BUAN 6341.005: Applied Machine Learning

Final Project Report

Automated Essay Scoring System

Prepared for:

Dr. Yingjie Zhang

Prepared by:

Dan Goldstein | dmg170030

Teenaz Ralhan | txr140230

Manisha Gupta | mxg176230

Jiayi Xu | jxx180009

May 11, 2019

Table of Contents

Introduction

Assessment of student writing in the form of essays is essential in the educational system. It is through the assessment of essays that higher-order learning--evaluate and create--can be demonstrated (Bloom, 1956). However, manual scoring of essays is an expensive and laborious task. If the manual grading process can be relieved by an efficient and accurate automated grading system, essays and other creative methods could be frequently used to indicate academic achievements. We are introducing models that form word representations by learning the extent to which specific words contribute to the text's score and predict a final essay grade on a marking scale. Our models use a large range of textual features that correspond to different properties of text, such as grammar, vocabulary, style, topic relevance, and discourse coherence and cohesion.

Related Work

Neural AES Models:

Alikaniotis et al (2016) presented a deep neural network model using bidirectional Long Short-Term Memory (LSTM) that tapped into not only the context but also the usage of words through a score-specific word embedding (SSWE). Alikaniotis et al (2016) experimented with a number of neural network architectures and noted that the LSTM without the SSWE resulted in a significant drop in the level of accuracies. In the same year, using the same dataset, training protocol, and evaluation measurement, Ng and Taghipour (2016) surpassed the model performance of Alikaniosti et al (2016) by using a LSTM with a mean-over-time layer. Given the limitation of a stand-alone LSTM model in automatic essay scoring, Dong and Zhang (2016) used a hierarchical two-layer Convolutional Neural Network (CNN) without word embeddings with accuracies comparable to Alikaniosti et al (2016) model. Dasgupta et al (2018) developed a deep convolutional recurrent neural network that went beyond word and sentence embeddings to include advanced psycholinguistic features that are inherent in a given text and reported accuracies that best the previously discussed neural network models for automated essay scoring.

KNN AES Models

Bin, Li et al (2008) used K-Nearest Neighbor (KNN) to categorize text and Vector Space Model (VSM) to represent each essay. Features selected for the model included words, phrases and arguments. Term frequency and inversed document frequency (TF-IDF) and information gain (IG) operations were applied to the features. Reported accuracy was 76% using K= 3. Larkey (1998) trained "binary classifiers [Bayesian independence and k-nearest neighbor]to distinguish 'good' from 'bad' essays, and use[d] the scores output by these classifiers to rank essays and assign grades to them." They used essays with different grading scales and with different content domains in Social Science, Physics, and Law. Larkey (1998) then applied the binary classifier and k-nearest neighbor (KNN) scores as additional explanatory variables in a linear regression model. Their conclusion was that Bayesian classifiers outperformed k-nearest neighbor in both experiments they performed. However, they noted that KNN had been used in other essay scoring studies with better results, but those studies used other similarity metrics and more advanced features. Jiang, Hao and Yaru Jin (2016) used latent semantic analysis (LSA) and a tuned KNN method to score essays. The KNN method was modified in two key ways: "…firstly, the distance between test sample and training samples is [was] calculated by weighting the features of essays according their information gain, the other is [was] to weight the scores of k-nearest training samples according their distance from the test sample" (Jiang, Hao and Yaru Jin, 2016). They concluded that the tuned KNN method boosted accuracy significantly compared to the standard KNN method with LSA.

LDA/LSA AES Models

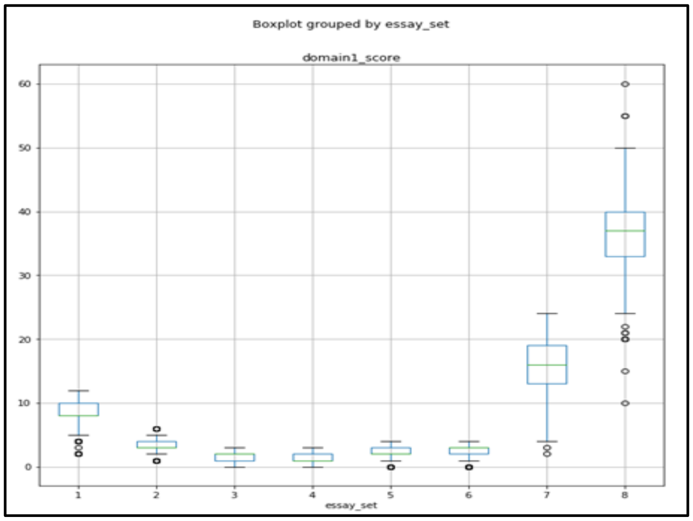
Kakkonen et al (2006) ran a comparative study on essay scoring using Latent Dirichlet Allocation (LDA), Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA) on 283 essays with five different prompts. They used Spearman correlation to assess the inter-rater reliability of each method. After running the experiments they found that LSA and PLSA outperformed LDA, especially on rather small tests sets (100-150 essays). "Although LDA achieved worse results, LDA has some theoretical advantages (*e.g.* dimensionality selection, being not so prone to model overfitting, the consistency of the generative model) compared to the two other methods" (Kakkonen, Tuomo et al. p 110). Hoque, Latiful, and Monjurul Islam (2012) replaced word by document matrix with n-gram by document matrix to transform Latent Semantic Analysis (LSA) into Generalized Latent Semantic Analysis (GLSA), which surpassed the accuracies of LSA. They tested on essays with 800 to 3,200 characters in length and results are similar to human grades with smaller standard deviations than the LSA-based system.

