Decreasing the workload for overloaded teachers with automated short answer response grading

IAN MCNAIR
NORTHEASTERN ILLINOIS UNIVERSITY

Abstract

Grading is a task that should already be automated for teachers everywhere. This study looks at two simple machine learning models for a total of three approaches for automated grading: clustering (K-Means), classification (Logistic Regression), and a combination method. The combination method is a novel way of using clustering to define a training set for the classification method despite its very average performance. Expectedly, classification yields the highest accuracy of 78%, whereas clustering and the combination score essentially the same at 64% across 84 different datasets. These models are also embedded into an online web application that allows them to be used by anyone. Overall, these results are very promising because of the simple machine learning models used. If better models are developed, they can easily be injected into the online application, allowing the automation to grow in its performance democratizing automated grading for teachers everywhere.

Contents

I. Introduction

Teachers have a lot on their plates. On top of creating and curating curriculum, they grade, manage, discipline, console, as well as other duties given out by administration. Many of these skills are difficult, if not impossible to automate, but there is one area that has hope: grading. Although questions come in various shapes and sizes, to some degree, it should be possible to automate some, if not all the process. This project aims to address one very narrow area of this spectrum, short answer responses. They involve more complicated processes than choice-based questions in terms of grading, but potentially also allow for a teacher to gather much more information about student understanding.

Question types very greatly, from multiple choice, which can already be automated, to essays. [17] outlines twelve different question types divided into six categories, encompassing two levels of depth of learning, “Recognition” and “Recall”. Recognition questions, also known as closed or passive questions according to [6] and [7], usually only require the responder to select or organize information e.g. multiple choice, ranking. Recall questions require responders to come up with original answers, which can vary greatly between respondents while all being correct (or incorrect). These also vary greatly, from essay answers to math or even coding, but short answer responses also fall into this category. [17] defines short answers as one paragraph to one sentence, where the focus is on content, and the openness of the question is closed. Essays for example would be two or more paragraphs, where the question is very open.

Although automated short response grading has been thoroughly investigated at a variety of levels and complexities, this project aims to present machine learning models that are easily accessible and usable by teachers, which can be easily upgraded as more research is done. Many of these more complex systems are not readily usable by actual teachers and the ones that are available tend to be

hidden behind paywalls. The goal then is to see if an adequate, accessible, automated system can be created, democratizing automated short answer response grading for all teachers.

II. Related Work

[17] organizes the history of automated short answer response grading into five distinct eras: Concept mapping, information extractions, corpus-based methods, machine learning, and evaluation. Each of these eras made significant improvement towards automating the short answer grading process. The below figure outlines this, describing dates and models or methods created within that era.

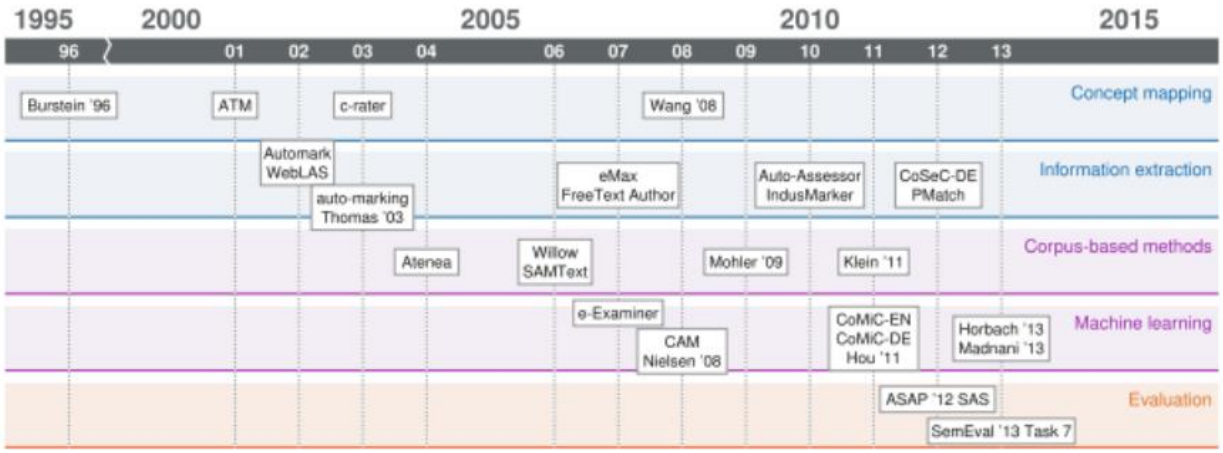

Figure 1: From [17]. Describes the eras and trends of short answer response grading.

The concept mapping era is described as considering "the students answers as made up of several concepts, and to detect the presence or absence of each concept when grading." An example of a model in this category is ATM created by [4], or the Automatic Text Marker. This system is one that broke down student and teacher answers into conceptual words, which then created a score based on the similarities.

The information extractions era is characterized by methods which search the student answer for certain concepts, which amounts to pattern matching techniques for key words or phrases. eMax

[18], a process from that era, required educators to identify and assign weights to certain elements in their answer, which were then used for calculating the student's final score.

Corpus-based methods are ones which utilize larger documents in which aid in comparing to the teacher answer. Since one answer from a teacher is limited, the added document allows of a larger space of correct answers, including synonyms. [12] uses a process which uses the student answers as the document instead of another external one. Selected answers are then manually graded which allow for the automated grading of the rest.

During the machines learning era, features are developed using natural language processing techniques, which are all then utilized to form a score through use of some type of model. Specifically, this era only contains supervised methods and not unsupervised ones. An example of this is from [8], which utilize a support vector machine classifier with features extracted from a variety of sources, importantly term frequency, inverse document frequency.

Lastly, the era of evaluation is unique in that the method of grading is independent. What mostly matters in this era is the shared use of datasets. Kaggle, an online community which hosts many machine learning contests, is an example of this shared data concept.

Within some of these eras, although unsupervised learning was used to some extent, clustering was largely left out. [2] and [3], both from Microsoft Research, discuss the use of clustering not in automated grading, but as a process to reduce the grading effort. They mention that most teachers did like the impacts of clustering, which allowed teachers to either grade at scale more quickly via assigning the same score to groups of answers or just by grading similar answers together, which is also appeared to decrease the effort spent on grading.

