
Automatic coding of students' writing via Contrastive Representation Learning in the Wasserstein space

Ruijie Jiang, Julia Gouvea, David Hammer, Shuchin Aeron *
Tufts University
Medford, MA -2155

Abstract

Qualitative analysis of verbal data is of central importance in the learning sciences. It is labor-intensive and time-consuming, however, which limits the amount of data researchers can include in studies. This work is a step towards building a statistical machine learning (ML) method for achieving an automated support for qualitative analyses of students' writing, here specifically in score laboratory reports in introductory biology for sophistication of argumentation and reasoning. We start with a set of lab reports from an undergraduate biology course, scored by a four-level scheme that considers the complexity of argument structure, the scope of evidence, and the care and nuance of conclusions. Using this set of labeled data, we show that a popular natural language modeling processing pipeline, namely vector representation of words, a.k.a word embeddings, followed by Long Short Term Memory (LSTM) model for capturing language generation as a state-space model, is able to quantitatively capture the scoring, with a high Quadratic Weighted Kappa (QWK) prediction score, when trained in via a novel contrastive learning set-up. We show that the ML algorithm approached the inter-rater reliability of human analysis. Ultimately, we conclude, that machine learning (ML) for natural language processing (NLP) holds promise for assisting learning sciences researchers in conducting qualitative studies at much larger scales than is currently possible.

1 Introduction

In recent years, science, engineering and mathematics education has emphasized supporting students' disciplinary "practices" of inquiry [1, 2]. These practices—such as of formulating questions, designing investigations, or arguing from evidence—are more difficult to identify and assess than the traditional objectives of particular correct content knowledge. In order to study students' practices, researchers rely mainly on qualitative analyses of naturalistic data [3–5]. These studies have advanced the field's understanding of practices.

These studies have been limited in scope, however, because they are extremely labor-intensive: Analysis of naturalistic data requires significant and extensive effort by trained researchers, from transcribing to coding to the construction of meaning. It has been time and cost-prohibitive to conduct qualitative studies with large samples of data. Our purpose in this project is to develop computational tools that can support qualitative research at large scales on students' inquiry practices in science. In this paper we report on initial progress towards applying natural language processing (NLP) techniques to research on students' written arguments in college biology laboratory reports.

*Research supported by NSF 1931978. S. Aeron is also supported by NSF CCF:1553075, NSF TRIPODS grant HDR:1934553, and AFOSR FA9550-18-1-0465. emails: ruijie.jiang@tufts.edu, julia.gouvea@tufts.edu, david.hammer@tufts.edu, shuchin@ece.tufts.edu

In this work, we report on our success in designing NLP that approached the reliability of trained, human coders. Specifically, we show that contrastive learning in the Wasserstein space is able to achieve a high level of agreement (as measured by QWK) on average.

The rest of the paper is organized as follows. In section 2 first overview current state-of-art in automating assessment of writing in science using machine learning for natural language processing. Following that we outline the writing assessment setting particular to our case in section 3. In section 4 we briefly survey the relevant literature in machine learning for NLP and introduce our novel approach for automatic scoring. In section 5 we evaluate the performance of the proposed approach and discuss the results in detail.

2 Examining science writing using NLP machine learning approaches

The most common use of ML/NLP in science education has been to assess the conceptual correctness of students' written explanations or arguments [6]. For example, Nehm and colleagues [7] have used ML techniques to assess students' explanations of evolutionary change. They and other researchers have found that ML can match the reliability of human coders in scoring or categorizing written explanations [7–9].

These results are encouraging for the general goal of conducting qualitative research at scale and they provide a useful complement to existing large-scale methods of assessing conceptual understanding. The same algorithms, however, cannot apply to research on students' inquiry practices. Assessing conceptual correctness can rely on using ML to recognize a limited set of target terms or phrases. For example, explanations of evolutionary change should include ideas about variation, heritability and selection [8]. Typically, responses will receive a higher score if they contain more of these target ideas and fewer incorrect or irrelevant ideas. Computational algorithms can learn to code for the presence or absence of these targets fairly easily. For researchers interested in expressions of scientific practice, that may contain novel ideas, to be effective the ML approach cannot rely on specific target terms.

