Learning from Graph Propagation via Ordinal Distillation for One-Shot Automated Essay Scoring

Zhiwei Jiang*†
jzw@nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Meng Liu*
mf1933061@smail.nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Yafeng Yin
yafeng@nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Hua Yu
huayu.yh@smail.nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Zifeng Cheng
chengzf@smail.nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

Qing Gu
guq@nju.edu.cn
State Key Laboratory for Novel Software Technology, Nanjing University
Nanjing, China

ABSTRACT

One-shot automated essay scoring (AES) aims to assign scores to a set of essays written specific to a certain prompt, with only one manually scored essay per distinct score. Compared to the previous-studied prompt-specific AES which usually requires a large number of manually scored essays for model training (e.g., about 600 manually scored essays out of totally 1000 essays), one-shot AES can greatly reduce the workload of manual scoring. In this paper, we propose a Transductive Graph-based Ordinal Distillation (TGOD) framework to tackle the task of one-shot AES. Specifically, we design a transductive graph-based model as a teacher model to generate pseudo labels of unlabeled essays based on the one-shot labeled essays. Then, we distill the knowledge in the teacher model into a neural student model by learning from the high confidence pseudo labels. Different from the general knowledge distillation, we propose an ordinal-aware unimodal distillation which makes a unimodal distribution constraint on the output of student model, to tolerate the minor errors existed in pseudo labels. Experimental results on the public dataset ASAP show that TGOD can improve the performance of existing neural AES models under the one-shot AES setting and achieve an acceptable average QWK of 0.69.

CCS CONCEPTS

• **Computing methodologies** → **Natural language processing**;
• **Information systems** → **Clustering and classification**.

*Both authors contributed equally to this research.
†Corresponding author.

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3450017

KEYWORDS

Essay Scoring, One-Shot, Graph Propagation, Ordinal Distillation

ACM Reference Format:
Zhiwei Jiang, Meng Liu, Yafeng Yin, Hua Yu, Zifeng Cheng, and Qing Gu. 2021. Learning from Graph Propagation via Ordinal Distillation for One-Shot Automated Essay Scoring. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442381.3450017

1 INTRODUCTION

Automated Essay Scoring (AES) aims to summarize the quality of a student essay with a score or grade based on the factors such as grammaticality, organization, and coherence. It is commercially valuable to be able to automate the scoring of millions of essays. In fact, AES has been developed and deployed in large-scale standardized tests such as TOEFL, GMAT, and GRE [2]. Besides evaluating the quality of essays, as an evaluation technique of text quality, AES can also be used conveniently to evaluate the quality of various Web texts (e.g., news, responses, and posts).

Research on automated essay scoring has spanned the last 50 years [25], and still continues to draw a lot of attention in the natural language processing community [17]. Traditional AES methods mainly rely on various handcrafted-features and score essays based on regression methods [2, 19, 26, 32, 48]. Recently, with the development of deep learning technology, many models based on LSTM and CNN have been proposed [7, 8, 10, 39, 41]. These models can automatically learn the features of essays and achieve better performance than traditional methods.

However, to train an effective neural AES model, it often needs a large number of manually scored essays for model training (e.g., about 600 manually scored essays out of totally 1000 essays in a test), which is labor intensive. This limits its application in some real-world scenarios. To this end, some recent work considers using the scored essays under other prompts (i.e., topic of writing essay) to alleviate the burden of manual scoring under target prompt. But due to the difference among prompts such as genre, score range,

topic, and difficulty, these cross-prompt methods often perform worse than the prompt-specific methods [9]. Tackling the domain adaptation among prompts is a challenging problem and there are some recent studies focusing on this line of work [5, 14].

In this paper, we consider another way without using data from other prompts. Given a set of essays towards a target prompt, we consider if we can score all essays only based on a few manually scored essays. Extremely, we consider the one-shot scenario, that is, only one manually scored essay per distinct score is given. In practical writing tests, scoring staff usually evaluates the essays by first designing a criteria specific to the current test and then applying the criteria for essays scoring. To alleviate the burden of scoring staff, we expect to firstly let the scoring staff express the criteria by one-shot manual scoring, and then use a specially-designed AES model to scoring the rest essays based on the one-shot data.

One-shot AES is a challenging task, since the one-shot labeled data is insufficient to train an effective neural AES model. To solve this problem, our intuition is whether we can augment the one-shot labeled data with some pseudo labeled data, and then perform model training on the augmented labeled data. Obviously, there are two challenges: one is how to acquire the pseudo labeled data, and the other is how to alleviate the disturbance brought by error pseudo labels during model training.

To this end, we propose a Transductive Graph-based Ordinal Distillation (TGOD) framework for one-shot automated essay scoring, which is designed based on a teacher-student mechanism (i.e., knowledge distillation) [13]. Specifically, we employ a transductive graph-based model [52, 53] as the teacher model to generate pseudo labels, and then train the neural AES model (student model) by combining the pseudo labels and one-shot labels. Considering that there may exist many error labels among the pseudo labels, we select the pseudo labels with high confidence to improve the quality of pseudo labels. Besides, considering that the score is at ordinal scale and an essay is easily to be assigned a score near its ground-truth score (e.g., 3 is easily to be predicted as 2 or 4), we proposed an ordinal-aware unimodal distillation strategy to tolerate some pseudo labels with minor errors.

The major contributions of this paper are summarized as follows:

- For the one-shot automated essay scoring, we propose a distillation framework based on graph propagation, which alleviates the requirement of supervised neural AES model on labeled data by utilizing unsupervised data.
- We propose the label selection and the ordinal-aware unimodal distillation strategies to alleviate the effect of error pseudo labels on the final AES model.
- The TGOD framework has no limitation on the architecture of student model, thus can be applied to many existing neural AES models. Experimental results on the public dataset demonstrate that our framework can effectively improve the performance of several classical neural AES models under the one-shot AES setting.

