Automated Essay Grading CISC 351 Report

Umur Gokalp - 20002381

Abstract—Automated essay grading is underrepresented in the realm of natural language understanding. While the automation of grading multiple choice prompts are well explored, research on essay grading is lacking. The advances in NLP urge an investigation of the uses of deep learning methods. This report explores the use of deep learning methods such as network embedding and the RoBERTA model. Additionally, this report uses the mentioned deep learning methods to compare to older representations of text, namely the bag of words representation.

Index Terms—group 13, NLP, Essay grading,survey of methods,deep nlp

1 Introduction

Automated grading tools for multiple-choice questions have existed for over 40 years. This type of automation was intended to alleviate manual labour borne to the teacher. The motivation for research contained in this report comes from the desire to reallocate teachers' time and effort to what truly matters—teaching.

Automated essay grading is much more complex than grading multiple choice questions, which requires only Optical Mark Recognition (OMR). For automated essay grading, machines must be taught to understand English. A couple of years ago, models beat the human benchmark in the General Language Understanding Evaluation (GLUE) using transformer architecture. The increase in the capabilities of Natural Language Processing (NLP) models have encouraged an investigation into their uses in automated essay grading. Automated essay grading can curb the manual labour of teachers reading through numerous, long essays. The automation of grading may also help students be evaluated in real-time, removing the disconnect between when the essay is written, marked, and available to the student.

2 Related Work

Unlike the other topics in NLP, like those listed in the GLUE benchmark, automated essay grading is not a popular topic. This may be due to the marketing of automated essay grading as a commercial product, and thus arousing little enthusiasm and interest in academia.

2.1 Classical Methods in Essay Grading

In "Identifying Important Factors in Essay Grading Using Machine Learning" [11], an approach of exploring the factors that correlate with essay grades is taken. Given a dataset of scores of high school students from the Netherlands ranging from zero to five, the basic algorithms such as classification and regression trees (CART) and Naive Bayes were used. The preprocessing steps used hand-coded features from applied linguists to assess the complexity of the observed corpus. The mentioned distinctive approach in the mentioned paper was to use adjacent agreement. Adjacent agreement is a measurement of when the predicted score is one step to the right or the left of the true score. This is particularly useful because manual instructor grading can be subject to change by a noticeable error margin. While the models in this paper did not reach an accuracy higher than 62%, the adjacent agreement of their best model reached 98% with all scores combined.

- *Umur Gokalp is with School of Computing at Queen's University E-mail: umur.gokalp@queensu.ca*

2.2 The Neural Network Approach

The difficulty of hand generating features for machine learning models is well known. The authors of the paper "Automatic Text Scoring Using Neural Networks" [1] designed a new neural network architecture specifically for essay scoring, using the same dataset as this paper. They also used a more rigid implementation of the competition-specific metrics and transformation, keeping in mind to capture score-specific word embeddings. The score-specific word embeddings are used such that synonymous words are clustered together with the same synonymous misspelled words. This is best illustrated with the example "computer" and"laptop" and their spelling mistakes "copmuter" and "labtop". Both pairs "computer" and "laptop" and "copmuter" and "labtop" are embedded or "kept in" the same vector space as they provide the same information. The final model is a score-specific word embedding combined with two-layer bidirectional LSTM. This architecture dramatically outperformed the next best available model in the Kaggle competition.

3 Methodology

This report will be surveying some of the methodologies in Natural Language Processing, beginning with the earlier language models and ending with the state of the art model available. The models and architectures that will be exploring are:

- Random Forest with Bag of Words Representation
- Random Forest with Tf idf representation

- Neural Network with Bag of Words representation
- Neural Network with Tf idf representation
- Random Forest with Generated Features
- Fully connected network with Generated Features
- Embedding network with Generated Features
- Average Stochastic Gradient Decent Weight-Dropped Long Short Term Memory model
- A robustly optimized BERT pretraining approach (RoBERTa) on just the essay
- RoBERTA with essay text and the prompt id

3.1 Pre-processing and Evaluation

When generating scores in the interval of zero to five, the scores outside the range were discretized.

