Predicting Organization Score for Student Essays

Kexin Yang
Carnegie Mellon University
Human-Computer Interaction Institute
Dec. 2018

Abstract

In process of automatic writing evaluation (AWE), predicting scores for students essay is an indispensable step. Among other aspects to assess students essays, organization or structure of the essays is not a low-hanging fruit as other more surface level aspects for essays, such as language. This paper explores methodology of predicting the organization score for students essays by empirical experiments, given the raw text of students essay. This paper is a final project for a class on machine learning, and follows the methodology of that discipline. The results may be of interest to researchers and practitioners in text mining field, especially those who are interested in mining text data in context of students essays.

1 Introduction

Writing is an indispensable part in education and is also not easy to assess. In traditional, non-automated essay scoring process, two trained experts are typically involved in scoring essays, and a third one to resolve any disagreement of opinion between them. While this can often yield reliable scores, it is time-consuming and labor-intensive. This paper explores ways to lessen human labor in the grading process by training machine learning models to predict the scores of students writing.

Previous papers by Mayfield et al discussed methods and systems to automate the process of essay scoring and assessing, which are promising in speeding up the essay grading process. Along the same line, this work aims at predicting one of the four aspects of that measures the quality of students essays. The data that is used, described in section 2, was collected by a language technology company, Turnitin, as part of an ongoing research project at Carnegie Mellon University, concerning giving automatic structural feedback to students essays. The project began with decomposing students essay using Rhetorical Structure Theory, in which we are exploring the predictability of structure on students essays. While we have no publications yet using the dataset used in this paper, we are working on one paper.

The work presented here relates to predicting students organization score. Though essay scoring is a well defined supervised learning task, many efforts, however, involve predicting the combination of different aspects of students score, which would require a different technique. In 2017, the work by Wood et al explored ways to give formative essay feedback using predictive Scoring models. In 2015, Mayfield et al also investigated method and system for automated essay scoring using nominal classification.

The rest of the paper is organized as follows. Data preprocessing including data partition, cleansing, and basic feature representation and encoding are covered in Section 2. In Section 3, it talks about the experiment, including the baseline models, feature extraction, error analysis, and some exploration in ensemble learning. In Section 4, it focuses on the parameter tuning and its evaluation. Section 5 and 6 talk about the final evaluation of a final test set, as well as the implication, limitation and reflection of this work.

2 Data Processing

2.1 Features and Class Value

The data I am using is the raw data that contains students essays, scores and prompt information from Turnitin, a language technology company in Pittsburgh. Raw features given to me are in three types. The first type of features related to essays, including text that students write, which is the most valuable feature that we can get insight from, and some other identification features of essays, such as essay ID. The second type of features relates to prompt and describe the nature of the prompt, type of the prompt and source of the prompts. In these features one that plays an important role is the id of the prompt, this is important because later when building the model, it is needed to do cross-validation by annotation, which in our case, we will categorize by the prompt version, the reason for which will be explored later. The third type of features relates to students, concerning student ID, student Class, etc. which is not in our concern since the data is anonymized and we want to make a generalizable prediction. The fourth type of features concerns the scoring of different aspects of students essay, which are also features we pay close attention to, and also contain our class value.

There are four different column features regarding students essay scoring, namely analysis, language, organization and claim. These four scorings aim at judging four different aspects of students writing. The analysis score judges whether the essays analyze the topic in a comprehensive and well-rounded way. The language score targets at gauging the proficiency, coherence and cohesiveness of language use in the essays. The claim score measures how well-supported is the claim in the essay, and the organization score assesses whether the essay has a balanced, clear structure that helps to communicate ideas.

The class value I choose is the organization score, which assesses the general clarity of the structure in the students essays. This is annotated, as mentioned in the introduction, by trained experts who are experienced in essay grading. The reason I choose this as my class value is, organization score is often harder to assess than other scores as language and analysis, which can be more easily predicted by the sophistication of students word choice, or the length of the article. Organization score, on the other hand, requires higher-level features that may not be easily spotted by human eyes, which evoke my interest in exploring whether the machine can tackle this challenge.