2 PROBLEM DEFINITION

We first introduce some notation and formalize the one-shot automated essay scoring (AES) problem. Let $\mathcal{X} = \{x_i\}_{i=1}^{N}$ denote a set of essays written to a certain prompt, $\mathcal{Y} = \{1, 2, ..., K\}$ denote a set of pre-defined scores (labels) at ordinal scale, and (x, y) denote an essay and its ground-truth score (label) respectively. For one-shot AES, we assume that we are given a set of one-shot labeled data $\mathcal{D}_o = \{(x_i, y_i = i)\}_{i=1}^{K}$, where the set $\mathcal{X}_o = \{x_i | (x_i, y_i) \in \mathcal{D}_o\}$ is a subset of $\mathcal{X}$ (i.e., $\mathcal{X}_o \in \mathcal{X}$), and the essay $x \in \mathcal{X}_o$ with $y = i$ is the one-shot essay for the distinct score (label) $i \in \mathcal{Y}$. Apart from the one-shot labeled essays $\mathcal{X}_o$, the rest essays in $\mathcal{X}$ constitute the unlabeled essay set $\mathcal{X}_u = \{x_i\}_{i=1}^{N_u}$, and thus $\mathcal{X}_u \cup \mathcal{X}_o = \mathcal{X}$. The goal of one-shot AES is to learn a function F to predict the scores (labels) of the unlabelled essays $x \in \mathcal{X}_u$, based on the one-shot labeled data $\mathcal{D}_o$ and essay set $\mathcal{X}$, by

$$\hat{y} = F(x; \mathcal{D}_o, \mathcal{X}). \tag{1}$$

Typical AES approaches based on supervised learning would remove $\mathcal{X}$ and replace $\mathcal{D}_o$ with a statistic $\theta^* = \theta^*(\mathcal{D}_o)$ in Eq. 1, since they can usually learn a sufficient statistic θ^* for prediction $p_{\theta^*}(y|x)$ only based on labeled data $\mathcal{D}_o$. However, the one-shot setting is never the case, since only few labeled data is given in $\mathcal{D}_o$, which is insufficient to train a statistic θ^* with good generalization. We therefore exploit both the one-shot labeled data $\mathcal{D}_o$ and the unlabeled essays $\mathcal{X}_u \in \mathcal{X}$ to learn the prediction function F, and thus adopt the more general form of F in Eq. 1.

3 THE TGOD FRAMEWORK

In this section, we introduce the proposed TGOD framework, followed by its technical details.

3.1 An Overview of TGOD

TGOD is designed based on the teacher-student mechanism. It can enable a supervised neural student model to benefit from a semi-supervised teacher model under the one-shot essay scoring setting. While the one-shot labeled data is insufficient to train the supervised neural student model, the student model can be trained by distilling the knowledge of the semi-supervised teacher model on the unlabeled essays. Through a specially-designed ordinal distillation strategy, the supervised neural student model can even outperform the semi-supervised teacher model.

Specifically, as shown in Figure 1, TGOD contains three main components: the *Teacher Model* which exploits the manifold structure among labeled and unlabeled essays based on graphs and generates pseudo labels of unlabeled essays for distillation; the *Student Model* which tackles the essay scoring problem as an ordinal classification problem and makes a unimodal distribution prediction for essays; the *Ordinal Distillation* which distills the unimodal smoothed *Teacher Model*'s outputs into the *Student Model*. In the following, we introduce these components of TGOD with technical details.

3.2 Graph-Based Label Propagation (Teacher)

We introduce the *Teacher Model* illustrated in Figure 1, which is a graph-based label propagation model and consists of three components: multiple graph construction that models the relationship among essays from multiple aspects; label propagation that spreads

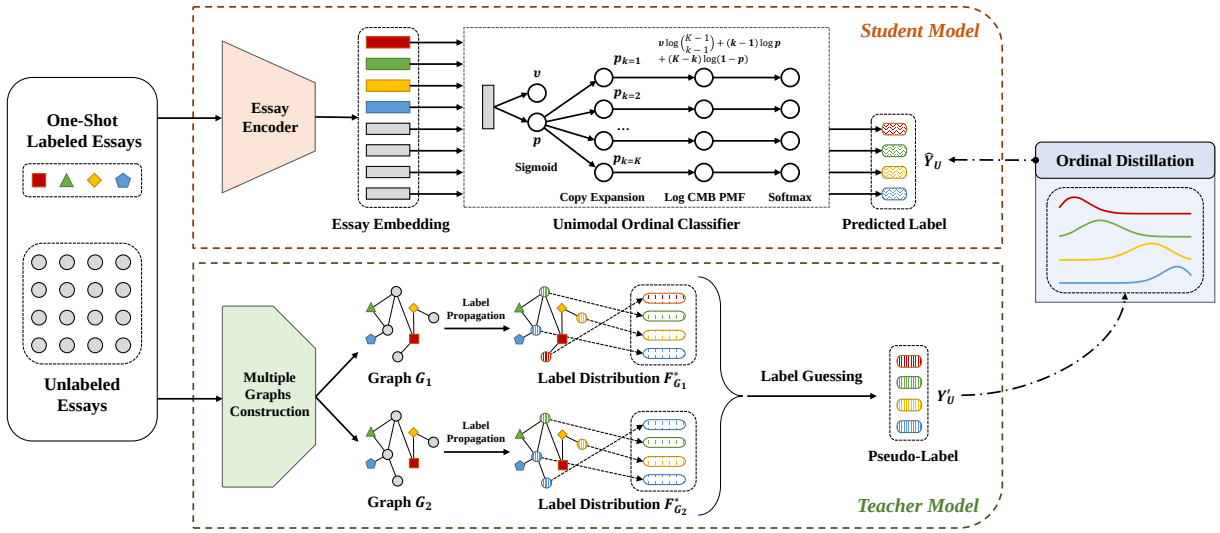

Figure 1: Architecture of the Transductive Graph-Based Ordinal Distillation (TGOD) framework.

labels from the one-shot essays to the unlabeled essays; label guessing that generates the pseudo labels of unlabeled essays from results of multiple graph propagation.

3.2.1 ***Multiple Graphs Construction.*** To construct a graph on the essay set $\mathcal{X}$, we need first to extract the feature embedding of each essay $x_i \in \mathcal{X}$. Specifically, we employ an embedding layer followed by a mean pooling layer as the essay encoder $f_e(\cdot)$ to extract the feature embedding $f_e(x_i)$ of essay x_i.

Based on the feature embedding of essays, we then construct a neighborhood graph $G = (V, E, W)$ for the essay set $\mathcal{X}$, where $V = \mathcal{X}$ denotes the node set, E denotes the edge set, and W denotes the adjacent matrix. To construct an appropriate graph, we employ the Gaussian kernel function [53] to calculate the adjacent matrix W:

$$W_{ij} = \exp\left(-\frac{d\left(f_e(x_i), f_e(x_j)\right)}{2\sigma^2}\right), \tag{2}$$

where $d(\cdot,\cdot)$ is a distance measure (e.g., Euclidean distance) and σ is a length scale parameter.

To construct a k-nearest neighbor graph, we only keep the k-max values in each row of W, and then apply the normalized graph Laplacians [6] on W:

$$S = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}, \tag{3}$$

where D is a diagonal matrix with its (i,i)-value to be the sum of the i-th row of W.

While using different pre-trained word embeddings as the embedding layer may result in different k-nearest neighbor graphs, we can construct B graphs by using B types of pre-trained word embeddings (e.g., Word2Vec [20], GloVe [28], ELMo [31], BERT [43]).