To study these aspects of scientific practice, researchers look for evidence of how students support, relate and evaluate scientific claims [10]. Interpreting the disciplinary value of students' scientific writing requires more attention to context and the ability to evaluate the value of novel or unexpected ideas. This creates a new set of challenges for NLP/ML and raises questions about the ability of ML to deal with the more creative aspects of scientific practice.

Lee and colleagues [11] have made progress in this direction by using NLP/ML to score the quality of students' uncertainty-infused scientific arguments. Lee [11] scored students' explanations environmental phenomena according to the quality of evidence and reasoning. They also scored students' reflections on the certainty of their explanations. Using an NLP engine called c-rater-ML, they found relatively high human-machine agreement ($QWK > 0.70$) on both constructs. Rather than identify pre-specified target words, c-rater-ML extracts features of text including characters, word sequences and word associations to model responses, and then uses support-vector methods to make score predictions. This work suggests that automated techniques can be successful in detecting qualitative markers of sophisticated science writing in text.

Building on this work, our aim was to explore the possibility that NLP techniques could identify markers of quality in university students' laboratory reports. In addition, our intention in using ML was to explore its role in aiding researchers. ML has commonly been positioned as a substitute for an instructor or grader, making it possible to provide feedback or grades to large numbers of students. Our aim has been to explore the utility of ML as a research tool that can allow researchers to detect patterns and consider meaning in qualitative data.

	Level 1	Level 2	Level 3	Level 4
Argument Structure	Tautological/ Simple	Additive, One-sided	Relational, Two-sided	Compound/ Conditional
Scope of Knowledge/ Evidence	Relies on given information and/or vague reference to results	Uses subset of available data as evidence to support inference	Uses more of the data (including contradictory patterns) to support inferences	Uses data (including contradictory patterns) as well as outside information or hypotheticals/ thought experiments
Consistency & Closure	Closed without rationale, ignores inconsistencies	Closed with some rationale, ignores inconsistencies	Closed, addresses and reconciles inconsistencies	Open-ended or appropriately qualified

Table 1: Description of dimensions that comprise each level.

3 Complexity of argumentation in biology lab reports

The present work concerns analyses by (Gouvea et al. in prep) to track shifts in the quality of students' writing due to curricular innovations. (Gouvea et al. in prep) adapted the domain-general Structure of Observed Learning Outcomes (SOLO) taxonomy [12] to study the quality of argumentation in college biology lab reports. The scheme defines quality in terms of three features of written arguments:

1. **Argument structure** refers to the number and inter-relatedness of claims.
2. **Scope of knowledge/evidence** refers to the sources of evidence used the degree to which that evidence is reconciled with other forms of evidence.
3. **Consistency & closure** refers to the extent to which conclusions are theoretically consistent and appropriately closed or left open in the face of uncertainty.

These features inform the choice of a score on a four-level scale. Table 1 describes and provides examples of student writing for each of four levels within each category. For example, in the category of argument structure, Level 1 was assigned to tautological arguments that repeated back given facts as claims. Level 2 was assigned to claims that made one-sided inferences. Level 3 was assigned to arguments that considered at least two possible claims, but ultimately sided with one claim. Finally, Level 4 arguments integrated claims to make compound or conditional arguments. In practice, it was not always possible for coders to clearly assign a level to each of the three categories. Instead, coders evaluated each response according to each category as much as possible and then assigned an overall score of 1-4.

3.1 Data description

This data set is composed of the Discussion sections of 146 lab reports from a Biology course. The reports were scored by human education researchers using a rubric adapted from Biggs and Collis [12]. The properties of the lab reports are shown in Table 2.

