Natural Language Processing System for Self-Reflection and Peer-Evaluation

Rui Wang
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47906, USA
ruiw@purdue.edu

Siqing Wei, Matthew W. Ohland and Daniel M. Ferguson
School of Engineering Education
Purdue University
West Lafayette, IN 47906, USA
wei118@purdue.edu, ohland@purdue.edu, dfergus@purdue.edu

Abstract

Peer evaluation has been well established as an effective method to motivate team members to reflect their contribution and performance, to enforce sense of responsibility, to act as an incentive for demonstrating good interpersonal skills and to help the team achieve its goals. Behaviorally anchored rating scale is generally considered as an efficient and fair method to measure certain scores. However, in the application of peer evaluation, numerical rating could be influenced by raters' biased understanding of the scale based on their cultural background. Supplementing peer-to-peer comments with numerical peer evaluation system could remediate rater bias effect. In this paper, we propose a natural language processing model that (1) processes the peer-to-peer comments about rater's teammates' teamwork behaviors; (2) converts comments into numerical space that allow for computation We evaluate our results against CATME data and validate our proposed system.

Keywords
Natural language processing, peer evaluation, peer comments, CATME, teamwork

1. Introduction

Teamwork is rated as the most important competency by engineering graduates based on their professional practice and the Accreditation Board for Engineering and Technology (ABET) also lists teamwork as one of required competency that engineering graduate should demonstrate throughout college training (Wei, 2019). To facilitate improving teamwork skills, the Comprehensive Assessment of Team Member Effectiveness (CATME) was developed to help instructors better manage teams and help students improve teamwork behaviors by its embedded peer evaluation system based on a behaviorally anchored rating scale (Ohland et al. 2012). CATME has currently over 7,000 active instructor accounts for instructors across multiple disciplines in over 2,000 institutions worldwide. CATME evaluation tool includes five dimensions of teamwork behaviors on which team members are asked to rate themselves and their teammates:

- Contributing (C) to the Team's Work: being able to add value to a team's work/project. Team members are rated on how well they meet their commitments, do their share of the work, and help their teammates.
- Interacting (I) with Teammates: how individuals communicate within their teams. It includes encouraging teammates, communicating ideas clearly, and listening respectfully to others' ideas.
- Keeping the Team on Track (K): being aware of milestones and deadlines and ensuring that the team is making appropriate progress.
- Expecting Quality (E): taking steps to ensure that the team meets or exceeds all requirements for project outcomes.

- Having (H) Relevant Knowledge, Skills, or Attributes (KSAs): the base knowledge of individual team members. It means having the required KSAs to solve the problems at hand or being willing to learn the KSAs the ratee lacks. (Loughry et al. 2007)

Peer evaluation has been well established as an effective method to improve teamwork experience among team members and boost team performance over time. (Gueldenzoph and May 2002; Hansen 2006) Peer evaluation mechanisms enforces a sense of responsibility and encourages team members to demonstrate adequate interpersonal skills and make contributions to the team to promote team success in achieving its goals. (Loignon et al. 2017) Peer evaluation can be conducted both in numerical scoring ratings, as giving peer a score on certain aspects of his or her performance, and in comments describing, discussing and evaluating teamwork behaviors of a given team member.

Previous research work in peer evaluation data, however, has primarily delved into numerical ratings where it is straightforward to perform computations with quantitative analysis (Wei 2019; Ferguson et al. 2018) and the peer-to-peer comments have been largely left out with a few works investigating the value of this certain section (Brawner et al. 2018). Nevertheless, peer-to-peer comments and self-reflections are free of the influence of both unintended or intended raters' bias towards peer rating and understanding of behavioral scale. We argue that the value which peer-to-peer textual comment data possesses deserves more attention from the research community

In this work, we propose a model through which we are capable of mapping peer-to-peer comments to scores in CATME teamwork dimensions by firstly determining whether the words used to describe behaviors in each dimension are existing and secondly assigning scores to corresponding dimensions. By assuming that such mapping exists, we approximate this mapping with the state-of-the-art model in the field of natural language processing (NLP) and then validate the proposed model.