Dataset Description

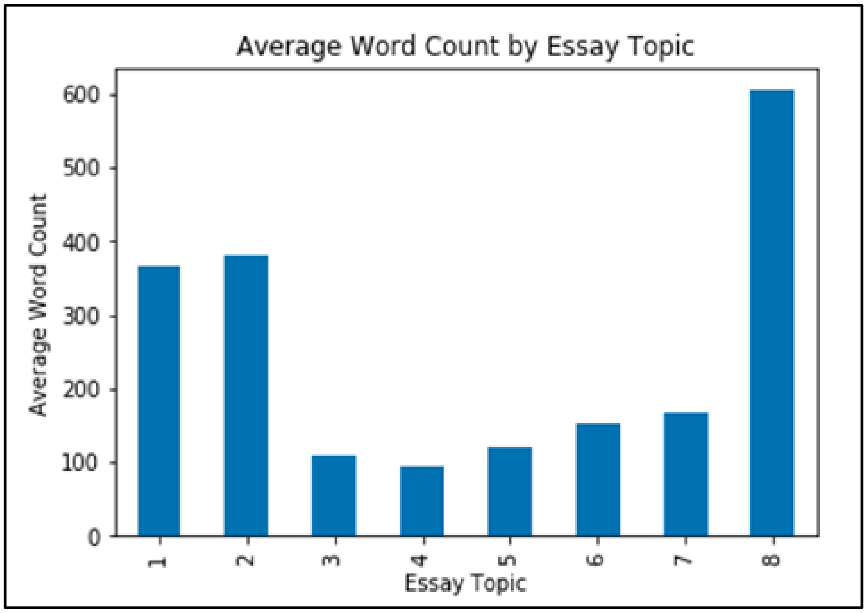
The dataset for this study is from The Hewlett Foundation Automated Essay Scoring competition in 2012 downloaded from Kaggle.com. There are 12,976 records and 8 datasets, with each dataset representing one essay prompt. Respondents were students in the 7th to 10th grades. Essay response length was determined by the given prompt and ranged from 150 to 550 words. We used 20% for testing, the other 80% was used for both training and validation. Essay types were divided into two types of prompts: Persuasive/Narrative/Expository, and Source Dependent. The Persuasive/Narrative/Expository type taps in the respondent's general declarative knowledge while Source Dependent focuses on the respondent's domain-specific knowledge such as Literature, Physics, or Social Studies. Each prompt varied in the degree of domain knowledge that was required. Essays were holistically graded by two graders. Efforts were made to anonymize the data using the Named Entity Recognizer from the Standard NLP group.

Preprocessing Techniques

To provide an overview of our dataset, two graphs were made showing the score scales and the average word count for the eight essay sets:



Essay set 1 was scored on a 10-mark scale and almost all essays were scored from 7 to 8. Set 2 through set 6 were scored on a 5-mark scale. Sets 7 and 8 were scored on a on a wider range with a difference of about 7 points.

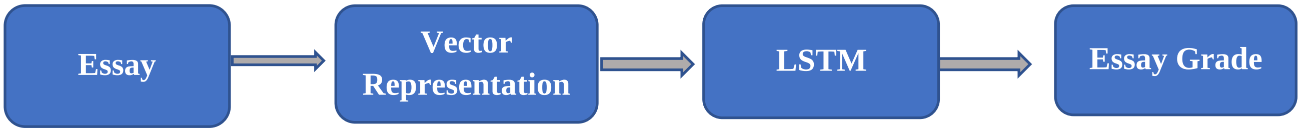


In essay Set 1 and Set 2, all the essays have between 350 and 400 words. Set 3, Set 4, and Set 5 contain less words of over 100. Essays from Set 6 and Set 7 were between 150 and 200 words. Essay Set 8 contained the longest essays with the number of words ranging between 550 and 600.

Proposed Solutions

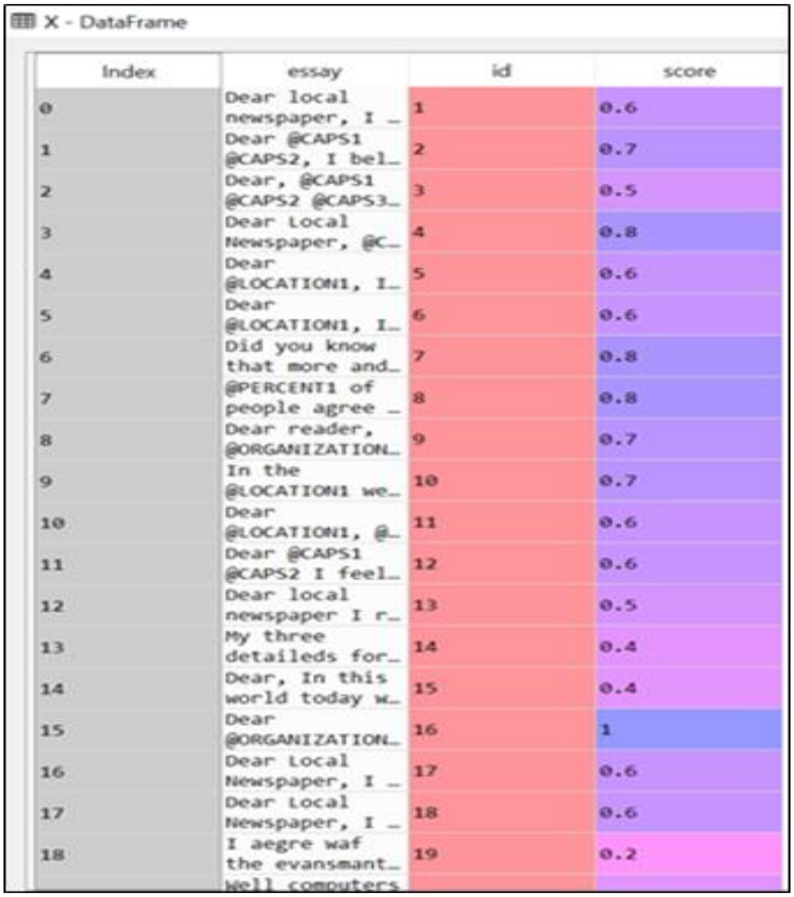
1. Long Short-Term Memory

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. To apply LSTM on this dataset, we divided our task into three steps: data processing, training the models with our loss functions, and tuning of hyperparameters.