The data was pre-processed differently for each task, and 5-fold cross-validation was applied for each model described below. A holdout set of size 2596 was used to produce the confusion matrices.

For all of the models three metrics were considered; negative logarithmic loss, accuracy and adjacent agreement. Negative logarithmic loss is especially useful because the confidence of the model is more important than predicting the true score. In parallel, the adjacent agreement is a worthwhile metric as it better mimics the real world application.

To evaluate the strength of the predictors, permutation ranking was used.The simple idea behind permutation ranking is, measuring the change in the evaluation metric when a specific feature is not available versus when it was available. A more concise description of the algorithm is available on Permutation importance: a corrected feature importance measure. [2]

3.2 Sparse Matrix Representation of Text

For the baselines, a bag of words representation (BoW) was chosen. The implementation of the representation is as follows: *Count the usage of every word for every essay. Words become features and counts become its values.*

As for the pre-processing steps, each word in each essay was lemmatized, stemmed, and stop words were removed. This was done to reduce the words into their purest form such that "zeppelin" and "zeppelins" would not be considered two different features. The explicit assumption here is that using unique words should be enough, since the words themselves do not convey any information other than being available in a one-hot encoded matrix. This approach excludes the transformations applied by the context to the part of speech. Finally, the frequency of each word, used at least 10 times, was used as a feature.

A random forest model was trained on the BoW representation of the essays.

As a second representation, Term Document Inverse Document Frequency (TFIDF) representation of the essay was used. The reason being, TFIDF is able to generate scores based on the relative usage of words, which in some cases may be more beneficial to the model.The TFIDF score of a word is calculated as follows:

$$tf(w) * idf(w) \tag{1}$$

where $tf(w)$ is the number of times the word appears in the document / the total number of words in the document And

	mistake	maria	canada	medic	common
1175	0	0	0	0	0
9551	0	0	0	0	0
10183	0	0	0	0	0
6841	0	0	0	0	0
3209	0	0	0	0	1

Fig. 1. Sparse matrix representation of Essays, 5 columns selected on random.

$idf(w)$ is the natural logarithm of number of documents / the number of documents that contain the word w

The same preprocessing and modeling steps were used for TFIDF. Just like BoW representation, a random forest model was trained. Permutation importance of the features of the BoW model can be found below.

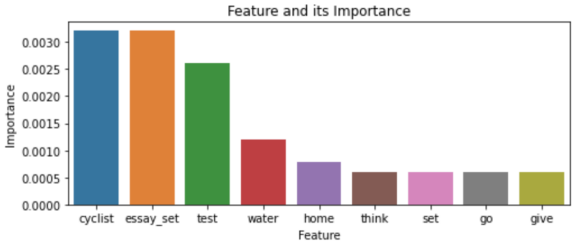

Fig. 2. Feature importance of Document term matrix modeled by Random Forest

3.3 Deep Neural Network

A Tabular Learner from the Fastai library [5] was chosen. The general structure of the model is as follows: Linear layers with batch normal and Rectified Linear Unit (ReLU) activation. The model was tested on BoW and TFIDF.

3.4 Generating Features

Noticing the limitations of the previous representations of the underlying text, a personal decision to generate new features on observed importance in this report was made.

These new features are:

- Counts of parts of speech such as count of nouns: Using a readily available part of speech tagger (POS) from the NLTK library [7] these features were generated. Author thought these features would be able to assess observations in the edge cases.
- The grade level of the essay such as Grade 10: To relax the grading.
- The prompt id such as prompt 1: Useful for knowing which essay prompt the essay is is with regards to.
- The type of essay such persuasive or not: To assess what type of essay is expected.
- Average length of words: To assess the complexity of words used.

