Final Report

Parth Tamane
ptamane3@gatech.edu

Gaurav Pande
gpande@gatech.edu

Naila Fatima
nfatima3@gatech.edu

Abstract

Automatic essay scoring (AES) aims to objectively assign scores to essays. We explored machine learning and deep learning techniques to achieve this. We found that deep learning performs better than machine learning when the size of dataset is larger, but simpler models give better results on smaller sets. We also tried to understand if there exists any bias towards a gender and an age group in the grading process. Since personally identifiable information in student records is closely guarded by privacy acts like FARPA we used a transfer learning based approach. We found that certain words used more often by males were ranked higher leading to higher average scores for males. This might indicate the presence of gender bias but it's necessary to examine the contents of the essays to determine conclusively. We detected no possibility of a bias for the age groups.

Code repository: `https://github.com/parthv21/NLP-Project-AES`
Video Link:
`https://youtu.be/ZQ4pyrMSCgg`

1 Introduction and Related Work

Essay based questions are a common place in various levels of education. Manual grading by humans is a tedious task and subjective. There has been a lot of interest in creating AES systems for various domains as human grading can be biased towards a group of people based on their gender, age, country, etc. Hence we wanted to explore ML and DL based approaches for AES and check for presence of biases on a standard dataset. To our knowledge we are the first team to use a transfer learning based approach on bias analysis in essay scoring.

In scoring the essays, there are 2 main methods of extracting features from them: 1) Style based features 2) Content based features (Salvatore et al., 2003). In (Kakkonen et al., 2005) (Kakkonen et al., 2006), a term document matrix was used for finding the similarity of the essays with the reference materials like textbooks. We wanted to explore how effective such an approach would be in the setting where a single reading was used as the reference material.

(Larkey, 1998) explored the use of 11 style based features (number of sentences, unique words, etc.) with a Bayesian classifier to predict the quality of an essay. These values were supplied as features to a regression classifier. We identified a gap in the breadth these style based features covered and decided to extend them. Our approach relied solely on linear regression.

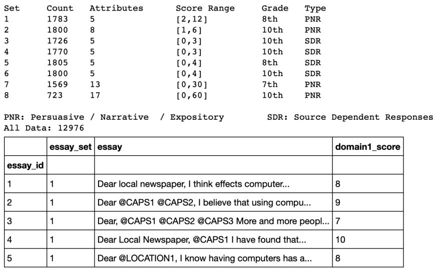
A lot of the literature focuses on using DL based approaches on the task of AES. For instance, the paper by Taghipour and Ng attempted the problem setwise using CNNs and LSTMs. Our approach used original architecture consisting of CNN, LSTM, BiLSTM which were trained for whole dataset as well. In (Liu et al., 2019), the authors used 3 stage learning: first using sentence embedding, then score generation and finally using XGboost to enhance the score. However, we experimented using 2 stage learning only using word2vec and Bert embedding.

The paper by (Yatam and Reddy, 2014) uses n-grams to perform gender prediction in absence of information such as author names, profile photos and profile colors. We used n-grams with different classifiers in order to improve our gender model. (Pentel, 2015) introduced text-readability features which performed better than traditional POS or n-gram features in age prediction models. We were able to use these models in a transfer learning approach to check for the presence of bias in essay scoring.

2 Methods

2.1 Data

For the task of essay grading, we used the Automated Student Assessment Prize (ASAP) dataset by The Hewlett Foundation. (Hewlett, 2012: accessed March 12, 2020) This dataset consists of essays written by students from 7th - 10th grade. The essays are divided into 8 sets with an associated prompt. There are 2 types of prompts, Type 1: Persuasive / Narrative / Expository and Type 2: Source Dependent Responses. Type 1 asks students to state their opinion about certain topic. Type 2 has a required reading associated with it and the students are expected to answer a question based on their understanding of this reading. Each set has a combined domain 1 score. The essays have on an average 275 words.



```
Set     Count   Attributes      Score Range     Grade   Type
1       1783    5               [2,12]          8th     PNR
2       1800    8               [1,6]           10th    PNR
3       1726    5               [0,3]           10th    SDR
4       1770    5               [0,3]           10th    SDR
5       1805    5               [0,4]           8th     SDR
6       1800    5               [0,4]           10th    SDR
7       1569    13              [0,30]          7th     PNR
8       723     17              [0,60]          10th    PNR

PNR: Persuasive / Narrative  / Expository       SDR: Source Dependent Responses
All Data: 12976
```

	essay_set	essay	domain1_score
essay_id			
1	1	Dear local newspaper, I think effects computer...	8
2	1	Dear @CAPS1 @CAPS2, I believe that using compu...	9
3	1	Dear, @CAPS1 @CAPS2 @CAPS3 More and more peopl...	7
4	1	Dear Local Newspaper, @CAPS1 I have found that...	10
5	1	Dear @LOCATION1, I know having computers has a...	8

