G9: Assessing English Language Learning using Semi-supervised Support Vector Regression

Yulin Zhao
yulin6@illinois.edu
University of Illinois Urbana-Champaign
Urbana, Illinios, USA

Haonan Sun
haonan6@illinois.edu
University of Illinois Urbana-Champaign
Urbana, Illinios, USA

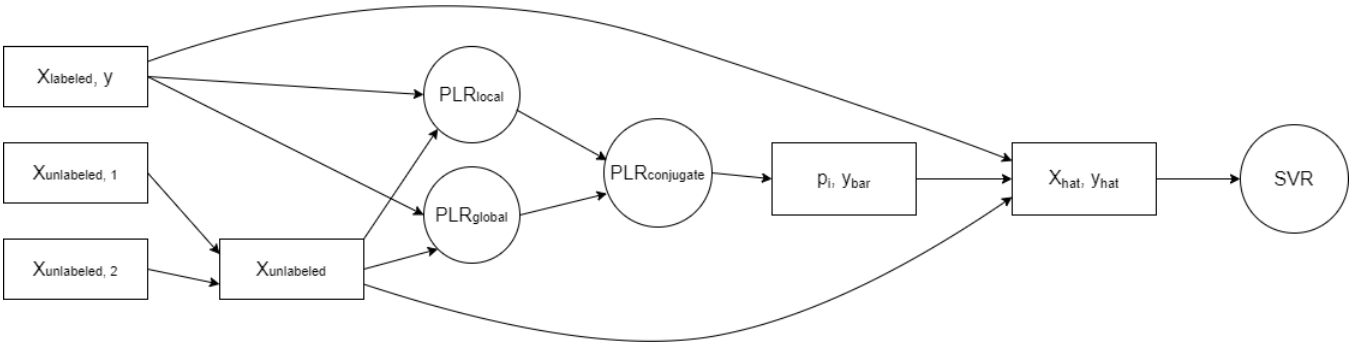

Figure 1: General Steps of the Algorithm

ABSTRACT

Currently, the way language proficiency is assessed for English language learners in grades 8-12 is still primarily through standardized assessments by English language educators through a combination of personal or institutional criteria and experience. In this project, we developed a Support Vector Regression (SVR) machine learning model which could provide English language assessment automatically. To improve our model performance, we introduced RoBERTa to embed the texts and semi-supervised learning to allow our model to train on much larger datasets.

CCS CONCEPTS

• **Information systems → Data mining**.

KEYWORDS

Semi-supervised SVR, SVR, Co-training, DeBERTa-V3, DeBERTa, RoBERTa, SVM, Bert

ACM Reference Format:
Yulin Zhao and Haonan Sun. 2022. G9: Assessing English Language Learning using Semi-supervised Support Vector Regression. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

This is an ongoing Kaggle Competition "Feedback Prize - English Language Learning." It aims to build an ML model to automatically provide a six-dimension assessment for the language proficiency of English Language Learners. Given a text, our model is expected to provide predictions on cohesion, syntax, vocabulary, phraseology, grammar, and conventions. The competition will measure the model using the mean columnwise RMSE and also time efficiency. We define this task to be a regression and NLP task.

2 MOTIVATION

With the increasing request about the strength of the English writing capability for English learners, the current grading system is no longer effectively making reasonable and efficient judgments for language ability for ELLs. Thus, we are trying to build more efficient automated writing evaluation (AWE) systems using ML models and NLP techniques. We believe our work will contribute to this community and help to support the integration of technology and education. Besides, our group is interested in dealing with NLP tasks and wants to learn more about semi-supervised models. In this project, we could be exposed to these fields and apply similar techniques to solve problems.

3 RELATED WORK

3.1 Datasets

The labeled dataset we used is the English Language Learning[1] provided by the Kaggle competition. It contains 3911 English essays with evaluations completed by humans, which are rated on a scale out of five on six different dimensions.

[1] https://www.kaggle.com/competitions/feedback-prize-english-language-learning

The other two unlabeled datasets are previous Evaluating Student Writing[2] and Automated Essay Scoring[3] also provided by the Kaggle competition. Evaluating Student Writing dataset contains 15,594 essays and we will not introduce other features. Automated Essay Scoring contains 12,976 essays and we will not introduce the corresponding ratings in that dataset. These two datasets will be unlabeled for our project.