III. Methodology

The process followed a standard set of procedures: data acquisition, data cleaning, feature engineering, modelling, evaluation, and application. Some of these processes required much more time and effort than others, either because of the creativity involved or simply because of the potential size of the data. The below figure details the general timeline for this project.

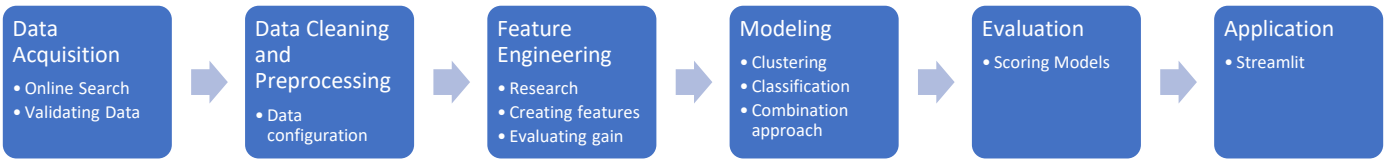

Figure 2: Describes the general process and tasks associated with this project.

The Data

The raw data was taken from [15]. An example is shown in the figure below. It consists of student answers as well as the teacher's correct answer. Student answers were also prelabeled as correct or incorrect. Finally, only single sentence answers were studied. In total, 84 different question

student_answer	teacher_answer	label	question_id
Because they are repelling each other always.	Like poles repel and opposite poles attract.	0	0
The magnets are not touching because they cannot attract to each other if they are north to north or south to south.	Like poles repel and opposite poles attract.	1	0
The magnets will maybe not stick because the force of the magnets will maybe the pencil is long and the magnets cannot feel the other magnets.	Like poles repel and opposite poles attract.	0	0

Table 1: Sample of raw data used.

sets were studied. Each of the answers, student or teacher, is also preprocessed in three ways, stop word removal, ordering, and stemming, although others were considered and ultimately not used.

Stop words are ones that have little meaning or add very little information in NLP processes. Although each package is different in what is considered a stop word, Spacy, the package used in this project, includes all the words found here [21]. Ordering is the process of breaking each answer into alphabetical order. This process sometimes helps when similarity measures are used. Finally stemming is the process of reducing words to their root, or "stem" word. For example, the word contain has variations: contains, contained, container, containing, and so on. Each of these, at least for purposes of the project, give the same amount of information, but would not result in a true statement if simply compared. Stemming would reduce each of these words to "contain" which would be equivalent. However, it should be noted that the stemming process is algorithmic and does not always result in a stem that matches the unstemmed word. For example, catty and catlike would result in the stem of "cat", making these two words equivalent. The Porter Stem [20] is specifically used in this project. There does exist other methods for removing the endings of words, like lemmatization, which also generates the root word. Although lemma's have the benefit of generating actual words, for the purposes of this project, it was not utilized because the information gain was not as significant as the stemmed words.

q_stopwords	q_stemmed	q_stem_ordered
repelling	repel	repel
magnets touching attract north north south south	magnet touch attract north north south south	attract magnet north north south south touch
magnets maybe stick force magnets maybe pencil long magnets feel magnets	magnet mayb stick forc magnet mayb pencil long magnet feel magnet	feel forc long magnet magnet magnet magnet mayb mayb pencil stick

Table 2: Same sample as figure 1 after data preprocessing.

An example of the final preprocessed form of the data is shown below in the table above.
Along with the student's original question, the student's answer has stop words removed, is stemmed, and is also ordered. The teacher's answer is also preprocessed the same way and will be discussed more in the feature engineering section.

Feature Engineering

The most time-consuming portion of the project, feature engineering entails “the process of using domain knowledge to extract features from raw data via data mining techniques.” [13] The student answers, even in their processed form, cannot be utilized by the machine learning algorithms. In some way, shape, or form they need to be turned into numerical values. Figure 3 details each of the created features.

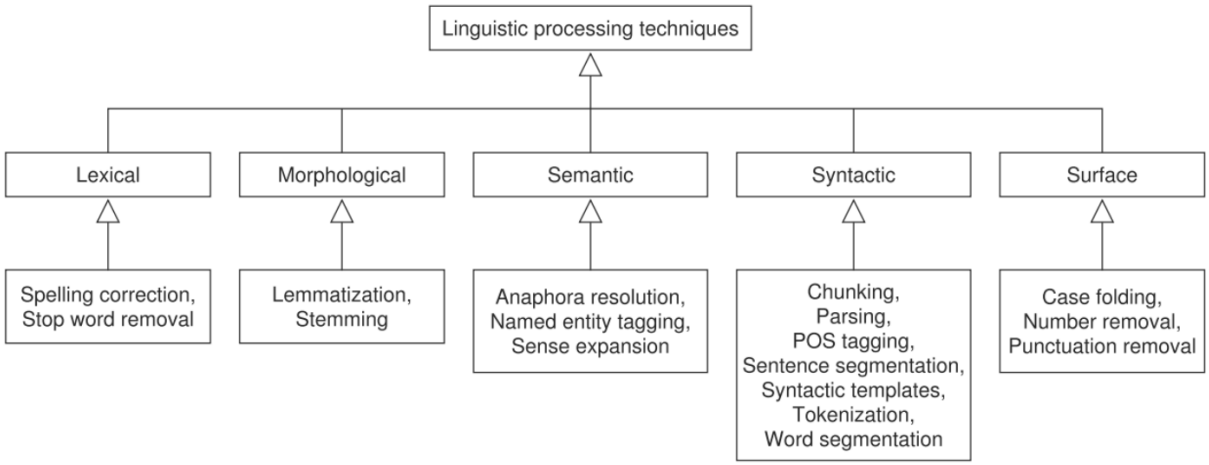

Figure 3: From S. Burrows et al. Describes the various groups of features and processes used in the automated short answer response area.

The features come from three basic areas, sentence statistics, similarity measures, and n-gram/bag of words. [17] outlines many of the various areas and types of feature engineering that has been done in the past.

Sentence statistics are essentially numerical values that describe the sentence. Number of words, count of commas, number of verbs, etc. Usually these types of features give information about the structure of the sentence.

Similarity measures utilize word vectors in order to compare separate sentences. The three primary similarity measures used were generic, Jaccard, and cosine similarity. Generic similarity is not an

official term, but something that was developed to measure sameness between two answers. It is a simple count between the number of words in the student's answer that is also in the teacher's answer, divided by the total number of words contained in the teacher's answer.

$$G = \frac{|S \cap T|}{|T|} \tag{1}$$

Equation 1: Equation representing generic similarity. S represents the word vector of the students answer while T represents the teacher's answer. The magnitude of these vectors is the length of them.