Essays	Avg length	Max length	Min length	Scores
146	512	1449	95	1-4

Table 2: Statistics of the dataset

In this study we explore the ability of NLP techniques to differentiate among samples of students' scientific writing by Gouvea et al s scheme. The target is to match human-human inter-rater reliability.

4 Proposed Approach

Our approach departs from existing ML work in examining science writings as surveyed previously and heavily based on using numerical representations or vector embeddings of words and documents,

popularly called as word2vec [13] or doc2vec [14], which capture lexical and semantic properties of words. These methods employ a self-supervised learning mechanism to create universally useful representations that have shown to be useful for tasks such as sentiment analysis, topic segmentation. With these word embeddings/representations, both Recursive Neural Networks (RNN) and Convolution Neural Networks (CNN) can then be used for modeling the input document. It has been commonly understood that CNN can capture the local information [15] and RNN can capture the sequential information, capable of modeling long term dependencies using architecture such as Long-Short-Term-Memory (LSTM) [16], bi-LSTM [17]. These models are further refined to handle longer-range dependencies, e.g. contextual LSTM (CLSTM) [18] and attention-based LSTM [19]. A relevant work that uses vector embeddings for Automatic Essay Scoring (AES) is Kaveh [20], where the authors employed RNN for automated scoring. However, as we show below the approach in Kaveh [20] does not capture all the temporal information from the RNN and therefore suffers from low accuracy, especially in cases with low amounts of training data.

In our context, the prime use of these embeddings as well as the subsequent use of CNN and RNN as encoders, is to measure similarity between documents, and using this similarity employ the contrastive representation learning with triplets loss [21, 22] for learning discriminative representations, across different classes. As a similarity measure we propose to use the Word Mover’s Distance [23]), which is the Wasserstein distance [24, 25] between document representations, and develop the novel approach of contrastive document representation learning in the Wasserstein space. Embedding natural language data into Wasserstein spaces have recently been shown to be useful for NLP tasks [26], and we are motivated by these developments. We next describe the proposed workflow in detail.

4.1 Proposed workflow for automatic scoring

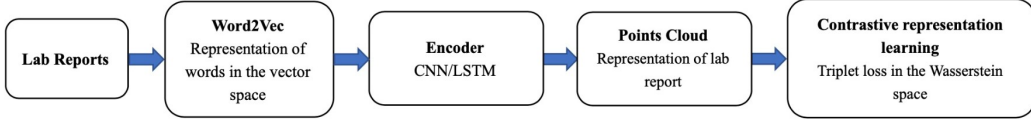


Figure 1: Overview of the main workflow – Contrastive Representation Learning in Wasserstein Space (CoReL-W).

Figure 1. shows the main workflow for Contrastive Representation Learning in Wasserstein Space (CoReL-W) and consists of the following modules.

Word Representation: The lab report is modeled as a sequence of words w_t , and each word, in turn, is represented by a vector $x_t \in \mathbb{R}^{d_w}$. So, a report with N words is represented by a matrix $X = (x_1, x_2, \dots, x_N) \in \mathbb{R}^{d_w \times N}$.

Encoder: This matrix X is then fed into an encoder module, that computes a representation of the entire document. We use two different encoders in our model, viz. Convolution Neural Network (CNN) and Recurrent Neural Network (RNN), and we describe the details below.

- **Convolution Neural Network (CNN):** In order to extract local features from the sequence [15], a shared linear transformation matrix S is applied to all windows. We define $2l + 1$ as the window size. The words in the window can be represented as a vector $x_{tl} = (x_{t-l}^\top, x_{t-l+1}^\top, \dots, x_{t+l}^\top) \in \mathbb{R}^{(2l+1) \cdot d_w}$. For matrix $S \in \mathbb{R}^{(2l+1) \cdot d_w \times d_c}$ and bias $g \in \mathbb{R}^{d_c}$, the output vector $\tilde{x}_t \in \mathbb{R}^{d_c}$:

$$\tilde{x}_t = x_{tl}S + g \tag{1}$$

To guarantee the number of words within the window remains the same, we did zero padding for each document. The output of this encoder is the set $\tilde{\mathcal{X}} = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N\}$, which can be viewed as the representation of the lab report for downstream task.