2.2 Dataset preparation and partition

There are 1130 instances in the whole dataset. The data is partitioned into three sets, namely the development set, cross-validation set and the final test set. These two set takes up roughly 20%, 70% and 10% of the whole dataset. The development data is used to do qualitative analysis, such as choosing what attribute to take into consideration. Examining the development data for error analysis purpose is also important, and the only set that we should look closely at. The cross-validation data set is used to train models, and takes up the majority of data. The final test set is not looked at during the whole feature extracting and model building process and is used to assess the final performance of the model, to ensure the validity and generalizability of the learning process.

2.3 Data cleansing

From a test run and doing error analysis, the problem of having instances being categorized in a null class value is alarming. This leads me to do data cleansing in which I deleted unlabeled data, and preserved only labeled data. I also randomized the data by inserting a random number and sort by that column, to avoid the circumstance of text with the same prompts gathering together which may harm the data partition and error analysis process.

2.4 Feature representation and encoding

I have tried three different representation of feature space, the bag of words, one-hot multi-hot, and TF-IDF, using Sklearn in Python. These three representations each has its own features, while can all be categorized as the bag of words representation, simply with different feature encoding. In one-hot multi-hot representation, each feature extracted are presented in a binary way, taking into account only whether the essay instance has or hasnt that particular feature. This is the simplest, plain vanilla representation of text features, which can be problematic in that it can yield too sparse features given large corpus, and it cannot scale well. The bag of words has non-binary feature encoding, and uses the counting of occurrence as a numeric value for the feature value. It can take into consideration the frequency of certain features and add in more complexity than the plain-vanilla representation of one-hot.

One more sophisticated representation, frequently used in information retrieval and text mining is TF-IDF. In information retrieval, tfidf or TFIDF, short for term frequencyinverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. Tfidf is one of the most popular term-weighting schemes today; 83% of text-based recommender systems in digital libraries use tfidf (Rajaraman). The advantage of this representation is that it not only takes into account the frequency of certain features, but also see the relative frequency of this feature in all documents.

I tested the performance of these three three feature encoding using four different algorithm in Sklearn, LinearSVC, Logisticregression, MultinomialNB and RandomForestClassifier, and TF-IDF representation has a slightly better performance, yet it is not significant improvement in statistics, after doing a two-tailed type 1 t-test and assessing the p-value. Also, since TF-IDF representation is currently not supported by LightSIDE, a text-mining and machine learning software that will be used predominantly for error analysis, this work uses the representation of bag of words with either binary or numeric feature values, in order to allow for more visibility of raw document and features during error analysis process.

3 Baseline Experiment

The baseline experiment uses the plain vanilla version of feature representation and algorithm, as well as the default setting for all the parameters. The baseline model uses the One-hot representation of text features, extracts only unigram features from the text field, and applies Naive Bayes as the algorithm. The baseline performance evaluated using cross-validation yields a performance of 0.57 as accuracy, and 0.37 as Kappa. If the model is built on the cross-validation set and tested on the held-out dataset (in this case the development dataset), then it yields a performance of 0.65 as accuracy, and 0.47 as Kappa.

What is worth noticing is, the subpopulation in my data leads me to use a different cross-validation method other than the regular 5 or 10 folds. Because in the dataset there is one column that can be used to distinguish sets of documents from each other (prompt version), I used cross-validation by annotation with the column of prompt version selected. This ensures that every time of doing cross-validation, the documents belonging to all but one prompt version will be used for training, and documents matching the remaining label of the prompt version are for testing. The reason for this is to prevent the model from overfitting to a particular prompt version and ensure generalizability of the model.

When evaluated using cross-validation by annotation, the baseline model using the plain vanilla One-hot unigram feature representation yield a rather low performance of 0.3 as accuracy and 0.14 as Kappa. Since the class value that we are predicting is organization score, a low kappa isnt particularly unexpected for a unigram model as one would expect that interword relationships would be useful for determining the quality of organization of the writing.