Jaccard similarity, also known as the Jaccard index, is the intersection of two vectors divided by the union of these same vectors.

$$J = \frac{|S \cap T|}{|S \cup T|} \tag{2}$$

Equation 2: Jaccard index (or similarity) equation.

The highest score of one would mean that each word vector had the same words. One big difference between this similarity measure and the generic one mentioned above is that Jaccard similarity ignores multiple occurrences of the same word. For example, "Cat jumped over the dog" would be perfectly similar to "dog jumped over the cat over the dog" according to Jaccard similarity but not generic similarity because duplicity matters in generic similarity. Finally, cosine similarity measures the direction both vectors point in space, regardless of the magnitude. Mathematically it is the dot product of the two vectors divided by the product of the magnitude of each vector.

$$C = \frac{S \cdot T}{\|S\|\|T\|} \tag{3}$$

Equation 3: Cosine similarity equation. The magnitude represents the vector magnitude.

Each vector is constructed differently from the two above in that the number of repeated words matter. Each vector contains numerical values, with each dimension representing a unique word in the feature space, like the bag-of-words document vectors. Since each unique word indicates a different direction

the vector could point, the order of the sentences does not matter. They are simply counted and added to the appropriate column, as seen below.

VECTOR	CAT	DOG	JUMPED	MOON	OVER
DOG JUMPED OVER MOON	0	1	1	1	1
CAT JUMPED OVER DOG OVER MOON	1	1	1	1	2

Table 3: Describes examples of two document vectors.

As seen above, although magnitude does not matter (two vectors that point in the same direction, with differing magnitudes will score the highest similarity), the counts in each column affect the overall direction and thus the similarity. Scores for each of the methods based on the example word vectors are seen below.

METHOD	SCORE
GENERIC	0.5
JACCARD	1
COSINE	0.88

Table 4: Scores of each method used in this project utilizing table 3.

The last group of features are n-gram based features. All of these features are essentially flags which denote whether a student's sentence contains certain words also in the teacher's answer. N-grams are specifically a sequence of *n* number of words. A tri-gram would look at each group of three sequential words within a sentence. For example, a tri-gram from the sentence "This can automatically grade answers" would be "can automatically grade". The three types of n-grams used are uni-grams, bi-grams, and tri-grams. Uni-grams are essentially keyword flags for every word in the sentence. The bi-gram feature goes through each pair of words within the teacher's answer and checks if the student's answer contains that same bi-gram. The tri-gram features are essentially the same, but with a sequence of three words instead of two.

VECTOR	TRI-GRAM 1	TRI-GRAM 2	TRI-GRAM 3
CAT JUMPED OVER THE DOG	Cat jumped over	Jumped over the	Over the dog

Table 5: Example of tri-grams from a vector.

Machine Learning Models

The models utilized within this project are k-means clustering and logistic regression. K-means clustering is known as an unsupervised model. Unsupervised models are distinct in that they try to find patterns in the data themselves without any additional data to train on. Supervised models, which is what logistic regression is a part of, requires a training set and a test set. There is finally a third method, which involves a combination of both, but would still be considered an unsupervised method. These is discussed in more detail below.

K-Means Clustering

K-Means clustering is an unsupervised method that tries to determine clusters within the feature space. The K-Means algorithm works by trying to place centroids at the center of groups of data points. Initially these centroids are randomly chosen, and depending on the distance metric, try to find the center of mass of various groups of data. Since there exists data sets which may not have any structure to find, the initialization of these centroids greatly affects which points belong to which cluster. As explained by [14], this algorithm aims to choose centroids that minimize the inertia, or within-cluster sum-of-squares criterion.

$$\sum_{i=0}^{n} \min_{\mu_j \epsilon C} \left(\|x_i - \mu_j\|^2 \right) \quad (4)$$

Equation 4: Within-cluster sum-of-squares criterion that K-Means tries to minimize.

Inertia is not a normalized metric. In general, the closer to zero inertia is, the better, but it depends on the data. For example, suppose you have one centroid per observation. Inertia in this case would be zero, because each centroid would converge exactly on each data point, but this is not a helpful use of clustering. Yet in high dimensional space, Euclidean distances tend to explode, but this does not mean a high inertia value means bad clustering. In general, it is all very relative. Dimensionality reduction techniques like principal component analysis (PCA) are helpful in this case, not just for helping the optimization process but also for the visualization process. If there are clusters to be found in high dimensional space, first using PCA to reduce to two or three dimensions, then visualizing it in that space

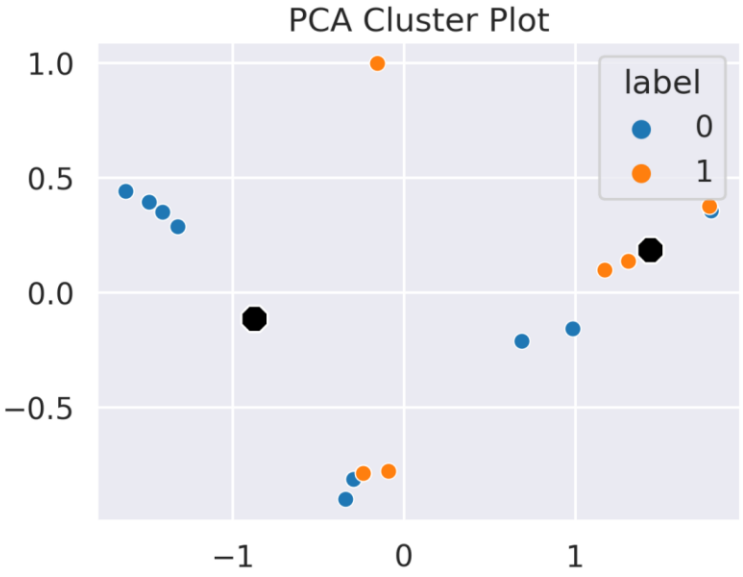

Figure 4: Example of PCA reduced cluster plot. Blue dots represent "wrong" answers in the data while orange represent "correct".

will sometimes yield good results. t-Distributed Stochastic Neighbor embedding (t-SNE) also seems to yield good results when clustering in high dimensions if there are distinct groups within the data.