Vector Representation of Essays

We represented each essay as a vector using Global Vectors for Word Representation (GloVe) algorithm. "We take [took] the GLoVe word vectors of all the words in an essay, averaged them to get the essay vector" (Dery, Lucio and Huyen Nguyen, 2016). In order to find the best performance parameters, we chose dimensions of GloVe vectors like 50, 100, 200, and 300. Given that the goal is to build an automated essay scoring model and that the essays being graded have been written by middle-school and high-school students, the grading scale can vary. "Our models are capable of acknowledging the difference in scale and outputting the corresponding grade" (Dery, Lucio, and Hyen Nguyen, 2016).

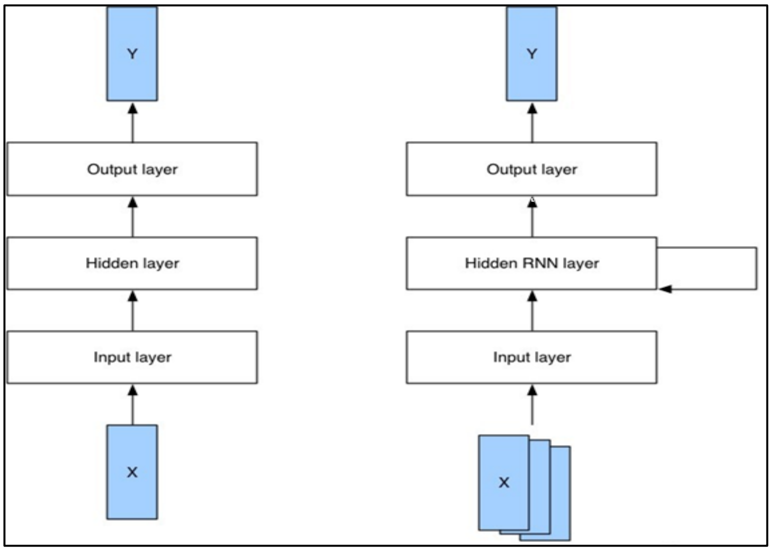


Index	essay	id	score
0	Dear local newspaper, I …	1	0.6
1	Dear @CAPS1 @CAPS2, I bel…	2	0.7
2	Dear, @CAPS1 @CAPS2 @CAPS3…	3	0.5
3	Dear Local Newspaper, @C…	4	0.8
4	Dear @LOCATION1, I…	5	0.6
5	Dear @LOCATION1, I…	6	0.6
6	Did you know that more and…	7	0.8
7	@PERCENT1 of people agree …	8	0.8
8	Dear reader, @ORGANIZATION…	9	0.7
9	In the @LOCATION1 we…	10	0.7
10	Dear @LOCATION1, @…	11	0.6
11	Dear @CAPS1 @CAPS2 I feel…	12	0.6
12	Dear local newspaper I r…	13	0.5
13	My three detaileds for…	14	0.4
14	Dear, In this world today w…	15	0.4
15	Dear @ORGANIZATION…	16	1
16	Dear Local Newspaper, I …	17	0.6
17	Dear Local Newspaper, I …	18	0.6
18	I aegre waf the evansmant…	19	0.2
	Well computers		

Models

First, we created a readout layer to predict the probability of the essays have one of the scores given. We used Categorical Cross-Entropy to minimize the distance between the predicted probability and the actual probability. Then, knowing that y^ represents the predicted grade and the y represents the true grades (each represented as One Hot Vectors), we decided to minimize the equation below

$$\mathrm{CE}(y, \hat{y}) = -\sum_{i} y_i log(\hat{y}_i)$$



The model took in a sequence of word vectors, each of which corresponds to words in the essay sets. Then, it outputted another vector, which has all the data contained within each essay. This output vector was then converted into a score within the given range. We had the following layers:

- 300 unit LSTM input layer
- 64 unit LSTM in a hidden layer

Hyperparameter Tuning

Once we finished building the LSTM model, we tuned Learning Rate (lr), Regularization (l2), Dimension of Essay Vectors (50, 100, 200, 300), and the percentages of sample taken for the training and validation stages in order to gain the most optimal results. However, due to the system memory constraints and the fact that the dataset is big, we were not able to conduct an extensive hyperparameter search on this model.

2. K-Nearest-Neighbors/Lasso Regression

K-Nearest-Neighbors (KNN) is a method for classification. We find the best number of neighbors to use and assign each point to its nearest neighbor. After KNN classification, a class membership is shown with every object classified by calculating Euclidean distances between each neighbor.

Least Absolute Shrinkage and Selection Operator (Lasso) is a regression analysis method which is formulated for least squares models. Different from ridge regression, it uses the sum of the absolute value of coefficients and it is able to set part of a variables' coefficient to zero. Its goal is to minimize prediction error. By setting an appropriate constraint to the model, the coefficients of some variables in a regression will shrink to zero, so that the response will mostly be generated from variables with non-zero coefficients. Before Lasso, stepwise selection was widely used to improve prediction accuracy, but it only works when a few variables are highly correlated with the results.