To realize projecting textual comments to the numerical space, we introduce several plausible natural language processing techniques. The field of natural language processing or computational linguistics has seen a tremendous development along with the resurgence of artificial neural network. After the introduction of word embeddings (Mikolov et al. 2013; Pennington et al. 2014) and large language models (Peters et al. 2018; Radford et al. 2018; Devlin et al. 2019; Radford et al. 2019; Yang et al. 2019; Liu et al. 2019), some of which are capable of surpassing human performance on certain tasks like sentence classification, where each sentence is assigned a label by humans and the task is to devise an automated system to classify each sentences into the correct category. With the help of these techniques, we developed the model shown in Section 4. With empirical validation evidence, we conclude that we have reached a model with great performance.

We have our literature review in Section 2 and some preliminaries in Section 3. In Section 4, we present our model and how it came to be. In Section 5 we show our evaluation of our proposed model and in Section 6 and 7 we discuss the limitations and the future improvements of the current model.

2. Literature Review

2.1 Peer Evaluation and Peer-to-Peer Comments

Investigation into peer-to-peer comments has been of interests for recent studies (Brawner et al. 2018), which observe that good quality written comments are congruent with peer ratings and also very informative about individual dysfunctionality or dysfunctional team behaviors. Such comments can indicate certain behaviors that is not noticeable through ratings alone as evidenced in Table 3 of the study. This study also shows that it is difficult for exact agreement on short snippets of comments. Although this study also suggests training for students to improve the quality of comments, it does not, however, provides means to analyze the comments automatically, which is very important for class with a large number of teams.

2.2 Natural Language Processing

With the advancement in machine learning field, especially the resurgence of artificial neural networks, natural language processing community has evidenced tremendous progress in recent years. Word2Vec (Mikolov et al. 2013) and other methods for word embeddings encodes natural language words into continuous domain of lexical information with each of the vector encodes syntactical and semantical information of its corresponding word. Large language models like ELMo (Peters et al. 2018), GPT (Radford et al. 2018), BERT (Devlin et al. 2019), GPT-2 (Radford et al. 2019), XLNet (Yang et al. 2019), ERNIE (Sun et al. 2019) and RoBERTa (Liu et al. 2019), pre-trained on unprecedented amount of data with the single objective of predicting a single word given its context, continues to push the boundaries of state-of-the-art results on NLP tasks sentiment analysis, sentence classification, semantical similarity detection, question answering, and reading comprehension.

In this study we mostly use RoBERTa as our backbone model as it is the most successful model at the time of this work (More information is provided in Section 2.2.1 and 2.2.2). RoBERTa is short for a robustly optimized BERT pre-training approach, which is a reiteration of BERT architecturally but a more robust way to perform the optimization, or training, procedure. BERT is regarded one of the milestones in NLP community because of its state-of-the-art and even super-human performance on a great variety of tasks. We give a brief high-level overview of the architecture and training procedure of RoBERTa in Section 3, but there are more left for the interested readers in the thread of work (Vaswani et al. 2017; Radford et al. 2018; Devlin et al. 2019; Yang et al. 2019; Liu et al, 2019) as it requires significant effort and space to elaborate in a detailed fashion.

2.2.1 BERT

BERT (Devlin et al. 2019) is short for Bidirectional Encoder Representations from Transformers. *Bidirectional* means this model extract information from one sentence not only in left-to-right direction but also the other way around, mimicking what we often find ourselves doing when reading a sentence. *Encoder* is simply describing BERT is encoding each input sentence into numerical space with continuous vector representations. Transformer (Vaswani et al. 2017) is an architecture devised, in the first place, for neural machine translation, in which field despite its smaller model size and faster training time surpassed then state-of-the-art results. Researchers since then have been trying to incorporate the Transformer architecture into other tasks in addition to neural machine translation. Because of its ability to model long term dependencies (Hochreiter and Schmidhuber 1997), a problem faced by NLP community from the start, and to parallelize computation for faster runtime with multiple computation cores (Vaswani et al. 2017), Transformers have gained tremendous focus and proved the architecture's performance is better than those of the previous models.

At the time of its release, BERT topped GLUE (Wang et al. 2019), RACE (Lai et al. 2017) and SQuAD (Devlin et al. 2016) leader boards. Some of the performance even match or surpass that of humans.

