Aggregating Multiple Heuristic Signals as Supervision for Unsupervised Automated Essay Scoring

Cong Wang, Zhiwei Jiang*, Yafeng Yin, Zifeng Cheng, Shiping Ge, Qing Gu
State Key Laboratory for Novel Software Technology, Nanjing University, China
cw@smail.nju.edu.cn, {jzw,yafeng}@nju.edu.cn,
{chengzf,shipingge}@smail.nju.edu.cn, guq@nju.edu.cn

Abstract

Automated Essay Scoring (AES) aims to evaluate the quality score for input essays. In this work, we propose a novel unsupervised AES approach ULRA, which does not require groundtruth scores of essays for training. The core idea of our ULRA is to use multiple heuristic quality signals as the pseudo-groundtruth, and then train a neural AES model by learning from the aggregation of these quality signals. To aggregate these inconsistent quality signals into a unified supervision, we view the AES task as a ranking problem, and design a special Deep Pairwise Rank Aggregation (DPRA) loss for training. In the DPRA loss, we set a learnable confidence weight for each signal to address the conflicts among signals, and train the neural AES model in a pairwise way to disentangle the cascade effect among partial-order pairs. Experiments on eight prompts of ASPA dataset show that ULRA achieves the state-of-the-art performance compared with previous unsupervised methods in terms of both transductive and inductive settings. Further, our approach achieves comparable performance with many existing domain-adapted supervised models, showing the effectiveness of ULRA. The code is available at https://github.com/tenvence/ulra.

1 Introduction

Automated Essay Scoring (AES) that aims to score the writing quality of essays without human intervention, is an important application of natural language processing in education. State-of-the-art AES models are typically trained in a supervised way with large labeled corpora, comprising essays and their groundtruth quality scores (Cozma et al., 2018; Ke and Ng, 2019; Kumar et al., 2022; Wang et al., 2022). However, collecting labeled essays is time-consuming and labor-intensive, especially for essays written specific to new prompts and when there is no professional scoring staff available.

Unsupervised AES can get rid of the requirement of groundtruth scores for training, and thus has significant potential in both scientific research and practical applications. Its importance can be summarized in three key aspects: 1) Unsupervised AES models can handle special scenarios that lack labeling resource, such as the absence of professional scoring staff, the need for rapid essay scoring without timely labeled data, or the cold start scoring of an AES system without historical labeled data; 2) Unsupervised AES models can serve as pseudo-label generators or validators for essay scoring based on semi-supervised learning, few-shot learning, or transfer learning; 3) In practical writing tests, unsupervised AES models can rapidly provide a preliminary decision-making basis for scoring staff prior to scoring.

Early work tackles unsupervised AES by using the clustering method (Chen et al., 2010). To solve the problem of *unordered* clusters (i.e., cannot map clusters to ordinal scores), Chen et al. (2010) propose to use a heuristic quality signal *the number of unique term in essay* as the initial score of each essay, and then iteratively propagate the scores to other essays in the same cluster. However, such unsupervised clustering process is *uncontrollable* (i.e., there is no guarantee that clusters are generated towards to essay quality). Recently, researchers propose to use a heuristic quality signal *word count* as the weak supervision to train a neural AES model (Zhang and Litman, 2021). However, they demonstrate that directly regressing the predicted score to a real-valued quality signal (i.e., word count) leads to poor performance.

The above unsupervised AES methods provide a good idea that heuristic quality signals can be used as an alternative of groundtruth scores for model training, but have two major drawbacks. 1) Signal values are too noisy to be directly regressed to. Considering that the quality signal and groundtruth score may have completely different values but sim-

* Corresponding author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 13999–14013

ilar partial orders, it is better to utilize the partial orders in signal rather than the values. 2) Single signal is too weak to provide good supervision. Since a single quality signal cannot comprehensively describe the quality of essay, more quality signals should be introduced to bring stronger and more robust crowdsourcing-like supervision.

To this end, we propose a novel framework for Unsupervised AES by Learning from Rank Aggregation (ULRA). The core idea of our ULRA is to introduce multiple heuristic quality signals as the pseudo-groundtruth, and then train a neural AES model by learning from the aggregation of these quality signals. Specifically, our ULRA contains a Heuristic Essay Ranking (HER) module which views each signal as a ranking metric to generate multiple rank lists, and a Deep Pairwise Rank Aggregation (DPRA) module which can aggregate the inconsistent quality signals for model training. In the HER module, we introduce three types of classic quality signals for essay ranking. In the DPRA module, we set a learnable confidence weight for each signal to address the conflicts among signals, and train the neural AES model in a pairwise way to disentangle the cascade effect among partial-order pairs. We conduct experiments on eight prompts of ASAP dataset, which demonstrate that our proposed ULRA significantly outperforms previous unsupervised methods, and can even achieve comparable performance to many existing domain-adapted supervised methods.

2 Related Work

Automated Essay Scoring. Early supervised AES systems are developed with handcrafted features (Attali and Burstein, 2004; Phandi et al., 2015; Yannakoudakis et al., 2011). While with the development of deep learning, most of recent AES methods try to use neural model to solve the problem (Taghipour and Ng, 2016; Alikaniotis et al., 2016; Dong et al., 2017; Nadeem et al., 2019; Tay et al., 2018; Wang et al., 2018; Liu et al., 2019; Uto et al., 2020). These methods are effective but labor-intensive for labeling. To reduce the reliance on labels, the generic method (Attali et al., 2010), cross-prompt methods (Cao et al., 2020; Dong et al., 2017; Jin et al., 2018), and one-shot method (Jiang et al., 2021) are proposed. For unsupervised AES setting, Chen et al. (2010) propose a voting algorithm that iteratively updates the scores by the heuristic quality signals. Zhang and Litman (2021) try to directly regress the predicted score to a real-valued quality signal, but achieve poor performance, because of the weak information of the quality feature. Overall, the effective unsupervised AES methods have not been widely explored.

Rank Aggregation. Rank aggregation (RA) is to aggregate multiple ranked list (i.e. base rankers) into one single list (i.e. aggregated ranker), which is intended to be more reliable than the base rankers (Deng et al., 2014). Many prior work have been proposed to effectively and efficiently solve the RA problem. These works can be roughly divided into three categories, which are the permutation-based methods (Mallows, 1957; Qin et al., 2010), the matrix factorization methods (Gleich and Lim, 2011), and the score-based probabilistic methods (Bradley and Terry, 1952; Luce, 2012; Pfeiffer et al., 2012; Thurstone, 1927; Chen et al., 2013). The score-based methods are gradually developed, which usually predicts a score for each object based on the base rankers, and obtains the aggregated ranker based on the scores. Bradley-Terry (BT) model (Bradley and Terry, 1952) is an early work that is instructive for the following researches. It is proposed to model a probabilistic relationship between objects, according to the achieved scores, which is suitable for pairwise comparison. Then, Thurstone model (Thurstone, 1927) extends the BT model by assuming that the score for each object has a Gaussian distribution. Another important extension is Crowd-BT (Chen et al., 2013) model, which assigns a learnable weight for each base ranker, and optimizes the scores and the weights by active learning and online learning. Crowd-BT achieves competitive performance on the task of reading difficulty ranking, and directly inspires our work.