In our project, KNN and Lasso were used on each subset of the data to classify and analyze all essays. First thing we did here was feature extraction, which meant we needed to transform all essays into feature vectors, so that our models could read and score essays in a standardized and digitized way. Next, we utilized the Natural Language Toolkit (NLTK) package for lemmatization and tokenization. Additionally, a text file from Project Gutenberg was read in to check spell errors for each essay. This file contains a million words which almost covers all the words that can be used in a high school level essay. To analyze the essays at a granular level, ten key features were extracted here: characters count; word count; sentence count; average of word length; lemma count; spelling error count; noun count; adjective count; verb count; and adverb count. Because there are different topics and scoring scales for each essay set, two models were applied set by set.

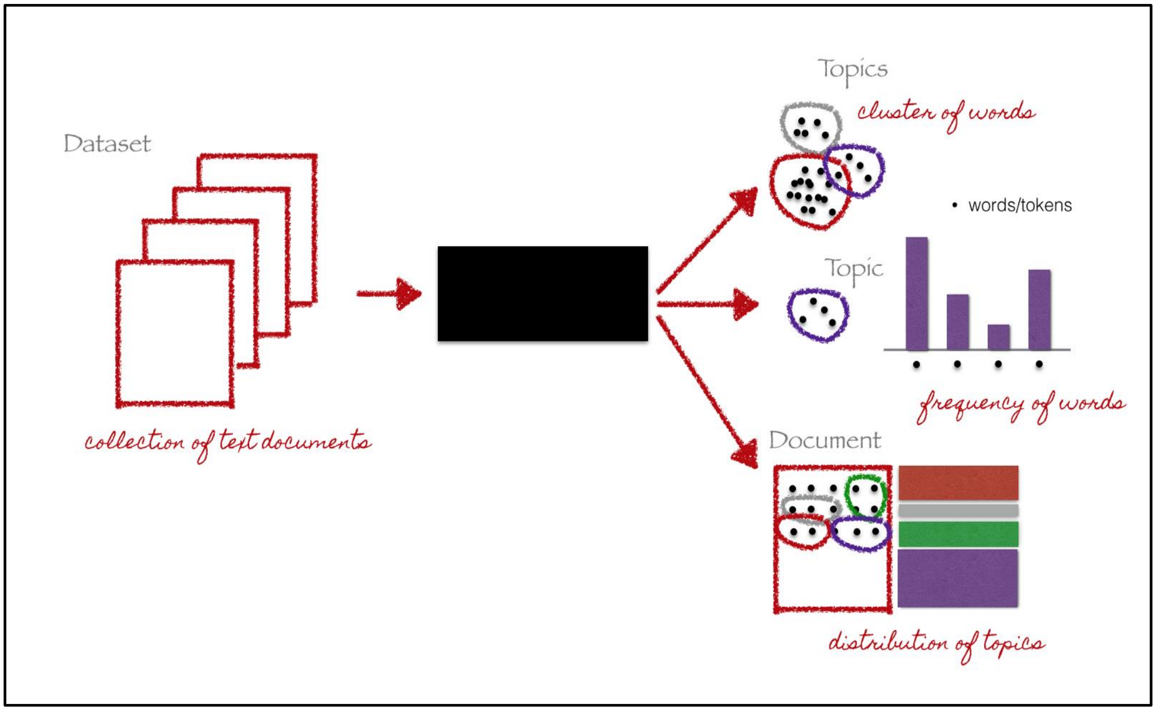
Training and test sets were split into a 7:3 proportion as X_train, X_test, y_train, y_test, with X contained all feature vectors and y contained the domain1 score. Grid search method chose 23 as the best number of neighbors within a range of 5 to 25 for applying KNN for all essay sets except set 3 with 17 neighbors and set 7 with 11 neighbors. For applying the Lasso model, the alpha value was chosen from 3, 2, 1, 0.6, 0.3, and 0.1. Grid search set the best alpha value as 0.1 for all sets except set 8 with an alpha value of 1.0.

3. Topic Modeling

Topic Modeling is an Unsupervised Natural Language Processing (NLP) technique that is charged with discovering "abstract" topics in a set of documents. This form of text mining is a statistical technique that obtains recurring patterns in the works of literature and combines documents that display the most similarities together. Besides its applications in text-mining, topic modeling is also widely used in working with images, genetic information, and bioinformatics (Bansal).

Several different approaches to Topic Modeling exist, including SVD (Singular Value Decomposition), Method of Moments, LSA (Latent Semantic Analysis), and LDA (Latent Dirichlet Allocation). LDA is by-far the most popular approach to text-mining. It gleans the topic of a document or set of documents through the words that appear in the document. LDA finds a correlation between the types of words that appear in a document and then assigns the topic to the document (Xu).

Topic Modeling Process (Doig)



In this project, we applied Topic Modeling to the Automated Essay Scoring dataset as an extension. Through our research, we found that Topic Modeling is not a good method for assigning scores to different essays: however, it is a good method to glean the topics of various essays. In our application of Topic Modeling, we used LDA to develop a model that found the most frequently occurring words in the overall dataset and assigned topics to different essays based on this information.

The first step applied was to clean the data and ensure that there was a base standard that all the essays could be brought to. To accomplish this task, we applied the language check function to standardize the level of grammar in each essay by improving the grammar in some essays. The next step in the standardization process was to lemmatize and tokenize the words in each essay. We used the NLTK (Natural Language Toolkit) available with Python to perform these tasks. The aim of lemmatization is to combine various inflectional forms of words and instead have one standard form of each type of word appear as a method of Text Normalization. For example, the word ‘play’, is the common root form for the words ‘playing’, ‘plays’, and ‘played’. Similarly, the aim of tokenization is to reduce sentences or paragraphs (units) in the essays into a standard form (here, sentences). Both these processes help to convert the essays into a more computer-readable form so that the dataset is fit for performing LDA on (Jabeen).