4 Feature Extraction

In this section, I iteratively extracted different features and tested them on simple models to see the performance change. The simple model I choose is the logistic regression, as known as the workhorse of natural language processing. One I choose logistic regression over Naive Bayes when evaluating the features to extract is that logistic regression is a linear model, which enable me to see the feature weight in the Explore Result panel, and avoid more interpretability during the error analysis process.

4.1 Basic Features

When evaluating what basic features should be chosen, different combination of them are evaluated using a simple model logistic regression, and the model that yields the best performance is deemed as having the optimal basic feature setting. In this case, features containing unigram, bigram, POS bigram and line length are considered the optimal setting. It comes as a surprise that the feature count occurrence, which takes into account how many time each text feature appear should have no improvement on the feature space, while there seems to be no proxies for that feature. My conjecture is that the dataset is still not big enough for the frequency of each text feature to form a meaningful pattern and make a difference on the performance.

The specific error analysis process is discussed later in later section.

4.2 Stretchy Pattern

Since basic features may not give enough context, stretchy pattern plugin was selected to allow potentially rich features that might capture structure or style despite simple variations in surface presentation. The stretchy pattern allows the model to extract a range of possible patterns, instead of specifying each one individually. I chose the default setting with a pattern length of 2-4 and gap length of 1-2 in configuring Stretchy Patterns. It yields a better performance than merely applying the basic features, though it is an insignificant improvement. This feature combined with the best setting selected in 3.2.1 yields an accuracy of 0.54 and Kappa of 0.24.

4.3 Column Features

In this context, there is additional information, namely meta-data in my file. Besides text, column features of scores in other aspects, such as language, analysis and claim are potentially useful indicators of the organization score. Since these column features are usually highly correlated with the class value that the model is trying to predict, exploration has been made of including these strong column features in the feature space.

Unsurprisingly, including the column features would result in a significant better model (p value = 0.039), with so far the best performance of 0.55 as accuracy and 0.26 as Kappa.

But since in reality, each of these column features should ideally be independent of each other, since they are supposed to be assessing students essays from different angles, even though in reality they may have a strong correlation. The question has come in terms of whether or not to include these highly indicative column features? The decision at last is not to include them, because the objective for these project is to explore ways to gain insight on text itself to predict the organization score. Including column features may result in the model relying overly on them, which may not yield practicality in reality, since in the context of students essays, usually essays either have scores from all aspects, or have none of them. The circumstances of using other aspects of scores to predict one particular aspect of score are highly unlikely to happen in reality, therefore, this kind of high-performing model, will not be very useful when applied to real-life context, such as used in Automated Writing Evaluation (AWE) tools.

4.4 Restructure Data

Following feature extraction is the process of restructuring data, multilevel modeling is used in this step. The multilevel modeling plugin works by creating copies of features based upon the domains each document occurs within. In this context, the prompt version can serve as the different domains for the instances. Since different prompts require students to talk about different topics, features extracted may have different significance in one or more domains, thus making a globally-defined feature confuse a model with noise that is actually meaningful variation by domain. After restructuring data using multilevel modeling, the model yields performance of 0.50 as accuracy and 0.22 as Kappa, which is a step backward from features that are not restructured, which may indicate the multilevel modeling is not very useful.

Feature Extracted	Accuracy	Kappa
1gram	0.44	0.12
1gram + 2gram	0.50	0.16
123gram	0.50	0.14
12 gram+ POS bigram	0.51	0.16
12gram,+POS bigram+ linelength	**0.53**	**0.23**
12 gram+POS bigram+wordpair	0.51	0.17

Table 1: Fig.1 Performance for different features

5 Error Analysis

In this section, error analysis process and changes it made to the model, as well as the evaluation of the changes are discussed.