3 Problem Definition

We firstly introduce some notation and formalize the unsupervised AES problem. Let $\mathcal{X} = \{x_i\}_{i=1}^{N}$ be a set of essays which are written to a prompt, and $\mathcal{Y} = \{1, 2, \cdots, L\}$ be the pre-defined scores. For unsupervised AES, we are given a set of unlabeled essays $\mathcal{D} = \{x_i\}_{i=1}^{N_\mathcal{D}} \subseteq \mathcal{X}$ for model training. The purpose of unsupervised AES is to train a model F_θ with parameters θ to predict the score of each essay $x_i \in \mathcal{T} \subseteq \mathcal{X}$ into the score set $\mathcal{Y}$, by

$$\hat{y}_i = F_\theta(x_i; \mathcal{D}) \in \mathcal{Y} \,. \tag{1}$$

In this paper, the model F_θ is a neural AES model, and we consider two settings of unsuper-

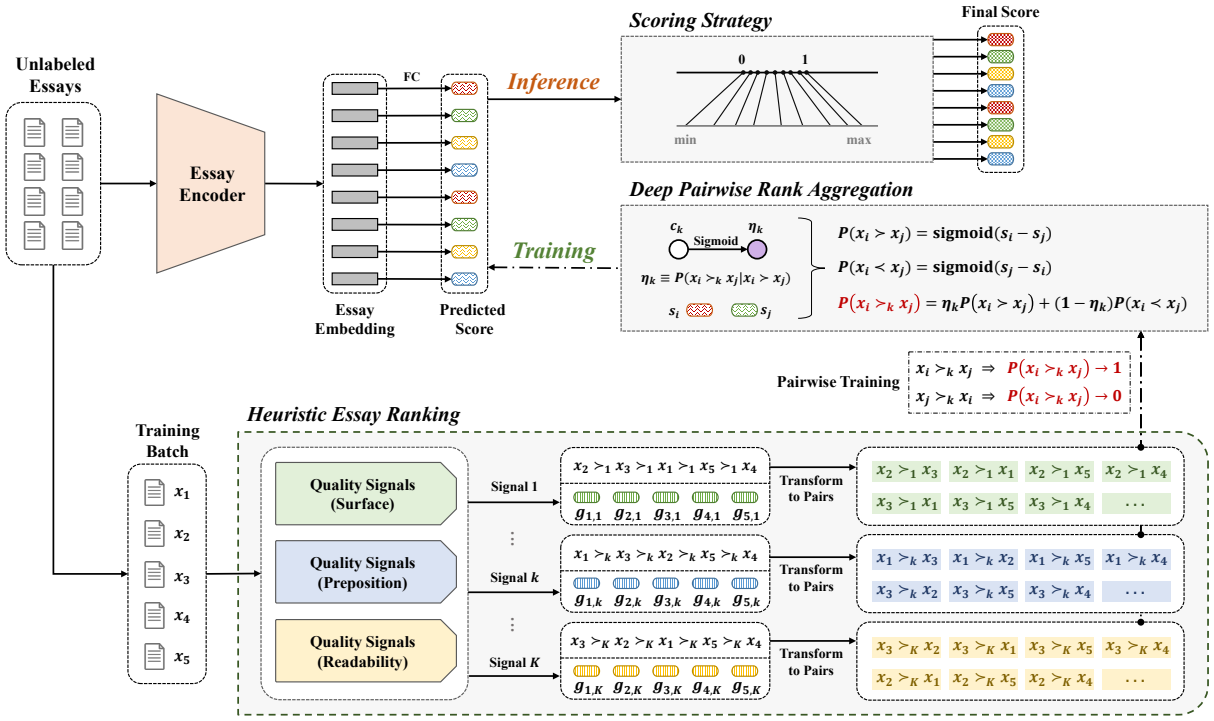


Figure 1: Illustration of the proposed ULRA framework for unsupervised AES task.

vised AES, transductive and inductive. For the transductive setting, the test set is just the training set $\mathcal{T} = \mathcal{D}$. For the inductive setting, the test set does not intersect the training set $\mathcal{T} \cap \mathcal{D} = \emptyset$.

4 The ULRA Framework

4.1 An Overview of ULRA

Our ULRA framework involves two stages, model training and model inference. As shown in Figure 1, in the model training stage, the ULRA framework contains two modules: 1) **Heuristic Essay Ranking** module, which can generate partial-order pairs by ranking essays according to heuristic quality signals, and 2) **Deep Pairwise Rank Aggregation** module, which trains a neural AES model by aggregating the partial-order pairs derived from multiple quality signals into a unified supervision. In the model inference stage, considering that the essay scores predicted by the neural AES model may have a different range from the pre-defined score set $\mathcal{Y}$, we propose a **Scoring Strategy** to transform the predicted scores given by the neural AES model into the range of the pre-defined score set. In addition, it should be noted that Figure 1 only shows the case of transductive setting, while for inductive setting, the trained neural AES model can be directly used to score unseen essays.

4.2 Heuristic Essay Ranking

As illustrated in Figure 1, the heuristic essay ranking module contains three components: quality signals, essay ranking, and partial-order pairs generation. Among them, multiple classic quality signals are introduced to describe the quality of essays from different aspects. Each quality signal can then be used to rank essays according to signal values and generate a rank list. Finally, each rank list can be transformed into many partial-order pairs for later model training.

Quality Signals. The quality signals are important to the ULRA framework, since it provides all supervision for model training. To obtain high-quality supervision, we investigate a lot of studies based on handcrafted quality signals (Lagakis and Demetriadis, 2021; Chen and He, 2013; Uto et al., 2020; Phandi et al., 2015). Considering that in practical unsupervised AES, there is no labeled data available as standard to select the most relevant signals, we just employ a set of quality signals designed in a classic work (Chen and He, 2013), which contains three aspects of signals, i.e., surface, preposition, and readability. In experiments, we find that not all of these signals are highly correlated with the groundtruth score. Moreover, for many of these signals, they may be highly correlated with the groundtruth score in one prompt but less correlated

in other prompts. Such quality signals with unstable supervision pose great challenges for the robustness of model training.

Essay Ranking. Compared with calculating the quality score of an essay based on its quality signals, it is easier to judge the relative quality of two essays based on their quality signals. Therefore, for each quality signal, we only reserve the partial-order relationship among essays by ranking the essays. Specifically, we rank the essays in a batch-wise way. Let $\mathcal{B} = \{x_i\}_{i=1}^{N_\mathcal{B}}$ denote an essay batch with $N_\mathcal{B}$ essays, and $\mathcal{R} = \{r_k\}_{k=1}^{K}$ denote K heuristic quality signals. As shown in Figure 1, we can calculate each quality signal r_k on each essay x_i to get a signal value $g_{i,k} = r_k(x_i)$, which can then be used for essay ranking. For the k-th quality signal, we can rank all essays x_i based on their signal values $g_{i,k}$ to get a partial-order rank list $x_{p_1} \succ_k \cdots \succ_k x_{p_{N_\mathcal{B}}}$, where $\succ_k$ denotes the partial-order relation defined by the k-th quality signal and p_j denotes the index of j-th essays in the rank list. Finally, we can get K rank lists $\{x_{p_1} \succ_k \cdots \succ_k x_{p_{N_\mathcal{B}}}\}_{k=1}^{K}$.