3.2.2 ***Label Propagation.*** We now describe how to get predictions for the unlabeled essays set $\mathcal{X}_u$ using label propagation [23].

Let $\mathcal{F}$ denote the set of $N \times K$ sized matrix with nonnegative entries. We define a label matrix $Y \in \mathcal{F}$ with $Y_{ij} = 1$ if x_i is from the one-shot essays $\mathcal{X}_o$ and labeled as $y_i = j$, otherwise $Y_{ij} = 0$. Starting from Y, label propagation iteratively determines the unknown labels of essays in $\mathcal{X}_u$ according to the graph structure using the following formulation:

$$F^{t+1} = \alpha S F^t + (1-\alpha) Y, \tag{4}$$

where $F^t \in \mathcal{F}$ denotes the predicted labels at the timestamp t, S denotes the normalized weight, and $\alpha \in (0,1)$ controls the amount of propagated information. It is well known that the sequence $\{F^t\}$ has a closed-form solution as follows:

$$F^* = (I - \alpha S)^{-1} Y, \tag{5}$$

where I is the identity matrix [52].

3.2.3 ***Label Guessing.*** For each unlabeled essay in $\mathcal{X}_u$, we produce a "guess" for its label based on the predictions of label propagation on multiple graphs. This guess is later used as pseudo label of unlabeled essay for knowledge distillation.

To do so, we first compute the average of the label distributions predicted by label propagation on all the B graphs by

$$Y' = \frac{1}{B} \sum_{b=1}^{B} F^*_{G_b}, \tag{6}$$

where Y' denotes the averaged label distribution matrix, and $F^*_{G_b}$ denotes the final label distribution matrix generated by applying label propagation on graph G_b.

Then, for each unlabeled essay $x_i \in \mathcal{X}_u$, its pseudo label y'_i is obtained as follows:

$$y'_i = \arg\max_{1 \le j \le K} Y'_{ij}, \tag{7}$$

where Y'_{ij} denotes the j-th element of the i-th row vector of Y'.

3.3 Ordinal-Aware Neural Network (Student)

We introduce the *Student Model* illustrated in Figure 1, which is an ordinal-aware neural network model and consists of two main components: essay encoder that extracts the feature embedding

of the input essay; ordinal classifier that predicts a unimodal label distribution on the pre-defined scores for each input essay.

3.3.1 ***Essay Encoder.*** We employ a neural network $f_\varphi(\cdot)$ to extract features of an input x_i, where $f_\varphi(x_i;\varphi)$ refers to the essay embedding and φ indicates the parameters of the network. This module is not limited to a specific architecture and can be various existing AES encoders. To demonstrate the universality of our framework and provide more fair comparisons in the experiments, we adopt the encoders adopted in some recent work (e.g., CNN-LSTM-Att [9], HA-LSTM [5], BERT [5]).

3.3.2 ***Unimodal Ordinal Classifier.*** Unlike previous neural network based AES models which predict the score of the input essay by using a regression layer (i.e. a one-unit layer), we view the essay scoring as an ordinal classification problem and adopt an ordinal classifier [3] for prediction.

To capture the ordinal relationship among classes, the unimodal probability distribution (i.e., the distribution has a peak at class k while decreasing its value when the class goes away from k) is usually used to restrict the shape of the predicted label distributions. According to previous studies [3, 22], some special exponential functions and the probability mass function (PMF) of both Poisson distribution and binomial distribution can be used to enforce discrete unimodal probability distribution.

In our framework, we choose an extension of the binomial distribution, Conway–Maxwell binomial distribution (CMB) [16], as the base distribution, and employ the PMF of the CMB to generate the predicted unimodal probability distribution of essay x_i:

$$P(y_i = k) = \frac{1}{S(p,v)}\binom{K-1}{k-1}^v p^{k-1}(1-p)^{K-k}, \tag{8}$$

where

$$S(p,v) = \sum_{k=1}^{K}\binom{K-1}{k-1}^v p^{k-1}(1-p)^{K-k}. \tag{9}$$

Here $k \in \mathcal{Y} = \{1,2,\ldots,K\}$, $0 \le p \le 1$, and $-\infty \le v \le \infty$. The parameter v can be used to control the variance of the distribution. The case $v = 1$ is the usual binomial distribution.

To be more specifically, we now describe the neural network architecture of the employed ordinal classifier based on the PMF of the CMB. As shown in Figure 1, the essay encoder is followed by a linear layer which transforms the essay embedding into a number $v \in \mathbb{R}$ and a probability $p \in [0,1]$ (by using sigmoid activation function). The linear layer is then followed by a 'copy expansion' layer which expands the probability p into K probabilities corresponding to K distinct scores, that is, $p_{k=1} = p_{k=2} = \cdots = p_{k=K}$. The following layer then applies the 'Log CMB PMF' transformation on these probabilities with different k:

$$\begin{aligned} LCP(k;v,p) = v\log\binom{K-1}{k-1} + (k-1)\log p \\ + (K-k)\log(1-p), \end{aligned} \tag{10}$$

where the log operation is used to address numeric stability issues. Finally, a softmax layer is applied on the logit, $LCP(k;v,p)$, to produce a unimodal probability distribution $\hat{Y}_i$ for essay x_i:

$$\hat{Y}_{ik} = \frac{e^{LCP(k;v,p)}}{\sum_{k=1}^{K} e^{LCP(k;v,p)}}, \tag{11}$$

where $\hat{Y}_{ik}$ denotes the k-th element of $\hat{Y}_i$. Based on $\hat{Y}_i$, the final predicted label $\hat{y}_i$ of essay x_i can be obtained by:

$$\hat{y}_i = \arg\max_{1\le k\le K} \hat{Y}_{ik}. \tag{12}$$

3.4 Ordinal Distillation

We introduce the *Ordinal Distillation* illustrated in Figure 1, which distills the pseudo-label knowledge of *Teacher Model* into the *Student Model*, and consists of three main steps: label selection that selects high confidence pseudo-labels for later distillation; unimodal smoothing that enforces the label distribution of pseudo-label to be a unimodal probability distribution; unimodal distillation that minimizes the KL divergence between the predicted label distribution of *Student Model* and the unimodal smoothed label distribution of *Teacher Model*.

3.4.1 ***Label Selection.*** Considering that only one-shot labeled data is available for label propagation, the pseudo labels generated by *Teacher Model* may be noisy. Therefore, we propose a label selection strategy to select a subset of pseudo labels with high confidence.