5.1 Problematic Feature 1

When examining the error cell containing instances predicted to be 3 but are actually 2 in organization score, one problematic feature on is found. ”On” has a high horizontal absolute difference, quite high feature weight, and high frequency of 7 out of 8 instances in total. After looking at the specific document, it is found that text besides ”on” can sometimes tell something about the general quality of the essay. For example, one document contains an ungrammatical sentence-what I am feeling on the subject while the correct way should be what I am feeling about this subject. In contrast, another instance uses a quite advanced word for elementary school students, impede on. Since former data analysis showed argument organization score is often positively correlated with the score of other aspects of students writing, such as language. Therefore, ungrammatical features in the writing may be an indicator for a lower organization score. Similarly, the second students used impede on, an advanced word for authors their age, could potentially be an indicator of higher score in language, which is positively correlated with organization score. From exploring the instances, my insight is giving more context about the text can generally make the model more accurate.

5.2 Problematic Feature 2

From looking at this error cell containing instances predicted to be 3 but are actually 4 in organization score. One problematic feature patriot was found, with a high horizontal absolute difference, quite high feature weight (-0.3821), and high frequency of 8 out of 8 instances in total. This word exists in every instance, since the topic is to have students evaluate the Patriot Act. My hypothesis about this was the word patriot, a mis-spelling from Patriot Act, indicates the writers are not meticulous writers, which should be negatively correlated with their organization score. But a later exploration led me to believe that LightSide is ignoring capitalization, which trivializes the former finding. Yet from examining the essays containing patriot, it is found that some writers did not separate the passage into paragraphs, which would be a bad sign in terms of organization. There are also multiple instances exhibiting the wrong usage of punctuation. This leads to potential features of part of speech tag and punctuation may help the models.

5.3 Potential Improvement and evaluation

From problematic feature 1 it is concluded that giving more context may be helpful, for instance adding bigram and creating richer feature space. From problematic feature 2, it suggests including word/POS pairs and including punctuations may help us gain more insights about the general coherence of essays.

From unigram to bigram, the model yields highly significantly improvement, with p value = 0.001. But including word/POS pairs did not make the model improve significantly.

6 Algorithm choosing

From comparing several models on the best representation of features acquired in previous stages, including Naive Bayes, Logistic Regression, Support Vector Machines and Decision Trees, it turned out that the algorithm that yields the best is logistic regression.

6.1 Naive Bayes

The non-linear, probabilistic model of Naive Bayes yields the worst performance. This is perfectly within expectation. While being efficient, naive bayes strong assumption of conditional independence is likely violated in terms of text data, which is highly dependent on context.

6.2 Logistic Regression

The workhorse of natural language processing research, work to our advantage here. Logistic regression, also known as a maximum entropy or log-linear model, has many of the same design benefits as Nave Bayes it scales well to multiple classes, its extraordinarily efficient, and will often give you the best performance for text data. Yet being a linear model, it can yield more interpretability than non-linear models. Explainability is the main advantage of linear classifiers. And we should not underestimate the importance of explainability in applied machine learning. In our model, it yields the best performance of 0.53 accuracy adn 0.22 Kappa.

6.3 Support Vector Machine

Support vector machines focus only on the marginal instances, places where decisions for a classifier are going to be hard, and mostly ignores the simple cases. But, being optimized for yes/no choices, SVM is not likely the proper algorithm here, where we have four class value to choose from.

6.4 Decision Tree

Decision tree J48 is tried. In our particular case, the decision tree has its advantage, because all of the above algorithms, to a greater or lesser extent, treat each feature as being independent. They dont vary a features importance based on its context. Decision trees try to account for that information when assigning labels, which is useful in the context of this work, since the text features extracted from students essays are likely not independent from each other. However, the decision tree is fairly slow and ineffective when working with sparse, high-dimensional feature tables, as with the feature we obtained from students' essays. Decision tree being also unstable and fairly unpredictable, it falls short of predictability compared with other classifiers such as logistic regression. These disadvantageous features make it very slow working with our sparse data, and is not our most optimized models.

