REFNET: AUTOMATIC ESSAY SCORING BY PAIRWISE COMPARISON

Anonymous authors
Paper under double-blind review

ABSTRACT

Automatic Essay Scoring (AES) has been an active research area as it can greatly reduce the workload of teachers and prevents subjectivity bias . Most recent AES solutions apply deep neural network (DNN)-based models with regression, where the neural neural-based encoder learns an essay representation that helps differentiate among the essays and the corresponding essay score is inferred by a regressor. Such DNN approach usually requires a lot of expert-rated essays as training data in order to learn a good essay representation for accurate scoring. However, such data is usually expensive and thus is sparse. Inspired by the observation that human usually scores an essay by comparing it with some references, we propose a Siamese framework called *Referee Network (RefNet)* which allows the model to compare the quality of two essays by capturing the relative features that can differentiate the essay pair. The proposed framework can be applied as an extension to regression models as it can capture additional relative features on top of internal information. Moreover, it intrinsically augments the data by pairing thus is ideal for handling data sparsity. Experiment shows that our framework can significantly improve the existing regression models and achieve acceptable performance even when the training data is greatly reduced.

1 INTRODUCTION

Automatic Essay Scoring (AES) is the technique to automatically score an essay over some specific marking scale. AES has been an eye-catching problem in machine learning due to its promising application in education. It can free tremendous amount of repetitive labour, boosting the efficiency of educators. Apart from automation, computers also prevail human beings in consistency, thus eliminate subjectivity and improve fairness in scoring.

Attempts in AES started as early as Project Essay Grade (PEG) (Page, 1967; 2003), when the most prevalent methods relied on hand-crafted features engineered by human experts. Recent advances in neural networks bring new possibilities to AES. Several related works leveraged neural networks and achieved decent results (Dong et al., 2017; Taghipour & Ng, 2016; Tay et al., 2018; Liu et al., 2019). As is shown in Figure 1, these approaches generally follow the 'representation + regression' scheme where a neural network reads in the text embeddings and generates a high level representation that will be fed to some regression model for a score. However, such model requires a large amount of expert-rated essays for training. In reality, collecting such dataset is expensive. Therefore, data sparsity remains a knotty problem to be solved.

Inspired by the observation that human raters usually score an essay by comparing it to a set of references, we propose to leverage the pairwise comparisons for scoring instead of regression. The goal of the model is shifted from predicting the score directly to comparing two essays, and the final score will be determined by comparing new essays with known samples. In order to achieve this, we designed a Siamese network called *Referee Network* (RefNet) and corresponding scoring algorithms. RefNet is a framework so that it can use various representation encoders as backbones. What's more, though this model is designed to capture mutual features, it can also benefit from essay internal information via transfer learning.

Scoring essays by comparison has various benefits. First, *RefNet* is incredibly strong in dealing with data sparsity problem. Essays are paired with each other to form the training data for RefNet, which significantly augmented the data size. Experiments show that our model achieve acceptable

performance even when the training data is radically reduced, while regression models are subject to drastic performance degeneration. Second, unlike end-to-end black-box models, our system scores an essay by comparing it with a set of labeled anchors, providing transparency to a certain degree during inference process. Last but not least, with information in both internal and mutual perspective, RefNet can have better insight into the quality of essays.

Our contributions can be summarized as follows:

- We designed *Referee Network (RefNet)*, a simple but effective model to compare two essays, and Majority Probability Voting Algorithm to infer the score from pairwise comparison results. To the best of our knowledge, it is the first time a Siamese neutral network is used in AES.

- Our model intrinsically solves the problem of data sparsity. It achieves acceptable performance even when the training data is greatly reduced, while regression models are impaired a lot. Its efficacy in few-shot learning makes it an ideal solution for real applications where labelled data is usually limited.

- *RefNet* exploits a new realm of information, mutual relationship, by pairwise comparison. With transfer learning, it also leverages internal features captured by regression. Moreover, *RefNet* can be applied as an extension to various regression models and consistently improve the performance.