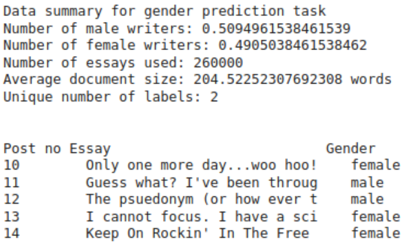
Figure 1: ASAP Dataset Summary

For the author profiling task, we used the Kaggle dataset ”Blog Posts labeled with Age and Gender” (J. Schler and Pennebaker, 2004). This dataset contains blog post entries along with labels specifying the author's gender and age. Since the ASAP dataset contains entries written by 7th-10th graders, we only included blogs written by that age group in order to train the age prediction models. Since the ASAP dataset does not have gender labels, we checked if our gender model generalized well on the dataset ”Blog Author Gender Classification Dataset” (Mukherjee and Liu, 2010a). This dataset consists of 3100 blogs and each blog is labeled with the author's gender- with around 51.8% entries by men and 48.8% entries by women (Mukherjee and Liu, 2010b).

2.2 Models/Analysis

2.2.1 Evaluation Metrics

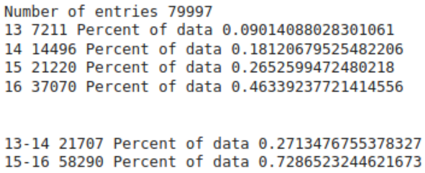
In the task of rating, absolute rating accuracy is a less useful evaluation metric. Instead, it is more useful to see how the 2 ratings correlate. We decided to use Mean Quadratic Weighted Kappa (MQWK) score which was recommended by the The Hewlett Foundation. The Quadratic Weighted Kappa (QWK) score varies from 0 when there is random agreement between raters to 1 when there is complete agreement between raters. If there is less agreement between the raters than expected by chance, then this metric can be negative. QWK score is calculated as follows:



```
Data summary for gender prediction task
Number of male writers: 0.5094961538461539
Number of female writers: 0.4905038461538462
Number of essays used: 260000
Average document size: 204.52252307692308 words
Unique number of labels: 2


Post no Essay                                   Gender
10          Only one more day...woo hoo!        female
11          Guess what? I've been throug        male
12          The psuedonym (or how ever t        male
13          I cannot focus. I have a sci        female
14          Keep On Rockin' In The Free         female
```

Figure 2: Gender Dataset Summary



```
Number of entries 79997
13 7211 Percent of data 0.09014088028301061
14 14496 Percent of data 0.18120679525482206
15 21220 Percent of data 0.2652599472480218
16 37070 Percent of data 0.46339237721414556


13-14 21707 Percent of data 0.2713476755378327
15-16 58290 Percent of data 0.7286523244621673
```

Figure 3: Age Dataset Summary

$$\kappa = 1 - \frac{\sum_{ij} W_{ij} O_{ij}}{\sum_{ij} W_{ij} E_{ij}}$$

Matrix W of weights is calculated using the difference between rater scores. If there are N scores 1, 2, ... , N.

$$W_{ij} = \frac{(i-j)^2}{(N-1)^2}$$

O_{ij} gives the count of essays that received a rating of i by grader A and a rating of j by grader B. While E is an N-by-N histogram matrix of the expected ratings which is calculated by evaluating the outer product of histogram for each graders ratings. It is normalized so E and O have the same sum. (Huyen Nguyen, 2020)

In order to test our gender and age prediction models, we used accuracy as the evaluation metric.