2.2.2 RoBERTa

RoBERTa (Liu et al. 2019) is another iteration of BERT which uses a more robust training recipe. It drops the objective of *next sentence prediction*, in which a model predicts whether two input sequence are supposed to be consecutive. It also utilizes a significantly larger dataset from 16GB to 160GB and a larger batch size for a longer period of time for training. Upon its release, RoBERTa topped BERT GLUE (Wang et al., 2019), RACE (Lai et al. 2017) and SQuAD (Devlin et al. 2016) leader boards.

3. Preliminaries

3.1 Deep Learning

3.1.1 Machine Learning

Machine learning (ML) is commonly viewed as a subfield of artificial intelligence, where practitioners and researchers do not hard-code rules into machines to perform certain intelligent actions or inference but instead make the model to learn from data (LeCun et al. 2015). Canonical machine learning problems can be divided into three subsets, namely supervised learning where data are labeled usually by humans (as supervised by humans) (LeCun et al. 2015), unsupervised learning where the data is not labeled and reinforcement learning where a model performs a certain action and receives a reward or punishment somewhere in the future (Kaelbling et al. 1996). In our paper, we mostly leverage the advances from the supervised learning branch of the community where the advancements are arguably most fruitful.

In a typical supervised learning fashion, when one is given a dataset, we split the dataset into training set and testing set with techniques like bootstrapping, or just simple random selection. We also determine what model architecture to use. During training, we try to optimize the parameters of the model based on certain objective function, also referred to as loss function in ML community, with the information provided by both the inputs and the corresponding labels of the training set. The idea is that after the model is trained on the training set, the model parameters are fixed, and hopefully it will be capable of performing reasonably well with respect to the objective function, on the testing set which it has not seen during training time. Usually such objective functions are based on the idea to improve alignment between the model prediction of certain inputs and their corresponding labels, as is our setting.

3.1.2 Deep Learning

Supervised machine learning is about building models that take in data as input and output predictions. In this paradigm, the most significant advancement is the resurgence of deep learning, which uses deep artificial neural network (DNN) as the model that learns from the data. DNN can be viewed as a universal function approximator, which means that we can approximate the true mapping with such model given enough data. However, the performance such approximator depends largely on the dataset chosen, the compatibility of the model architecture, the dataset and the given tasks. The elaboration of deep learning is beyond the scope of this work, we refer the readers to this review (LeCun et al. 2015).

3.1.3 Training

The term training in the setting of machine learning generally means optimization with objective functions that align with the goal of the objective of the machine learning research or practice. The optimization, however because of the complexity of the surface of the objective function, mostly ends up in local minima, which have been suggested to be good enough in most situations. Training is performed on the dataset. However, not all data can be used for training, as we also need some parts of the data to validate the performance of developed model and, sometimes, inference to guarantee the quality of the training. Hence, we usually divide our dataset into training set, validation set and testing set.

The optimization algorithm used ubiquitously by ML community is stochastic gradient descend (SGD) or its variants such as adaptive momentum (Kingma and Ba 2015), which is one of the most prevalent optimizers and the one we employ in this work. In its core, SGD is about to perform the following update to parameter w,

$$w = w - \alpha \times w',$$

where w is the original parameter value, alpha is called learning rate and w' is the derivative of w. We use the chain rule of differentiation to propagate loss signal Variants of SGD usually perform some augmentation with momentum or utilize the approximation of the second order derivative since the exact computation is computationally inefficient. The differentiation is performed with respect to the loss function, which is usually computed by a mini-batch of the entire dataset for its feasible memory requirements.

Training is performed, in an ideal setting, as long as the model is not overfitting, which is determined by the trend of the validation loss, that is, the loss function value computed using the validation dataset. During training, the validation set has never been seen by the model, which guarantees the determination process of the overfitting criterion is adequate. Before the overfitting, both the training loss, computed by training dataset, and the validation loss should share the same trend of decreasing. For overfitting, however, one can observe that with the decrease of training loss, which is guaranteed by the algorithm, the same no longer holds for validation loss.

3.1.4 Inference

After training ended, the model performs inference to either carry out its design purpose or to evaluate its performance. We use the testing set to report the performance of our model. How these test, training and validation set came to be will be discussed in Section 4.

3.2 Natural Language Processing

The most exiting advancement recently in NLP is the advent of large pre-trained language models. Because its performance, we use RoBERTa as our base pre-trained model. Modern language models are based on layers of artificial neural networks of Transformer layer (Vaswani et al. 2017), which has a representation each layer that we can extract and feed as features into the classifiers.