2 Related Works

Existing AES solutions fall into two categories depending on how essays are represented: feature-engineered models and end-to-end models.

In Yannakoudakis et al. (2011), 9 handcrafted features are fed into a Support Vector Machine(SVM). Those features range from simple ones like script length to elaborated ones like phrase structure rules and grammatical relation distance measure. However, no matter how many features are used, it cannot develop expressive representations of the essays. Besides, extracting handcrafted features often relies on other models, resulting in highly coupled systems that are slow and sophisticated.

Recent models based on neural networks can capture the features automatically. Taghipour & Ng (2016) uses a self-trained look-up table for embedding and investigates a variety of neural networks, among which LSTM yields the best performance. Several variations of neural network were also explored: Dong et al. (2017) applies CNN with attention pooling on sentence level and LSTM with attention pooling on essay level. Tay et al. (2018) attached LSTM to a special network architecture called SkipFlow, which models the relationship between snapshots of hidden states of LSTM as it reads. All of these methods achieved decent results, which proved the effectiveness of neural network in this problem. However, several problems remind to be solved. The first one is that no significant improvement over vanilla ones has been observed. The second problem is the scarcity of labeled data casts doubt on whether elaboration of neural networks has reached its bottleneck.

A stark contrast to researchers' delicate designs in generating better representations is the rudimentary regression methods used to obtain the final prediction. Of course, a regression layer satisfies the fundamental need of this task: mapping a representation to a scalar. However, considering how complicated the relationship between essay contents and its score can be, the capability of such simple method is questionable. Pairwise difference between essays, on the contrary, is intuitively more understandable and, according to Yannakoudakis et al. (2011), outperforms mere regression in experiment. Instead of directly mapping to the grade, learning rank preference is used to explicitly model the grade relationships between essays. Yannakoudakis et al. used a special kind of SVM which outputs a real number with the ϵ-intensive loss function. However, its only distinction from regression is including the degree of misclassification as part of the loss function so that the SVM can maximize the difference between closely-ranked data pairs. It is more an amendment for regression than a new approach and cannot fully release the power the of comparison based methods.

3 METHODS

In the last section, we argue that the potential of ranking preference methods in AES has not yet been fully exploited. In this paper, we will try to exhaust the information in representations using a novel approach. As depicted in Figure 1, to score an essay in our system, *Referee Network* will compare it with known samples. Then probability majority voting will be used to infer the score from pairwise rank preference. Details will be elaborated in this section.

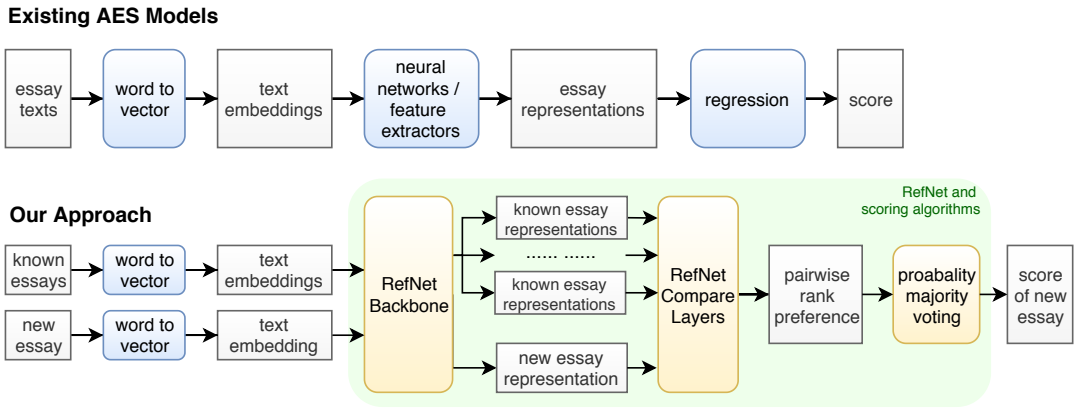

Figure 1: The workflow of our approach and comparison with existing AES models

3.1 SENTENCE EMBEDDINGS

We use the same scheme as Liu et al. (2019) to obtain the embeddings at sentence level. It is proven that Transformers, an attention mechanism that learns the contextual relations between words, can effectively make use of the textual information. The model Bidirectional Encoder Representations from Transformers (BERT) which is built on this this approach achieved state-of-the-art results on various downstream tasks such as reading comprehension (Devlin et al., 2018). Therefore, we directly retrieve the word embeddings from pre-trained BERT model.