Partial-Order Pairs Generation. Considering that only part of the partial-order information in each rank list is correct, we transform each rank list into a set of partial-order pairs, which allows the incorrect partial-order pairs to be corrected by other rank lists. Specifically, for a batch $\mathcal{B}$, each rank list can be transformed into a total of $N_\mathcal{B}(N_\mathcal{B} - 1)/2$ partial-order pairs. Then, we can use a binary matrix with size of $N_\mathcal{B}(N_\mathcal{B} - 1)/2 \times K$ to record the transformed partial-order pairs, that is,

$$\mathbf{M} = \left[\mathbb{1}_{x_i \succ_k x_j}\right]^{\frac{N_\mathcal{B}(N_\mathcal{B}-1)}{2} \times K} , \tag{2}$$

where $i \neq j$ and $\mathbb{1}$ is an indicator function. $\mathbf{M}$ reflects the partial-order relationship between two essays in terms of different heuristic quality signals, and will be used as the supervision information to train the neural AES model in the next module.

4.3 Deep Pairwise Rank Aggregation

This module mainly deals with how to address the inconsistent partial-order supervision from multiple quality signals, so that the neural AES model can learn how to judge the partial-order relationship of essay quality. To address this problem, we design a deep pairwise rank aggregation loss, which set a learnable confidence weight for each signal to measure the importance of each signal.

Neural AES Model. We denote the neural AES model as $F_\theta(\cdot)$ with learnable parameters θ. By feeding an essay x_i into the model, we can get the predicted score $s_i = F_\theta(x_i) \in \mathbb{R}$ (not the final score) for the essay x_i. The neural AES model consists of two components, an essay encoder which maps the essay into an essay embedding, and a fully-connected (*fc*) layer which maps the embedding into a predicted score.

Confidence Weight. Considering that two signals may provide opposite partial order for an essay pair, we expect to measure which one is more trustworthy. Therefore, we set a learnable confidence weight η_k for the k-th quality signal to measure its confidence. The learnable weight η_k can be defined as the probability that the partial-order information in the k-th rank list agrees with the groundtruth score. Inspired by Crowd-BT (Chen et al., 2013), we formalize η_k as

$$\eta_k \equiv \mathrm{P}\left(x_i \succ_k x_j \mid x_i \succ x_j\right) , \tag{3}$$

where $x_i \succ_k x_j$ and $x_i \succ x_j$ denote the partial-order relationship between two essays in the k-th rank list and the ground truth score, respectively. In ULRA, η_k is generated by applying sigmoid function on learnable parameters $\mathcal{W} = \{c_1 \cdots, c_K\}$,

$$\eta_k = \mathrm{sigmoid}(c_k) \in [0, 1] , \tag{4}$$

where c_k is a learnable parameter and is optimized with model parameter θ together.

Deep Pairwise Rank Aggregation Loss. Based on the partial-order pairs derived from multiple signals and the confidence weight corresponding to each signal, we can define a special Deep Pairwise Rank Aggregation (DPRA) loss for model training.

Specifically, given an essay pair (x_i, x_j), we can use the neural AES model to get their predicted scores s_i and s_j, respectively. Inspired by the Bradley-Terry model for paired comparisons (Bradley and Terry, 1952), we can define the predicted probability of $x_i \succ x_j$ as

$$\mathrm{P}\left(x_i \succ x_j\right) = \mathrm{sigmoid}\left(s_i - s_j\right) . \tag{5}$$

If $s_i \gg s_j$, $\mathrm{P}(x_i \succ x_j)$ tends to be 1; If $s_i \ll s_j$, $\mathrm{P}(x_i \succ x_j)$ tends to be 0; While $s_i = s_j$, $\mathrm{P}(x_i \succ x_j) = 0.5$.

To further get the predicted probability of the partial-order pair $x_i \succ_k x_j$ generated by the k-th

Algorithm 1: The Scoring Process of ULRA

Input: A set of unlabeled essays $\mathcal{D} \subseteq \mathcal{X}$
Output: The final score of each essay in $\mathcal{T} \subseteq \mathcal{X}$
▷ **Training**: learning from rank aggregation;
Initialize a neural AES model;
for *each epoch* **do**
  for *each batch of* $\mathcal{D}$ **do**
    ▷ *Heuristic Essay Ranking*;
    Extract quality signals for each essay;
    Rank essays based on each signal;
    Generate partial-order pairs from rank lists;
    ▷ *Deep Pairwise Rank Aggregation*;
    for *each partial-order pair* **do**
      Predict probability of partial-order pair;
      Calculate the pairwise loss by Eq.7;
    end
    Optimize the neural AES model by Eq.8;
  end
end
▷ **Inference**: scoring essays by trained neural model;
for *each essay in* $\mathcal{T}$ **do**
  ▷ Here $\mathcal{T}$ can just be $\mathcal{D}$;
  Predict score of essay by the neural AES model;
end
Get final scores for all essays by the scoring strategy;
return all final scores;

quality signal, we can make use of the confidence weights and apply the law of total probability as

$$\begin{aligned}
&\mathrm{P}(x_i \succ_k x_j) \\
=&\mathrm{P}(x_i \succ_k x_j \mid x_i \succ x_j) \cdot \mathrm{P}(x_i \succ x_j) \\
&+ \mathrm{P}(x_i \succ_k x_j \mid x_i \prec x_j) \cdot \mathrm{P}(x_i \prec x_j) \\
=&\mathrm{P}(x_i \succ_k x_j \mid x_i \succ x_j) \cdot \mathrm{P}(x_i \succ x_j) \\
&+ \mathrm{P}(x_j \prec_k x_i \mid x_j \succ x_i) \cdot \mathrm{P}(x_i \prec x_j) \\
=&\eta_k \cdot \mathrm{P}(x_i \succ x_j) + (1-\eta_k) \cdot \mathrm{P}(x_i \prec x_j) \, .
\end{aligned} \tag{6}$$

Here, $\mathrm{P}(x_i \succ_k x_j)$ is the predicted probability of $x_i \succ_k x_j$, the label of which is $\mathbb{1}_{x_i \succ_k x_j}$. Then, the loss function for s_i, s_j, and η_k can be formulated as a negative log likelihood loss, which is

$$\mathcal{L}(s_i, s_j, \eta_k) = -\mathbb{1}_{x_i \succ_k x_j} \log \mathrm{P}(x_i \succ_k x_j) \, . \tag{7}$$

For each essay batch $\mathcal{B}$, a set of $N_{\mathcal{B}}(N_{\mathcal{B}}-1)/2$ partial-order pairs are obtained, which is denoted as $\mathcal{S}_{\mathcal{B}}$. Based on the supervision of $\mathbf{M}$, the loss function can be formulated as

$$\mathcal{L} = \sum_{(x_i, x_j) \in \mathcal{S}_{\mathcal{B}}} \sum_{k=1}^{K} -\mathbf{M}_{(i,j),k} \cdot \log \mathrm{P}(x_i \succ_k x_j) \, . \tag{8}$$

4.4 Scoring Strategy

Considering that the range of predicted score is not constrained during training process, the predicted score can be any real number. Therefore, we should cast the predicted score s_i of an essay x_i into the range of the pre-defined score set $\mathcal{Y} = \{1, \cdots, L\}$ to get the final scores $\hat{y}_i \in \mathcal{Y}$. Here, we can get the final score $\hat{y}_i$ of x_i by min-max transformation $\left[(L-1)\frac{s_i - \min(s_1, \cdots, s_N)}{\max(s_1, \cdots, s_N) - \min(s_1, \cdots, s_N)}\right] + 1$.