- Average length of sentences: To assess the observations in the edge cases.
- Number of sentences: A proxy to asses the complexity of the text.
- Count of commas: To assess the structure of the sentences
- Count of quotations: Some essays require quoting from other pieces of text.
- Count of spelling mistakes: Spelling is an important piece of grading.
- The sentiment of the essay (Generated by BERT [12]): Assesses the sentiment of opinionated essays.
- Count of words: To see if the essay is at correct length.
- Count of SAT words: To test the level of vocabulary used.

3.5 Neural Network with Embeddings

A Tabular Learner from the Fastai library [5] was chosen. This particular model utilizes embeddings for categorical variables. The benefits of using such a layer is explained in Entity Embeddings of Categorical Variables [4]. The general structure of a Tabular Learner is as follows: Embedding layers for each categorical variable,a dropout layer for the embedding layer, followed by Linear layers with batch normal and Rectified Linear Unit (ReLU) activation between. An illustration of the network and the permutation importance of features can be found below.

```
TabularModel(
  (embeds): ModuleList(
    (0): Embedding(9, 5)
    (1): Embedding(4, 3)
    (2): Embedding(3, 3)
  )
  (emb_drop): Dropout(p=0.04, inplace=False)
  (bn_cont): BatchNorm1d(16, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
  (layers): Sequential(
    (0): Linear(in_features=27, out_features=1000, bias=True)
    (1): ReLU(inplace=True)
    (2): BatchNorm1d(1000, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (3): Linear(in_features=1000, out_features=500, bias=True)
    (4): ReLU(inplace=True)
    (5): BatchNorm1d(500, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (6): Linear(in_features=500, out_features=32, bias=True)
    (7): ReLU(inplace=True)
    (8): BatchNorm1d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (9): Linear(in_features=32, out_features=6, bias=True)
  )
)
```

Fig. 3. Layers of the FastAI Tabular Network.

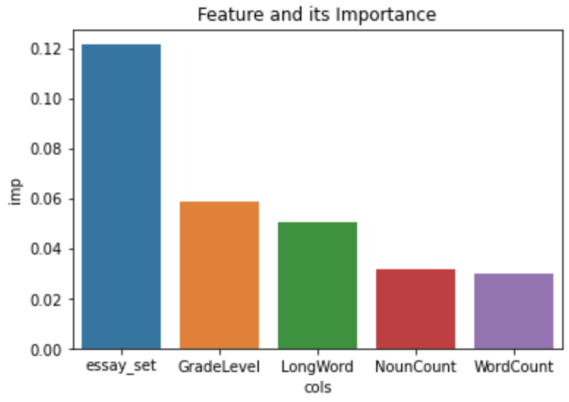

Fig. 4. Permutation feature importance of top 5 features.

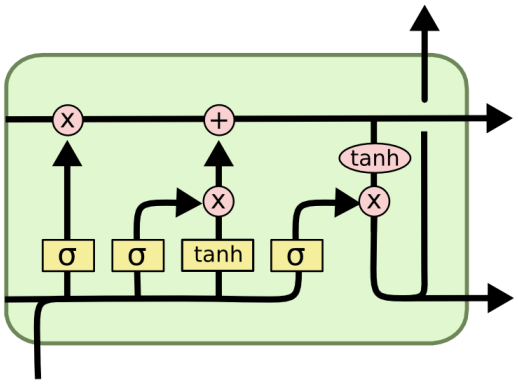

Fig. 5. A visual representation of the LSTM model.

3.6 Average Stochastic Gradient Decent Weight-Dropped Long Short-Term Memory model

This model has been first introduced by Stephen Merity et al. in 2017 and was considered a state of the art language model(AWD-SGD) for word-level models. [8] This language model is known for its special optimizer and regularization techniques, though it is otherwise much more complex. To get an idea of the architecture, the readers are encouraged to read the paper.

As a part of Fastai text library, The AWD-SGD model was used to first tokenize the inputs and then was used as a classifier. No modifications were added to the model, it was used in its vanilla form.