3.2 NLP Transformers

Compared with other previous models, BERT (Bidirectional Encoder Representation from Transformers) [2] introduced bidirectional conversion decoding and a much larger pre-training scale. BERT creatively introduced MLM (Masked Language Model) to capture information from both left and right tokens, and has a certain probability to substitute some tokens with masks or other tokens.

RoBERTa (A Robustly Optimized BERT) [8] is an improved version of BERT, which uses a larger model parametric size, larger batch size, and more training data. Roberta has also some improvements in its training methods: it discards the next-sentence prediction (NSP) task and enables dynamic masks and new text encoding strategies. Thus, RoBERTa represents a better generalization to downstream tasks than BERT.

DeBERTa (Decoding-enhanced BERT with disentangled attention) [5] has two main improvements, compared with RoBERTa. The first one is disentangled attention, where each word uses two vectors respectively, and the attention weights between the words are computed by using their text and relative position disentanglement matrices respectively. The second technique uses an enhanced mask decoder Electra that introduces absolute positions in the decoding layer to predict masked tokens. DeBERTa-V3 [4] has further improvements on MLM and Electra. It introduced the new disentangled attention mechanism RTD instead of MLM and an enhanced mask decoder to replace the softmax layer to solve "tug-of-war" problems.

3.3 Support Vector Regression

SVM[3] (Support vector machine) is a very efficient algorithm that tries to find the best decision boundary while also maximizing the margin. The introduction of kernel functions allows SVM to capture nonlinear patterns and thus achieve great performance in the real world. The basic idea of SVM is to push points away from the margin boundary and enlarge the width of the margin. The optimization problem for binary classification can be written as Eq. 1.

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^t w + C \sum_{i=1}^{m} \xi_i \\ \text{s.t.} \quad & y_i (w^t x_i + b) \geq 1 - \xi_i, \quad \forall i \\ & \xi_i \geq 0, \quad \forall i \end{aligned} \tag{1}$$

[2]https://www.kaggle.com/competitions/asap-aes
[3]https://www.kaggle.com/competitions/feedback-prize-2021/

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} w^t w + C \sum_{i=1}^{m} \xi_i + \xi_i^* \\ \text{s.t.} \quad & y_i - (w^t x_i + b) \geq \varepsilon - \xi_i, \quad \forall i \\ & (w^t x_i + b) - y_i \geq \varepsilon - \xi_i^*, \quad \forall i \\ & \xi_i \geq 0, \xi_i^* \geq 0 \quad \forall i, \varepsilon \geq 0 \end{aligned} \tag{2}$$

This simple but efficient assumption makes SVM also adapts to regression tasks. SVM for regression[10], or SVR, on the other hand, try to put points inside the margin boundary and also narrow the margin. The optimization problem can be written as Eq. 2.

3.4 Semi-supervised Learning based on SVM

Due to the great extensibility and performance of SVM algorithms, many papers are purposed to adapt SVM to semi-supervised learning (SSL) tasks. Transductive SVM (TSVM) [11] is one of the most popular methods for low-density separation, where it estimates labels for unlabeled data with max margin on both label and unlabeled data. Given that TSVM is np-hard, many studies focused on introducing new optimization methods or using graph techniques to tackle this problem, such as a Bayesian version of TSVM [1] and compact graph-based semi-supervised learning (CGSSL) [12].

Many studies focus on classification tasks, instead of regression. Among algorithms of semi-supervised regression based on SVM, COREG[13] is a co-training algorithm that adapts to self-training by using the k-NN regression model to co-train with SVR. S3VR[9], instead, modifies the optimization problem and constructs a new Lagrangian function to consider the label estimation of unlabeled data. S3VR based on self-training[6] (S3VR-SL) is one of the most well-developed algorithms that deploys two Probabilistic Local Reconstruction[7] (PLR) models to estimate a probability distribution for each unlabeled data. PLR_{local} will capture local information and PLR_{global} will capture more general information. S3VR-SL will estimate labels and choose useful data points as Eq. 4, based on the conjugated distribution as Eq. 3.