Prompt	Numbers of Essays	Genre	Average Length	Score Range
1	1783	ARG	350	[2, 12]
2	1800	ARG	350	[1, 6]
3	1726	RES	150	[0, 3]
4	1772	RES	150	[0, 3]
5	1805	RES	150	[0, 4]
6	1800	RES	150	[0, 4]
7	1569	NAR	250	[0, 30]
8	723	NAR	650	[0, 60]

Table 1: Statistics of the ASAP dataset. In the column Genre, ARG, RES, and NAR denote argumentative essays, response essays, and narrative essays, respectively.

	P1	P2	P3	P4	P5	P6	P7	P8
CH	.414	.399	.244	.409	.267	.212	.225	.413
W	.437	.432	.261	.401	.300	.208	.253	.330
CO	.140	.195	.115	.180	.111	.121	.073	.144
UW	.314	.375	.394	.471	.344	.323	.294	.221
NNP	.204	.294	.168	.223	.255	.161	.131	.064
DT	.287	.351	.223	.371	.198	.182	.163	.291
NN	.256	.298	.217	.422	.253	.164	.231	.339
RB	.221	.251	.207	.198	.248	.217	.151	.166
JJ	.217	.373	.184	.269	.228	.232	.179	.228
IN	.305	.320	.214	.353	.227	.190	.223	.382
GF	.179	.202	.253	.304	.258	.194	.152	.168
SMOG	.387	.398	.243	.413	.271	.204	.228	.418
RIX	.366	.433	.236	.404	.269	.216	.215	.416
DC	.425	.437	.260	.404	.302	.204	.256	.321
WT	.469	.536	.370	.449	.375	.308	.295	.458
S	.179	.202	.253	.304	.258	.194	.152	.168
LW	.237	.319	.197	.379	.223	.178	.141	.284
CW	.165	.325	.170	.309	.220	.157	.094	.146
NBW	.282	.324	.243	.367	.230	.188	.194	.181
DW	.169	.186	.226	.428	.212	.272	.122	.141

Table 2: QWKs between the groundtruth scores and the employed 20 quality signals under 8 prompts, which are obtained by viewing the quality signal as the predicted score and applying the scoring strategy.

5 Experiments

5.1 Dataset and Evaluation Metric

Experiments are conducted on the Automated Student Assessment Prize[1] (ASAP) dataset, which is widely used for the AES task. A total of 12,978 essays in ASAP are divided into 8 different sets, each of which corresponds to a prompt. The statistics of the dataset is shown in Table 1.

Quadratic Weighted Kappa (QWK) is adopted to be the evaluation metric. Specifically, given a score set $\mathcal{Y} = \{1, \cdots, L\}$, QWK is calculated to measure the automated predicted scores (Rater A)

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

Setting	Method	P1	P2	P3	P4	P5	P6	P7	P8	Avg.
One-Shot	**TGOD** (Jiang et al., 2021)	.772	.581	.690	.725	.776	.691	.766	.505	.688
Unsupervised	**Mean of the 20 quality signals**	.283	.333	.234	.353	.253	.206	.189	.264	.264
	Maximum of the 20 quality signals	.469	.536	.394	.471	.375	.323	.295	.458	.415
	Signal Clustering (Chen et al., 2010)	.355	.386	.370	.446	.509	.425	.428	.334	.407
	Signal Clustering w/ averaged signal as supervision	.393	.408	.383	.480	.500	.425	.470	.354	.427
	Signal Clustering w/ averaged output as prediction	.405	.413	.384	.498	.509	.435	.473	.370	.436
	Signal Clustering w/ aggregated signal as supervision	.359	.425	.404	.466	.535	.461	.465	.371	.436
	Signal Clustering w/ aggregated output as prediction	.363	.419	.397	.467	.544	.464	.467	.379	.438
	Signal Regression (Zhang and Litman, 2021)	.224	.321	.264	.404	.301	.441	.292	.353	.325
	Signal Regression w/ averaged signal as supervision	.232	.326	.271	.415	.303	.451	.304	.368	.334
	Signal Regression w/ averaged output as prediction	.249	.342	.289	.430	.311	.470	.316	.374	.348
	Signal Regression w/ aggregated signal as supervision	.246	.342	.263	.434	.309	.454	.304	.349	.338
	Signal Regression w/ aggregated output as prediction	.256	.344	.284	.451	.333	.496	.341	.345	.356
	Signal Aggregation (Chen et al., 2013)	.435	.480	.454	.608	.452	.439	.489	.218	.455
	ULRA (Ours)	**.757**	**.621**	**.547**	**.628**	**.664**	**.562**	**.694**	**.450**	**.615**

Table 3: **Transductive** performance (QWK) of all comparison methods under 8 prompts of ASAP dataset. The best measures of the unsupervised methods are in **bold**.

Setting	Method	P1	P2	P3	P4	P5	P6	P7	P8	Avg.
Supervised	**BLRR** (Phandi et al., 2015)	.761	.606	.621	.742	.784	.775	.730	.617	.705
	CNN-LSTM-Att (Dong et al., 2017)	.822	.682	.672	.814	.803	.811	.801	.705	.764
	TSLF (Liu et al., 2019)	.852	.736	.731	.801	.823	.792	.762	.684	.773
	HA-LSTM (Cao et al., 2020)	.828	.718	.711	.787	.808	.814	.786	.734	.773
	R^{2}BERT (Yang et al., 2020)	.817	.719	.698	.845	.841	.847	.839	.744	.794
	(Uto et al., 2020)	.852	.651	.804	.888	.885	.817	.864	.645	.801
Cross-Prompt	**CNN-LSTM-Att** (Dong et al., 2017)	.592	.553	.666	.680	.690	.656	.640	.565	.630
	HA-LSTM (Cao et al., 2020)	.633	.545	.685	.683	.729	.629	.281	.436	.578
	BERT (Cao et al., 2020)	.661	.669	.651	.698	.709	.599	.725	.574	.661
Unsupervised	**Mean of the 20 quality signals**	.320	.408	.285	.419	.262	.296	.305	.272	.320
	Maximum of the 20 quality signals	.511	.606	.420	.549	.368	.464	.427	.444	.474
	Signal Regression (Zhang and Litman, 2021)	.244	.309	.216	.338	.234	.189	.151	.247	.241
	Signal Regression w/ averaged signal as supervision	.253	.328	.219	.355	.247	.183	.162	.248	.249
	Signal Regression w/ averaged output as prediction	.269	.341	.213	.364	.239	.193	.180	.248	.256
	Signal Regression w/ aggregated signal as supervision	.252	.314	.239	.351	.246	.198	.167	.271	.255
	Signal Regression w/ aggregated output as prediction	.258	.319	.250	.365	.248	.216	.191	.300	.268
	ULRA (Ours)	**.759**	**.508**	**.608**	**.644**	**.711**	**.577**	**.661**	**.446**	**.614**

Table 4: **Inductive** performance (QWK) of all comparison methods under 8 prompts of ASAP dataset. The best measures of the unsupervised methods are in **bold**.

and the resolved human scores (Rater B),

$$\kappa = 1 - \frac{\sum_{i,j} w_{i,j} \cdot O_{i,j}}{\sum_{i,j} w_{i,j} \cdot E_{i,j}} \tag{9}$$

where $w_{i,j} = (i-j)^2/(L-1)^2$ is the difference between scores of the raters, O is an L-by-L 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 the normalized outer product between each rater's histogram vector of scores.