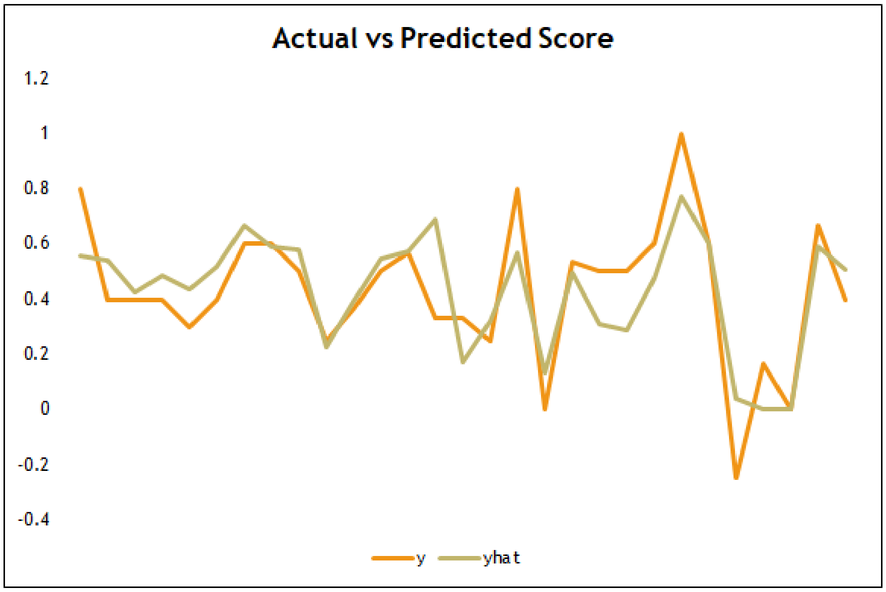
The next step was to apply the Topic Modeling on the dataset. After defining the list of topics and applying a vectorizer to the dataset to convert the dataset into a computer-understandable matrix, we found the most frequently appearing words in the dataset and the probability of each of these words appearing in each of the defined topics. Finally, based on these probabilities and the most frequently appearing words, we were able to classify different essays into each topic. We applied this methodology to all eight topics as they appear in the dataset, five of the eight topics that appear in the dataset, and two of the eight topics that appear in the dataset (Nicolich).

Experimental Results

1. Long Short-Term Memory

We evaluated the performance of the LSTM model with the kappa metric which measures inter-rater agreement for categorical items. “It is generally thought to be a more robust measure than simple percent agreement calculation, as *k* takes into account the possibility of the agreement occurring by chance” (Cohen’s kappa). “This metric typically varies from 0 (random agreement between raters) to 1 (complete agreement between raters)” (Kaggle, 2019). It is possible for the value of the Kappa Metric to be negative which implies that there is no effective agreement between the two raters or the agreement is worse than random. Our best LSTM model achieved a score of 0.82.

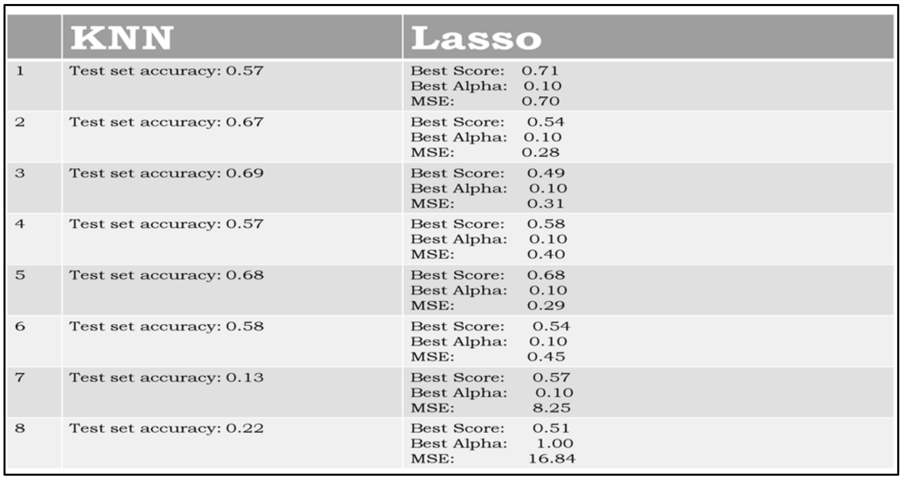
$$\kappa \equiv \frac{p_o - p_e}{1 - p_e} = 1 - \frac{1 - p_o}{1 - p_e},$$



Model	Kappa Score
KNN	51.4%
LASSO	57.8%
LSTM	82.4%

2. K-Nearest-Neighbors/Lasso Regression

After applying KNN and Lasso regression to the eight essay sets, the results from each set are shown in the following table:



	KNN	Lasso
1	Test set accuracy: 0.57	Best Score: 0.71 Best Alpha: 0.10 MSE: 0.70
2	Test set accuracy: 0.67	Best Score: 0.54 Best Alpha: 0.10 MSE: 0.28
3	Test set accuracy: 0.69	Best Score: 0.49 Best Alpha: 0.10 MSE: 0.31
4	Test set accuracy: 0.57	Best Score: 0.58 Best Alpha: 0.10 MSE: 0.40
5	Test set accuracy: 0.68	Best Score: 0.68 Best Alpha: 0.10 MSE: 0.29
6	Test set accuracy: 0.58	Best Score: 0.54 Best Alpha: 0.10 MSE: 0.45
7	Test set accuracy: 0.13	Best Score: 0.57 Best Alpha: 0.10 MSE: 8.25
8	Test set accuracy: 0.22	Best Score: 0.51 Best Alpha: 1.00 MSE: 16.84

As shown in the above table, KNN trained with accuracy score from 0.57 to 0.69 for essay set 1 to 6; however, the accuracy scores were markedly lower on set 7 and set 8. Scores for Lasso hovered in the 50’s, with best score in set 1 with 0.71. MSE scores were not perfect, especially for set 7 and set 8. The reason that both models were not working well on the last two sets might be: the last two set both have a wider range for scoring, instead of 2 to 3 scores range from the first 6 sets, the last two got a score range of 7, which may improve the difficult for prediction. Also, since set 1 and set 2 had longer essays than set 7 but still got higher accuracy, we would say that the length of essay did not affect the accuracy results. In other words, KNN and Lasso may work well when scoring essays with a 10-mark scale or a 5-mark scale.