$$y_{conjugate} = \frac{\frac{y_{global}}{\sigma^2_{global}} + \frac{n y_{local}}{\sigma^2_{local}}}{\frac{1}{\sigma^2_{global}} + \frac{n}{\sigma^2_{local}}}, \sigma^2_{conjugate} = \frac{1}{\frac{1}{\sigma^2_{global}} + \frac{n}{\sigma^2_{local}}} \tag{3}$$

$$X_{hat} = X_{labeled} \cup (X_{unlabeled}, p_i \geq r), p_i = \frac{\sigma_i^2 - min(\sigma^2)}{max(\sigma^2) - min(\sigma^2)} \tag{4}$$

4 METHODOLOGY

There is no change in scope compared with our proposal. The only difference is that one of the teammates has left the team (now our team only have two students). With one labeled dataset and two unlabeled datasets, we cleaned and transformed the raw datasets mentioned in Chapter 3 and stored them as Numpy matrices for the following training and testing purposes. Since the inputs are texts, the NLP Transformer RoBERTa was introduced to help us embed the texts into vectors. The mean columnwise RMSE was developed as the metrics of our regression task. We then tested the performance of Support Vector Regression (SVR), which evolved

from the Support Vector Machine (SVM) model learned in Chapter 7, using only the labeled dataset. It would be our baseline model for the following optimization and tests. Then we implemented the S3VR based on self-training [6] as shown in Algo 1, where it uses two PLR[7] models to estimate the labels and select the training samples. The PLR models were inspired by the KNN, which was also learned in Chapter 7, but instead were used to estimate a probability distribution. In the future, we will implement our own SVR algorithm. We will test different NLP transformers and fine-tune the hyperparameters. Then we will review our codes and optimize the training performance.

Algorithm 1 S3VR based on self learning

procedure S3VR-SL($X_{labeled}, y, X_{unlabeled}$) For
  $N_{local}(y_{local}, \sigma^2_{local}) \leftarrow PLR_{local}(X, y, k_{local})$
  $N_{global}(y_{global}, \sigma^2_{global}) \leftarrow PLR_{global}(X, y, k_{global})$
  $y_{conjugate} = \frac{\frac{y_{global}}{\sigma^2_{global}} + \frac{n y_{local}}{\sigma^2_{local}}}{\frac{1}{\sigma^2_{global}} + \frac{n}{\sigma^2_{local}}}$
  $\sigma^2_{conjugate} = \frac{1}{\frac{1}{\sigma^2_{global}} + \frac{n}{\sigma^2_{local}}}$
  $p_i = \frac{\sigma_i^2 - min(\sigma^2)}{max(\sigma^2) - min(\sigma^2)}$
  $X_{hat} = X_{labeled} \cup (X_{unlabeled}, p_i \geq r)$
  $y_{hat} = y \cup (y_{conjugate}, p_i \geq r)$
  $model \leftarrow SVR(X_hat, y_hat)$
end procedure

5 CURRENT PROGRESS

So far, we have implemented most of the work except the SVR algorithm. We wrote our mean columnwise RMSE (MCRMSE) metrics. We deployed the Roberta model to embed NLP texts into vectors. For SVR, we temporarily introduce sklearn algorithm for test purposes. We gained the training MCRMSE 0.4452 as in Table 1. Then, we implemented the paper [6] to adapt the SVR model to semi-supervised tasks using two PLR models to estimate labels and pick valuable unlabeled data points. We have tested several combinations of hyperparameters as shown in Table 1. We can see it has very similar training loss with just SVR. The reason could be the hyperparameter combination is not good enough or maybe the unlabelled dataset increases the robustness (i.e. better in the testing dataset) but not decreases the training loss much.

6 PLAN OF WORK

In the following weeks, we will implement our own SVR algorithm, further improve the performance of the current model, and test a more state-of-the-art NLP transformer. Afterward, we will focus on collecting results and preparing for the final report.

By Nov 9th, we will optimize the current calculation performance using NumPy and implement most of the SVR algorithm except the optimization (we will expect to use the SMO algorithm to optimize). By Nov 16th, we will finish the SVR implementation and fine-tune the hyperparameters for SVR and two PLRs. By Nov 23rd, we will try to substitute the Roberta with Deberta-v3 for better embedding representations. It will also be our buffer time if we were behind the schedule. By Nov 30th, we will collect more experiment results. By Dec 7th, we will complete and submit the final report.