3.7 A robustly optimized BERT pretrained approach (RoBERTa)

This model can be thought as a modification of pre-training of Deep Bidirectional Transformers for Language Understanding (BERT) [3]. RoBERTa is trained longer with a more "optimal" learning rate and with dynamic masking technique [6]. RoBERTa is considered to be state-the-art in language modeling due to its performance on GLUE.

In order to use RoBERTa, the package simpletransformers [10] on PyPI was used. The model was tested with two different inputs.

The first one included, using the text as an input, transforming it into sequences of integers with special tokens native to RoBERTa.

The second use of RoBERTa included a modification to the text, namely appending the text "This essay belongs to set x." with the x being the index of the associated essay prompt to every essay. This attempt provided a small decrease in negative log loss and a slight increase in accuracy though it may not be very significant.

4 Dataset

The dataset is from Kaggle, made available by The Hewlett Foundation. There are a total of 12978 observations in this dataset. The original dataset contains the different sets of essay prompts, the essays, two scores from independent

graders and domain-specific scores. There are a total of 8 essay sets, each with different prompts, essay lengths and grading schemes. The essays were written by students in grades 7 to 10.

The length of essays ranges from 150 to 550 words per response. The data has gone through as a pre-processing step of anonymization before making it available. The process includes tagging named entities using Named Entity Recognizer from the Stanford NLP group. Some of the named entities are "person" and "organization." These appear in the dataset as "@PERSON" and "@ORGANIZATION." If an essay is talking about two organizations the naming convention, "@ORGANIZATION1" and "@ORGANIZATION2" still retains some information about the text.

The rubric for the different set of essay prompts are also included.The essay rubrics include the rubric guidelines, the essay question, and the type of essay. To keep it simple, the author of this report decided to exclude domain-specific scores and kept only one of the scores as the final score. The scores for each essay set differ, as noted in the associated rubrics. This is, of course, something the author had to deal with in order to provide predictions on the scores.

5 Experiments and Results

5.1 Experiments

An experiment was to convert text into their vector representations in Glove [9] 50, 100, 300. Applying machine learning algorithms on the vector representations were not helpful and performed significantly worse than the baseline.

Deep learning method with bidirectional LSTM model with pre-trained embedding layer was tested but was not found to be generalized well. Only the tokenized essay were provided to the model, with a maximum sequence length of 550 words.

The same model was also tested with a pre-trained embedding layer such as Glove, but it suffered from a similar issue. This may have been due to lack of context in that the essay prompt wasn't provided or the problem with RNNs with long sequences.

5.2 Results

Here the top 2 performing models are contrasted.

Figure 6 shows how the model wasn't able to generalize well. It can be seen that most predictions are made around 2, 3, 4 where most of the observations lie.

Figure 7 shows how the model was able to generalizes cases with scores 2, 3 and 4 but not 0, 1 and 5.

6 Discussion

Looking at the results section above, it can be seen that while RoBERTa delivers low NLL and high accuracy, it struggles to make accurate predictions. Pointing at the heavily shaded area, RoBERTa model seems to be most confused. This can be seen in both the vanilla RoBERTa model and the RoBERTa model with the augmented text.

The second-best model is FastAI network. With embedding layers, the FastAI network is able to distinguish between essays of grades 4 and 5 better than the rest of the models.

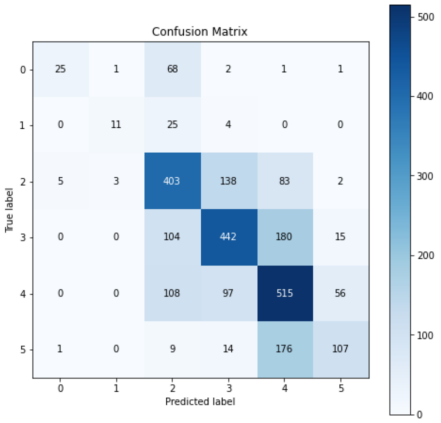

Fig. 6. Confusion matrix of RoBERTa model.