Specifically, for each distinct score $k \in \mathcal{Y}$, we first collect all corresponding pseudo labels, that is, $C_k = \{y'_i | y'_i = k, x_i \in \mathcal{X}_u\}$, and then rank these pseudo labels C_k according to their confidence. We measure the confidence of a pseudo label y'_i by calculating the negative Shannon entropy of its corresponding label distribution (Eq. 13), so that a peaked distribution may tend to get a high confidence.

$$Confidence(y'_i) = -\mathbb{H}(Y'_i) = \sum_{j=1}^{K} Y'_{ij}\log_2 Y'_{ij} \tag{13}$$

After that, we select top m_k pseudo labels with high confidence from C_k by

$$m_k = \min\left(|C_k|, \max(a, |C_k| \times \gamma)\right), \tag{14}$$

where the threshold ratio γ and the threshold number a are set to ensure a sufficient number of pseudo labels are selected in the end and avoid serious class imbalance problem.

3.4.2 ***Unimodal Smoothing.*** Previous studies on knowledge distillation [13, 49] have shown that a soft or smoothed probability distribution from teacher model is more suitable for knowledge distillation than one-hot probability distribution. Considering that essay scoring is an ordinal classification problem and an essay is more likely to be mispredicted as a score close to the ground-truth score, we enforce the distribution of pseudo labels produced by *teacher model* to be a unimodal smoothed probability distribution.

As mentioned before, some special exponential functions [22] can be used to enforce discrete unimodal probability distribution. Therefore, we employ an exponential function to perform the unimodal smoothing on both one-shot labels and pseudo labels:

$$q'(y_i = k|x_i) = \begin{cases} \frac{\exp(\frac{-|k-y_i|}{\tau})}{\sum_{j=1}^{K}\exp(\frac{-|j-y_i|}{\tau})} & x_i \in \mathcal{X}_o \\ \frac{\exp(\frac{-|k-y'_i|}{\tau})}{\sum_{j=1}^{K}\exp(\frac{-|j-y'_i|}{\tau})} & x_i \in \mathcal{X}_u \end{cases}, \tag{15}$$

Algorithm 1 The Training Flow of TGOD

Input: The whole set of essays $\mathcal{X}$, one-shot labeled data $\mathcal{D}_o$.
Output: An optimized student model.
Run the Teacher Model:
Construct multiple graphs $G^* = \{G_1, G_2, \ldots, G_B\}$ on $\mathcal{X}$.
for each $G_b \in G^*$ **do**
Apply label propagation algorithm on G_b as Eq. 5.
end for
Generate pseudo labels by label guessing as Eq. 6 and 7.
Train the Student Model by Ordinal Distillation:
Select the pseudo labels with high confidence by Eq. 13 and 14.
Smooth the selected labels as Eq. 15.
Split selected essays into training set $\mathcal{D}_t$ and validation set $\mathcal{D}_v$.
for all iter=1,...,MaxIter **do**
Optimize the student model on $\mathcal{D}_t$ by minimizing Eq. 16.
Validate the student model on $\mathcal{D}_v$
end for
return The student model with best performance on $\mathcal{D}_v$

where $k \in \mathcal{Y}$ and τ is a parameter used to control the variance of the distribution.

3.4.3 ***Unimodal Distillation***. Since the one-shot labeled data $\mathcal{D}_o$ is not sufficient to train a neural network, we use the pseudo labels produced by *teacher model* as a supplement to train the *student model*.

Specifically, we train the *student model* by matching the output label distribution of *student model* $\hat{q}(x_i) = \hat{Y}_i$ and the unimodal smoothed pseudo label of *teacher model* $q'(x_i)$ via a KL-divergence loss:

$$\mathcal{L}_{OD} = \sum_{x_i \in \mathcal{X}_s} D_{KL}\left(\hat{q}(x_i) || q'(x_i)\right), \tag{16}$$

where $\mathcal{X}_s$ denotes the set of essays from either one-shot data or the selected essays after label selection.

3.5 Training Flow of TGOD

In summary, there are two steps in TGOD to train the *Student Model* under the one-shot setting, i.e., first generating pseudo labels of unlabeled essays by running the *Teacher Model*, and then training the *Student Model* by *Ordinal Distillation*. The whole training flow of TGOD is illustrated in Figure 1 and Alg. 1.

In particular, considering that model selection is difficult to implement under the one-shot supervised setting, we design a model selection strategy based on pseudo labels, which validates the model on a subset of pseudo labels.

4 EXPERIMENTS

In this section, we first introduce the dataset and evaluation metric. Then we illustrate the experimental settings, the implementation details, and the performance comparison. Finally, we conduct ablation study and model analysis to investigate the effectiveness of our proposed approach.

Table 1: Statistics of the ASAP datasets. For column Genre, ARG denotes argumentative essays, RES denotes response essays, and NAR denotes narrative essays. The last column lists the score ranges.

Prompt	#Essay	Genre	Avg Len	Range
1	1,783	ARG	350	2-12
2	1,800	ARG	350	1-6
3	1,726	RES	150	0-3
4	1,772	RES	150	0-3
5	1,805	RES	150	0-4
6	1,800	RES	150	0-4
7	1,569	NAR	250	0-30
8	723	NAR	650	0-60

4.1 Dataset and Evaluation Metric

We conduct experiments on a public dataset ASAP (Automated Student Assessment Prize[1]), which is a widely-used benchmark dataset for the task of automated essay scoring. In ASAP, there are eight sets of essays corresponding to eight different prompts, and a total of 12,978 scored essays. These eight essay sets vary in essay number, genre, and score range, the details of which are listed in Table 1.

To evaluate the performance of AES methods, we employ the quadratic weighted kappa (QWK) as the evaluation metric, which is the official metric of ASAP dataset. For each set of essays with possible scores $\mathcal{Y} = \{1, 2, \ldots, K\}$, the QWK can be calculated to measure the agreement between the automated predicted scores (Rater A) and the resolved human scores (Rater B) as follows:

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}}, \tag{17}$$

where $w_{i,j} = \frac{(i-j)^2}{(K-1)^2}$ is calculated based on the difference between raters' scores, O is a K-by-K histogram matrix, $O_{i,j}$ is the number of essays that received a score i by Rater A and a score j by Rater B, and E is calculated as the normalized outer product between each rater's histogram vector of scores.

4.2 Experimental Settings

For the setting of '*one-shot*', we conduct experiments by randomly sampling the one-shot labeled data to train the model and test the model on the rest unlabeled essays. To reduce randomness, under each case, we repeat the sampling of one-shot labeled data 20 times, and the average results are reported. For our proposed framework, we perform model selection based on the pseudo validation set. For other baseline methods, since one-shot labeled data is used for training and no extra labeled data can be used as a validation set to perform model selection, we report their best performance on test set as their upper bound performance for comparison.

For the setting of '*one-shot+history prompt*', we combine the one-shot labeled data and the labeled data in a history prompt of the similar score range (e.g., P1 → P2, P2 → P1, P3 → P4, P4 → P3, and

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

Table 2: The performance (QWK) of all comparison methods on ASAP dataset. The best measures are in bold. † denotes that the data is referenced from previous studies and the setting is 'one history prompt + 10 essays from target prompt'. 'T(·)' refers to teacher model and 'S(·)' refers to student model.