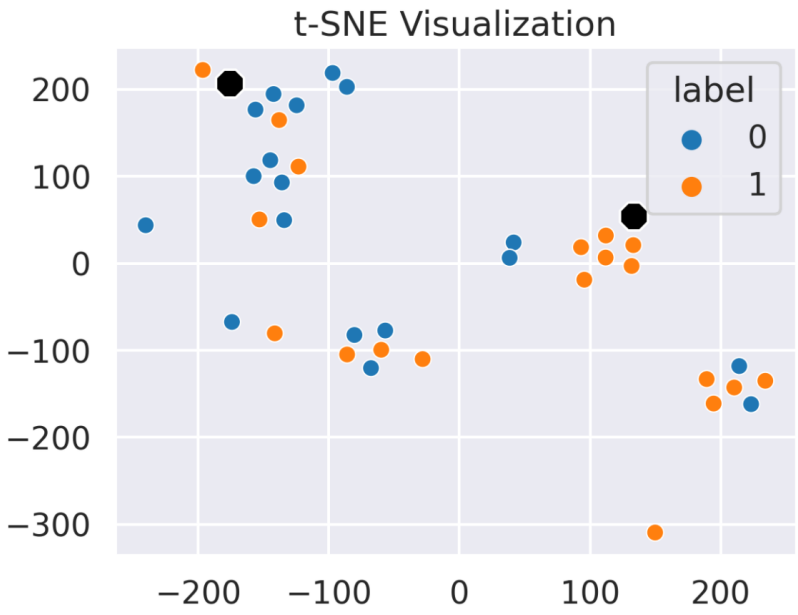

Figure 5: Example of t-SNE plot, on the same data as figure 4. Although some mismatch there is consistency to the groups of data points.

Logistic Regression

Logistic Regression, despite the name, is used for classification rather than regression. It is a supervised method, which means that initial training needs to be done before the model can begin to predict. This training tunes the parameters of the model to allow it to predict more accurately on the rest of the data. Logistic regression models probabilities by using the logistic function:

$$f(x) = \frac{L}{1 + e^{-k(x-x_0)}} \tag{5}$$

Equation 5: The Logistic function.

Where L is the curves maximum value, k is the growth rate, x_0 is the midpoint of the curve, and x are the values of the domain [19].

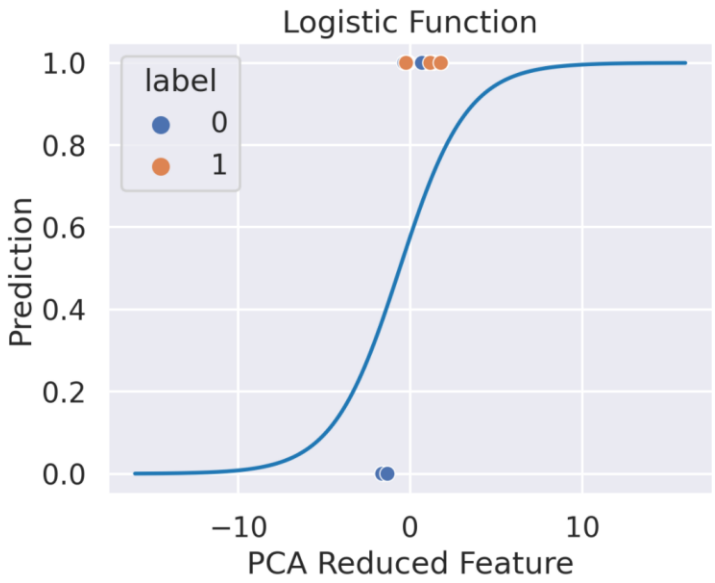

Figure 6: Sigmoid curve of logistic regression model displaying the correct and incorrect predictions for one of the question datasets.

In figure 6 above, the probabilities of the points are not actually forced to one or zero, but can be labeled through using a threshold (e.g. if p > 0.5, label data as 1). Figure 6 is an actual model created using one of the question datasets created by using PCA to reduce the data from the entire feature space down to just one feature. Even with this single feature, that classification does a relatively good job, despite the small data size. Logistic regression has been improved over the years, through using different solvers and regularization. The model in this project does not attempt to tune or use any of these additional improvements, but just the vanilla logistic regression model.

Application

Earlier it was mentioned that automated short answer response grading has been tackled in a variety of ways and methods. Many of these methods are in research form, such that they would need to be recreated in order to be used again, or cost money in order to access. This issue of access is the primary aim of this project. In order to address this, Streamlit [23], a relatively new python package was

utilized to create an accessible web app that would allow teachers to upload their question datasets in csv form, have them graded, and returned to them, with no upfront charge or cost. The models used, although basic in their complexity, are used in such a way that they can be easily replaced with better, more complex, or better tuned models. The hope is that overtime, this application can utilize better models, opening up automated grading to teachers everywhere. More about this application and Streamlit can be found in the references below.

The application contains six pages as well as a navigation bar. It is organized to allow the user to explore the findings of the project, specifically the data as well as the models. The final portion of the applications allows the user to upload their own dataset and download an automatically graded one. The data exploration portion enables to explore any of the datasets used in this project, displaying additional information like descriptive statistics, heatmaps of the dataset, and datatypes of the features created.

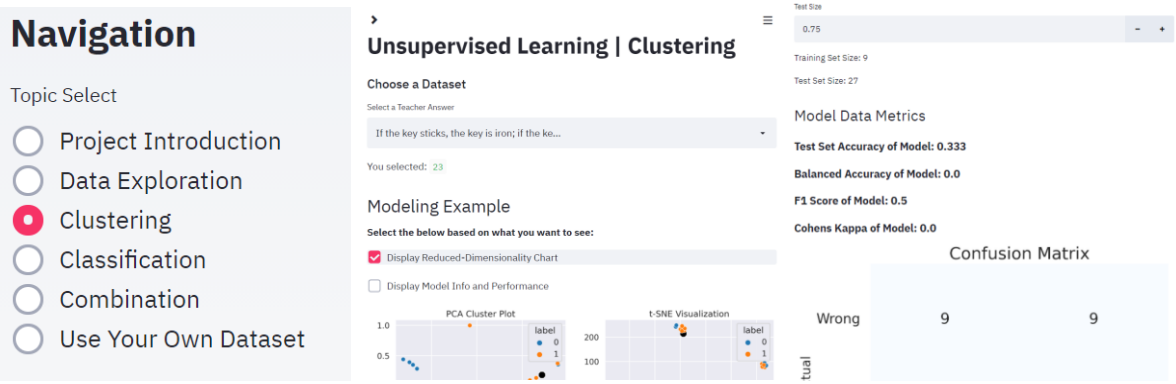

Figure 7: Various screenshots of the application displaying the navigation bar, part of the clustering page, and the part of the classification page.