2.2.2 AES using Machine Learning

The machine learning approach used style and content based features. We trained linear regression models using 5 fold cross validation and took the mean value of QWK scores across the folds (scores from regression were rounded off). Figure 4 shows

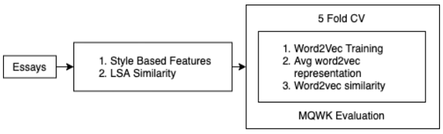


Figure 4: Steps for AES using ML

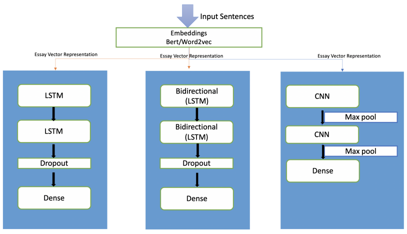


Figure 5: Three separate model architecture consist of BiLSTM, LSTM, and CNN

the steps followed. Essays in set 3 consistently performed poorly for linear regression. Further analysis on this set indicated that a better model to use would be Random Forest.

2.2.3 AES using Deep Learning

In Deep learning we used BERT and word2vec embedding to generate vector representations of essays, and then we input these arrays to the model architecture consisting of BiLSTM, LSTM, CNN and Dense network. The model pipeline includes of below steps:

Essay Vector Representation: We used 2 approaches: 1) Word2Vec: We trained word2vec model on training essay sets to generate word2vec embeddings which serve as input to our deep learning model. 2) Bert: We used Bert and DistilBERT(Sanh et al., 2019) to processes the sentences and pass along information from it to the next layer. DistilBERT is a smaller version of BERT developed by the team at HuggingFace. We used the output corresponding to the [CLS] token as an embedding for the entire sentence.

Model architecture: We wanted to explore LSTM, BiLSTM, and CNNs so we created 3 separate architectures to train the model. As seen in figure 5 the model consists of stacked layers of LSTM/BiLSTM/CNN, dropout/maxpool and Dense network. The output of the Dense network is fed to a Relu activation unit to give the predicted score. Since we are not normalizing the score we didn't use sigmoid activation.

Hyperparameters that were used are shown in figure 8. Other non-extensive parameters like minimum number of words for word2vec didn't make any difference.

2.2.4 Age and Gender Prediction

The gender and age prediction models made use of several features which are explained in detail in 3.1.3. We trained different models on our features and selected the two best-performing models for evaluation. The models that we have used are shown in Table 7.

Gender Prediction For the gender prediction model, we used a Naive Bayes classifier, in combination with n-gram (unigrams) features, as it performed best with a 62.02 % accuracy on the test dataset. We evaluated this model on the generalization dataset in order to ensure that it was properly able to distinguish between the two genders. We were able to get an accuracy of 61.63 % on the generalization dataset, thereby proving that our model could be applied on the ASAP dataset for further analysis.

Age Prediction For the age prediction model, we observed that SVMs, in combination with features described by (Pentel, 2015), performed best with a 74.11 % accuracy on the test dataset.

2.3 Baseline Models

We trained different original models and compared their MQWK score for the task of AES. For machine learning the baseline model was built on the 11 style based features (explained in 3.1.1) using linear regression. For deep learning, we built 3 separate model architectures using the inspiration drawn from Taghipour and Ng. Later we also found that, a team(Kag, 2012) at Carnegie Mellon University built a model using dense and sparse features, trained on the same dataset to achieve the MQWK score of 0.833. For the gender and age prediction task, we considered the SVM model using n-grams by (Yatam and Reddy, 2014) to be our baseline. Yatam's model had a 57.8% accuracy for predicting gender and a 37.78% accuracy for predicting age on a blog corpus. We adjusted the age groups to correspond to school grades 7th-10th.

3 Results

3.1 Experiment setup

3.1.1 AES using Machine Learning

The machine learning approach relied on style and content based features. We used a total of 19 style based features and 3 content based features. The essays were converted in to sentence with "punkt" tokenizer and processed word by word.

Style based features are used to capture information about things of interest in grading like fluency. These can't be directly measured and hence are evaluated using correlated proxies like number of words. (Page, 1994) We added 8 features to the basic 11 style based features used by (Larkey, 1998) (first 11 below) which helped capture more information about the essay.