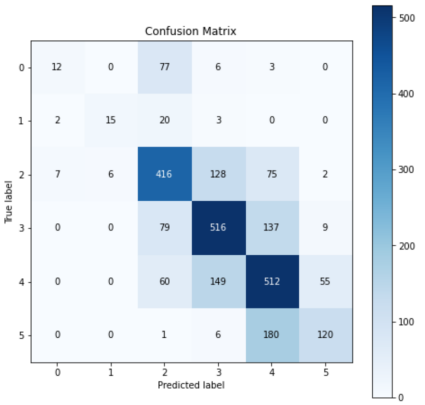

Fig. 7. Confusion matrix of Fastai Embedding Network.

By looking at the models built from a bag of words representation of the text, it can be observed that only the grades between 2 and 4 are correctly classified. That is to say, the grades in the edge cases are mostly misclassified.

The inability for all models to assign 5 to a grade effectively is concerning. It raises several issues such as the incorrectness of the labels, the problem in data collection/ manual grading, and the ineffectiveness of the representation of the underlying text or the incomplete selection of the loss function.

Expanding on the first and the third issue, grading of essays sometimes may seem arbitrary. To combat this, the multiple grades for each essay could be aggregated in some way. In the original competition where this data is taken from, a transformation is applied to the data and a different metric is chosen. The author has not implemented those methods because of unfamiliarity with those concepts. A rigorous exploration of those methods is included in the related works section.

7 Group Member Contributions

This is a one person project. All work was done by the author.

Models	Neg Log Loss (5-fold)	Accuracy (5-fold)	Epochs
RoBERTa + Essay Text + Prompt Id	0.83917	0.64407	1
Fastai Embedding + Custom Features	0.85022	0.63029	5
RoBERTa + Essay Text	0.86970	0.63389	1
Random Forest + Generated Features	0.97735	0.62927	NA
Random Forest + Document Term Matrix	1.00563	0.56908	NA
Random Forest + tfidf	1.04500	0.55770	NA
Fully Connected Network + Generated Features	1.48430	0.42530	10
AWD-LSTM	1.57982	0.26975	3
Fastai Embedding + Document Term Matrix	2.92468	0.43984	5
Fastai Embedding + Tfidf	3.08870	0.43387	5

TABLE 1
Results of different models on the dataset.

	0	1	2	3	4	5
Fastai	12%	93%	87%	99%	92%	98%
RoBERTa	27%	90%	86%	98%	86%	92%

TABLE 2
Results of adjacent agreement on top performing models on the dataset. Column numbers represent the essay grade.

8 Replication Package

The code for the project can be found at https://github.com/uGokalp/Cisc-351-Essay-Project

9 Conclusion and Future Work

The author notes that his work is far from complete and requires more exploration of methods before attempting to use such a model for automated essays. Such methods may include engineering new and improved features that are statistically significant to explain the variance in scores, surveying models made available by researchers in the domain and tweaking existing models such as RoBERTA to achieve state-of-the-art results in essay grading.

References

[1] D. Alikaniotis, H. Yannakoudakis, and M. Rei. Automatic text scoring using neural networks. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

[2] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer. Permutation importance: a corrected feature importance measure. *Bioinformatics*, 26(10):1340–1347, 04 2010.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[4] C. Guo and F. Berkhahn. Entity embeddings of categorical variables, 2016.

[5] J. Howard et al. fastai. https://github.com/fastai/fastai, 2018.

[6] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

[7] E. Loper and S. Bird. Nltk: The natural language toolkit. In *In Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics. Philadelphia: Association for Computational Linguistics*, 2002.

[8] S. Merity, N. S. Keskar, and R. Socher. Regularizing and optimizing lstm language models, 2017.

[9] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[10] T. Rajapakse et al. simpletransformers. https://github.com/ThilinaRajapakse/simpletransformers, 2019.

[11] V. Sanchez, J. Nerbonne, and M. Verspoor. *Identifying Important Factors in Essay Grading Using Machine Learning*. 01 2013.

[12] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.