Setting	Method			P1	P2	P3	P4	P5	P6	P7	P8	Avg.
One-Shot	TGOD (Ours)	T(4 Graphs)		0.667	0.525	0.648	0.693	0.734	0.570	0.619	0.447	0.613
		S(CNN-LSTM-Att)	OCLF + Distill(unimodal)	0.784	**0.626**	0.652	0.689	**0.777**	0.651	0.723	0.619	0.690
			REG + Distill(score)	0.772	0.617	0.649	0.694	0.773	0.606	0.721	0.608	0.680
		S(HA-LSTM)	OCLF + Distill(unimodal)	**0.792**	0.593	0.661	0.689	0.759	0.674	0.738	**0.635**	**0.693**
			REG + Distill(score)	0.780	0.565	0.674	0.678	0.741	0.667	0.700	0.581	0.673
		S(BERT)	OCLF + Distill(unimodal)	0.772	0.581	**0.690**	**0.725**	0.776	**0.691**	**0.766**	0.505	0.688
			REG + Distill(score)	0.752	0.571	0.665	0.644	0.773	0.668	0.691	0.577	0.668
	AES Model	BLRR	REG	0.731	0.553	0.578	0.644	0.623	0.581	0.583	0.574	0.608
		CNN-LSTM-Att	OCLF	0.626	0.443	0.352	0.526	0.643	0.475	0.170	0.145	0.422
			REG	0.545	0.477	0.202	0.569	0.671	0.493	0.580	0.641	0.522
		HA-LSTM	OCLF	0.576	0.507	0.617	0.553	0.635	0.585	0.620	0.222	0.539
			REG	0.616	0.515	0.338	0.531	0.746	0.649	0.555	0.480	0.554
		BERT	OCLF	0.695	0.535	0.629	0.621	0.748	0.660	0.706	0.447	0.630
			REG	0.704	0.562	0.648	0.631	0.775	0.647	0.687	0.568	0.653
	Semi-Supervised Model	Label Propagation	Word2Vec-MoT	0.703	0.525	0.654	0.657	0.627	0.571	0.540	0.429	0.588
			GloVe-MoT	0.675	0.552	0.642	0.668	0.686	0.546	0.588	0.385	0.593
			ELMo-MoT	0.658	0.382	0.577	0.635	0.583	0.640	0.443	0.422	0.543
			BERT-MoT	0.668	0.467	0.603	0.641	0.753	0.545	0.615	0.471	0.595
		TSVM	Word2Vec-MoT	0.167	0.423	0.479	0.507	0.619	0.474	0.215	0.188	0.384
			GloVe-MoT	0.152	0.435	0.386	0.530	0.547	0.488	0.131	0.135	0.350
			ELMo-MoT	0.189	0.327	0.480	0.573	0.541	0.412	0.224	0.109	0.357
			BERT-MoT	0.201	0.193	0.523	0.561	0.611	0.450	0.175	0.202	0.365
One-Shot + History Prompt	AES Model	CNN-LSTM-Att	Reference Data †	–	0.552	–	0.691	–	0.669	–	0.603	–
			Re-Implement	0.592	0.553	0.666	0.680	0.690	0.656	0.640	0.565	0.630
		HA-LSTM	Reference Data †	–	0.570	–	0.681	–	0.704	–	0.605	–
			Re-Implement	0.633	0.545	0.685	0.683	0.729	0.629	0.281	0.436	0.578
		BERT	Reference Data †	–	0.552	–	0.705	–	**0.725**	–	0.600	–
			Re-Implement	0.661	**0.669**	0.651	0.698	0.709	0.599	0.725	0.574	0.661
	Few-Shot Model	PROTO NET	Meta-training	0.693	0.599	0.676	0.714	0.735	0.612	0.545	0.415	0.624
		TPN	Meta-training	0.648	0.479	0.663	0.681	0.704	0.575	0.514	0.402	0.583

so on) to train the baseline AES model. For the few-shot models, we use the data of history prompt as their meta training data.

4.3 Implementation Details

In our TGOD framework, for the teacher model, we adopt four types of word embeddings (i.e., Word2Vec, GloVe, ELMo, and BERT) to construct four graphs for label guessing. The dimension of word embedding is 200. We fix the word embedding during training. The k for constructing k-nearest neighbor graph is set 20. For label selection, γ is set to 0.25 and a is set to 50. For label smoothing, τ is set to 30. For the student model, we adopt three neural AES models (i.e., CNN-LSTM-Att, HA-LSTM, and BERT) as the student model. We test the cases of using either the ordinal classifier (OCLF, adopted by our framework) and the regression layer (REG, used by the baseline AES models). While using the regression layer, the smoothed label distribution is replaced by the score of pseudo labels.

For the training of regression based AES models, the ground-truth scores of essays are rescaled into $[0, 1]$ for regression. To evaluate the results, the predicted scores are rescaled to the original score range of the corresponding prompts. For the hyper-parameters of CNN-LSTM-Att and HA-LSTM, the hidden size is set to 100, dropout is set to 0.5, the Adam optimizer is adopted, and the learning rate is set to 0.001. For BERT, the 'uncased BERT-base model' is adopted, the Adam optimizer is adopted, and the learning rate is set to 0.001.

4.4 Comparison Methods

As described in Section 3, our framework employs a graph-based label propagation method as the teacher model and an ordinal-aware neural network as the student model, which are transductive semi-supervised model and supervised AES model respectively. Thus, under the one-shot setting, we compare our models with existing supervised *AES models* and *semi-supervised models*. Considering that some previous studies on AES have often focused on combining few data in target prompt and the data in history prompts to perform essay scoring, we view them as a different one-shot like setting, named one-shot plus history prompt. In this setting, we consider the existing *AES models* and classical *few-shot models*.

We implement four existing *AES models*:

• **BLRR** [32] is based on hand-crafted features, and uses correlated linear regression for prediction.

• **CNN-LSTM-Att** [9] is a neural AES model based on hierarchical architecture and attention mechanism.

• **HA-LSTM** [5] is a neural AES model based on hierarchical architecture and self-attention mechanism.

Table 3: Ablation study of TGOD. The setting '– US&OC' means that both unimodal smoothing and ordinal classifier are ablated from framework and general classification layer is adopted for prediction.