The clustering page included reduced dimensionality visualizations, the same scoring metrics discussed above, a confusion matrix, as well as a section that allows the user to answer the question from the dataset. This allows the user to try and beat the model. Finally, the last section of this page includes another data exploration section, allowing the user to see what was correctly and incorrectly clustered, giving some intuition as to the nature and tendencies of the model. The classification and combination

pages are extremely similar. The main difference for the classification is a logistic function visualization instead of clustering-based ones, as well as a test size adjuster, which allows the model to be trained on different data sizes. The combination page's only main difference is that the visualizations only display the three largest and smallest observations, which shows the user where these are in reduced dimensionality space relative to the created cluster and teachers answer. Finally, the last page allows users to have their own questions graded. After the grading process, the files can then be downloaded as well.

IV. Experimental study

As mentioned previously, in order to generate features, each student answer was compared to a singular teacher answer. Other than sentence statistics methods, similarity methods and n-gram methods were always based on and compared to the teacher's answer. Any prepossessing done to the student's answer was also always done to the teacher's answer. The result gives a dataset which looks like the one below.

Wordcount	stem_g_ similarity	stem_j_ similarity	stem_c_ similarity	stem_ordered_ g_similarity	q_stemmed_ has_water	q_stemmed_ has_evapor	q_stemmed_ has_water_ evapor
0.375	0.5	0	0	0.5	0	0	0
0.5	0.833333	0.5	0.6708	0.833333	1	1	1
0.5	0.833333	0.5	0.6708	0.833333	1	1	1
0.625	0.666667	0.111111	0.2041	0.666667	1	0	0
0.375	0.571429	0.142857	0.25	0.571429	1	0	0

Figure 8: Sample of features after they have been generated for a random question set.

Because clustering is sensitive to scale, all features always need to be normalized. Every feature can only range between 0 and 1. The feature set for each question set varies on the size of the teachers answer because of the n-gram features. For example, if a teacher's answer has a length of five words, it will

create 11 unique features (5 unigram, 4 bigram, 2 trigram). Once this process is complete, modeling can begin.

Clustering posed a unique issue since it is an unsupervised method. Essentially, how can one say one cluster contains the right answer in the actual application of this model, one where there are no labels? To get around this, the teacher answer is included in the student answer column and the centroid that the teacher's answer belongs to becomes the "correct" centroid. The results of this method are included below.

The classification method came together much more easily. The only issue is that in the actual application of this model, the teacher does need to do some initial grading. Not only this, but the potential uploaded data is a bit more complicated as well. Using this process, it was found that about six student answers and one teacher answer could be used in the training set to achieve average results. These results are included below.

The last method employed was a combination method utilizing both clustering and classification. Classification on its own did do better than clustering overall, so this combination aims to combine the benefits of an unsupervised model with the benefits of a classification one. The main assumption of this combination model is that the observations closest to the teacher's answer in feature space were probably correct, and the ones furthest were probably incorrect. This assumption is not always true because of a combination of simple features as well as poor questions. After using clustering to only identify the closest (correct) and furthest (incorrect) observations, they are then labeled along with the teacher's answer and used as the training set for the logistic regression model. The rest of the data is then classified. The results of this method are provided below.

Results

The metrics utilized are classic accuracy, balanced accuracy, and F1-score. In order to further breakdown and interpret f1-score, precision and recall scores are also reported. Accuracy is used as a standard metric to see how well the models predict the correct classes. The standard accuracy used is defined as the percent of total right predictions. Adjusted balanced accuracy is also utilized because it is sensitive to datasets with imbalanced classes. Adjusted balanced accuracy is defined as

$$acc = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right) \tag{6}$$

Which allows for the minority class to be better represented as an accuracy score. The adjusted aspect means that it is rescaled between -1 and 1, such that a score of zero represents random chance. F1-score is the harmonic mean between recall and precision, defined as

$$F1 = 2\left(\frac{precision * recall}{precision + recall}\right) \tag{7}$$

Where precision and recall are defined as

$$precision = \frac{TP}{TP + FP} \tag{8}$$

$$recall = \frac{TP}{TP + FN} \tag{9}$$

SciKit-Learn describes precision as “intuitively the ability of the classifier not to label as positive a sample that is negative” and recall as “intuitively the ability of the classifier to find all the positive samples.” F1-score then balances the ability of a classifier to find the positive class as well as penalizes when it mislabels something as positive when it is negative.

In terms of benchmarking, in the paper by [15] and accuracy of 76% is reported, which is considered by the authors to be state-of-the-art, at least in terms of LSTM based models. Since the dataset is the same one used by the authors, a comparison is appropriate, mostly just to get a sense of whether the outcome is feasible to continue pursuing. The authors however use purely supervised methods, whereas unsupervised methods are also considered here.

The following figures display the performance of across all 84 question sets, with horizontal lines representing the mean of the performance. The actual mean scores are also provided. The box and whisker plots display the quartile information, with the center line being the median. Median is generally good to utilize when the data is not normally distributed, which for some scores is true. The edges of the box represent the second and third quartile, with the edge of the whiskers representing the minimum and maximum. Any data outside this range is considered an outlier.