3. Topic Modeling

We applied topic modeling to three different scenarios within the dataset. The first scenario dealt with all eight datasets and mapped the probabilities of the top words occurring in each topic. Then, we found the top 15 words for each of the eight topics.

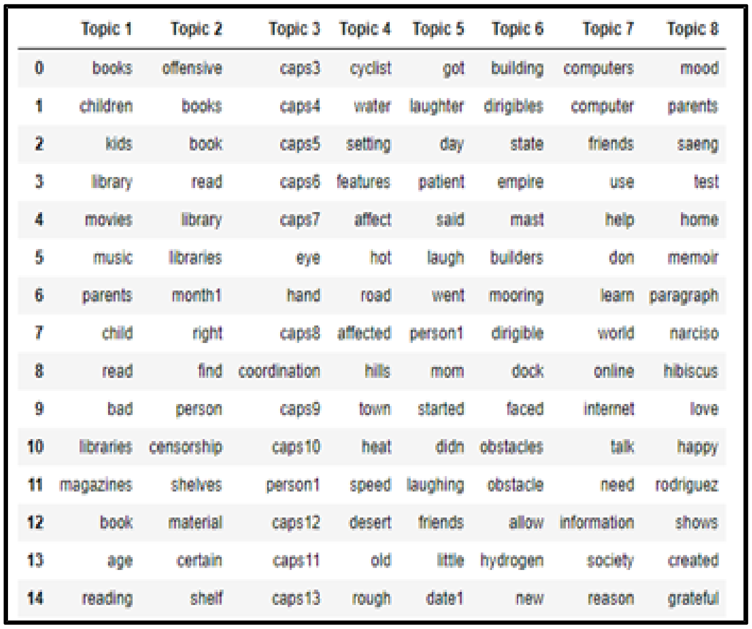
Finally, the model assigned the topic labels to different topics using heatmaps. We performed LDA on both the training and testing datasets as provided in the dataset. We followed this procedure using all eight topics, five of the eight topics (randomly selected), and two of the eight topics (randomly selected). The images below show the heatmaps for each scenario for the testing dataset and the top 15 words for each scenario.

Scenario 1: All Eight Topics

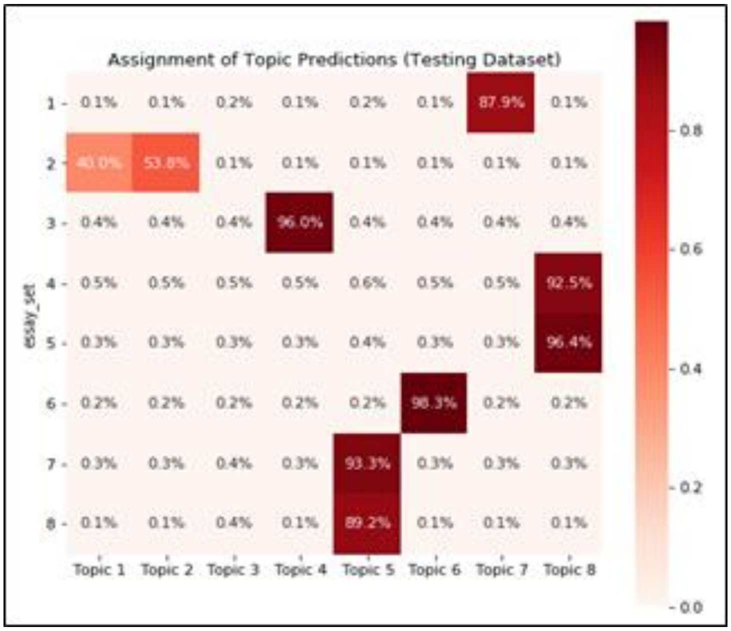
In this scenario, we took all eight topics:

1. Effect of Computers
2. Censorship Views
3. Effect of setting on a cyclist
4. Winter Hibiscus
5. Mood in memoir
6. Empire State Building and Dirigibles
7. Patience
8. Laughter

The images below show the Top 15 words for each of the eight topics and the heatmap of Topic Assignments.



	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6	Topic 7	Topic 8
0	books	offensive	caps3	cyclist	got	building	computers	mood
1	children	books	caps4	water	laughter	dirigibles	computer	parents
2	kids	book	caps5	setting	day	state	friends	saeng
3	library	read	caps6	features	patient	empire	use	test
4	movies	library	caps7	affect	said	mast	help	home
5	music	libraries	eye	hot	laugh	builders	don	memoir
6	parents	month1	hand	road	went	mooring	learn	paragraph
7	child	right	caps8	affected	person1	dirigible	world	narciso
8	read	find	coordination	hills	mom	dock	online	hibiscus
9	bad	person	caps9	town	started	faced	internet	love
10	libraries	censorship	caps10	heat	didn	obstacles	talk	happy
11	magazines	shelves	person1	speed	laughing	obstacle	need	rodriguez
12	book	material	caps12	desert	friends	allow	information	shows
13	age	certain	caps11	old	little	hydrogen	society	created
14	reading	shelf	caps13	rough	date1	new	reason	grateful