Model Setting	P1	P2	P3	P4	P5	P6	P7	P8	Avg.
TGOD(CNN-LSTM-Att)	**0.784**	**0.626**	0.652	**0.689**	**0.777**	0.651	**0.723**	**0.619**	**0.690**
– label selection (LS)	0.729	0.514	**0.665**	0.664	0.743	0.668	0.697	0.574	0.657
– unimodal smoothing (US)	0.754	0.532	0.614	0.675	0.743	0.635	0.633	0.351	0.617
– ordinal classifier (OC)	0.736	0.576	0.541	0.656	0.731	**0.670**	0.705	0.604	0.652
– US&OC (= only label selection)	0.696	0.500	0.630	0.658	0.637	0.458	0.556	0.439	0.572
– LS&OC (= only unimodal smoothing)	0.689	0.522	**0.665**	0.652	0.750	0.622	0.596	0.553	0.631
– LS&US (= only ordinal classifier)	0.700	0.467	0.654	0.657	0.746	0.547	0.548	0.492	0.601
– all (LS & US & OC)	0.680	0.472	0.661	0.658	0.743	0.556	0.528	0.421	0.590

• **BERT** [5] is the widely-used pre-training model, which has been used as an encoder for the task of AES.

We implement two classical *semi-supervised models*:

• **Label Propagation** [52] is a graph-based classification method under transductive setting.

• **TSVM** [15] is a margin-based classification method under transductive setting.

We implement two classical *few-shot models*:

• **Protopical Network** [37] is a few-shot model based on metric learning and adopts the episodic training procedure.

• **TPN** [24] is a transductive few-shot model based on label propagation and adopts the episodic training procedure.

For our TGOD framework, we implement a baseline that replaces the ordinal-aware unimodal distillation with linear regression.

4.5 Performance Comparison

As shown in Table 2, the best performance is mostly achieved by our *TGOD* framework with using different essays encoders (i.e., *CNN-LSTM-Att*, *HA-LSTM*, and *BERT*). By observing *TGOD*, we can find the performance of the teacher model (i.e., graph-based label propagation) with 4 graphs is an average QWK of 0.613, based on which, the student models can greatly outperform the teacher model. This indicates that the design of learning from graph propagation is effective for one-shot essay scoring.

By observing the AES models under 'One-Shot' setting, we can find that among the four AES models, *BERT* performs best, which can achieve a QWK of 0.630 (by REG, i.e., regression) and 0.653 (by OCLF, i.e., ordinal classification). Besides, the hand-crafted features based method *BLRR* performs better than the *CNN-LSTM-Att* and *HA-LSTM*, but worse than *BERT*. This may be because that *BLRR* does not need to train an essay encoder, and *BERT* has a pre-trained encoder. By comparing these three neural AES models to our *TGOD*s with the corresponding essay encoder, we can find that *TGOD* can greatly improve their performance under the one-shot setting. To be more detailed, for each neural AES model, we can find that the performance of using OCLF (i.e., 0.422, 0.539, and 0.630 for three neural AES models) is worse than the performance of using REG (i.e., 0.522, 0.554, and 0.653 for three neural AES models) when directly trained on one-shot data, but under our TGOD framework, the performance of using OCLF (i.e., 0.690, 0.693, and 0.688 for three neural AES models) is better than the performance of using REG (i.e., 0.680, 0.673, and 0.668 for three neural AES models). This may be because that ordinal classification is more robust to the weak labels.

By observing the semi-supervised models, we can find that by just using word embedding to get the feature of essays, *Label Propagation* can achieve a better performance than the supervised AES models. By comparing *Label Propagation* with single graph to the teacher model in *TGOD* with 4 graphs, we can find that an ensemble of these graphs can produce a better teacher for TGOD than using only one graph.

By observing the models under the setting 'One-Shot + History Prompt', we can find that even with more labeled data from other prompt, these models do not outperform our *TGOD*.

4.6 Ablation Study

We explore the effects of the components designed specific to the one-shot setting, by removing each of them from *TGOD* individually. These components include: *label selection* (**LS**), *unimodal smoothing* (**US**), and *ordinal classifier* (**OC**). We remove them from TGOD in three ways: remove one of them, remove two of them, and remove all of them.

As shown in Table 3, after removing one of them from *TGOD*, the performance decreases a lot. This indicates that all of the three components are important to *TGOD*. After removing another one of them from *TGOD*, the performance continues to decrease. After removing all of them from *TGOD*, the performance decreases to a QWK of 0.590, which is even worse than the teacher model in *TGOD*. This indicates that distilling the pseudo labels to a classification model without label processing can not prevent the model from being disturbed by the noises or errors in pseudo labels. In addition, the performance of '– US&OC' (which means only label selection is used) is even worse than the performance of '– all'. This indicates that *label selection* should be used along with other two components (i.e., **US** and **OC**), otherwise, it would fail to benefit the general classification model (not ordinal aware), and even have a negative impact on the final performance.

4.7 Model Analysis

In this part, we analyze the effects of the one-shot labeled data and the graph construction on the performance of TGOD.

4.7.1 ***Effect of one-shot data selection.*** For one-shot labeled data, we first study the impacts of data selection on the performance of *TGOD*, that is, whether our *TGOD* framework is sensitive to the selection of one-shot essays. To this end, we repeat the sampling of

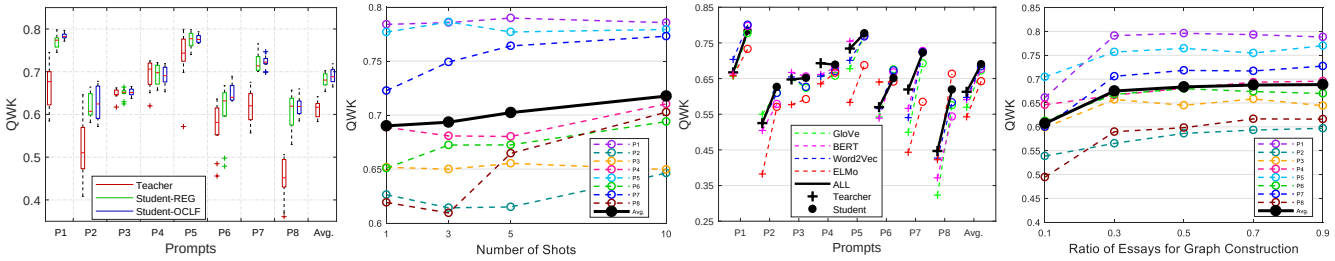

Figure 2: Effects of the one-shot labeled data and the graph construction on the performance of TGOD.

one-shot labeled data 20 times, and record the corresponding performance of *TGOD* (*CNN-LSTM-Att* is adopted as the essay encoder) for each sampling. For comparison, we record the performance of teacher model (*Teacher*), regression based student model (*Student-REG*), and ordinal classification based model (*Student-OCLF*).

As shown in Figure 2(a), the red boxes often have a large variance, which means that the performance of teacher is sensitive to the selection of the one-shot labeled data. The blue and green boxes have an obviously smaller variance than the corresponding red boxes. This indicates that after the process of label selection and distillation, the student model is no longer as sensitive to the selection of one-shot labeled data as teacher model. By comparing the blue boxes and green boxes, we can find that *Student-OCLF* is more robust to the selection of one-shot data than *Student-REG*.