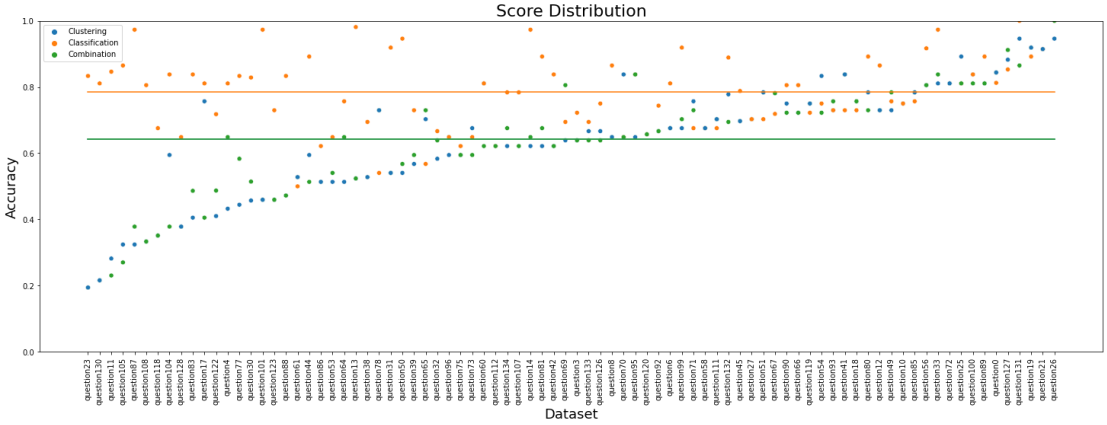

Figure 9: Distribution of accuracy scores

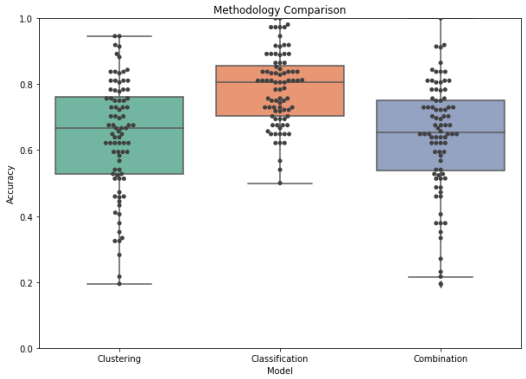

Figure 10: Box and whisker plot with swarm plot overlayed which approximately shows the distribution of the datasets of the accuracy score.

METHOD	MEAN ACCURACY SCORE	MEDIAN ACCURACY SCORE
CLUSTERING	0.642	0.667
CLASSIFICATION	0.785	0.806
COMBINATION	0.641	0.653

Table 6: Central tendency of accuracy scores.

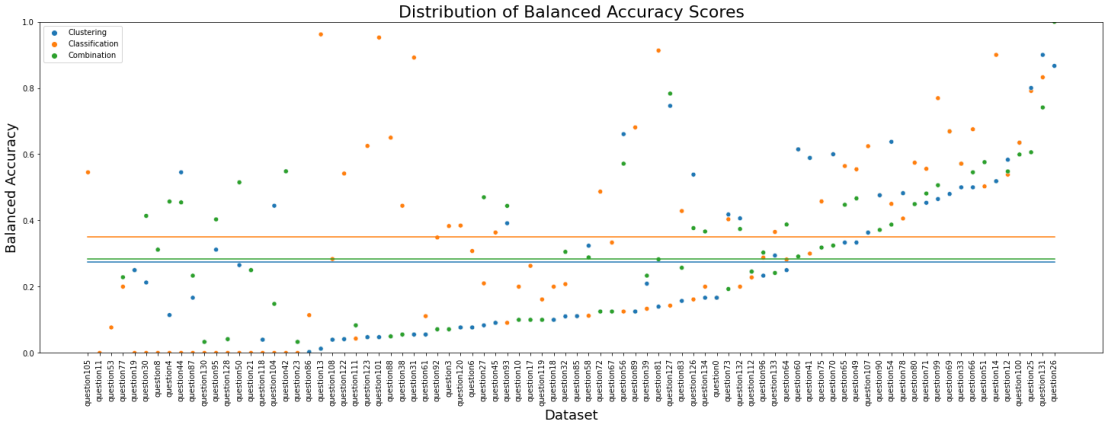

Figure 12: Distribution of balanced accuracy scores.

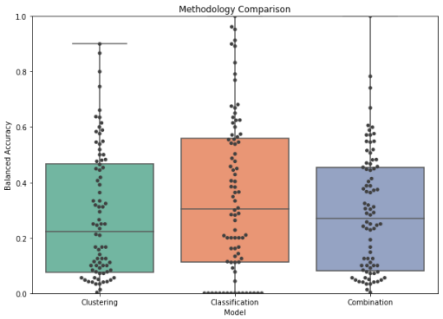

Figure 11: Box and whisker plot with swarm plot overlayed which approximately shows the distribution of the datasets of the balanced accuracy score.

METHOD	MEAN BALANCED ACCURACY SCORE	MEDIAN BALANCED ACCURACY SCORE
CLUSTERING	0.274	0.223
CLASSIFICATION	0.350	0.304
COMBINATION	0.284	0.270

Table 7: Central tendency of balanced accuracy scores.

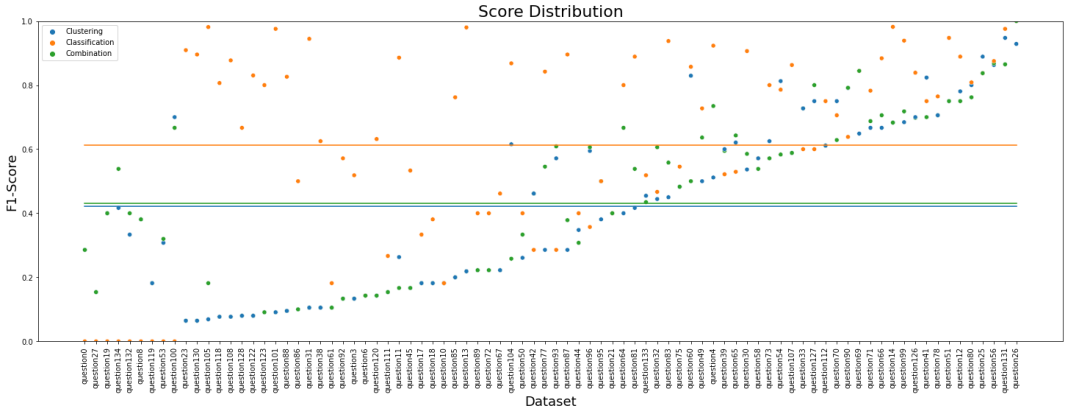

Figure 13: Distribution of F1-scores.

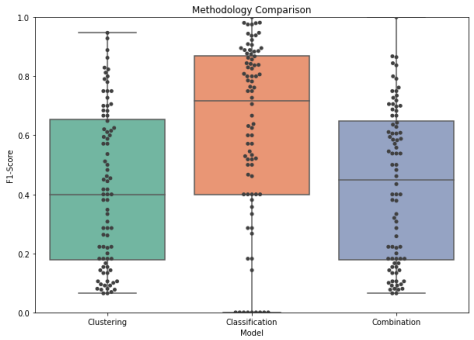

Figure 14: Box and whisker plot with swarm plot overlayed which approximately shows the distribution of the datasets of the F1- score.