1. Count of characters in an essay
2. Count of words in an essay
3. Count of unique words in an essay
4. Fourth root of number of words in an essay
5. Count of sentences in an essay
6. Char count / Word count (avg. word length)
7. Word count / Sent. count (avg. sent. length)

8-11. Count of words longer than 5, 6, 7, 8 chars.

12-15. Count of nouns, adjectives, verbs, adverbs

16. Count of small sentences (less than 4 words)
17. Count of punctuations (. / ! / ? / : / ;)
18. Count of number of spelling errors
19. Count of number of stop words

NLTK's PerceptronTagger was used for POS tagging and spelling errors were identified with pyspellchecker.

Content based features are concerned with what the essay actually says. We computed 3 content based features.

Average word2vec representation (avg_word2vec) was generated by training a word2vec model on the training data. Then the word list of the essay was used to generate word vectors and its average was taken for training and testing data.

Similarity Measures are calculated for essay sets 3, 4, 5, and 6. The prompt for each of these sets have a reading associated with it which tells a story and the students are expected to give examples from it. Since we only had one story per set, we divide the story in to 4 paragraphs: Para 1 - story introduction, Para 2 - first anecdote, Para 3 - second anecdote, Para 4 - story conclusion. The splits were made based on how the story was logically progressing. The word lists from these paragraph splits were used to generate 2 different similarity measures:
A. LSA Similarity (lsa_sim): There are total 5 similarity values. This similarity was generated by creating a word count vector representation of the individual paragraphs and one for the whole story. Then cosine similarity was computed between the vectors and word count vector for the essay after LSA was used to reduce the dimensionality of the vector to 6. The word-to-index map used to generated word count vectors was created for each set separately after stemming the word using Porter stemmer.
B. Word2vec Similarity (word2vec_sim): There are 5 similarity values. Unlike LSA similairty, we used the average word2vec representation of each paragraph, the whole story and calculated the cosine similarities with the average word2vec representation of the essay.
We experimented to see how the linear regression model performed on these features using all of the essay sets together and on individual essay sets. The feature combinations used were: 1) Basic 11 style based, 2) All style based, 3) All style based + avg_word2vec, 4) All style and content based.

3.1.2 AES using Deep Learning

We trained deep learning models separately on individual sets and whole dataset. We used BERT and word2vec embeddings for each of these data sets. Reported values are MQWK scores calculated across 5 folds. We trained all the models using Adam optimizer, means-square-error as loss function and relu as activation unit. All these values can be also fed using hyperparameters. We reported the best parameters we found. We removed named entity tags like "@CAPS", "@LOCATION" from the dataset so that these irrelevant tags wont impact our model.

3.1.3 Gender and Age Prediction Models

For our demographic prediction models, we made use of features in table 7. Each essay was tokenized and represented as a list of words and all punctuation marks were removed. For the gender prediction task, we preprocessed the gender labels to have binary values (0 for male, 1 for female). For the age prediction model, we only considered the texts written by authors who belong to the age group 13-16 (corresponding to 7th-10th graders) as the ASAP dataset contains essays written by this age group. Initially, we used 4 labels (corresponding to ages 13, 14, 15 and 16), but we realized that students belonging to the same grade can have different ages. We then divided the essays into two age groups to train our age prediction models: one for 13-14 year olds (7th-8th grade) and the other for 15-16 year olds (9th-10th grade).

Possessive features, used by (Rao et al., 2010), are the counts of two word patterns where the first word is ”my” (such as ”my books”). According to Rao et al., the possessives used by different genders and ages vary- for example, ”my jeep” is more commonly used by men whereas ”my research” is more commonly used by women. The n-gram character and word features were used as they contain information about the number of times a pattern (pattern of characters and words, respectively) is seen in the text. In order to construct our POS counts (unigrams) and bigram features, we used NLTK's PerceptronTagger along with the UPenn Tagset. According to (Mukherjee and Liu, 2010b), using POS features helps as women tend to use emotionally intensive adverbs and adjectives often. The f-measure feature introduced by (Heylighen and Dewaele, 2002) is defined as: 0.5*((f.noun + f.preposition + f.articles + f.adjective)-(f.pronoun + f.verb + f.adverb + f.interjection)+100) where f.i refers to the frequency of the POS tag i. This feature captures contextuality and formality- a lower score indicates contextuality (more pronouns and verbs) whereas as a higher score indicates formality (more nouns). According to Heylighen, this feature can be used to predict the gender of an author as women prefer to use a more contextual style of writing (lower score) whereas men prefer a more formal style of writing (higher score).