5.2 Implementation Details

Quality Signals Setting. The employed 20 quality signals, which are commonly used in some earlier work (Chen and He, 2013; Uto et al., 2020; Phandi et al., 2015), fall into following three categories:

- **Surface Signals**: character number (CH), word number (W), commas number (CO), and number of unique words (UW);
- **Preposition Signals**: number of noun-plural words (NNP), number of determiner words (DT), number of noun-singular words (NN), number of adverb words (RB), number of adjective words (JJ), and number of preposition/subordinating-conjunction words (IN);
- **Readability Signals**: Gunning Fog (GF) index (Gunning, 1969), SMOG index (Mc Laughlin, 1969), RIX (Anderson, 1983), Dale-Chall (DC) index (Dale and Chall, 1948), wordtype number (WT), sentence number (S), number of long words (LW), number of complex words (CW), number of non-basic words (NBW), and number of difficult words (DW).

In Table 2, we demonstrate QWK between each signal and the groundtruth of each prompt. It indicates that single quality signal carry noisy partial-order

	P1	P2	P3	P4	P5	P6	P7	P8	Avg.
Full Model	**.757**	**.621**	**.547**	.628	**.664**	**.562**	**.694**	.450	**.615**
– **learnable** η_k (fix $\eta_k = 1$)	.702	.610	.504	.610	.651	.547	.610	.380	.577
– **pretrained neural model** (using CNN-LSTM-Att)	.634	.599	.501	.628	.411	.553	.641	.419	.548
– **pretrained neural model** (using HA-LSTM)	.653	.613	.513	.605	.600	.501	.615	.436	.567
– **neural model** (all s_i are set as learnable parameters)	.432	.481	.451	.519	.600	.450	.484	.213	.454
– **surface signals** (preposition & readability signals)	.714	.610	.419	.593	.623	.541	.585	**.451**	.567
– **preposition signals** (surface & readability signals)	.694	.584	.504	.613	.649	.515	.643	**.451**	.582
– **readability signals** (surface & preposition signals)	.712	.584	.471	.626	.631	.500	.683	.431	.580
– **preposition & readability signals** (only surface signals)	.672	.588	.543	.628	.597	.497	.612	.434	.571
– **surface & readability signals** (only preposition signals)	.691	.553	.441	.518	.483	.429	.677	.403	.524
– **surface & preposition signals** (only readability signals)	.654	.627	.464	.563	.598	.514	.661	.444	.566
w/ averaged signal as supervision	.524	.541	.501	.615	.646	.542	.545	.245	.520
w/ averaged output as prediction	.536	.542	.519	.621	.632	.561	.553	.270	.529
w/ aggregated signal as supervision	.548	.544	.531	.624	.648	.548	.562	.262	.533
w/ aggregated output as prediction	.573	.544	.530	**.629**	.649	.551	.566	.260	.538

Table 5: Ablation Study of ULRA under 8 prompts. The demonstrated measures are QWK under the transductive setting. The best measures are in **bold**. The second-best measures are underlined.

information of groundtruth scores, which results in poor performance.

Dataset Setting. For the transductive setting, the model is trained on the entire dataset (w/o labels), and is tested on the entire dataset, which means that all test essays have been *seen* during training. For the inductive setting, the dataset (w/o labels) is divided into the training set, the validation set, and the test set in a ratio of 6:2:2, which means that all test essays have not been *seen* during training. Due to the unsupervised setting, the validation set is useless and therefore discarded for ULRA.

Model Setting. The model is implemented by PyTroch (Paszke et al., 2019) and Higgingface Transformers (Wolf et al., 2020) libraries. BERT (Devlin et al., 2019) with pretained parameters *bert-base-uncased* is adopted as the essay encoder, whose hidden size is 712. The essay embedding is achieved by mean-pooling the token embeddings of BERT output. The *fc* layer of the neural AES model maps the essay embeddings into scalars. Each confidence weight η_k is initialized as 0.9.

Training. AdamW (Loshchilov and Hutter, 2017) is adopted as the optimizer, whose weight decay is set as 5e-4. The learning rates for the neural AES model and all η_k are 5e-5 and 5e-2, respectively. The batch size is set as 32. Our model is trained for 30 epochs, and the model which achieves minimum loss is selected to report the results.

5.3 Comparison Methods

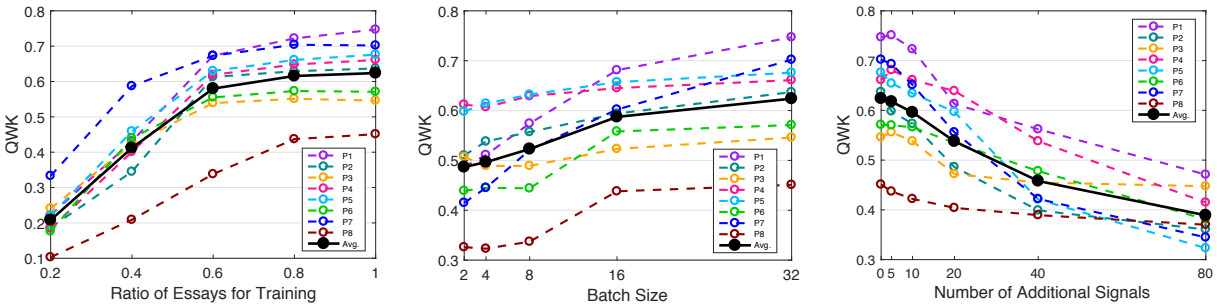
We mainly compare our method with previous unsupervised AES methods, Signal Clustering (Chen et al., 2010) and Signal Regression (Zhang and Litman, 2021). Considering that they only employ one quality signal as supervision, we extend them by introducing the 20 signals we used into their method. Four variants are tested: (1) *averaged signal as supervision*, (2) *averaged output as prediction*, (3) *aggregated signal as supervision*, and (4) *aggregated output as prediction*. Here, *aggregated* means that multiple rank lists are aggregated into one rank lists based on a rank aggregation algorithm (Chen et al., 2013). We also list two additional baselines, which directly apply the mean and maximum of the 20 quality signals as the predicted scores, respectively. Moreover, we list the performance of several state-of-the-art supervised methods (including general supervised, cross-prompt, and one-shot).