For a sentence with l words, BERT will output a set of low-dimensional representations $s = \{w_0, w_1, ..., w_l, w_{l+1}\}$, where $w_i (1 \leq i \leq l)$ denotes the embedding for the i^{th} word, w_0 is a special tag CLS for classification tasks and w_{l+1} is a SEP tag for separating sentences. We use the average of the hidden states of the penultimate layer along the time axis to represent a sentence:

$$s = \frac{1}{n+2} \sum_{i=0}^{l+1} w_i^{-2} \tag{1}$$

Here the superscript -2 means that the embedding is retrieved from the second last layer in BERT. We use the penultimate layer instead of the final one because according to Liu et al. (2019), the representations in the final layer fit too closely to the pre-training tasks including masked language modelling and next sentence prediction. Embeddings in the penultimate layer are flexible enough for the model to fit to our AES task while at the same time semantically meaningful.

3.2 THE REFEREE NETWORK (REFNET)

3.2.1 NETWORK ARCHITECTURE

As is mentioned above, we will not built a model that directly infers the score via any kind of regression. Instead, we built a model that compares essays, i.e takes two essays as input and outputs the rank preference. In order to achieve this goal, we borrowed the idea from *Siamese Network* Koch et al. (2015) and designed the *Referee Network* (RefNet). As shown in Figure 2, RefNet is actually embarrassingly simple. The pair of essays, which are both in the form of a set of sentences embeddings, will be encoded into essay representations through a backbone network. The backbone network can be anything that outputs a vector from a matrix with certain dimension, in our case, the size of BERT outputs. In this paper, we will try the following backbones:

- Average Embeddings. For each input essay $e = \{s_1, s_2, s_3, ..., s_m\}$, it will just compute the average of all sentences $r = (\sum_{i=1}^{m} s_i)/m$ as representation.
- One layer of Simple Recurrent Neural Networks (RNN).
- One layer of Long Short-Term Memory (LSTM).

Due to the prevalence of RNN and LSTM, we shall skip their descriptions here. Details of these two network architectures can be found in Sherstinsky (2018). Note that for a pair of essays $[e^1, e^2] \in \mathbb{R}^{2m \times d}$, the RNN and LSTM backbones will generate two sets of hidden states $\{\{h_1^1, h_2^1, ..., h_m^1\}, \{h_1^2, h_2^2, ..., h_m^2\}\}$, and we will use only use the state at the final time step because its superiority over average state over time has been presented by Liu et al. (2019). Then the representations of the two essays will be concatenated along their first dimension $r = [h_m^1, h_m^2]$. After that, one fully connected layer with leaky linear rectifier activation will be applied to extract higher level features. Finally, another fully connected layer with two units together with softmax activation will give a rank preference, indicating with essay is better written:

$$y = softmax(W_2 \cdot ReLU(W_1 \cdot r + b_1) + b_2) \in \mathbb{R}^2 \tag{2}$$

In this paper, we denote RefNet as $R(e_i, e_j)$ in mathematical formulas. To keep things simple, we simplify the output of $R(e_i, e_j)$ as a scalar in [0,1], indicating the probability that essay e_i is better written than e_j.