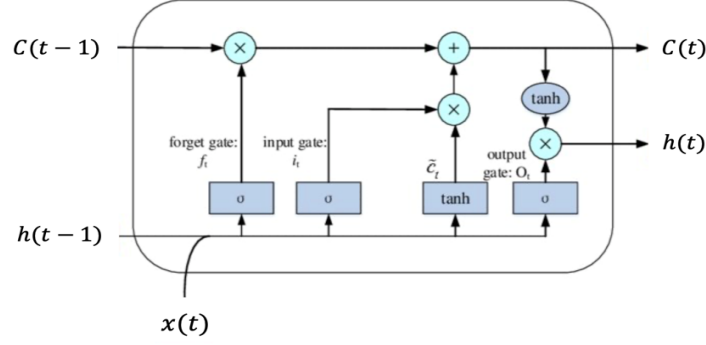


Figure 2: The LSTM architecture [27]

- **Recurrent Neural Network (RNN):** The RNN architecture that we will use here is the standard LSTM [16] architecture. The structure of a LSTM model is shown in Figure 2., and the Equations 2 describe how LSTM works:

$$\begin{aligned}
 \mathbf{i}_t &= \sigma(\mathbf{x}_t \mathbf{U}_i + \mathbf{h}_{t-1} \mathbf{W}_i + \mathbf{b}_i), \\
 \mathbf{f}_t &= \sigma(\mathbf{x}_t \mathbf{U}_f + \mathbf{h}_{t-1} \mathbf{W}_f + \mathbf{b}_f), \\
 \mathbf{o}_t &= \sigma(\mathbf{x}_t \mathbf{U}_o + \mathbf{h}_{t-1} \mathbf{W}_o + \mathbf{b}_o), \\
 \tilde{\mathbf{C}}_t &= \tanh(\mathbf{x}_t \mathbf{U}_g + \mathbf{h}_{t-1} \mathbf{W}_g + \mathbf{b}_c), \\
 \mathbf{C}_t &= \sigma(\mathbf{f}_t \circ \mathbf{C}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{C}}_t), \\
 \mathbf{h}_t &= \tanh(\mathbf{C}_t) \circ \mathbf{o}_t.
 \end{aligned} \tag{2}$$

σ represent the sigmoid activation function $\sigma(r) = \frac{1}{1 + e^{-r}}$, and \tanh represent the tanh activation function $\tanh(r) = \frac{e^{2r} - 1}{e^{2r} + 1}$:

The set of Equations (2) is depicted as a unit in Figure 2. For each unit, the input contains three parts, $\mathbf{x}_t, \mathbf{h}_{t-1}, \mathbf{C}_{t-1}$. \mathbf{x}_t corresponds to the t_{th} word vector, \mathbf{h}_{t-1} and \mathbf{C}_{t-1} outputs from the previous unit. \mathbf{U}, \mathbf{W} are matrices that are parameters to be learned, \mathbf{b} are learned bias vectors. The symbol \circ denotes the Hadamard product. The vector $\mathbf{h}_t \in \mathbb{R}^{d_h}$ corresponds to the hidden state at time step t .

The overall output of the LSTM is the set of hidden states $\mathcal{H} = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N\}$, which can be viewed as the representation of the lab report for downstream task. In our experiments, we also use biLSTM [17], which consists of two LSTMs, one that processes the input in forward direction $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N)$, one that processes the input in backward direction $(\mathbf{x}_N, \mathbf{x}_{N-1}, \dots, \mathbf{x}_1)$ separately.

Learning representation in the Wasserstein space: Given the representation of a lab report as a set of support points $\mathcal{Z} = \{z_1, z_2, \dots, z_N\}$ generated from the encoder. Our objective is to tune the parameters of the encoder to learn useful representation. Note that $\mathcal{Z} = \tilde{\mathcal{X}}$ when using the CNN as the encoder and $\mathcal{Z} = \mathcal{H}$ when using the RNN as the encoder. Please refer to the notation in the discussion above.