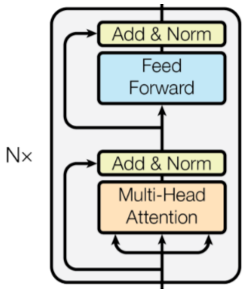

Figure 1. Transformer layer from (Vaswani et al. 2017)

3.2.1 Language Models

Language models (LM) are models that predict the words based on their context. LMs are generally trained through the techniques of machine learning and deep neural network. Next word prediction, for instance, where the model predicts the next word in a sentence provided the previous words of the sentence, is one of the earliest language model objectives. Cloze task, on the other hand, is to predict a masked token given the background. With an objective like the next word prediction and cloze task, LMs can be trained with unannotated data as the next word is automatically the label, which allows models of particular large architectures and allows the access to huge amount of training data.

3.2.2 Transformers

Before Transformers (Vaswani et al. 2017), NLP community mostly use recurrent neural networks (RNN) (Chung et al. 2014) and its variants like Long Short-Term Memory(LSTM) (Hochreiter and Schmidhuber 1997). Problems with this method includes issues in training for deeper models as people put more and more layers of LSTM on top of the lower ones, not capable of learning long term dependencies, and not able to parallelize its computation as the output of RNN on one token over one sequence depends on the output value of previous token.

Attention mechanisms (Bahdanau et al. 2015), on the other hand, generally do not have those constraints. Hence, Transformers use stacks of mutli-headed Attention layer, which is attention performed several times on one layer, and totally rid of RNNs. A typical transformer layer, as in Figure 1, consist of a mutli-headed attention layer with its inputs added onto its output, called residual connection (He et al. 2016), followed by a simple feed-forward layer with the same residual connection, both residual connections were followed by LayerNorm (Ba et al. 2016). With extensive empirical results and performance reports (Vaswani et al. 2017) (Radford et al. 2018), it is reasonable to trust that Transformers are a better alternative to RNNs.

3.2.3 Transformer-based Language Models

Though Transformer was designed as a model to perform natural language translations, which uses an encoder-decoder transduction structure, great efforts have been put in and tremendous results have been reported from using only the encoder part of Transformer as the basic building block of large language models. Transformer layers enable the models to go deeper. For example, ELMo (Peters et al. 2018) used only LSTMs and can only go as deep as three layers while the original Transformer has 12 layers in total and BERT (Devlin et al. 2019) has 24 in its larger version.

4. Methodology

Given the fact that trained expert raters are capable of performing such comments-to-scores mapping, we argue that our hypothesis holds. However, to hard code such a mapping requires tremendous effort. As an alternative, we leverage the recent advances in the deep learning and natural language processing community to build this mapping.

4.1 Data

4.1.1 Dataset details

We use the data from (Brawner et al. 2018), in which the process of obtaining expert-rating system is described in Section 3. After carefully selected teams to code, one of the authors of the paper calibrates her ratings with Dr. Ohland, CATME Principal Investigator, considered a subject matter expert. This author then went on and coded 46 teams. Another author, who was then a previous user of CATME, coded 15 of those teams and their results are compared. 77% exact agreement and 85% agreement within 1 on 5-point scale discrepancy was reported.

4.1.2 Data Augmentation

As per the general underlying assumption of training most, if not all, deep neural networks, we need a huge amount of data. In our settings, the requirement is even higher as we need labels to train the models. Based on the dataset that we have with its 480 instances, to develop any robust model, we need proper data augmentation techniques.

The first technique that we use is referred to as back translation (Sennrich et al. 2016) (Lample et al. 2018) in the natural language processing community where we translate an original sentence into another intermediate language like Chinese or French and then translate it back into English. In such a fashion, we obtain two sentences, though almost identical meaning, of two separate and different representations which we can use the same labels because while syntactically different, the new English sentence and the original sentence are semantically rather close. In our implementation, the intermediate language that we chose to perform back translations are Chinese, French, German, Spanish, Korean, Greek, Romanian, Hindi, Japanese, Italian, Irish, Georgian, Finnish and Czech, summing up to a total of 14 intermediate languages, which brings the total number to 7,200 instances.