5.4 Performance Comparison

We can find that ULRA outperforms all unsupervised methods with a large improvement, and achieves the average QWK of 0.615 and 0.614 under transductive (Table 3) and inductive (Table 4) settings, respectively. It indicates that ULRA can perform well on both *seen* and *unseen* essay sets.

Compared with the cross-prompt and one-shot methods, we can find that ULRA achieves competitive performance, which is only 0.047 and 0.073 lower than that of the cross-prompt and one-shot methods, respectively. By observing the general supervised methods, we can find that the performance of ULRA is still much lower than theirs, due to the lack of strong supervision. But on some prompts, ULRA achieves comparable performance with the handcrafted features-based supervised method BLRR (e.g., prompt 1 and 3).

By observing the variants of two unsupervised methods, we can find that both unsupervised methods achieve improvements after introducing 20



(a) More Unlabeled Essays (b) More Training Pairs (c) Weak Signals

Figure 2: Effects of more unlabeled essays, more training pairs, and weak signals on the performance.

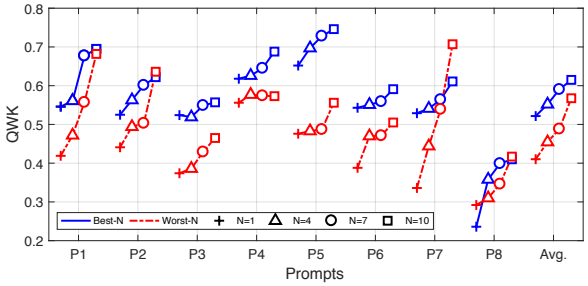


Figure 3: Effect of more quality signals on the performance. The results are reported by training with the N best or worst signals from the signal set.

	P1	P2	P3	P4	P5	P6	P7	P8
Transductive	.7438	.6855	.6677	.7813	.5365	.6033	.8360	.8932
Inductive	.7442	.6659	.6052	.7994	.5681	.6259	.8254	.9007

Table 6: Spearman's correlation coefficient between the learned confidence weights and the corresponding QWKs listed in Table 2 under each prompt.

quality signals. Among the four variants, two *aggregated* variants outperform two *averaged* variants. It indicates that the aggregation operation is better than the averaging operation, no matter as supervision or as prediction.

5.5 Ablation Study

We firstly study the effect of confidence weight η_k and neural model on the performance. As shown in Table 5, by replacing learnable η_k with fixed $\eta_k = 1$, the performance drops a lot. It indicates that the learnable η_k can address conflicts among inconsistent signals. The performance also drops a lot when using the non-pretrained encoder, or directly setting the essay scores s_i as learnable parameters. It indicates that a good essay encoder can make full use of the textual information of essays to improve scoring performance.

We then study the effect of signals on the performance by removing some types of signals from supervision. As show in Table 5, the performance drops by about 0.02 after removing one type, and continues to drop after further removing another one. It indicates that all three types of quality signals are useful for model training.

We finally study the effect of using the four variants on the performance. As shown in Table 5, using the same 20 signals, aggregating signals during training (i.e., ULRA) is superior to aggregating before training (i.e., *averaged/aggregated signal as supervision*) or during inference (i.e., *averaged/aggregated output as prediction*).

5.6 Model Analysis

Effect of More Unlabeled Essays. We study the impact of varying the number of unlabeled essays on the performance of ULRA, i.e., whether our ULRA requires numerous unlabeled essays to achieve a good enough performance. To this end, we vary the ratio of essays for training from 0.2 to 1.0 step by 0.2. As shown in Figure 2(a), we can find that the lines show a trend that goes up first and then keeps stable after the ratio about 0.6. It indicates that about 60% of unlabeled essays are sufficient to train a good enough ULRA model.

Effect of More Training Pairs. We study the impact of varying the number of training pairs on the performance of ULRA, i.e., whether our ULRA framework can benefit from more training pairs. To this end, we vary the batch size from 2 to 32, so that the number of training pairs in a batch will accordingly vary from 1 to 496. As shown in Figure 2(b), we can find that all the lines show a trend that goes up. It indicates that a larger number of training pairs can lead to better performance.

Effect of Weak Signals. We study the impact of weak signals (i.e., low correlation with groundtruth

	P1	P2	P3	P4	P5	P6	P7	P8	Avg.
G	.840	.693	.688	.730	.807	.704	.730	.610	.725
N	.545	.551	**.645**	**.729**	.736	.554	.601	.300	.583
T	.576	.595	.631	.727	**.742**	.553	.673	.346	.605
U	.543	.568	.632	.728	.730	.554	.586	.296	.580
O	**.757**	**.621**	.547	.628	.664	**.562**	**.694**	**.450**	**.615**

Table 7: Performance of different scoring strategies. **G**, **N**, **T**, and **U** denote the scoring strategies based on the groundtruth, normal, triangle, and uniform distributions, respectively. **O** denotes our scoring strategy.

		P1	P2	P3	P4	P5	P6	P7	P8	Avg.
T	**G**	.674	.789	.998	.999	.922	.897	.812	.585	.835
	O	.757	.621	.547	.628	.664	.562	.694	.450	.615
I	**G**	.635	.610	.567	.842	.713	.769	.717	.448	.663
	O	.759	.508	.608	.644	.711	.577	.661	.446	.614

Table 8: Comparison the performance of applying ground-truth score as the quality signal (**G**) with that of applying 20 heuristic quality signals (**O**) under all 8 prompts of the ASAP dataset. **T** and **I** denote under the transductive and inductive settings, respectively.

scores) on the performance of ULRA. To this end, we add additional 0 to 80 weak signal(s) to the 20 quality signals. Note that the details can be seen in Appendix A. As shown in Figure 2(c), we can find that almost all lines show an overall downward trend. It indicates that weak signals will weaken the supervision and thus reduce the model performance. However, when between 0 and 10, all lines do not go down too much, and some lines even go up (e.g. prompt 1, 3, and 4). It indicates that ULRA is robust to weak signals if the weak signals are not yet dominant the signal set.

Effect of More Signals. We study the impact of the number of employed signals on the performance of ULRA, assuming that these signals have similar correlations with the groundtruth scores. To this end, we conduct experiments based on the best-N quality signals and the worst-N quality signals, according to QWKs with the groundtruth scores which are shown in Table 2. By varying N from 1 to 10, as shown in Figure 3, we can find that all the lines of best-N and most lines of worst-N show an upward trend. It indicates that more signals can often lead to better performance. By comparing the lines of best-N (with blue color) and worst-N (with red color), we can find that in most prompts, the performance differences between best-N and worst-N decrease with the growth of N. This may be because that more signals can help better address conflicts among signals, and therefore achieve more robust performance.

Effect of Confidence Weights. We study the impact of the learnable confident weights in ULRA. To this end, we calculate the Spearman's correlation coefficient the learned confidence weights and the corresponding QWKs listed in Table 2 under each prompt. As shown in Table 6, we can find that they are highly correlated under both transductive and inductive settings, which indicate that the learned confidence weights can indeed reflect the confidences of quality signals.