To learn good encoder representations, we propose a novel approach, namely that of contrastive learning in the Wasserstein space that we outline below. The *key idea* is to view the output of the encoder as empirical distributions, supported on the finite sets of points \mathcal{Z} . For two different reports with representations $\mathcal{Z}^1 = \{z_1^1, z_2^1, \dots, z_{N_1}^1\}$ and $\mathcal{Z}^2 = \{z_1^2, z_2^2, \dots, z_{N_2}^2\}$, we denote the corresponding empirical distribution μ and ν via,

$$\mu = \sum_{i=1}^{N_1} \mathbf{u}_i \delta_{z_i^1}, \nu = \sum_{i=1}^{N_2} \mathbf{v}_i \delta_{z_i^2}. \tag{3}$$

Here, \mathbf{u} and \mathbf{v} are vectors of non-negative weights summing to 1 separately and δ is the dirac delta function. One can compare two distributions using the Wasserstein distance [25] defined as follows:

$$\mathcal{W}(\mu, \nu) = \inf_{\Gamma \in \mathbb{R}^{N_1 \times N_2}, \Gamma \geq 0} \sum_{i,j} d(\mathbf{z}_i^1, \mathbf{z}_j^2) \Gamma(i, j) \quad (4)$$

$$\text{s.t. } \sum_i \Gamma(i, j) = \mathbf{v}_j, \sum_j \Gamma(i, j) = \mathbf{u}_i,$$

where $d(\mathbf{z}_i^1, \mathbf{z}_j^2) = \frac{1}{2} \|\mathbf{z}_i^1 - \mathbf{z}_j^2\|^2$. To allow for end-to-end training, we use entropic regularization and use the Sinkhorn algorithm to compute the Wasserstein distance [24].

Given this set-up, our approach is to learn useful representations via contrastive learning using the triplet loss [21, 22]. In this approach, one defines triplets (a, a^+, a^-) , where a is the anchor sample (report), a^+ is the positive sample, a^- is the negative sample, and use the Wasserstein distances $\mathcal{W}(\mu_i^a, \mu_i^{a^+})$ and $\mathcal{W}(\mu_i^a, \mu_i^{a^-})$ as similarity measure, where μ^a, μ^{a^+} and μ^{a^-} are representations of the anchor, positive sample and negative sample respectively as empirical distributions. Given P such triplets, we minimize the following contrastive loss function [22]:

$$\mathcal{L}(\mu^a, \mu^{a^+}, \mu^{a^-}) = \frac{1}{P} \sum_{i=1}^P \max(\mathcal{W}(\mu_i^a, \mu_i^{a^+}) - \mathcal{W}(\mu_i^a, \mu_i^{a^-}) + m, 0), \quad (5)$$

where the constant m is a margin hyperparameter that controls the difference of the distance between positive pair and negative pair.

Noting that the distributions μ^a, μ^{a^+} and μ^{a^-} are function of the parameters θ of the encoder, these parameters are learned on a training set via solving the following optimization problem.

$$\theta_0 = \arg \min_{\theta} \mathcal{L}(\mu_{\theta}^a, \mu_{\theta}^{a^+}, \mu_{\theta}^{a^-}) \quad (6)$$

KNN Prediction: Given the learned parameters from a training set, say θ_0 , of the encoder, each report from training set can be represented by the corresponding representation $\mu_{\theta_0}^i, i = 1, 2, \dots, N_{train}$, where N_{train} is the number of reports in the training set. To predict the score s_{test} of a report from the test set, the Wasserstein distance between its representation $\mu_{\theta_0}^{test}$ and the representation of all reports from training set $\{\mu_{\theta_0}^i\}_{i=0}^{N_{train}}$ is calculated. Then K reports from the training set with the lowest Wasserstein distance to the test $\mu_{\theta_0}^{test}$ are selected. Let these scores be (s_1, s_2, \dots, s_K) . Then the declared score is simply the nearest integer rounding of the average score $s_{test} = \text{round}(\frac{1}{K} \sum_{i=1}^K s_i)$.