In order to extract the sentiment features, we used NLTK's WordNetLemmatizer and SentiWordNet on each word of a training sample. The SentiWordNet indicated the positive, negative and objective scores for each word. We used the sums of positive, negative and objective scores for each word in a training sample as its corresponding feature. We created text-readability features as defined by (Pentel, 2015). These features included information such as the average number of characters in a word, the average number of words in an entry, the complex (cplx) words (words with more than 2 syllables (syl)) to all words ratio, average number of syllables per word and ratios of n-syllable words to all words (we used n = 1 to 8). We used the Pyphen library to detect the number of syllables in a word. We used an 80-20 train-test split.

3.2 Result comparison

3.2.1 AES using Machine Learning

It was our hypothesis that style based features will improve the score for open ended prompts (type 1) and content based features will help improve the score for restricted prompts (type 2). In table 1 it can be seen how the performance improved for Set 1, 2, 7, and 8 when more style based features were added. On the other hand, these additions didn't help a lot for sets 3, 4, 5, 6 which are all type 2 prompts. But once the avg_word2vec feature was added, the performance improved for all type 2 prompts except set 3. We will get to set 3 in a bit. Getting these results required hyperparameter tuning. As the number of training samples in each individual sets were less, a larger size of the word2vec vector was leading to poor performance. With the standard 300 size, the results were quite poor. So we tested for lower values and got a good result with size 50. Also the count threshold for words to be considered needed to be kept lower as the essays were on an average 275 words in length. We kept the minimum words count to 10 words and context size was also 10.

Following this we tested the performance on sets 3, 4, 5, and 6 using all style and content based features (last row in table 1). But the added similarity measures didn't help improve the score a lot since we have one reference document per set. Hence the similarity measures could not capture a lot of useful information that hadn't already been captured by avg_word2vec.

Features	MQWK
Basic 11 style based	0.747
All style based	0.8221
All style based + avg_word2vec	0.9312

Table 2: Evaluation using all sets

Next, we evaluated performance using all the essay sets, first using basic 11 style based features, then using all style based features and finally using all style based features and avg_word2vec. It can be seen that the improvement in performance was quite significant in table 2. The size of the word vectors was kept to 300 since we had enough training data. The minimum word count was still 10 and context size was 10.

Set Number Features	1	2	3	4	5	6	7	8
Basic 11 style based	0.8388	0.6912	0.6438	0.6888	0.7777	0.6626	0.7303	0.7241
All style based	0.8422	0.6977	0.6458	0.688	0.7753	0.6778	0.7703	0.7266
All style based + avg_word2vec	0.8424	0.6921	0.6317	0.7486	0.7961	0.7734	0.7986	0.7115
All style and content based	-	-	0.6287	0.7547	0.7983	0.7897	-	-

Table 1: Mean Quadratic Kappa Scores per Set

Features	MQWK
All style based	0.7569
All style and content based	0.7981

Table 3: Evaluation using all source dependent sets

Set Number	3	4
Linear Regression	0.6287	0.7547
Random Forest	0.7	0.7383

Table 4: Linear Regression vs Random Forest - Set 3 and 4

Next we compared the performance on type 2 sets using all style based features vs all style and content based features. It can be seen in table 3, adding content based features helped boost the score.

Throughout these evaluations, set 3 consistently performed poorly. We believe there are 2 reasons for this. Firstly both set 3 and set 4 have been assigned scores in range 0 - 3. In such a small range, the margin of error is quite low. Hence even a small difference in assigned grade (due to rounding off) would lead to reduced correlation. In addition, the scores assigned to set 3 were quite skewed as compared to set 4 as you can see in figure 6. To address these issues, we tried using Random Forest Classifier. We expected that it would be able to predict the limited number of grades better. This intuition helped improve the score for set 3 but not for set 4 as seen in table 4. The most likely reason for this is grade assignment for set 4 was not skewed. For Random forest, we used 500 estimators, a max depth of 20 was used and minimum sample leaf was kept as 5.