Though a 14-time increasement may seem large at first, we must acknowledge the fact that though syntactically diverse, these instances share around only 480 clusters in the semantic space, which means we need to augment the data through other techniques. One of which is to utilize large pre-trained language models. Many works (LeCun et al. 2015) have been done in this fashion to augment small datasets, which constitutes a paradigm in machine learning called few-shot learning where a model learns the smaller dataset after trained on a large, general dataset.

RoBERTa (Liu et al. 2019), the large language model that we use in this work, was trained with 160 GB raw text data. We will use the last layer of the model as the sentence representation. While we are not directly augmenting the dataset itself, the fact that the language model is capable of correctly determining the word based on context words, it must learn to extract both syntactically and semantically meaningful representations through proper analysis of the context words. We expect it to perform similar, if not identical, analysis in our settings. It is a proper expectation to hold given the success of such procedure done for FewRel dataset (Han et al. 2018) by (Baldini Soares et al. 2019), some of which actually reports human-surpassing performance. Without such analysis, our whole model would have to learn to conduct these analyses on a very small and yet challenging dataset, which will almost be guaranteed to overfit the dataset, that is, the performance results only pertain to the 480 instances that we have.

4.1.3 Splitting data

To properly evaluate when to stop training and more importantly measure the end performance of our model, we need to split our data properly.

For the testing set that we use to test the performance of our model, we randomly select 80 instances out of the 480 original data instances without back translation. Data splitting is performed without replacement and the probability of each instance to be chosen following a uniform distribution.

If we simply divide the remaining data into training and validation set, we would lose another chunk of the data to validation rather than training. To avoid this issue, we instead split the data in k equal-sized splits. We use this as a method to train ensemble models where one trains many models and use the average of the output of each model as the output of his or her final model (Mosteller and Tukey, 1968). For each single model that we train, we take one of the splits as the validation set and train this particular model on the remaining splits. We perform such training method on k models in total. We made sure that one split only served once as the validation set. This allows the final ensemble model to leverage all the information in the training set without compromising validation.

It is worth noting that we perform all the above procedures on the original 480 instances and only the training set is back translated to boost performance.

4.1.4 Text Encoding

Before the text is ready to be processed by the core model of the pipeline, there are certain processes to be performed in order to make it easier for the model to understand the text.

4.1.4.1 Lowering the case

In a way, we can view our task in this paper as sentence classification, that it, we classify each sentence as whether it mentions anything related to the CATME teamwork dimensions behavior scale description, and if so, we classify the related information into actual scores on a scale of 1 to 5 with 1 being the lowest and 5 being the highest. From this standpoint, we argue that it loses some but far from significant information if we lower the cases of all the word in the comments, but this specific action will lower the pressure of the model as it does not have to learn a separate representation for the same word with different cases.

4.1.4.2 Byte-Pair Encoding

Byte-Pair Encoding (BPE) (Gage 1994) (Sennrich et al. 2016) is a word representation scheme, hybrid of character and word level representation. Depending on the word performed on, BPE might leave the word untouched, or separate it into two or more sub-word units, whether each word should be dissected and, if so, what it should be dissected into are learned with statistical analysis of a chosen corpus. This allows us to deal with unmet words, misspelled word, for example, during inference when we put our trained model into task.

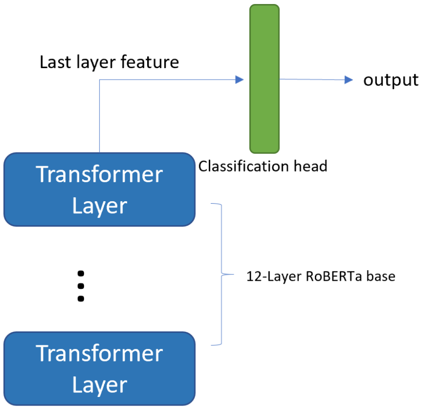

Figure 2. The general structure of our model

4.2 The Model

4.2.1 Base Model Configuration

Due to limitations of our computation resources, we have chosen the base model of RoBERTa rather than the better, larger version of it. However, a base model is far from inadequate. It still contains 12 Transformer layers and the last layer output has a dimension of 768. We add a classification head on top of the last layer for each dimension to get its score and another classification head for its threshold. The classification consists of three layers of fully connected neurons with ReLU (Gordon and Dunson 2011) activation and 0.7 probability of dropout (Srivastava et al. 2014). The middle layer is of dimension 3072. The classification heads for score values are activated through a LeakyReLU layer and the last layer is passed to a linear layer and the threshold heads were passed in through sigmoid function,

$$sigmoid(x) = \frac{1}{1 + e^{-x}},$$

which restrains the output into a single interval from 0 to 1. The model is as demonstrated in Figure 2.