5 Evaluation of the proposed approach on Biology lab report scoring

5.1 Evaluation Metric

We use Quadratic Weighted Kappa [28] to evaluate our system that is calculated as follows. First, a weight matrix $\mathbf{A} \in \mathbb{R}^{M \times M}$ is calculated in as follows:

$$\mathbf{A}_{p,q} = \frac{(p - q)^2}{(M - 1)^2} \quad (7)$$

where p and q are the true score and the predicted score respectively, and M is the number of possible ratings. A matrix $\mathbf{O} \in \mathbb{R}^{M \times M}$ is calculated such that $\mathbf{O}_{p,q}$ denotes the number of essays whose score is p and prediction q . An expected count matrix $\mathbf{E} \in \mathbb{R}^{M \times M}$ is calculated where $\mathbf{E}_{p,q}$ equals to the product of the number of essays whose true score is p and the predicted score is q . The matrix \mathbf{E} is then normalized such that the sum of elements in \mathbf{E} and the sum of elements in \mathbf{O} are the same. Finally, the QWK score is calculated as:

$$\text{QWK} = 1 - \frac{\sum_{p,q} \mathbf{A}_{p,q} \mathbf{O}_{p,q}}{\sum_{p,q} \mathbf{A}_{p,q} \mathbf{E}_{p,q}} \quad (8)$$

Selection of Triplets for contrastive learning: For each report, we randomly choose one report with the same score as a positive sample and choose another report with different score as a negative sample. This negative sample is selected based on the following probability:

$$\text{Prob}(D_j) = \frac{|s - s_j|}{\sum_{k=1}^M |s - s_k|}. \quad (9)$$

Here, D_j is the report to be chosen, s and s_j are the score of the original report and the report to be chosen. M is the number of possible ratings. In other words, the probability probability is proportional to the difference between the scores of the anchor and the sample to be chosen, which means the sample whose score is more different will have a higher probability to be chosen as the negative sample. In our experiment, we generate 8 triplets for each lab report in the training set. This makes the total number of triplets to be 816.

5.2 Setup

We used 10-fold cross-validation to evaluate our system. In each fold, 70% of the data is used as the training set and the other 30% as the test set. We use Quadratic Weighted Kappa to evaluate our system. A max number of epochs was set to train the model. To clean the text, we removed the punctuation and stop words, lowercase, stemming, and lemmatize the text by using NLTK [29].

Our system has some hyper-parameters that need to be set. The optimizer we used was Adam [30], the learning rate was set to 0.01. The mini-batch size was 408. The number of epochs was 5. We also employed batch normalization [31] during the training process. The length of the word embedding d_w was 300, the length of the hidden state for RNN d_h was 300. We did zero paddings for the CNN, and the window size we set was 3, the output length d_c was 300. We employed the pre-trained GLoVe [32] word vectors as the representation of words. Before being fed into the neural network, each word embedding was multiplied with a certain weight from referred to as the smooth inverse frequency (SIF) [33], this weight has a negative relevance to the frequency of the words. The parameter was initialized randomly. For prediction using the KNN, we set the number of nearest neighbors to $K = 7$.

5.3 Result and Discussion

The various methods we used for scoring are shown in Table 3. As a baseline, we also rebuild a widely used automated scoring system called "c-rater-ML" [34]. "c-rater-ML" is a key words and key sentences detector-based system, which has been proven to successfully score short answers automatically. As shown in Table 4, we can see "c-rater-ML" fails in our task compared to the proposed methods. For reference, note that the QWK for reliability between human coders was estimated at 0.839. According to Cohen [28], values of QWK between 0.41 - 0.6 are considered "moderate" and values between 0.61 - 0.8 are considered "substantial." There the proposed methods achieve a good level of performance.