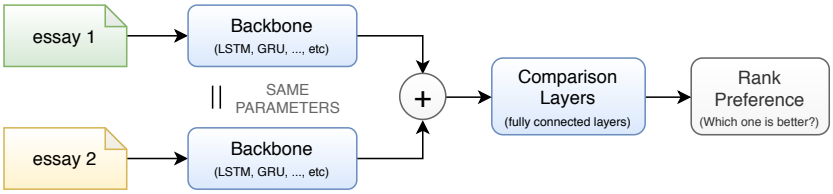

Figure 2: Architecture of *Referee Network*

3.2.2 TRAINING

To train RefNet, we firstly pair each essay with those with different scores within the same prompt to form essay pairs. Pairing is not conducted across prompts because essays from different prompts are not really comparable due to disparate writing requirements and scoring scales. We do not pair the essays with the same score because one essay can hardly be exactly as good as another: among identically scores essays, one can further distinguish one may be better than another. What's more, the inconsistent scoring schemes make concept of equal quality even more vague. In the ASAP dataset, for instance, the scale can be as wide as 0-60 or as narrow as 0-3. As a result, two essays with the same score in 0-3 scale may have drastically different scores in 0-60 scale. Therefore, it is foreseeable that requiring RefNet to categorize identically rated essays will frustrate the model during training and impair the performance. After pairing, RefNet is trained by minimizing the cross entropy between model output and true values.

3.2.3 TRANSFER LEARNING

We hope to make full advantage of the essay representations by exploiting both their internal and mutual information. And what enables RefNet to use internal information is transfer learning. As Figure 3 presents, we firstly train the backbone network with conventional end-to-end schemes where the model outputs the score directly by regression. After being trained in this pseudo task, the parameters in the LSTM are transferred to RefNet and fine tuned for essay comparison, which is the real task. In this way, what is learned in regression is more or less retained in the comparison model.

3.3 SCORING

After comparing the test essay against known samples, we designed an algorithm called probability majority voting to infer its final score. Suppose there are n_i anchors for score notch N_i. Each anchor is paired with the testing input x and fed into our referee network for comparison. By taking the average referee output over all the anchors, we get the probability of how likely that the essays at

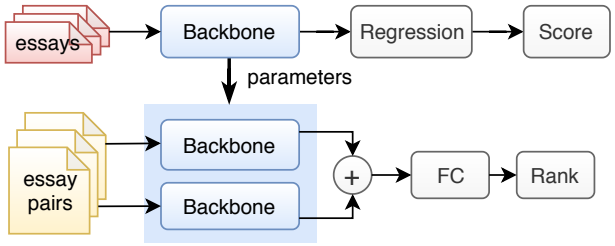

Figure 3: Transfer Learning

this score notch is better than the input:

$$p_i = \frac{1}{n_i} \sum_{j=1}^{n_i} R(e_j, x) \tag{3}$$

Two aspects are considered to form the voting criteria. First, according to the model, the test input is inferior than p_i percent of anchors with score label N_i. Intuitively, if we choose the anchors with higher scores, the test essay will also be beaten. Therefore, depending on how large p_i is, it will more or less add to the likelihood that $[N_{min}, N_i)$ contains the correct answer. In specific, votes of value p_i will be given to all the scores in $[N_{min}, N_i)$. Similarly, the test input is also superior than $1-p_i$ percent of anchors with score label N_i. Thus, a vote with value $1-p_i$ is added to $(N_i, N_{max}]$. The second consideration is that we hope the p for the predicted score is close to 0.5, as one score does not appear correct if most of its anchors are better or worse than the input essay. Therefore, we will penalize the distance from 0.5 by subtracting $|p_i - 0.5|$ from the notch's own votes.

Formally, total pairwise comparison result for some essay is $\{N_1 : p_1, N_2 : p_2, ..., N_k : p_k\}$, where k is the total number of score notches. Without loss of generality, we assume that $N_1 < N_2 < ... < N_k$. The total votes for score notch N_i is:

$$V(N_i) = \sum_{j=1}^{i-1}(1 - p_j) + \sum_{j=i+1}^{k} p_j - |p_i - 0.5| \tag{4}$$

Our final prediction is naturally the score notch with the highest votes. If more than one scores are the highest, we will take the average of all such scores and round to the nearest integer:

$$Score = avg(\arg\max_{N_i} V(N_i)) \tag{5}$$

4 Experiments

In this section, we will firstly present some background information regarding the dataset and performance metrics. Then we will show the performance of our model in two different tasks and compare with other models. After that, we will conduct ablation studies analyze the results.