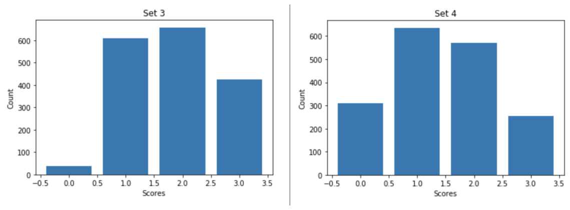


Figure 6: Histogram of Score Distribution

3.2.2 AES using Deep Learning

We tried different models and examined the MQWK score achieved for individual sets and whole dataset. As it can be seen from tables 5 and 6, the deep learning models perform best when trained (using python Keras library) on whole dataset giving 0.9678 MQWK using BiLSTM model and word2vec. It not only beats all other machine learning models used in this project but it also outperformed the best result on the dataset when it was presented in the 2012 Hewlett Foundation Automated Essay Scoring challenge (a MQWK of 0.84107).

The best hyperparameters we found are shown in figure 8. The activation unit ”relu” performed best when compared to others like tanh, prelu for all the models. The best model was BiLSTM, and one of the main reason for its performance is because this architecture traverses the essay in both forward and backward directions.

However the models did not perform well on individual sets for word2vec embeddings. One of the reasons Bert performed well in comparison to word2vec is because Bert is jointly conditioned on both left and right context in all layers with consideration to more historical words than word2vec. Also, it is an attention based mechanism, which focuses on similar words in different context in the sentence and assigns weights according to that. Bert model could have performed much better, but because of computational limits, we were not able to test the model for encoding above length 50. We feel that model overall for both word2vec and Bert did not perform well on sets in comparison to the whole dataset because the length of the individual sets is too small to train properly. Also, it might be due to insufficient search for hyperparameters, as can be seen in figure 9, the loss dropped abruptly in the first few epochs, whereas for full dataset the loss dropped gradually with each epochs as shown in Figure 7.

Set Number	Bert(QWK)	Word2vec(QWK)
1	0.5123	0.5120
2	0.4639	0.4491
3	0.5796	0.3729
4	0.6804	0.5927
5	0.6715	0.5505
6	0.6642	0.6762
7	0.7638	0.5315
8	0.5898	0.4152

Table 5: Set wise QWK score using BiLSTM model

Model	Bert(QWK)	Word2Vec(QWK)
CNN	0.6467	0.7823
LSTM	0.9123	0.9351
BiLSTM	0.9275	0.9678

Table 6: QWK score for whole dataset

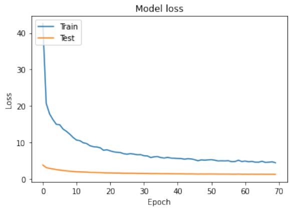
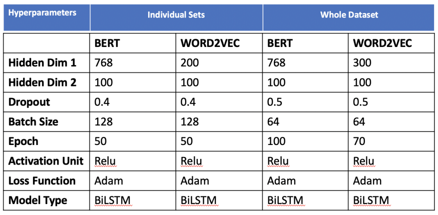


Figure 7: A sample loss curve for training BiLSTM on whole dataset



Hyperparameters	Individual Sets		Whole Dataset	
	BERT	WORD2VEC	BERT	WORD2VEC
Hidden Dim 1	768	200	768	300
Hidden Dim 2	100	100	100	100
Dropout	0.4	0.4	0.5	0.5
Batch Size	128	128	64	64
Epoch	50	50	100	70
Activation Unit	Relu	Relu	Relu	Relu
Loss Function	Adam	Adam	Adam	Adam
Model Type	BiLSTM	BiLSTM	BiLSTM	BiLSTM

Figure 8: Hyperparameters for Deep Learning Model

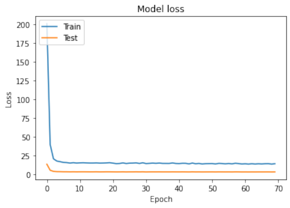


Figure 9: A sample loss curve for training BiLSTM on individual set

Features	Model accuracy			
	Gender		Age	
Possessives	LR 56.198	kNN 52.70	-	
POS counts	SVM 58.236	kNN 57.55	kNN 73.79	LR 73.693
POS bigrams	SVM 59.75		-	
F-measure	SVM 52.5		-	
N-gram (char)	NB 57.17		NB 74.064	
N-gram (word)	NB 62.068	SVM 58.606	NB 65.249	
Sentiment	SVM 51.0	kNN 52.16	-	
Text readability	-		SVM 74.15	