4.7.2 ***Effect of using more labeled data.*** We then study the impacts of using more labeled data on the performance of *TGOD*, that is, whether our *TGOD* framework can be further improved by providing more labeled data. To this end, we sample the labeled data by one-shot, three-shots, five-shots, and ten-shots, and record the corresponding performance of *TGOD* (*CNN-LSTM-Att* is adopted as the essay encoder) for each setting.

As shown in Figure 2(b), by observing the line of *Avg.* (with black color), the overall performance of TGOD shows an upward trend, and the performance on the ten-shot labeled data has an improvement of about 0.03 (on QWK) compared to that on the one-shot labeled data. By observing other eight lines, we can find that the overall performance of P3 shows a flat trend and P5 shows a slight downward trend. This may be because that when more labeled samples are added, the performance bottleneck may be the quality of graphs in teacher model, and thus the teacher model is not benefit from using more labeled data.

4.7.3 ***Effect of combining multiple graphs.*** For the graph construction in teacher model, we first study the impacts of adopting multiple word embeddings for graph construction on the performance of TGOD, that is, whether our TGOD framework is benefit from combining multiple graphs for label guessing. To this end, we record the performance of the teacher model (graph propagation) and the student model (CNN-LSTM-Att) when using each of the four types of word embeddings and using them together.

As shown in Figure 2(c), by observing the black line, we can find that its end point (*Teacher Model*) is higher than that of the other lines at most cases, regardless of the position of starting point. This indicates that combining multiple graphs for label guessing is an effective way to provide pseudo labels with stable quality and thus improves the performance of the *Student Model*.

4.7.4 ***Effect of the graph size.*** We then study the impacts of varying the number of essays for graph construction on the performance of *TGOD*, that is, whether our *TGOD* framework needs a large number of unlabeled data for graph construction and pseudo label generation. To this end, we vary the ratio of essays used for graph construction from 0.1 to 0.9 step by 0.2.

As shown in Figure 2(d), we can find that all the lines show a trend that goes up first and then keeps stable after the ratio about 0.3. This indicates that 30% unlabeled essays is enough to run the teacher model and generate pseudo labels for our *TGOD* framework.

5 RELATED WORK

In this section, we introduce briefly the following three research topics relevant to our work.

5.1 Automated Essay Scoring

Early research on AES mainly focused on the construction of automated composition scoring systems [11, 26], which mainly combined surface features with regression models for essay scoring. Since this century, feature engineering has been used to design abundant linguistic features for essay scoring [29, 30, 38]. More recently, many neural-based methods have been proposed to learn the features automatically [1, 8, 9, 34, 39, 41]. Among these methods, prompt-specific methods are effective but the process of manual scoring is labor intensive. Generic methods [48] and cross-prompt neural based methods [5, 9, 14] are thus proposed to alleviate the burden of manual scoring.

Most of the previous work tackled AES as a regression problem and used the Mean Square Error (MSE) as loss function for model training [7, 32, 39]. But they often used Quadratic Weighted Kappa (QWK) [17, 44] as their metric, which is a metric for ordinal classification problem. This inconsistency may be because that it is more complicated to tackle it as an ordinal classification problem [44, 47], and regression model can usually achieve good performance.

5.2 Knowledge Distillation

Knowledge distillation is originally proposed to transfer the knowledge of a complicated model to a simpler model by training the simpler model with the soft targets provided by the complicated

model [13]. Since then, it has been widely adopted in a variety of learning tasks [18, 35, 46]. Recently, several approaches [27, 36, 50] have been proposed to improve performance of knowledge distillation. They address how to extract information better from teacher networks and deliver it to students using the activations of intermediate layers [36], attention maps [50], or relational information between training examples [27]. Besides, instead of transferring information from teacher to student, Zhang et al. [51] proposed a mutual learning strategy. Our work differs from existing approaches in that we enforce the student model to learn a unimodal distribution but not the output distribution of teacher model.

5.3 Semi-Supervised Learning

Semi-Supervised Learning (SSL) aims to label unlabeled data using knowledge learned from a small amount of labeled data combined with a large amount of unlabeled data. SSL have two settings: transductive inference and inductive inference. The setting of transductive inference was first introduced by [42], which aims to infer the label of unlabeled data directly from the labeled data. The classical methods include the Transductive Support Vector Machine (TSVM) [15] and the graph-based label propagation [12, 52, 53]. Recently, the neural version of graph-based label propagation has been developed [24]. The setting of inductive inference aims to train an inductive model based on both labeled and unlabeled data. It has a great development in recent years and many effective methods have been proposed, such as Pseudo-Label [21], Γ Model [33], Mean Teacher [40], MixMatch [4], UDA [45].

6 CONCLUSION

In this paper, we aim to perform essay scoring under one-shot setting. To this end, we propose the TGOD framework to train a student neural AES model through a way of distilling the knowledge of a semi-supervised teacher model. In order to alleviate the negative effect of error pseudo labels on the student neural AES model, we introduce the label selection and ordinal distillation strategies. Experimental results demonstrate the effectiveness of the proposed TGOD framework for one-shot essay scoring. In the future, we will try to improve the performance of teacher model and student model by co-training or self-supervised learning.

ACKNOWLEDGMENTS

This work is supported by National Natural Science Foundation of China under Grant Nos. 61906085, 61802169, 61972192, 41972111; JiangSu Natural Science Foundation under Grant No. BK20180325; the Second Tibetan Plateau Scientific Expedition and Research Program under Grant No. 2019QZKK0204. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic Text Scoring Using Neural Networks. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 715–725.

[2] Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater®; v.2.0. *Journal of Technology Learning & Assessment* 4, 2 (2006), i–21.

[3] Christopher Beckham and Christopher J. Pal. 2017. Unimodal Probability Distributions for Deep Ordinal Classification. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017 (Proceedings of Machine Learning Research, Vol. 70)*, Doina Precup and Yee Whye Teh (Eds.). PMLR, 411–419.

[4] David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019. MixMatch: A Holistic Approach to Semi-Supervised Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*. 5050–5060.

[5] Yue Cao, Hanqi Jin, Xiaojun Wan, and Zhiwei Yu. 2020. Domain-Adaptive Neural Automated Essay Scoring. In *SIGIR '20: The 43rd International ACM SIGIR conference on research and development in Information Retrieval*.

[6] Fan Chung. 1997. *Spectral graph theory*. Published for the Conference Board of the mathematical sciences by the American Mathematical Society,.

[7] Madalina Cozma, Andrei M. Butnaru, and Radu Tudor Ionescu. 2018. Automated essay scoring with string kernels and word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 503–509.