Method	Description
c-rater-ML	Five features: Word n-grams, Character n-grams, Response length, Syntactic dependencies and Semantic role labels. Then use SVR to train the model.
LSTM	Use LSTM as an encoder, make prediction using the average of the hidden state vectors [20].
biLSTM	Use biLSTM in Kaveh's method [20].
CoReL-W+biLSTM	CoReL-W with biLSTM as the encoder + KNN for prediction.
CoReL-W+CNN	CoReL-W with CNN as the encoder + KNN for prediction.

Table 3: Methods we used in the experiment

System	QWK
c-rater-ML	0
LSTM	0.46
biLSTM	0.61
CoReL-W+biLSTM	0.628
CoReL-W +CNN	0.645

Table 4: The performance of the various models

A	Report ID	ID	Original score	ML prediction	Re-code score
ML prediction consistently differs from original score	3BS15	121	1	3	3*
	C10	13	2	3	2
	3BS16	107	2	3	2
	E5	26	2	3	3*
	B7	61	3	4	4*
	H20	3	4	3	3*
	3AS17	91	4	3	4
	3BS8	122	4	3	4
	3BS17	115	4	3	4
ML prediction consistently matches original score	K15	19	2	2	1
	3As7	89	2	2	1
	3AS6	90	2	2	2*
	G21	50	3	3	3*
	D15	62	4	4	4*

Table 5: Match between original human coder consensus, ML model prediction and human re-coding consensus. *indicates consistency between ML prediction and re-code score.

5.4 Re-coding Process

During the cross-validation process we evaluated the reliability of ML predictions against human scores. ML predictions consistently matched human coders for 5 lab reports; ML predictions consistently predicted a score that differed from human coders for 9 reports (Table 5). Three of the coders who had coded the original dataset participated in a re-coding process. This re-coding took place over a year after the original coding and coders were blind to both the original code and the ML prediction. Each report was re-coded by at least two coders independently and was then discussed by all three coders until a consensus score was agreed upon. Cohen’s (linear weighted) kappa amongst coders for this subset (N=14) was 0.566; while QWK was 0.701. Note that both values are lower than the kappa values for the full dataset, indicating that this subset of 14 reports may have been more challenging to code.

Table 5 compares the original consensus codes by human coders with the ML predictions and the consensus score from the re-coding process. Of the 9 reports for which the ML prediction consistently differed from the original human-coded consensus score, 4 were revised in re-coding in the direction of the ML prediction. The remaining 5 examples were not changed from the original. Of the 5 reports for which the ML prediction consistently matched the original human coded score, three remained matching in re-coding, while two reports were revised away from both the original code and the ML prediction.

It is noteworthy that the two instances in which the re-coding process resulted in revision away from the ML prediction involved reports coded as Level 1. The ML process was least successful in predicting Level 1 reports, likely because the data set was imbalanced (only 12 out of 146 were coded as Level 1 by human coders).

That the ML prediction matched revised scores from the re-coding process suggests the possibility that ML may be useful for identifying examples that human coders may want to revisit, either because they were coded inconsistently or because they represent difficult to code "borderline" examples.

6 Conclusions and future work

In this paper, we proposed a novel approach to get insights into student learning and outcomes based on a novel ML-NLP framework. Our results show that modern ML-NLP workflows may be effectively used as tools to support education research on student learning. The proposed method doesn’t need any feature engineering and is able to work on a small dataset. The achieved QWK of 0.65 is high and can provide a valuable feedback to the instructor.

7 Codes and Data Release

Coming soon...

8 Acknowledgements

We would like to thank Eric Miller, Mark Hempstead, and Kristen Wendell for helpful discussions.