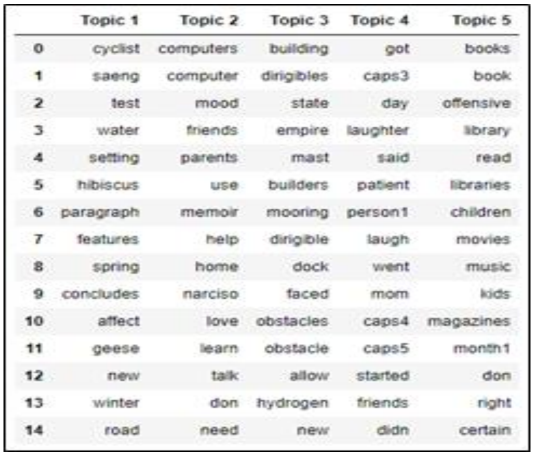
We can see that the model has the most success in correctly identifying Topic 6 while it does poorly at identifying the remaining topics. A contributing reason for this could be the Top 15 words found for each topic. We see that Topic 1 is about the effect of computers, but books, children, and library are some of the most frequently found words and these words are unrelated to computers which is why we see that the model identifies Topic 1 as being most related to Topic 7 because the most frequently found words in that topic are related to computers like computer, internet, and online.

Scenario 2: Five of the Eight Topics

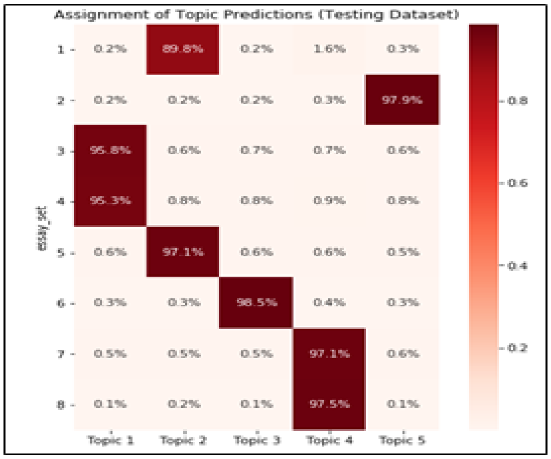
In this scenario, we randomly chose five of the eight topics:

1. Effect of Computers
2. Mood in memoir
3. Empire State Building and Dirigibles
4. Patience
5. Laughter

The images below show the Top 15 words for each of the five topics and the heatmap of Topic Assignments.



	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
0	cyclist	computers	building	got	books
1	saeng	computer	dirigibles	caps3	book
2	test	mood	state	day	offensive
3	water	friends	empire	laughter	library
4	setting	parents	mast	said	read
5	hibiscus	use	builders	patient	libraries
6	paragraph	memoir	mooring	person1	children
7	features	help	dirigible	laugh	movies
8	spring	home	dock	went	music
9	concludes	narciso	faced	mom	kids
10	affect	love	obstacles	caps4	magazines
11	geese	learn	obstacle	caps5	month1
12	new	talk	allow	started	don
13	winter	don	hydrogen	friends	right
14	road	need	new	didn	certain



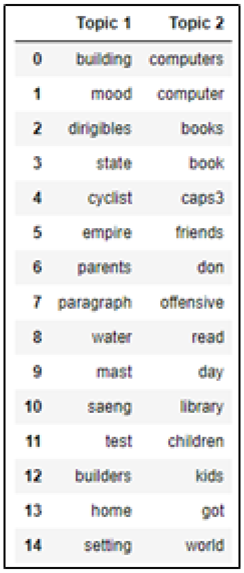
We can see that the model has minimal success correctly identifying the five topics. The reason for this is the Top 15 words found for each topic. We see that Topic 1 is about the effect of computers, but cyclist, hibiscus, and geese are some of the most frequently founds words, and these words are unrelated with computers which is why we see that the model identifies Topic 1 being most related to Topic 2 because the most frequently found words in that topic are related to computers like computers and computer.

Scenario 3: Two of the Eight Topics

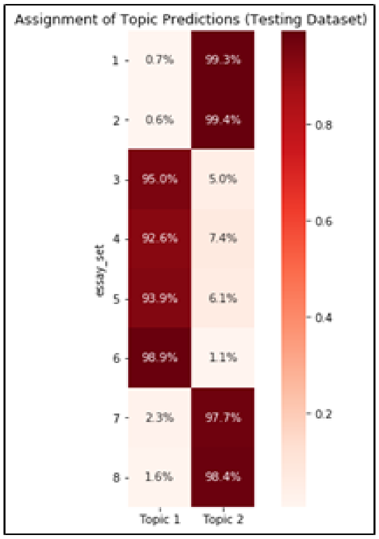
In this scenario, we randomly selected two of the eight topics:

1. Effect of Computers
2. Laughter

The images below show the Top 15 words for each of the two topics and the heatmap of Topic Assignments.



	Topic 1	Topic 2
0	building	computers
1	mood	computer
2	dirigibles	books
3	state	book
4	cyclist	caps3
5	empire	friends
6	parents	don
7	paragraph	offensive
8	water	read
9	mast	day
10	saeng	library
11	test	children
12	builders	kids
13	home	got
14	setting	world



We can see that the model has the most success in identifying Topic 2 correctly with an accuracy of over 99%. Once again, we see that the model has little success in identifying Topic 1 correctly.

Conclusion

In general, KNN and Lasso perform better when the total score range of all essays are less than 5 points. As the range becomes wider to 7 or 8 points, it may lose a considerable amount of accuracy. Topic Modeling does not provide a very good accuracy as far as preparing the essays for automated grading. We see that, while the model provides a very high accuracy in correctly assigning the topics in some cases, it performs poorly in other cases. This has to do primarily with the words identified as being the most frequent words for each topic. It is possible that a different approach to Topic Modeling like LSA (Latent Semantic Analysis) or SVD (Singular Value Decomposition) might provide better and more consistent accuracies. Those implementations could be an extension to this project.