[8] Fei Dong and Yue Zhang. 2016. Automatic Features for Essay Scoring - An Empirical Study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1072–1077.

[9] Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. 153–162.

[10] Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. 2018. Neural Automated Essay Scoring and Coherence Modeling for Adversarially Crafted Input. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018*. 263–271.

[11] Peter W. Foltz, Darrell Laham, and Thomas K Landauer. 1999. Automated Essay Scoring: Applications to Educational Technology. In *Proceedings of EdMedia + Innovate Learning 1999*, Betty Collis and Ron Oliver (Eds.). Association for the Advancement of Computing in Education (AACE), Seattle, WA USA, 939–944.

[12] Yanwei Fu, Timothy M. Hospedales, Tao Xiang, and Shaogang Gong. 2015. Transductive Multi-View Zero-Shot Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 37, 11 (2015), 2332–2345.

[13] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the Knowledge in a Neural Network. *CoRR* abs/1503.02531 (2015).

[14] Cancan Jin, Ben He, Kai Hui, and Le Sun. 2018. TDNN: A Two-stage Deep Neural Network for Prompt-independent Automated Essay Scoring. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. 1088–1097.

[15] T. JOACHIMS. 1999. Transductive Inference for Text Classification using Support Vector Machines. In *Proc of International Conference on Machine Learning*.

[16] Joseph B. Kadane. 2014. Sums of Possibly Associated Bernoulli Variables: The Conway-Maxwell-Binomial Distribution. *Bayesian Analysis* 11, 2 (2014).

[17] Zixuan Ke and Vincent Ng. 2019. Automated Essay Scoring: A Survey of the State of the Art. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. 6300–6308.

[18] Adhiguna Kuncoro, Chris Dyer, Laura Rimell, Stephen Clark, and Phil Blunsom. 2019. Scalable Syntax-Aware Language Models Using Knowledge Distillation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*. 3472–3484.

[19] Darrell Laham and Peter Foltz. 2003. *Automated scoring and annotation of essays with the Intelligent Essay Assessor*.

[20] Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*. 1188–1196.

[21] Dong-Hyun Lee. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, Vol. 3.

[22] Xiaofeng Liu, Fangfang Fan, Lingsheng Kong, Zhihui Diao, Wanqing Xie, Jun Lu, and Jane You. 2020. Unimodal regularized neuron stick-breaking for ordinal classification. *Neurocomputing* 388 (2020), 34–44.

[23] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. 2019. Learning to Propagate Labels: Transductive Propagation Network for Few-Shot Learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[24] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. 2019. Learning to Propagate Labels: Transductive Propagation Network for Few-Shot Learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[25] Ellis B Page. 1966. The imminence of... grading essays by computer. *The Phi Delta Kappan* 47, 5 (1966), 238–243.

[26] Ellis Batten Page. 1994. Computer Grading of Student Prose, Using Modern Concepts and Software. *Journal of Experimental Education* 62, 2 (1994), 127–142.

[27] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. 2019. Relational Knowledge Distillation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. 3967–3976.

[28] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1532–1543.

[29] Isaac Persing, Alan Davis, and Vincent Ng. 2010. Modeling organization in student essays. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. 229–239.

[30] Isaac Persing and Vincent Ng. 2014. Modeling prompt adherence in student essays. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*. 1534–1543.

[31] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*. 2227–2237.

[32] Peter Phandi, Kian Ming Adam Chai, and Hwee Tou Ng. 2015. Flexible Domain Adaptation for Automated Essay Scoring Using Correlated Linear Regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 431–439.

[33] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. 2015. Semi-supervised Learning with Ladder Networks. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. 3546–3554.

[34] Pedro Uria Rodriguez, Amir Jafari, and Christopher M. Ormerod. 2019. Language models and Automated Essay Scoring. *CoRR* abs/1909.09482 (2019). arXiv:1909.09482 http://arxiv.org/abs/1909.09482

[35] Haggai Roitman, Guy Feigenblat, Doron Cohen, Odellia Boni, and David Konopnicki. 2020. Unsupervised Dual-Cascade Learning with Pseudo-Feedback Distillation for Query-Focused Extractive Summarization. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. 2577–2584.

[36] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for Thin Deep Nets. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[37] Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical Networks for Few-shot Learning. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (Eds.). 4077–4087.

[38] Swapna Somasundaran, Jill Burstein, and Martin Chodorow. 2014. Lexical chaining for measuring discourse coherence quality in test-taker essays. In *Proceedings of the 25th International conference on computational linguistics*. 950–961.

[39] Kaveh Taghipour and Hwee Tou Ng. 2016. A Neural Approach to Automated Essay Scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 1882–1891.

[40] Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*. 1195–1204.

[41] Yi Tay, Minh C. Phan, Luu Anh Tuan, and Siu Cheung Hui. 2018. SkipFlow: Incorporating Neural Coherence Features for End-to-End Automatic Text Scoring. In *Proceedings of the 32nd Conference on Artificial Intelligence(AAAI-18)*. 5948–5955.

[42] Vladimir Vapnik. 1999. An overview of statistical learning theory. *IEEE Trans. Neural Networks* 10, 5 (1999), 988–999.

[43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*. 5998–6008.

[44] Yucheng Wang, Zhongyu Wei, Yaqian Zhou, and Xuanjing Huang. 2018. Automatic Essay Scoring Incorporating Rating Schema via Reinforcement Learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 791–797.

[45] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848* (2019).

[46] Ruochen Xu and Yiming Yang. 2017. Cross-lingual Distillation for Text Classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*. 1415–1425.

[47] Helen Yannakoudakis and Ted Briscoe. 2012. Modeling coherence in ESOL learner texts. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP, BEA@NAACL-HLT 2012, June 7, 2012, Montréal, Canada*, Joel R. Tetreault, Jill Burstein, and Claudia Leacock (Eds.). The Association for Computer Linguistics, 33–43.

[48] Attali Yigal, Bridgeman Brent, and Trapani Catherine. 2010. Performance of a Generic Approach in Automated Essay Scoring. *Journal of Technology Learning & Assessment* 10, 3 (2010), 17.

[49] Li Yuan, Francis E. H. Tay, Guilin Li, Tao Wang, and Jiashi Feng. 2020. Revisiting Knowledge Distillation via Label Smoothing Regularization. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 3902–3910.

[50] Sergey Zagoruyko and Nikos Komodakis. 2017. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[51] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. 2018. Deep Mutual Learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. 4320–4328.

[52] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Sch Olkopf. 2003. Learning with Local and Global Consistency. *Advances in neural information processing systems* 16, 3 (2003).

[53] X. Zhu. 2002. Learning from Labeled and Unlabeled Data with Label Propagation. *Tech Report* (2002).