METHOD	MEAN F1-SCORE	MEDIAN F1-SCORE
CLUSTERING	0.421	0.4
CLASSIFICATION	0.614	0.717
COMBINATION	0.430	0.448

Table 8: Central tendency of F1-scores.

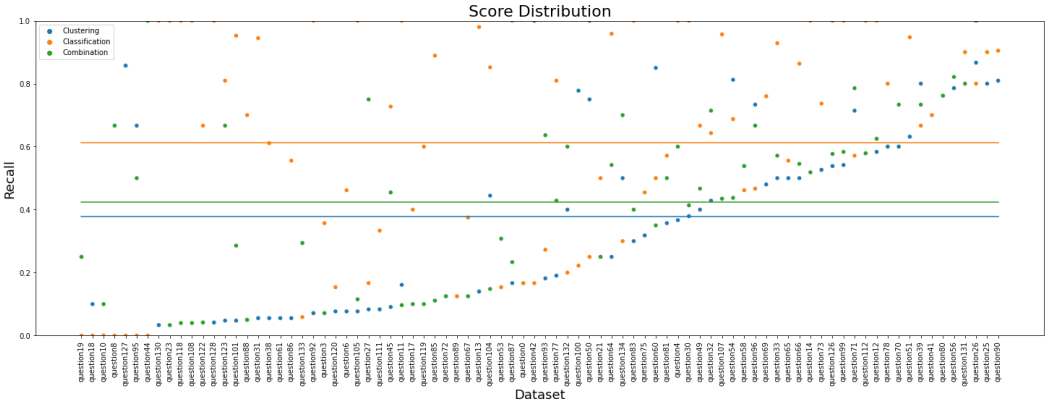

Figure 15: Distribution of recall scores.

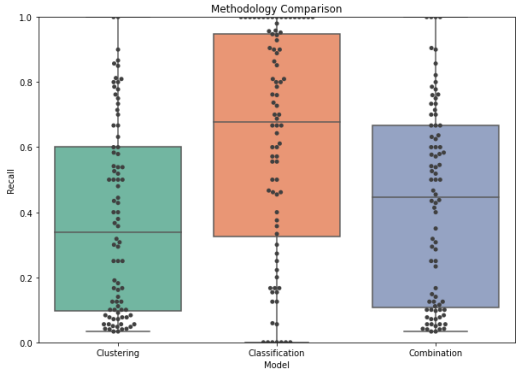

Figure 16: Box and whisker plot with swarm plot overlayed which approximately shows the distribution of the datasets of the recall score.

METHOD	MEAN RECALL SCORE	MEAN RECALL SCORE
CLUSTERING	0.377	0.337
CLASSIFICATION	0.613	0.677
COMBINATION	0.424	0.446

Table 9: Central tendency of recall scores.

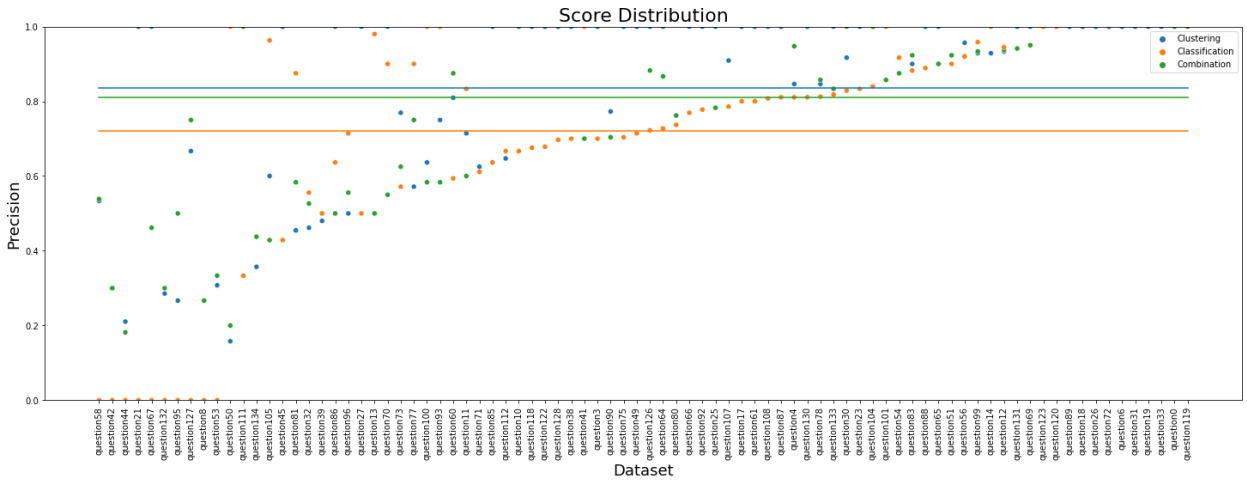

Figure 18: Distribution of precision scores.

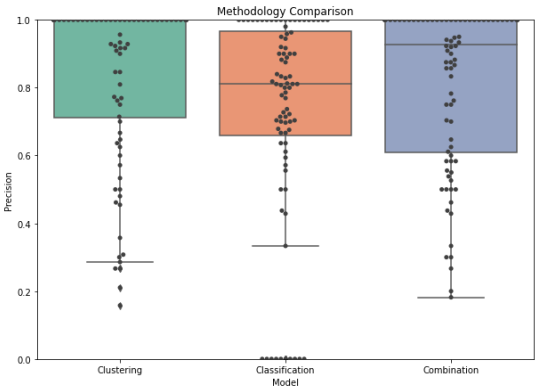

Figure 17: Box and whisker plot with swarm plot overlayed which approximately shows the distribution of the datasets of the precision score.

METHOD	MEAN PRECISION SCORE	MEDIAN PRECISION SCORE
CLUSTERING	0.835	1
CLASSIFICATION	0.722	0.811
COMBINATION	0.811	0.928

Table 10: Central tendency of precision scores.

In terms of comparing methods, classification constantly performs better or just as good as the other methods. This of course makes sense. In this method, the teacher or user of the application has to do some upfront work in labeling the grades. For accuracy, classification scored an average of 78% across all 84 datasets, having a median value of 80%, meaning although there are some low scoring datasets, most of the data is skewed above the mean.