The LSTM model greatly improved upon the standard achieved by KNN/Lasso. The best neural network models achieved a score of 0.82 on the Kappa metric as against the original best of 0.57. This verifies the vast potential of neural network architectures in solving natural language processing problems. It was surprising that our simple neural network model using 300-dimensional Glove as initialization to the embedding layer was our most successful model. We believe that a more patient hyperparameter search with our LSTM based models could outperform this result. However, due to computing resource constraints we were unable to find these optimal parameters set for the LSTM models.

Team Member Contribution

Team Member Name	Contribution/Role
Dan Goldstein	Final Report Presentation Project Topic Selection Research Fine-tuning aspects of final product
Teenaz Ralhan	Final Report Presentation Project Topic Selection Research Topic Modeling
Manisha Gupta	Final Report Presentation Project Topic Selection Research LSTM
Jiayi Xu	Final Report Presentation Project Topic Selection Research Lasso/KNN

References

- Alikaniotis, Dimitrios, et al. “Automatic Text Scoring Using Neural Networks.” *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* 2016, pp 715-725.

- Bansal, Shivam, et al. “Beginners Guide to Topic Modeling in Python.” *Analytics Vidhya*, Analytics Vidhya, 11 Jan. 2019, www.analyticsvidhya.com/blog/2016/08/beginners-guide-to-topic-modeling-in-python/.

- Bin, Li, et al. “Automated Essay Scoring Using the KNN Algorithm.” *2008 International Conference on Computer Science and Software Engineering.* 2008, pp 735-738.

- “Cohen’s kappa.” Wikipedia, Wikimedia Foundation, 11 May. 2019, en.wikipedia.org/wiki/Cohen%27s_kappa.

- Dasgupta, Tirthankar, et al. “Augmenting Textual Qualitative Features in Deep Convolution Recurrent Neural Network for Automatic Essay Scoring.” *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, 2018, pp 93-102.

- Dery, Lucio and Huyen Nguyen. “Neural Networks for Automated Essay Grading.” 6 June 2016. CS224n Deep Learning for Natural Language Processing, Stanford University, student paper.

- Doig, Christine. “Topic Modeling Process.” *Medium.com*, Ankur Tomar , 24 Nov. 2018, medium.com/@tomar.ankur287/topic-modeling-using-lda-and-gibbs-sampling-explained-49d49b3d1045.

- Dong, Fei, and Yue Zhang. “Automatic Features for Essay Scoring--An Empirical Study.” *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp 1072-1077.

- Dscape. “Dscape/Spell.” *GitHub*, 13 Dec. 2011, github.com/dscape/spell/tree/master/test/resources.

- Islam, Monjurul, and Latiful Hoque. “Automated Essay Scoring Using Generalized Latent Semantic Analysis.” *Journal of Computers*, vol. 7, no. 3, 2012, pp. 616-626.

- Jabeen, Hafza. “Stemming and Lemmatization in Python.” *DataCamp Community*, DataCamp, 23 Oct. 2018, www.datacamp.com/community/tutorials/stemming-lemmatization-python.

- Jiang, Hao, and Yaru Jin. “Research on the Automatic Scoring Method of English Essay based on the Improved K-Nearest Neighbor Algorithm.” *3rd International Conference on Education, Management, and Computing Technology*, 2016, pp 1297-1302.

- Kakkonen, Tuomo et al. “Applying Latent Dirichlet Allocation to Automatic Essay Grading.” *FinTAL*, 2006, pp 110-120.

- “k-Nearest Neighbors Algorithm.” Wikipedia, Wikimedia Foundation, 21 Apr. 2019, en.wikipedia.org/wiki/K-nearest_neighbors_algorithm#Properties.

- “Lasso (Statistics).” *Wikipedia*, Wikimedia Foundation, 28 Mar. 2019, en.wikipedia.org/wiki/Lasso_(statistics).

- Larke, Leah. “Automatic Essay Grading Using Text Categorization Techniques.” *21st ACM-SIGIR International Conference on Research and Development in Information Retrieval.*

- Ng, Hwee, and Kaveh Taghipour. “A Neural Approach to Automated Essay Scoring.” *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp 1882-1891.

- Nguyen, Huyen, and Lucio Dery. “Neural Networks for Automated Essay Grading.” *Stanford University* , Stanford University, cs224d.stanford.edu/reports/huyenn.pdf.

- Nicolich, Jeff. “Turanga1/Automated-Essay-Scoring.” *GitHub*, Github, 27 Dec. 2018, github.com/Turanga1/Automated-Essay-Scoring/blob/master/0_EDA_and_Topic_Modeling_with_LDA.ipynb.

- “Quadratic Kappa Metric Explained in 5 Simple Steps.” *Kaggle* 11 May 2019, https://www.kaggle.com/aroraaman/quadratic-kappa-metric-explained-in-5-simple-steps

- “Revised Bloom’s Taxonomy.” *Iowa State University*, 2019, http://www.celt.iastated.edu/teaching/effective-teaching-practices/revised-blooms-taxonomy.

- Shubhpawar. “Shubhpawar/Automated-Essay-Scoring.” *GitHub*, 26 Apr. 2018, github.com/shubhpawar/Automated-Essay-Scoring/blob/master/automated_essay_scoring.ipynb.

- Xu, Joyce. “Topic Modeling with LSA, PSLA, LDA & lda2Vec.” *Medium*, Medium, 25 May 2018, medium.com/nanonets/topic-modeling-with-lsa-psla-lda-and-lda2vec-555ff65b0b05.