Table 7: Age and Gender prediction accuracy (Blog dataset) [LR: Logistic Regression, NB: Naive Bayes, SVM uses RBF Kernel]

Set	Avg male score	Avg female score
1	8.66 (1461)	7.92 (322)
2	3.57 (1198)	3.10 (602)
3	1.88 (1568)	1.55 (158)
4	1.66 (477)	1.35 (1341)
5	2.60 (464)	2.34 (1341)
6	2.73 (1787)	1.54 (13)
7	14.67 (276)	16.36 (1293)
8	39.27 (52)	36.77 (671)

Table 8: Average score for each gender per essay set

3.2.3 Gender and Age Prediction

Gender prediction model performance is shown in table 7. As we can observe, the possessive features did not yield very good result. This might be the case as in essay writing, authors tend to opt for a more formal style instead of a personal one and therefore, they tend to talk less about themselves. Using both character-level and word-level n-grams we observed that the n-gram word features performed better as they contained more information about the content of the written text. For the gender prediction model, the n-gram features along with the Naive Bayes model had the highest test accuracy (62.068%) and did well on the generalization dataset (61.6318%).

We used f-measure scores for the gender prediction model but noticed that we were not getting good results as the average f-measure scores for both genders were nearly equal. We believed that these features would not work well for the ASAP dataset since essay prompts usually ask for a particular type of question (either contextual or formal) and the tone adopted by the author depends on what the essay asks for. Similarly, when we tested the sentiment features for the gender model, we observed that the values for the positive, negative and objective scores for an essay did not vary much according to the gender.

When we applied gender prediction model on the ASAP dataset, the average normalized male score was 0.646 (7283 males) and female score was 0.543 (5693 females). Since the maximum score for each set was different, we normalized the scores by dividing them with the maximum score for the corresponding set. The per set statistics are shown in table 8. The frequent words used by males were ’people’, ’would’, ’computers’, ’computer’ and ’many’ while those used by females were ’people’, ’like’, ’would’, ’get’ and ’one’. Table 9 shows the top 5 scoring words (which are often present in higher scoring essays) and the average number of times they are used by either gender in an es-

Words	Average usage	
	Male	Female
people	0.6895	0.6072
would	0.7115	0.6075
like	0.6661	0.5820
computers	0.7468	0.6794
one	0.6899	0.6042

Table 9: Top scoring words used in essays

Word feature	7-8 graders	9-10 graders	Normalized score ≥ 0.8
Chars	4.213	4.346	4.508
Essay words	220.933	224.858	285.465
Cplx : All	0.0121	0.0169	0.0224
Syl per	0.3351	0.3371	0.4021
1-syl : All	0.1801	0.1841	0.1900
2-syl : All	0.0577	0.0500	0.0701
3-syl : All	0.0089	0.01475	0.0183
4-syl : All	0.0031	0.0020	0.0039
5-syl : All	7.538-05	10.548 e-05	16.824 e-05
6-syl : All	6.046 e-06	12.458 e-06	15.5627 e-06
7-syl : All	60.762 e-07	5.612 e-07	67.894 e-07
8-syl : All	0	0	0

Table 10: Average feature values based on age group (ASAP dataset)

say. It is interesting to note that males use all of these words more often than females, which may explain why their average scores are higher than those achieved by females. This may indicate the presence of gender bias as certain words are being scored higher. However to make sure of this, we would have to look at the context of the essays.

Age prediction model used text-readability features defined by (Pentel, 2015). We observed that using SVM with these features gave the best performance (74.11 %) for the blog test data. We used this model on the ASAP dataset (which has annotations for grades- we assumed that 7th and 8th graders fall in the age group 13-14 and 9th and 10th graders fall in the age group 15-16) and got an accuracy of 60.257 %.

On the ASAP dataset, we got the results shown in Table 10. The average normalized score of 7th and 8th graders 0.6193 whereas that of 9th and 10th graders is 0.5885. We compared the average values of the text readability features for both age groups. We can observe from Table 10 that the average ratios of n-syllables words to all words (where n=6 and 7) differ slightly for both age groups. We computed the average text readability feature for essays with a normalized score greater than 0.80 (considered to be a good score). We observed that this feature was similar to the younger age group in ratios of 2, 4 and 7 syllable words to all words and similar to the older age group for other features. Since the average features are not completely similar to those of a specific group, we couldn't detect any age bias here.