4.1 Dataset and Evaluation Metrics

We use Automated Student Assessment Prize(ASAP) dataset[1], which is the standard dataset for developing and evaluating AES systems. It is composed of 12976 essays in 8 prompts (see Table 1 for detailed statistics), where prompt 1,2,7,8 are persuasive, narrative or expository essays while the rest are Source Dependent Response questions. In our experiments, we use the same 5-fold split as Taghipour & Ng (2016) did for fair comparisons. In this data split scheme, 60% of data is used for training, 20% for validation and the rest for testing.

Similar to (Dong et al., 2017; Taghipour & Ng, 2016; Tay et al., 2018; Liu et al., 2019), we use Quadratic Weighted Kappa (QWK), which is the official standard for ASAP dataset, as our evaluation metric. The QWKs are computed for each prompt in its original scale respectively. All the experiments are conducted over all 5 folds and the the average of each prompt's QWKs over all folds is calculated as the performance measure for the model.

[1]https://www.kaggle.com/c/asap-aes

Table 1: Statistics of ASAP dataset

Prompt ID	# Essays	Avg Length	Score Range
1	1783	350	2-12
2	1800	350	1-6
3	1726	150	0-3
4	1772	150	0-3
5	1805	150	0-4
6	1800	150	0-4
7	1569	250	0-30
8	723	650	0-60

4.2 TASK 1: TRAINING ON THE WHOLE DATASET

We firstly trained our model on the whole dataset and observed that RefNet tends to overfit when all the training data is used. It is not supervising as the total training data is amplified by approximately 300 times after pairing, which means that the model will see each essay around 300 times within each epoch. As a result, the model can overfit before the first epoch is finished.

To solve this problem, training samples are dropped randomly before training. To minimize the potentially negative impact, we set up different dropping rate for different essay pairs. As Figure 4 suggests, when training on the whole dataset, RefNet can easily give accurate comparisons between two essays with contrasting scores. However, distinguishing the pairs that have similar scores is much more challenging. Therefore, the pairs with larger score difference are more likely to be dropped while those with smaller difference is more likely to be kept. Different dropping rate are shown in Table 4.2. After the data adjustment, approximately only 15% of training set are kept.

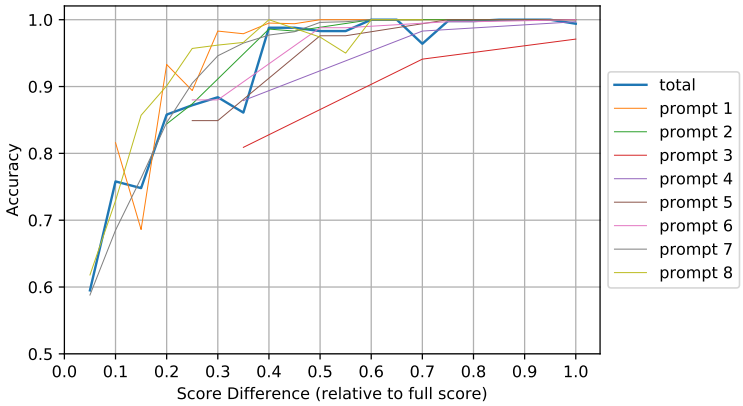

Figure 4: The model accuracy for the essays pairs with different score difference. The score difference here is relative to the scoring range of that prompt.

Table 2: Keep Rate for Different Essay Pairs

Score Difference Interval	Keep Rate
$(0, 0.1]$	0.25
$(0.1, 0.2]$	0.2
$(0.2, 0.4]$	0.15
$(0.4, 1]$	0.1

The first three blocks of Table 3 compares the of regression and RefNet with the same backbone. One can clearly see that RefNet constantly outperforms conventional regression approaches. Notice that our method offers the greatest improvement in for RNN backbone, which is the weakest of the three. That is because scoring by comparison is not very sensitive to the quality of representations. As long as the representation makes sense, RefNet is able to boost the performance to a high level.