6.5 Ensemble Learning

In order to maximize accuracy, ensemble learning is also applied to make the model more robust. Common methods include bagging, boosting and stacking, in this experiment, stacking is explored to combine the decision made by multiple classifiers.

Stacking is a method of using multiple classification models, to make the model more reliable and more sophisticated in classifying the data. However, it also has the disadvantage of overfitting problem. In this context, stacking of two classifiers, Naive Bayes and Locally Weighted Learning are used, in order to combine the advantage of both classifiers. However, due to the large data and sparse features, Weka is taking a long time to build a model. Though in this experiment ensemble learning methods are not fully explored, it is promising to try the ensemble, voting and stacking in the future, in order to preserve all the decision made and get a more robust model that can perform more stable.

Algorithm	Accuracy	Kappa
Logistic Regression	**0.53**	**0.22**
Naive Bayes	0.26	0.04
SVM	0.47	0.10

Table 2: Fig.2 Performance for different algorithm

7 Parameter Tuning

7.1 Stage 1 and 2- Choosing the best setting

For the parameter tuning, since the best performance is determined to be logistic regression, a parameter that we can fine-tune within this model is batch size. Three settings were chosen as batch size, respectively 50,100, 150. In stage 1, when doing the cross-validation on the whole dataset, the best performance appear under setting 2, where batch size is 100, with the accuracy and kappa being 0.53 and 0.22 respectively. Therefore in stage 2, the model was built under the best-performing setting in stage 1.

7.2 Stage 3- Evaluating the tuning process

To get a conservative estimate for the progress made from doing the parameter tuning. In stage 3, the evaluation process is divided as inner and outer loop. We use inner loop to select the best setting for each fold, and use the outer loop to identify the performance under the setting selected in inner loop. Then five performances for each fold recorded, so are the five baseline performance. A two-tailed, type 1 T-test is done to determine whether the tuning process is worth it. It turned out the p-value is 0.73, therefore, the tuning process is not yielding significant improvement, which suggests us to go with the default setting.

8 Final Evaluation on Final Test Set

To generate a final result, train and test were combined into a single training set. The testing set was a holdout, which had been otherwise unused. Using the best performing representation and algorithm identified in Section 3 and 6, the final evaluation was performed on the final test set. The performance is 0.46 of accuracy and 0.19 of Kappa. This is not as high as the performance achieved doing cross-validation by annotation, (0.53 accuracy). The drop of performance is puzzling, consider the presumably similar nature of the data in the training set and test set. One hypothesis is the instability of the algorithm. Though logistic regression is generally considered to be a rather stable algorithm, its linear assumption may not fit all circumstances. So in the future, it may be useful to add non-linearity in the model trained. Another potential solution to retain the robustness of the model is to use ensemble learning, where all the features from different classifiers can be better preserved to make a more comprehensive decision for testing instances.

9 Discussion

In this section we talk about some practical considerations for deployment, limitation of the work and reflection on the project.

9.1 Adversarial learning

When we deploy such models more broadly, we need to be careful to monitor their performance for unexpected patterns, especially as peoples behavior and learning strategies change over time. We may need to collect additional data from our user base and get it scored by experts, supplementing the original training data. It is important to treat scoring models as living algorithms, needing maintenance to stay at their best performance.
In this case, we can make use of instance-based learning, and try to be incremental. If we can augment the old model with new data over time and add in new examples, the model would be able to maintain its performance for a longer period of time.

9.2 Personalization

Since each learner may have their own personal characteristics, if this data is collected over a long period of time, it may make sense to take those individuals into account. If we are just observing the general outlook of students essay quality and not focused on individuals, then we dont need to observe that. But if the data is used to give specific feedback and recommendation for individual about their own writing strategies, features and tendencies in writing, it may make sense to pay attention to the personalization. While the data at hand is not annotated at the level that allows me to see these aspects of personalization and pay little attention to the individuality across different writers, in reality, it might be useful if the model knows different peoples characteristics. If in the future, we were to cater the model to a specific writer, we may need more data collected over time for one writer, and structure the data and models to learn about specific features of one learner. One way to achieve this may include, using incremental learning, and argument model with individual data.