References

- [1] Michael J Ford. Educational implications of choosing “practice” to describe science in the next generation science standards. *Science Education*, 2015.
- [2] National Research Council et al. *A framework for K-12 science education: Practices, crosscutting concepts, and core ideas*. National Academies Press, 2012.
- [3] Eve Manz. Understanding the codevelopment of modeling practice and ecological knowledge. *Science Education*, 96(6):1071–1105, 2012.
- [4] Kathleen Emlen Metz, Amy Cardace, Eric Berson, Uyen Ly, Nicole Wong, Stephanie Sisk-Hilton, S Emlen Metz, and Mark Wilson. Primary grade children’s capacity to understand microevolution: The power of leveraging their fruitful intuitions and engagement in scientific practices. *Journal of the Learning Sciences*, 28(4-5):556–615, 2019.
- [5] R Lehrer and L Schauble. The development of scientific thinking handbook of child psychology and developmental science ed rm lerner, 2015.
- [6] Xiaoming Zhai, Yue Yin, James W Pellegrino, Kevin C Haudek, and Lehong Shi. Applying machine learning in science assessment: a systematic review. *Studies in Science Education*, 56(1):111–151, 2020.
- [7] Ross H Nehm, Minsu Ha, and Elijah Mayfield. Transforming biology assessment with machine learning: automated scoring of written evolutionary explanations. *Journal of Science Education and Technology*, 21(1):183–196, 2012.
- [8] Minsu Ha, Ross H Nehm, Mark Urban-Lurain, and John E Merrill. Applying computerized-scoring models of written biological explanations across courses and colleges: prospects and limitations. *CBE—Life Sciences Education*, 10(4):379–393, 2011.
- [9] Ou Lydia Liu, Joseph A Rios, Michael Heilman, Libby Gerard, and Marcia C Linn. Validation of automated scoring of science assessments. *Journal of Research in Science Teaching*, 53(2):215–233, 2016.
- [10] Gregory J Kelly and Allison Takao. Epistemic levels in argument: An analysis of university oceanography students’ use of evidence in writing. *Science education*, 86(3):314–342, 2002.
- [11] Hee-Sun Lee, Amy Pallant, Sarah Pryputniewicz, Trudi Lord, Matthew Mulholland, and Ou Lydia Liu. Automated text scoring and real-time adjustable feedback: Supporting revision of scientific arguments involving uncertainty. *Science Education*, 103(3):590–622, 2019.
- [12] John B Biggs and Kevin F Collis. *Evaluating the quality of learning: The SOLO taxonomy (Structure of the Observed Learning Outcome)*. Academic Press, 2014.
- [13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [14] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196, 2014.
- [15] Mengyang Zhao, Chornng Hwa Chang, Wenbin Xie, Zhou Xie, and Jinyong Hu. Cloud shape classification system based on multi-channel cnn and improved fdm. *IEEE Access*, 8:44111–44124, 2020.
- [16] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [17] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

- [18] Shalini Ghosh, Oriol Vinyals, Brian Strope, Scott Roy, Tom Dean, and Larry Heck. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291*, 2016.
- [19] Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*, 2016.
- [20] Kaveh Taghipour and Hwee Tou Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1882–1891, 2016.
- [21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [22] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.
- [23] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. From word embeddings to document distances. In *International conference on machine learning*, pages 957–966, 2015.
- [24] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in neural information processing systems*, pages 2292–2300, 2013.
- [25] Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.
- [26] Charlie Frogner, Farzaneh Mirzazadeh, and Justin Solomon. Learning embeddings into entropic wasserstein spaces. *arXiv preprint arXiv:1905.03329*, 2019.
- [27] Xiaofeng Yuan, Lin Li, and Yalin Wang. Nonlinear dynamic soft sensor modeling with supervised long short-term memory network. *IEEE Transactions on Industrial Informatics*, 2019.
- [28] Jacob Cohen. Weighted kappa: nominal scale agreement provision for scaled disagreement or partial credit. *Psychological bulletin*, 70(4):213, 1968.
- [29] Edward Loper and Steven Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.
- [30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [31] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.
- [32] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.
- [33] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. 2016.
- [34] Claudia Leacock and Martin Chodorow. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities*, 37(4):389–405, 2003.