We also compare our model with the ensembled neural networks in Taghipour & Ng (2016) and SkipFlow LSTM networks proposed by Tay et al. (2018). The results show that we achieved the state-of-the-art average QWK score, and our edge is particularly large in prompt 8 which has much less data but more score notches than other prompts.

Table 3: Performance of different backbones in regression and RefNet, performance of typical existing AES models and human raters

Methods	Prompts								Average
	1	2	3	4	5	6	7	8	
Average (regression)	0.771	0.622	0.643	0.662	0.752	0.702	0.763	0.555	0.684
Average (RefNet)	0.822	0.694	0.664	0.789	0.800	0.802	0.802	0.727	0.762
Improvement	6.6%	11.5%	3.3%	19.1%	6.3%	14.2%	5.1%	30.1%	11.5%
RNN (regression)	0.622	0.530	0.618	0.747	0.711	0.696	0.700	0.288	0.614
RNN (RefNet)	0.791	0.672	0.665	0.782	0.782	0.783	0.790	0.627	0.737
Improvement	27.2%	26.8%	7.6%	4.7%	10.0%	12.5%	12.9%	117.7%	20.0%
LSTM (regression)	0.802	0.674	0.660	0.770	0.780	0.784	0.772	0.592	0.729
LSTM (RefNet)	0.822	0.695	0.676	0.799	0.800	0.811	0.807	0.711	**0.765**
Improvement	2.5%	3.1%	2.4%	3.8%	4.5%	2.6%	4.5%	16.7%	4.9%
10×CNN	0.804	0.656	0.637	0.762	0.752	0.765	0.750	0.680	0.726
10×LSTM	0.808	0.697	0.689	0.805	0.818	0.827	0.811	0.598	0.756
10×(CNN+LSTM)	0.821	0.688	0.694	0.805	0.807	0.819	0.808	0.644	0.761
SkipFlow (Bilinear)	0.830	0.678	0.677	0.778	0.795	0.807	0.790	0.670	0.753
SkipFlow (Tensor)	0.832	0.684	0.695	0.788	0.815	0.810	0.800	0.697	0.764
Human Raters	0.721	0.814	0.769	0.851	0.753	0.776	0.721	0.624	0.754

4.3 TASK 2: FEW-SHOT LEARNING

In this task, we created two mini-ASAP datasets by excerpting 25% and 10% of essays from the original training and validation dataset respectively while testing data remains the same. Noticing that the essays under each prompt are distributed evenly in the original train test split, we gave special attention to this issue and our mini-ASAP dataset is balanced among different prompts. And we will not use the drop training data as the excessive data problem no longer exists.

Table 4: QWKs of RefNet with different backbones in mini-ASAP dataset

Backbones	Data Size	Approach	Average QWK	Degeneration[1]	Improvement[2]
Average	25%	regression	0.650	5.0%	34.0%
		RefNet	0.737	3.3%	
	10%	regression	0.606	11.4%	32.5%
		RefNet	0.703	7.7%	
RNN	25%	regression	0.525	14.5%	68.3%
		RefNet	0.703	4.6%	
	10%	regression	0.493	24.5%	45.7%
		RefNet	0.639	13.3%	
LSTM	25%	regression	0.623	14.5%	75.9%
		RefNet	0.738	3.5%	
	10%	regression	0.542	25.7%	75.1%
		RefNet	0.716	6.4%	

[1] The percentage of QWK score decrease compared to full dataset scenario
[2] How much the degeneration of RefNet is less than that of regression method

From Table 4 we can see that with regression, the model will suffer from major performance degradation when the training data is reduced. On the contrary, RefNet is much more robust to scarce data. Even after 90% of data is dropped, RefNet can still offer high quality predictions.

4.4 ABLATION STUDIES

Several ablation tests are conducted to study the effects of individual components. In the first experiment, we remove the transfer learning and try to train RefNet from starch to test its efficacy. In the second experiment, data adjustment is disabled and RefNet is trained on the whole dataset. Finally, we hope to compare the performance of regression and pairwise methods under the same representations. We fix the transferred backbone parameters and train the fully connected comparison layers in RefNet only and see how QWK varies. All the ablation tests use LSTM as backbone.