9.3 Limitation

In this experiment, the class value is represented by nominal value, while if it is ordinal it may be more advantageous, in that it can denote direction and distance between different class value. Being limited by the tools used, it is not feasible now to achieve representing class value as ordinal, but if in the future, it would be interesting to try to represent the class value using ordinal representation.

Another aspect not explored in this project is unsupervised and semi-supervised learning. In addition to the scored essay for each prompt, we can also use an unscored collection of essays collected. In Wood et als work, the extra unsupervised data was used to inform rare feature removal and to provide the basis for off-topic essay detection. Future work in semi-supervised and unsupervised modeling can make even more use of the large pool of unscored data.

9.4 Reflection

From doing this project and exploring the prediction problem of students essays, I learned that in terms of using machine learning to achieve robust Automated Writing Evaluation (AWE), there is still a long way to improve the robustness and stability of models. Especially, since many text features are high-dimensional and sparse, it is especially important to carefully design our feature space and adjust our model in order to capture the useful features and avoid the noisy features. Another thing I learned is, since different prompts tend to have an influence on the things students write about, and will influence strongly features such as unigram and bigram, in this case we need to ensure generalizability and prevent our model from fitting too much to one specific prompt, one way to do this is do cross-validation by annotation.

One other thing I learnt is that while some features are very indicative and predictive for the class value, such as the column features of scores in other aspects, if they are infeasible in reality to obtain, or if they are not proper features to learn from, then when we train our model, we should exclude them and focus more on the features that can yield meaningful, interpretable and practical results.

Finally, as the class value distribution is not balanced in this context, some adjustment to data may be needed. In our case, intermediate scores of 2 and 3 are far more common than the extreme scores of 1 or 4. In addition, the resolution process tends to pull scores toward the center. The extreme scores are being underrepresented and training sets being very imbalanced. To combat this in practice, Mayfield et al include essays in our training sets with the extreme scores if at least one rater scored them as such. This inspired me that we should adjust our feature space and data representation, in accordance with our specific need and circumstance we run into while doing the machine learning problems.

Acknowledgments

Special thanks to Dr. Carolyn Rose and the teaching assistants, for their supportive help in achieving this projects, as well as Turnitin who has provided the data for this work.

References

Bridge. ”Data Mining” Mining of Massive Datasets. pp.1-17, 2011.

C.Breitinger, B.Gipp, S. Langer. Research-paper recommender systems: a literature survey. *International Journal on Digital Libraries*, 17(4) pp.305-338, Jul. 2015.

T.Hastie,R.Tibshirani,J. Friedman, The Elements of Statistical Learning. 2009.

E. Mayfield, D.Adamson, and C. Rose. LightSide Researchers Workbench User Manual. Spring. 2014

R.Shay. On predicting the final Character of Passwords. Dec. 2010.

R.Shams. Weka Tutorials. `https://www.youtube.com/watch?v=Nje8mblA7bs&list=PLJbE6j2EG1pZnBhOg3_Rb63WLCprtyJag&index=16`

K. Toutanova, D. Klein and C. Manning, Feature-rich part-of-speech tagging with a cyclic dependency network. *NAACL*, 2003.

I.Witten, E.Frank, M. Hall and C.Pal. *Data-mining practical machine learning tools.*. Morgan Kaufman Book Co., Cambridge, USA, 4rd edition, 2017.

B.Woods, D. Adamson, S. Miel and E. Mayfield, Formative Essay Feedback Using Predictive Scoring Models *KDD*, Mar. 2017.

E. Mayfield,D. Adamson, Method and System for Automated Essay Scoring using Nominal Classification *Patent Application Publication*, Jul. 2015.