Effect of Different Scoring Strategies. We study the impact of different scoring strategies on the performance of ULRA. To this end, we test other four scoring strategies, which conduct score transformation based on predefined distributions. These distributions include groundtruth distribution (G), normal distribution (N), triangle distribution (T), and uniform distribution (U). Note that the details can be seen in Appendix B. As shown in Table 7, we can find that our scoring strategy outperforms all strategies except for the strategy based on groundtruth distribution. It indicates that our scoring strategy can adaptively learn the score distribution and ULRA can achieve better performance once the distribution of groundtruth score is known.

Groundtruth as Signal. In our ULRA framework, the applied 20 heuristic quality signals contain the information about score ranking, but also noise. We want to explore how the model would perform if the input signal contains no noise at all. Therefore, we conduct experiments by feeding the groundtruth scores as the signal in our ULRA. As shown in Table 8, the average QWK of our ULRA is 0.220 and 0.049 lower than which applies the groundtruth scores under the transductive and inductive settings. It indicates that the signals with less noise can help model to achieve better performance.

6 Conclusion

In this paper, we aim to perform essay scoring under the unsupervised setting. To this end, we propose a novel ULRA framework to train a neural AES model by aggregating the partial-order knowledge contained in multiple heuristic quality signals. To address the conflicts among different signals and get a unified supervision, we design a deep pairwise rank aggregation loss for model training. Experimental results demonstrate the effectiveness of ULRA for unsupervised essay scoring.

Limitations

Although our ULRA outperforms all unsupervised baseline methods, there are still some limitations.

The first limitation is that there is still a gap between the performance of our unsupervised method and that of some supervised methods. Although our ULRA can complete the AES task without label annotations, it is still worth exploring an unsupervised AES method whose performance is comparable to the state-of-the-art supervised method.

The second limitation is that the essay encoder which adopted in our ULRA (i.e., BERT) is pretrained on the English-based corpora, and the essays for training is also written by English. Thus, our ULRA works mostly for English, which means a well-trained ULRA model may fail to perform well on the essays written by other languages. An unsupervised AES system which supports multiple languages needs to be further explored.

The third limitation is that it requires about 25G GPU memory for training, which may fail on devices with small GPU memory. A possible solution is to set a smaller batch size, but this may take longer time. However, the evaluation process only requires about 2G GPU memory, which can run in most of GPU devices, or even CPU devices.

Acknowledgements

This work is supported by National Natural Science Foundation of China under Grant Nos. 61972192, 62172208, 61906085, 41972111. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

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 (Volume 1: Long Papers)*, pages 715–725, Berlin, Germany. Association for Computational Linguistics.

Jonathan Anderson. 1983. Lix and rix: Variations on a little-known readability index. *Journal of Reading*, 26(6):490–496.

Yigal Attali, Brent Bridgeman, and Catherine Trapani. 2010. Performance of a generic approach in automated essay scoring. *The Journal of Technology, Learning and Assessment*, 10(3).

Yigal Attali and Jill Burstein. 2004. Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series*, 2004(2):i–21.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Yue Cao, Hanqi Jin, Xiaojun Wan, and Zhiwei Yu. 2020. Domain-adaptive neural automated essay scoring. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1011–1020.

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1741–1752, Seattle, Washington, USA. Association for Computational Linguistics.

Xi Chen, Paul N Bennett, Kevyn Collins-Thompson, and Eric Horvitz. 2013. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 193–202.

Yen-Yu Chen, Chien-Liang Liu, Tao-Hsing Chang, and Chia-Hoang Lee. 2010. An unsupervised automated essay scoring system. *IEEE Computer Architecture Letters*, 25(05):61–67.

Mădălina Cozma, Andrei 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 (Volume 2: Short Papers)*, pages 503–509, Melbourne, Australia. Association for Computational Linguistics.

Edgar Dale and Jeanne S Chall. 1948. A formula for predicting readability: Instructions. *Educational research bulletin*, pages 37–54.

Ke Deng, Simeng Han, Kate J Li, and Jun S Liu. 2014. Bayesian aggregation of order-based rank data. *Journal of the American Statistical Association*, 109(507):1023–1039.

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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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)*, pages 153–162, Vancouver, Canada. Association for Computational Linguistics.

David F Gleich and Lek-heng Lim. 2011. Rank aggregation via nuclear norm minimization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 60–68.

Robert Gunning. 1969. The fog index after twenty years. *Journal of Business Communication*, 6(2):3–13.

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*, pages 2347–2356.

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 (Volume 1: Long Papers)*, pages 1088–1097, Melbourne, Australia. Association for Computational Linguistics.

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, IJCAI-19*, pages 6300–6308. International Joint Conferences on Artificial Intelligence Organization.

Rahul Kumar, Sandeep Mathias, Sriparna Saha, and Pushpak Bhattacharyya. 2022. Many hands make light work: Using essay traits to automatically score essays. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1485–1495, Seattle, United States. Association for Computational Linguistics.

Paraskevas Lagakis and Stavros Demetriadis. 2021. Automated essay scoring: A review of the field. In *2021 International Conference on Computer, Information and Telecommunication Systems (CITS)*, pages 1–6. IEEE.

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

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

R Duncan Luce. 2012. *Individual choice behavior: A theoretical analysis*. Courier Corporation.

Colin L Mallows. 1957. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130.

G Harry Mc Laughlin. 1969. Smog grading-a new readability formula. *Journal of reading*, 12(8):639–646.

Farah Nadeem, Huy Nguyen, Yang Liu, and Mari Ostendorf. 2019. Automated essay scoring with discourse-aware neural models. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 484–493, Florence, Italy. Association for Computational Linguistics.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Thomas Pfeiffer, Xi Alice Gao, Yiling Chen, Andrew Mao, and David G Rand. 2012. Adaptive polling for information aggregation. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Peter Phandi, Kian Ming A. 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*, pages 431–439, Lisbon, Portugal. Association for Computational Linguistics.

Tao Qin, Xiubo Geng, and Tie-Yan Liu. 2010. A new probabilistic model for rank aggregation. In *Advances in neural information processing systems*, pages 1948–1956.

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*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.

Yi Tay, Minh 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 AAAI conference on artificial intelligence*.

Louis L Thurstone. 1927. The method of paired comparisons for social values. *The Journal of Abnormal and Social Psychology*, 21(4):384.

Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yongjie Wang, Chuang Wang, Ruobing Li, and Hui Lin. 2022. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3416–3425, Seattle, United States. Association for Computational Linguistics.

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*, pages

791–797, Brussels, Belgium. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. 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*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Haoran Zhang and Diane Litman. 2021. Essay quality signals as weak supervision for source-based essay scoring. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 85–96, Online. Association for Computational Linguistics.