Table 5: Results of Ablation Studies

Ablation	No transfer learning	No data adjustment	No fine tuning
QWK Score	0.746	0.758	0.754
Degeneration	-2.48%	-0.92%	-1.44%

4.5 Analysis

4.5.1 Regression vs Pairwise Ranking

RefNet has multiple advantages over regression. First, it leverages the mutual information between essays. Notice that in Table 5, the pairwise ranking approach achieved 0.754 on the fixed embeddings learnt from regression task, which is higher than pure regression performance 0.729 in Table 3. It shows that by taking mutual information into account, scoring by pairwise comparison can consistently improve the performance.

Second, internal information is also exploited by transfer learning. With information from both internal and mutual perspective, RefNet can have better insight into the quality of essays. Besides, considering the complexity of the task, it is hard to train RefNet from scratch. Results in Table 5 show that the performance of RefNet without transfer learning degenerates by 2.48%.

Third, RefNet is naturally invulnerable to cross-prompt noise. In reality, dataset of a single prompt is rarely large enough, so hybrid dataset such as ASAP dataset are commonly used. However, the hybrid dataset consisting of essays of different scoring range, written by students at different levels or with different backgrounds may be not self-consistent. For regression, the scores from different score range should be carefully aligned before training on the whole dataset because end-to-end method is sensitive to labels. However, such alignment can hardly be achieved. Current works just linearly rescale whatever original score range to [0,1], bringing noise to the system. In contrast, RefNet sees only the relative relation between essays within the same prompt, avoiding possible cross-prompt noise.

4.5.2 Few-shot Learning

RefNet shows even larger edge over regression in few-shot learning problems. On one hand, RefNet is intrinsically suitable for few-shot learning problems as the pairing operation can amplify the training data by one or multiple levels of magnitude. What's more, unlike some common data augmentation techniques such as random transforms, no noise is introduced in pairing.

On the other hand, the 'representation + regression' mechanism is highly data demanding. First, though both approaches needs to somehow extract the features from texts, predicting by regression imposes higher requirement in expressing those features in a numeric and explicit form. Second, since basic models such as LSTM and the features that can be learnt by those models have been well exploited, researchers may have to resort to deeper and more elaborated features to push the performance, resulting in more complicated models with more parameters. Both factors makes end-to-end model unable to be fully trained in small datasets.

5 Conclusion

In this paper we present *Referee Network*, a framework for automatic essay scoring using pairwise comparisons. We demonstrate that RefNet is expert in solving data sparsity problems. It can retain the performance at a high level even when the training data is significantly reduced, which outperforms regression models by a significant margin. We also show that RefNet can improve conventional regression models by leveraging the additional mutual information between representations. With only vanilla backbones, our model is able to obtain state-of-the-art results. Even if the essay representations are fixed and have mediocre quality, our model can still boost up the scoring accuracy.

Furthermore, the capacity of RefNet can go far beyond this context as it is an extendable framework that can be used with any kind of representation encoders. Besides the simple backbones we tried in this paper, one can by all means utilize more complicated and better performing models as backbones. In this way, the performance of AES systems can always be pushed to a new record.

REFERENCES

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Fei Dong, Yue Zhang, and Jie Yang. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 153–162, 2017.

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, 2015.

Jiawei Liu, Yang Xu, and Lingzhe Zhao. Automated essay scoring based on two-stage learning. *arXiv preprint arXiv:1901.07744*, 2019.

Ellis B Page. Grading essays by computer: Progress report. In *Proceedings of the invitational Conference on Testing Problems*, 1967.

Ellis Batten Page. Project essay grade: Peg. 2003.

Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *arXiv preprint arXiv:1808.03314*, 2018.

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*, pp. 1882–1891, 2016.

Yi Tay, Minh C Phan, Luu Anh Tuan, and Siu Cheung Hui. Skipflow: incorporating neural coherence features for end-to-end automatic text scoring. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pp. 180–189. Association for Computational Linguistics, 2011.