4.2.2 Loss function

We use mean square error (MSE) with mean reduction to compute the loss of scoring discrepancies and use Focal Loss (Lin, Goyal et al. 2017) to compute the loss for binary classification on whether to output this dimension score or not. The MSE loss is clipped where the model prediction is larger than 5 and the real expert rating is 5 and where the model prediction is smaller than 1 and the real expert rating is 1. It is also disregarded if the expert rating does not exist. The focal loss is computed as follows,

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t),$$

where

$$p_t = \begin{cases} p & if\ expert\ coding\ exists \\ 1 - p & otherwise \end{cases}$$

α_t is the scaling factor where it is set by inverse class frequency, p represents the output of binary classification heads and γ is a hyperparameter and is set as 2 in our setting. We use the sum of both MSE and Focal Loss functions as our final loss signal. Focal loss is better suitable for class imbalance problems in binary classification as it lowers the weight for easier examples that the model already learns well. We use this loss because it is suggested in previous study (Brawner et al. 2018) that certain dimensions are coded (rated) more regularly than not.

4.3 Training procedure

We use Adam optimizer (Kingma and Ba 2015) with the default parameters except learning rate α is configured to 10^{-5} using PyTorch framework to train 5 models of the same architecture. Hence, we split the data into $k = 5$ folds. For each fold, we use batch size of 64 for two full epochs before overfitting.

5. Evaluation

5.1 Metrics

For the decision on whether to code (whether there is relevant information), we use F_1 score as our metric,

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}},$$

where,

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive},$$
$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

which has good qualities for imbalanced classes. For a higher F_1 score, a binary classification model has to be able to produce both good precision and good recall metrics.

We also use agreement as found in (Brawner et al. 2018).

5.2 Results

We report our F_1 score as 0.67 on the testing dataset with the threshold of determination determined on the validation set. Since for each sentence this task contains 5 binary classification tasks for F_1 score to be computed on, we deem this result very positive. Though our model achieves good results for its design purpose, it is very reasonable to assume that with more data we are expecting substantial performance boost on the next iteration of the model. Moreover, we are excited to be able to bring this feature online for instructors to use.

For the agreements, the model agree with the expert rater 61.5% exactly and 71.5% with 1-point discrepancy on a 5 points scale (where we allow for right decision on whether to code and at most 1 points off from the expert rater Author Brawner in the original paper for this dataset). Given previous studies have recognized the difficulty of exact agreement (Brawner et al. 2018), we regard this result satisfactory. this We note that this is largely limited by the available dataset size and argue that with a bigger dataset we can have 1-point discrepancy over 80%, which is the threshold normally considered acceptable for inter-coder reliability (Fraenkel and Wallen, 1993).

6. Discussion

Though our model achieves good results for its design purpose, it is very reasonable to assume that with more data we are expecting substantial performance boost on the next iteration of the model. Later, we could utilize this instrument to analyze the relationship between students' written comments and provided numerical rating to provide constructive suggestion and tool to improve the quality of students' comment and perhaps, automating dimension grade assignment based on comment.

This pivot study also explores a new research method to investigate qualitative data. Hopefully, this work inspires lots of other novel research in this field.

7. Conclusion

We proposed a model capable of processing the peer-to-peer comments about rater's teammates' teamwork behaviors by converting comments into numerical space that allow for computation. The performance of the system is satisfying based on the dataset size that we have, which reaches 61.5% accuracy.

References

Ba, J. L., Kiros, J. R., and Hinton, G. E., *Layer Normalization*, Available: https://arxiv.org/abs/1607.06450?source=post_page, July 21, 2016.

Bahdanau, D., Cho, K., and Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate, *3rd International Conference on Learning Representations (ICLR),* San Diego, 2015.

