemented recurrent networks in addition to feed forward neural network in this project. We implemented a simple LSTM with a scoring layer at the end. The LSTM receives a sequence of word vectors corresponding to the words of the essay and outputs a vector that encapsulate in the information contained in the essay. The scoring layer at the end converts this vector into a score in the required range. The bag of words model used in the feed forward networks described above has limited power. The decision to implement the LSTM based architecture therefore, was to capture sequence information in the dataset.

8. Results

After analyzing the result of our implementation, we observe that the simple feed-forward neural network provides the best kappa score. However, both of our models were able to identify score ranges for the essay set and score them accordingly. We were able to achieve a Quadratic Weighted Kappa score of ~ 0.79.

As expected, the neural network that trains on word vectors formed from concatenating the pre-trained GloVe embeddings and feature matrix performed the best. Since essays were answers to specific prompts, features of essays do extract some of the properties that are relevant to the prompts.

We are also certain based on our understandings and research study on different methodologies that the implementation of the system over more sets of the essays in the dataset are bound to increase the accuracy of the system. However, due to limited availability of processing power, we were not able to implement all the features extraction over the entire set.

The overall findings of our project is presented in the evaluation matrix below.

Evaluation Matrix

Model	Word Vector	Kappa
3 – Layer NN	Features only	**~0.7812**
4 – Layer NN	GloVe	**~0.7173**
4 – Layer NN	GloVe+ Features	**~0.7886**
LTSM	GloVe	**~0.701**

9. Conclusion

As we hypothesized, features from each category contribute towards a good prediction. Our final model contains features across all the categories we tried to test for. Our model works relatively better on non-context specific essays. Performance on content specific and richer essays can be improved by incorporating content and advanced NLP features.

10. Improvements

The goal of the project was to implement an automated approach for grading of essays. Due to lack of practical hands-on experience with Deep Learning, a relatively simple set of models of neural networks were chosen for this project. The models use simple features that concentrate on the sentence structures of the essay and the similarities of the words used in the essay. The end results show us that this approach of involving Deep Learning, a model with a high accuracy can be generated for this project. Analyzing the sentence structure that serves as a benchmark for a good essay, exploring the latest researches in the NLP domain, the immense usefulness of the open source NLP libraries proved to be a great learning curve for us in this project.

Due to a confined data-set, essays similar to the training essays can be evaluated using this model directly with appreciable results. However, for essays in different domain, the model can serve as a benchmark for future tasks in the field of Automated Essay Grading.

Performance on content specific and richer essays can be improved by incorporating content and advanced NLP features. Also use of complex recurrent neural networks with added contextual features can improve the systems accuracy and provide more accurate results for essays belonging to different domains.

References

[1] M. J. A. A. Manvi Mahana, "Automated Essay Grading Using Machine Learning".

[2] L. D. Huyen Nguyen, "Neural Networks for Automated Essay Grading".

[3] B. Ellis, "Grading essays by computer: Progress report.".

[4] P. Foltz, D. Laham and T. Landauer, "Automated essay scoring: Applications to educational technology.".

[5] Y. Attali and J. Burstein, "Automated essay scoring with e-raterR v.2.0. ETS Res. Rep. Ser," 2004.

[6] S. Drolia, S. Rupani, P. Agarwal and A. Singh, "Automated Essay Rater using Natural Language Processing.".

[7] C. Santos and M. Gatti, "Deep Convolutional Neural Networks for Sentiment Analysis of Short Text".

[8] K. Taghipour and H. Ng, "A Neural Approach to Automated Essay Scoring.".

[9] B.-W. O. D. J. a. H.-C. K. Guoxi Liang, "Automated Essay Scoring: A Siamese Bidirectional".

[10] T. K. :. T. H. F. A. Essay, "https://www.kaggle.com/c/asap-aes/overview/evaluation," [Online].

[11] J. R. S. a. C. D. M. Pennington, "Glove: Global Vectors forWord Representation," 2014.

[12] H. T. N. Kaveh Taghipour, "A Neural Approach to Automated Essay Scoring".

[13] M. R. Md. Haider Ali Annajiat Alim Rasel Arshad Arafat, "Automated Essay Grading with Recommendat," 2016.