Balanced accuracy however presents another story. Because balanced accuracy takes into account the minor class, the question sets where there was a huge imbalance, classification performed the poorest, most likely because it turned into a “majority class” classifier. A majority class classifier is

one in which it also predicts whichever class appears the most. Balanced accuracy will score those classifiers very low, which from figure 11, it can be seen that the classification method has the most "scored 0%" datasets. A 0% balanced accuracy score means that its not better than ranodmly guessing. Even so, the average and median balanced accuracy for classification still performed better overall than the other two methods.

F1-score tells much of the same information. The classification model does very poorly for some of the imbalanced datasets, but overall does much better than clustering and the combination methods. The place where classification does the worst is with precision, which makes sense. Precision measures how good the classifier is at not predicting false positives. Since it sometimes does create a majority class model, it expectedly performs poorly.

Although classification did the best overall, it is not without concession. Looking at the score distributions for any of the metrics shows a highly sporatic spread. The question set itself however did represent a real world scenario very well. Because of the amount of work that teachers do, if this application were built out to scale, the question sets would probably look like this.

V. Conclusion

Grading is a task that should already be automated. During this project, it was seen that with little modeling effort, decent models, capable of grading well can be created. However, there is tremendous room for improvement. The models utilized in this project are some of the simplest in the category they belong to. Even so, accuracies that were good were achievable. There are however areas of concern. First, how good the question is as well as the teacher's

answer matters greatly. When a question simply does not do a good job of discriminating good answers from bad answers, many of the algorithms, especially the classification ones, will just guess the majority class. In fact, classification algorithms will be completely useless on question sets where the students either get them all correct or incorrect. Clustering algorithms will always break the sets into two categories even if all students answer the same way. Even so, for most questions types that do not fall into the extremes, these algorithms.

There are other areas where this process could be improved. Firstly, because the data size is so small, an abundance of good independent features is important. One area is possibly the use of a thesaurus for synonyms. The n-gram features ended up being predictive, so being able to identify words that are similar would improve the flagging aspect of the n-gram features. The use of a large teacher answer document could also prove to be beneficial. This would increase the number of n-gram features, but in combination with matching synonyms of words, could prove very affective. The use of word embeddings could also be effective. Word embeddings allow for there to be some level of association between words, which could have a similar effect as including synonyms. On the modeling side, there is also significant area for improvement. First, more sophisticated models could be used, including deep learning methods. For unsupervised learning, potentially using the DB-Scan method would allow the algorithm to find clusters based on the data than on a random initialization. For the classification side, basically anything else could be used and would probably improve the overall result. XG-Boost, a gradient boosted trees method, has recently had much success in various machine learning competitions.

VII. References

[1] Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python: analyzing text with the na…tural language toolkit. " O'Reilly Media, Inc."

[2] Basu S., Jacobs, C., Vanderwende, L., “Powergrading: a Clustering Approach to Amplify Human Effort for Short Answer Grading”. Microsoft Research.

[3] Brooks, M., Basu S., Jacobs, C., Vanderwende, L., “Divide and Correct: Using Clusters to Grade Short Answers at Scale”. Microsoft Research.

[4] Callear, D., Jerrams-Smith, J., Soh, V. (2001). CAA of short non-MCQ answers. In M. Danson & C. Eabry (Eds.), Proceedings of the 5th computer assisted assessment conference (pp. 1–14). Loughborough: Loughborough University.

[5] G. Hinton and S. Roweis. “Stochastic Neighbor Embedding”. New York University.

[6] Gay, L.R. (1980). The comparative effects of multiple-choice versus short-answer tests on retention. Journal of Educational Measurement, 17(1), 45–50.

[7] Gonzalez-Barbone, V., & Llamas-Nistal, M. (2008). eAssessment of open questions: an educator’s perspective. In C. Traver, M. Ohland, J. Prey, T. Mitchell (Eds.), Proceedings of the 38th annual frontiers in education conference (pp. F2B–1–F2B–6). Saratoga Springs: IEEE.

[8] Hochreiter, Sepp & Schmidhuber, Jürgen. (1997). Long Short-term Memory. Neural computation. 9. 1735-80. 10.1162/neco.1997.9.8.1735.

[9] Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.

[10] Hou, W.J., Tsao, J.H., Li, S.Y., Chen, L. (2010). Automatic assessment of students’ free-text answers with support vector machines. In M. Ali, C. Fyfe, N. Garc ́ıa-Pedrajas, F. Herrera (Eds.), Proceedings of the 23rd international conference on industrial engineering and other applications of applied intelligent systems (Vol. 1, pp. 235–243). Cordoba: Springer.

[11] John A. Hartigan , Clustering Algorithms, John Wiley & Sons New York , London , Sydney , Toronto,1975 .

[12] Klein, R., Kyrilov, A., Tokman, M. (2011). Automated assessment of short free-text responses in computer science using latent semantic analysis. In G. R ̈oßling, T. Naps, C. Spannagel (Eds.), Proceedings of the 16th annual joint conference on innovation and technology in computer science education (pp. 158–162). Darmstadt: ACM.

[13] Ng, A., "Machine Learning and AI via Brain simulations". Stanford University.

[14] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

[15] P. Patil, A. Agrawal. “Auto Grader for Short Answer Questions”. Stanford University.

[16] Peng, Joanne & Lee, Kuk & Ingersoll, Gary. (2002). An Introduction to Logistic Regression Analysis and Reporting. Journal of Educational Research - J EDUC RES. 96. 3-14. 10.1080/00220670209598786.

[17] S. Burrows, I. Gurevych, B. Stein, The Eras and Trends of Automatic Short Answer Grading, International Artificial Intelligence in Education Society (2015), 60-117

[18] Sima, D., Schmuck, B., Sz ̈oll, S., Mikl ́os, A. (2007). Intelligent short text assessment in eMax. In Proceedings of the 8th Africon conference (pp. 1–6). Windhoek: IEEE.

[19] Verhulst, Pierre-François (1838). "Notice sur la loi que la population poursuit dans son accroissement" (PDF). Correspondance Mathématique et Physique. 10: 113–121.

[20] Willett, Peter. (2006). The Porter stemming algorithm: Then and now. Program electronic library and information systems. 40. 10.1108/00330330610681295.

[21] Spacy: https://github.com/explosion/spaCy/blob/master/spacy/lang/en/stop_words.py

[22] Project Github: https://github.com/ian-mcnair/Answer_Clustering

[23] Streamlit: https://www.streamlit.io/