3.3 Work Division

Naila examined presence of the grading bias, Parth examined AES using ML, and Gaurav examined AES using DL.

4 Conclusion

Any prediction model requires data. The amount of data can crucially affect the kind of results we get. In our experiments, we observed that ML models worked better when the number of training example were lesser. Across sets, the performance was better than (Taghipour and Ng, 2016). On the other hand, DL based approaches gave a good result when evaluating the performance over the entire dataset beating the kaggle score (Kag, 2012). Hence the choice of the model to use will depend on size of the data. For an essay grading application that has just been set up, it would be wise to use ML models. On the other hand, if the software for grading is being transitioned from a human grader to machine grader then deep learning could be used since there would be plenty of essays already graded by humans.

Another interesting result that we noticed was the dependence of grading performance on the distribution of grades. Such effect of grading imbalance was not something we had expected and this certainly opens new avenues of investigation. A possible fix to this problem could involve scaling the grades up so that instated of a score range of 0-3, we could have a score range of 0-100. This will penalize regression based models lesser in terms of MQWK. A further investigation would involve finding the original grade to assign when the predicted score is scaled back to 0-3 range.

The gender prediction model, which performed better than that used by (Yatam and Reddy, 2014), showed that the top 5 scoring words are used more often by males when compared to females. This might indicate why the average normalized score of males is higher and may reflect an underlying bias. However, in order to conclusively determine, we would have to analyze the way these words have been used. The age prediction model showed that the average text readability feature for a normalized score greater than 0.8 is similar to both age groups which is why we do not detect any bias related to age in essay scoring.

References

2012. *Develop an automated scoring algorithm for student-written essays*.

Hewlett. 2012: accessed March 12, 2020. *The Hewlett Foundation: Automated Essay Scoring*.

Francis Heylighen and Jean-Marc Dewaele. 2002. Variation in the contextuality of language: An empirical measure. *Context in Context. Special issue Foundations of Science*.

Lucio Dery Huyen Nguyen. 2020. *Neural Networks for Automated Essay Grading*.

S. Argamon J. Schler, M. Koppel and J. Pennebaker. 2004. *Blog Posts Labeled with Age and Gender*.

Tuomo Kakkonen, Niko Myller, and Erkki Sutinen. 2006. Applying latent dirichlet allocation to automatic essay grading. *Lecture Notes in Computer Science*, 4139:110–120.

Tuomo Kakkonen, Niko Myller, Jari Timonen, and Erkki Sutinen. 2005. Automatic essay grading with probabilistic latent semantic analysis. In *Proceedings of the Second Workshop on Building Educational Applications Using NLP*, pages 29–36, Ann Arbor, Michigan. Association for Computational Linguistics.

Leah S. Larkey. 1998. Automatic essay grading using text categorization techniques. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, page 90–95, New York, NY, USA. Association for Computing Machinery.

Jiawei Liu, Yang Xu, and Yaguang Zhu. 2019. Automated essay scoring based on two-stage learning.

A. Mukherjee and B. Liu. 2010a. *Blog author gender classification data set*.

Arjun Mukherjee and Bing Liu. 2010b. Improving gender classification of blog authors. *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Ellis Batten Page. 1994. Computer grading of student prose, using modern concepts and software. In *The Journal of Experimental Education 62, no. 2*, pages 127–42.

Avar Pentel. 2015. Automatic age detection using text readability features. *Workshop on Tools and Technologies in Statistics, Machine Learning and Information Retrieval for Educational Data Mining*.

Delip Rao, David Yarowsky, Abhishek Shreevats, and Manaswi Gupta. 2010. Classifying latent user attributes in twitter. *Proceedings of the 2nd International Workshop on Search and Mining User-Generated Contents - SMUC 10*.

Valenti Salvatore, Neri Francesca, and Alessandro Cucchiarelli. 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education*, 2.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*.

S. Yatam and T. Reddy. 2014. Author profiling: Predicting gender and age from blogs, reviews social media. *International Journal of Engineering Research Technology (IJERT)*.