A Details of Weak Signals

In Section 5.6, we study the impact of using weak signals on the performance of our ULRA. We add additional 0 to 80 weak signals into the set of employed 20 quality signals. These 80 weak signals include: (1) mean of characters per word, (2) variance of characters per word, (3) mean of word per sentence, (4) variance of word per sentence, (5) mean of NNP per sentence, (6) number of PRP, (7) mean of PRP per sentence, (8) number of NNS, (9) mean of NNS per sentence, (10) number of VBZ, (11) mean of VBZ per sentence, (12) mean of DT per sentence, (13) mean of NN per sentence, (14) mean of RB per sentence, (15) number of POS, (16) mean of POS per sentence, (17) number of TO, (18) mean of TO per sentence, (19) number of VB, (20) mean of VB per sentence, (21) mean of JJ per sentence, (22) number of VBP, (23) mean of VBP per sentence, (24) mean of IN per sentence, (25) number of CC, (26) mean of CC per sentence, (27) number of VBG, (28) mean of VBG per sentence, (29) number of VBN, (30) mean of VBN per sentence, (31) number of WP, (32) mean of WP per sentence, (33) number of MD, (34) mean of MD per sentence, (35) number of WRB, (36) mean of WRB per sentence, (37) number of VBD, (38) mean of VBD per sentence, (39) number of NNPS, (40) mean of NNPS per sentence, (41) number of CD, (42) mean of CD per sentence, (43) number of JJR, (44) mean of JJR per sentence, (45) number of JJS, (46) mean of JJS per sentence, (47) number of RBR, (48) mean of RBR per sentence, (49) number of RBS, (50) mean of RBS per sentence, (51) number of RP, (52) mean of RP per sentence, (53) number of EX, (54) mean of EX per sentence, (55) number of WDT, (56) mean of WDT per sentence, (57) number of UH, (58) mean of UH per sentence, (59) number of PDT, (60) mean of PDT per sentence, (61) number of LS, (62) mean of LS per sentence,(63) mean of clause per sentence, (64) mean of clause length, (65) number of maximum of clause per sentence, (66) mean of tree depth of sentences, (67) mean of average leaf depth of sentences, (68) number of error words, (69) ratio of stop words, (70) ratio of positive sentiment, (71) ratio of negative sentiment, (72) ratio of neutral sentiment, (73) ratio of compound sentiment, (74) Kincaid index, (75) ARI index, (76) Coleman-Liau index, (77) Flesch Reading Ease index, (78) LIX, (79) sentence beginnings with pronoun, and (80) sentence beginnings with preposition.

Through comparing Table 9 and Table 2, we can find that the 80 weak signals are less correlated with the groundtruth scores, compared with the 20 quality signals used in ULRA.

	P1	P2	P3	P4	P5	P6	P7	P8
Max	.240	.317	.187	.348	.245	.179	.248	.173
Avg	.025	.039	.029	.039	.032	.018	.012	.013

Table 9: The maximum and mean of QWKs between the groundtruth scores and the employed 80 weak signals under 8 prompts, which are obtained by viewing the quality signal as the predicted score and applying the scoring strategy described in Section 4.4.

B Details of Different Scoring Strategies

In Section 5.6, we study the impact of different scoring strategies on the performance of our ULRA. We test other four scoring strategies, which conduct score transformations based on predefined distributions (i.e., groundtruth distribution, normal

distribution, triangle distribution, and uniform distribution).

For the scoring strategy based on **groundtruth distribution**, we denote the distribution of the groundtruth labels in $\mathcal{X}$ as $\{a_1, \cdots, a_L\}$, where a_i is the number of essays with the score $i \in \mathcal{Y}$. We fist rank all essays in $\mathcal{X}$ according to their predicted scores $\{s_i\}_{i=1}^N$, and get a rank list of essays $\{x_{r_1}, \cdots, x_{r_N}\}$, where r_i is the rank index. Finally, for each essay x_{r_i}, if its ranking index r_i satisfies $\sum_{j=1}^{t-1} a_j < r_i \leq \sum_{j=1}^{t} a_j$ for $t \in \mathcal{Y}$, the corresponding final score $\hat{y}_{r_i}$ is set as t.

For the scoring strategy based on **normal distribution**, we first rank all essays in $\mathcal{X}$ according to their predicted scores $\{s_i\}_{i=1}^N$, and get a rank list of essays $\{x_{r_1}, \cdots, x_{r_N}\}$, where r_i is the rank index. Next, we use the normal distribution $\mathcal{N}(\frac{L-1}{2}, 1)$ to calculate the proportion[2] of samples in i-th final score to the total number of samples after the score transformation for all $i \in \mathcal{Y}$, which is

$$\phi_i = \exp\left(-\left(i - 1 - \frac{L-1}{2}\right)^2 / 2\right) , \quad (10)$$

$$\Phi_i = \phi_i / \sum_{j=1}^{L} \phi_j , \quad (11)$$

where Φ_i is the proportion of samples in i-th final score to the total number of samples. Then, the sample number in i-th final score after the score transformation is

$$\Psi_i = \lfloor N\Phi_i \rfloor , \quad (12)$$

where $\lfloor \cdot \rfloor$ is the floor function. Finally, for each essay x_{r_i}, if its ranking index r_i satisfies $\sum_{j=0}^{t-1} \Psi_j < r_i \leq \sum_{j=0}^{t} \Psi_j$ for $t \in \mathcal{Y}$, the corresponding final score $\hat{y}_{r_i}$ is set as t. Note that we additionally define $\Phi_0 = 0$.

For the scoring strategy based on **triangle distribution**, we first rank all essays in $\mathcal{X}$ according to their predicted scores $\{s_i\}_{i=1}^N$, and get a rank list of essays $\{x_{r_1}, \cdots, x_{r_N}\}$, where r_i is the rank index. Then, we use the triangle distribution to calculate the proportion of samples in i-th final score to the total number of samples after the score transformation for all $i \in \mathcal{Y}$, which is

$$\phi_i = -\left|i - 1 - \frac{L-1}{2}\right| + \frac{L+1}{2} . \quad (13)$$

[2]Since the calculation is a proportion (e.g., Equation 11), the term $1/\sqrt{2\pi}$ is removed for simplicity in Equation 10.

Same as the scoring strategy based on normal distribution, the sample number in i-th final score after the score transformation is

$$\Psi_i = \lfloor N\phi_i / \sum_{j=1}^{L} \phi_j \rfloor . \quad (14)$$

Finally, for each essay x_{r_i}, if its ranking index r_i satisfies $\sum_{j=0}^{t-1} \Psi_j < r_i \leq \sum_{j=0}^{t} \Psi_j$ for $t \in \mathcal{Y}$, the corresponding final score $\hat{y}_{r_i}$ is set as t. Note that we additionally define $\Psi_0 = 0$.

For the scoring strategy based on **uniform distribution**, we fist rank all essays in $\mathcal{X}$ according to their predicted scores $\{s_i\}_{i=1}^N$, and get a rank list of essays $\{x_{r_1}, \cdots, x_{r_N}\}$, where r_i is the rank index. The final score $\hat{y}_{r_i}$ of x_{r_i} is set as $\lfloor \frac{L}{N} r_i \rfloor + 1$.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations

☒ A2. Did you discuss any potential risks of your work?
Our work provides methodological contributions that do not have direct boarder impacts. Although our work might indirectly lead to future researches and applications, it is premature to predict their positive or negative impacts.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 5.1

☑ B1. Did you cite the creators of artifacts you used?
Section 5.1

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 5.1

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 5.1

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section 5.1

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 5.1

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 5.1

C ☑ Did you run computational experiments?

Section 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 5.2 & Section Limitations

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 5.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5.2

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 5.2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.