Baldini Soares, L., FitzGerald, N., Ling, J., and Kwiatkowski, T., Matching the Blanks: Distributional Similarity for Relation Learning, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy: Association for Computational Linguistics, pp. 2895-2905, 2019.

Brawner, C. E., Olivia, M. W., Ferguson, D. M., and Ohland, M. W., Comparing Peer-to-Peer Written Comments and Teamwork Peer Evaluations, *presented at 2018 ASEE Annual Conference and Exposition*, Salt Lake City, Utah, 2018.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y., *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*, Available: https://arxiv.org/abs/1412.3555, December 11 2014.

Devlin, J., Chang, M.-W., Kenton, L., and Toutanova, K. (2019, May 24), *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. Retrieved from BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding: https://arxiv.org/abs/1810.04805.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., SQuAD: 100,000+ Questions for Machine Comprehension of Text, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas: Association for Computational Linguistics, pp. 2383-2392, 2016.

Ferguson, D. M., Ohland, M., Lally, C., Ibriga, H., and Cao, Y., Evaluating the effect of different teamwork training interventions on the quality of peer evaluations, *Proceedings of the Frontiers in Education Conference*, San Jose, CA, pp. 1-5, 2018.

Fraenkel, J. R., and Wallen, N. E., *How to Design and Evaluate Research in Education*, New York: McGraw-Hill, 1993.

Gage, P., A New Algorithm for Data Compression. *C Users Journal*, 23-38, 1994

Gordon, G., and Dunson, D. D., Deep Sparse Rectifier Neural Networks, *Proceedings of Machine Learning Research*, Fort Lauderdale, FL: PMLR, pp. 315-323, 2011.

Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., and Sun, M., FewRel:A Large-Scale Supervised Few-Shot Relation Classification Dataset with State-of-the-Art Evaluation, *EMNLP*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J., Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, pp. 770-778, 2016.

Hochreiter, S., and Schmidhuber, J., Long Short-Term Memory, *Neural Computation*, 1735-1780, 1997

Kaelbling, L. P., Littman, M. L., and Moore, A. W., Reinforcement Learning: A Survey *Journal of Artificial Intelligence Research*, 237-285,1996.

Kingma, D. P., and Ba, J., Adam: A method for stochastic optimization, *Proceedings of 3rd International Conference on Learning Representations (ICLR)*, San Diego, 2015.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E., RACE: Large-scale ReAding Comprehension Dataset From Examinations. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, Available: https://www.cs.cmu.edu/~glai1/data/race/, pp. 785-794, 2017.

Lample, G., Conneau, A., Denoyer, L., and Ranzato, M., Unsupervised Machine Translation Using Monolingual Corpora Only, *International Conference on Learning Representations*, Vancouver, Canada, 2018.

LeCun, Y., Bengio, Y., and Hinton, G., Deep Learning, *Nature*, pp. 436–444, 2015.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P., Focal Loss for Dense Object Detection, *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, pp. 2999-3007, 2017.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., . . . Stoyanov, V., *RoBERTa: An optimized method for pretraining self-supervised NLP systems*, Available: https://ai.facebook.com/blog/roberta-an-optimized-method-for-pretraining-self-supervised-nlp-systems/, July 29, 2019.

Loughry, M. L., Ohland, M. W., and Moore, D. D., Development of a Theory-Based Assessment of Team Member Effectiveness. *Educational and Psychological Measurement*, pp. 505-524, 2007.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J., Distributed Representations of Words and Phrases, *Advances in Neural Information Processing Systems 26*, Harrahs and Harveys, Lake Tahoe: Curran Associates, Inc, pp. 1-9, 2013.

Mosteller, F., and Tukey, J., *Handbook of Social Psychology*, MA: Addison-Wesley, Reading, 1968

Ohland, M. W., Loughry, M. L., Woehr, D. J., Felder, R. M., Finelli, C. J., Layton, R. A., . . . Schmucker, D. G. The Comprehensive Assessment of Team Member Effectiveness: Development of a Behaviorally Anchored Rating Scale for Self- and Peer Evaluation, *Academy of management Learning and Education*, pp. 609-630, 2012.

Pennington, J., Socher, R., and Manning, C. D., GloVe: Global Vectors for Word Representation, *Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp. 1532-1543, 2014.

Peters, M. E., Neumann, M., Iyyer, M., and Gardner, M., Deep contextualized word representations, *Proceedings of NAACL*, New Orleans, 2018.

Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I., *Improving Language Understanding by Generative Pre-Training*, Retrieved from Improving Language Understanding with Unsupervised Learning: https://openai.com/blog/language-unsupervised/, June 11, 2018

Radford, A., Wu, J., Child, R. L., and Amodei, D., *Better Language Models and Their Implications*, Available: https://openai.com/blog/better-language-models/, February 14, 2019.

Sennrich, R., Haddow, B., and Birch, A. Improving neural machine translation models with Monolingual Data, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany: Association for Computational Linguistics, pp. 86-96, 2016.

Sennrich, R., Haddow, B., and Birch, A., Neural Machine Translation of Rare Words with Subword Units, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Berlin, Germany: Association for Computational Linguistics, pp. 1715-1725, 2016.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R., Dropout: A Simple Way to Prevent Neural Networks from. *Journal of Machine Learning Research*, pp. 1929-1958, 2014.

Sun, Y., Wang, S., Li, Y., Feng, S., Tian, H., Wu, H., and Wang, H., *ERNIE 2.0: A Continual Pre-training Framework for Language Understanding*. Available: https://arxiv.org/abs/1907.12412, July 29, 2019

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I., Attention is All you Need. *Advances in Neural Information Processing Systems 30,* Long Beach: Curran Associates, Inc, pp. 5998-6008, 2017.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R., GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding, *International Conference for Learning Representations* New Orleans, 2019.

Wei, S. a., Examining the Cultural Influence on Peer Ratings of Teammates between International and Domestic Students, *Proceedings of American Society for Engineering Education Annual Conference and Exposition.* Tampa, FL, 2019.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V., *XLNet: Generalized Autoregressive Pretraining for Language Understanding*, Available: https://arxiv.org/abs/1906.08237, June 19, 2019.

Biographies

Rui Wang is a junior undergraduate student and Research Assistant in Engineering Education at Purdue University. He is currently majoring in Computer Engineering. His research interests include natural language processing, adversarial examples in computer vision and machine translation. Rui is currently conducting research at CATME with emphasis on the CATME peer comment data.

Siqing Wei received both bachelor's and master's degrees in electrical and Computer Engineering from Purdue University. He is currently pursuing Ph.D degree in Engineering Education at Purdue University. After years of experience of serving a peer teacher and a graduate teaching assistant in first year engineering courses, he is a research assistant at CATME research group studying the existence, causes and interventions on international engineering teamwork behaviors, the integration and implementation of team-based assignments and projects into STEM course designs and using mixed-method, especially natural language processing to student written research data, such as peer-to-peer comments. Siqing also works as the technical support manager at CATME research group.

Prof. Daniel M. Ferguson is CATME Managing Director and the recipient of several NSF awards for research in engineering education and a research associate at Purdue University. Prior to coming to Purdue he was Assistant Professor of Entrepreneurship at Ohio Northern University. Before assuming that position he was Associate Director of the Inter-Professional Studies Program [IPRO] and Senior Lecturer at Illinois Institute of Technology and involved in research in service learning, assessment processes and interventions aimed at improving learning objective attainment. Prior to his University assignments he was the Founder and CEO of The EDI Group, Ltd. and The EDI Group Canada, Ltd, independent professional services companies specializing in B2B electronic commerce and electronic data interchange. The EDI Group companies conducted syndicated market research, offered educational seminars and conferences and published The Journal of Electronic Commerce. Dr. Ferguson is a graduate of Notre Dame, Stanford and Purdue Universities, a special edition editor of the Journal of Engineering Entrepreneurship and a member of Tau Beta Pi.

Prof. Matthew W. Ohland is Professor of Engineering Education at Purdue University. He has degrees from Swarthmore College, Rensselaer Polytechnic Institute, and the University of Florida. His research on the longitudinal study of engineering students, team assignment, peer evaluation, and active and collaborative teaching methods has been supported by the National Science Foundation and the Sloan Foundation and his team received Best Paper awards from the Journal of Engineering Education in 2008 and 2011 and from the IEEE Transactions on Education in 2011 and 2015. Dr. Ohland is an ABET Program Evaluator for ASEE. He was the 2002–2006 President of Tau Beta Pi and is a Fellow of the ASEE, IEEE, and AAAS.