Table 1: Perfomance of some combinations of hyperparamters

Experiment	k_{local}	k_{global}	r	β	MCRMSE
SVR	None	None	None	None	0.4452
S3VR-1	5	10	0.5	1	0.4473
S3VR-2	5	10	0.5	10	0.4474
S3VR-3	5	10	0.8	1	0.4453
S3VR-4	5	10	0.8	10	0.4453
S3VR-5	5	20	0.5	1	0.4473
S3VR-6	5	20	0.5	10	0.4474
S3VR-7	5	20	0.8	1	0.4453
S3VR-8	5	20	0.8	10	0.4453
S3VR-9	10	10	0.5	1	0.4482
S3VR-10	10	10	0.5	10	0.4488
S3VR-11	10	10	0.8	1	0.4454
S3VR-12	10	10	0.8	10	0.4454
S3VR-13	10	20	0.5	1	0.4483
S3VR-14	10	20	0.5	10	0.4488
S3VR-15	10	20	0.8	1	0.4454
S3VR-16	10	20	0.8	10	0.4454

7 CONCLUSIONS

So far, we have completed the implementation of the S3VR-SL algorithm. We temporarily use sklearn SVM due to the time limitation. We have tested several hyperparameters but not found one with very great performance. In the following weeks, we will develop our own SVR algorithm and continuously optimize the performance, such as fine-tuning hyperparameters and change to DeBERTa-v3. We believe semi-supervised SVR is suitable for this task and we are very excited to develop our own semi-supervised SVR from scratch.

REFERENCES

[1] Sounak Chakraborty. 2011. Bayesian semi-supervised learning with support vector machine. *Statistical Methodology* 8, 1 (2011), 68–82.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, 4171–4186. https://doi.org/10.18653/v1/n19-1423

[3] Theodoros Evgeniou and Massimiliano Pontil. 2001. Support Vector Machines: Theory and Applications. In *Machine Learning and Its Applications, Advanced Lectures (Lecture Notes in Computer Science, Vol. 2049)*, Georgios Paliouras, Vangelis Karkaletsis, and Constantine D. Spyropoulos (Eds.). Springer, 249–257. https://doi.org/10.1007/3-540-44673-7_12

[4] Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. DeBERTaV3: Improving DeBERTa using ELECTRA-Style Pre-Training with Gradient-Disentangled Embedding Sharing. *CoRR* abs/2111.09543 (2021). arXiv:2111.09543 https://arxiv.org/abs/2111.09543

[5] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021. Deberta: decoding-Enhanced Bert with Disentangled Attention. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net. https://openreview.net/forum?id=XPZIaotutsD

[6] Pilsung Kang, Dongil Kim, and Sungzoon Cho. 2016. Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing. *Expert Syst. Appl.* 51 (2016), 85–106. https://doi.org/10.1016/j.eswa.2015.12.027

[7] Seung-kyung Lee, Pilsung Kang, and Sungzoon Cho. 2014. Probabilistic local reconstruction for k-NN regression and its application to virtual metrology in semiconductor manufacturing. *Neurocomputing* 131 (2014), 427–439. https://doi.org/10.1016/j.neucom.2013.10.001

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[9] Kyungha Seok. 2014. Semi-supervised regression based on support vector machine. *Journal of the Korean Data and Information Science Society* 25, 2 (2014), 447–454.

[10] Alexander J. Smola and Bernhard Schölkopf. 2004. A tutorial on support vector regression. *Stat. Comput.* 14, 3 (2004), 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88

[11] Vladimir Vapnik. 1998. *Statistical learning theory*. Wiley.

[12] Ming-Bo Zhao, Tommy W. S. Chow, Zhao Zhang, and Bing Li. 2015. Automatic image annotation via compact graph based semi-supervised learning. *Knowl. Based Syst.* 76 (2015), 148–165. https://doi.org/10.1016/j.knosys.2014.12.014

[13] Zhi-Hua Zhou and Ming Li. 2007. Semisupervised Regression with Cotraining-Style Algorithms. *IEEE Trans. Knowl. Data Eng.* 19, 11 (2007), 1479–1493. https://doi.org/10.1109/TKDE.2007.190644