Generalizable Automatic Short Answer Scoring via Prototypical Neural Network

Chapter · June 2023

DOI: 10.1007/978-3-031-36272-9_36

5 authors, including:

Li Lin
Shenzhen University
13 PUBLICATIONS **82** CITATIONS
SEE PROFILE

Quanlong Guan
Jinan University (Guangzhou, China)
70 PUBLICATIONS **486** CITATIONS
SEE PROFILE

Dragan Gasevic
The University of Edinburgh
747 PUBLICATIONS **22,741** CITATIONS
SEE PROFILE

Generalizable Automatic Short Answer Scoring via Prototypical Neural Network

Zijie Zeng[1], Lin Li[1], Quanlong Guan[2], Dragan Gašević[1], and Guanliang Chen[1,✉]

[1] Centre for Learning Analytics, Monash University, Melbourne, Australia
[2] Jinan University, Guangzhou, China
{zijie.zeng, lin.li, dragan.gasevic, guanliang.chen}@monash.edu, gql@jnu.edu.cn

Abstract. We investigated the challenging task of generalizable automatic short answer scoring (ASAS), where a scoring model is tasked with generalizing to target domains (provided only with limited labeled data) that have no overlap with the auxiliary domains on which the model is trained. To address this, we introduced a framework based on Prototypical Neural Network (PNN). Specifically, for a target short answer instance whose score needs to be determined, the framework first calculates the distance between this target instance and each cluster of support instances (support instances are a set of labeled short answer instances that are grouped to different clusters according to their labels, i.e., the ground-truth scores). Then, it rates the target instance using the ground-truth score of the cluster that has the closest distance to the target instance. Through extensive empirical studies on an open-source ASAS dataset consisting of 10 different question prompts, we observed that the proposed approach consistently outperformed other baselines across settings concerning different numbers of support instances. We further observed that the proposed approach performed better when with wider training data sources than when with restricted data sources for training, showing that including more data sources for training may add to the generalizability of the proposed framework.

Keywords: Generalizability · Automatic Short Answer Scoring · Few-shot Learning · Prototypical Neural Network.

1 Introduction

Short answer scoring aims at accurately evaluating the short textual responses written by learners based on certain grading criteria [1]. The completion of this task used to rely on the manual efforts of human instructors. On the one hand, manual grading can be rather accurate and reliable when performed by human experts (with enough time for scoring each assignment). On the other hand, the lack of qualified human experts may hinder the timely and accurate assessment of short text answers when facing a large number of students (e.g., the student-teacher ratio in MOOC can be up to 10,000:1 [18]). Moreover, it has

been documented that human graders can hardly keep their scoring precision and consistency at high levels throughout the whole grading process, i.e., grading performance can be of high quality at the beginning and then gradually becomes poor, which can result in unfairness to some students [9, 28]. Therefore, researchers have been exploring the feasibility of applying computational techniques to score short written text automatically, i.e., automatic short answer scoring (ASAS). ASAS approaches vary from the early rule-based methods [15] that rate a textual short answer based on the extent the answer matches the specific rules, to the traditional machine learning approaches [23, 17] based on hand-crafted features, then to the deep learning-based approaches [29, 25, 24] that can automatically engineer features from raw input text.

Certain ASAS approaches have demonstrated impressive scoring performance [29, 25, 24], e.g., the ASAS model proposed in [25] achieved predictive performance up to human-agreement levels on the task of scoring short textual responses from psychology domain. However, there remain issues that may prevent educators from adopting such models, i.e., their inabilities to generalize to unseen domains [5, 33] and the significant amount of manual efforts on labeling data required by machine learning or deep learning-based ASAS approaches [17, 29, 25] to train reliable models. To address this challenge, existing studies attempt to develop generalizable ASAS systems [5, 33], which are capable of generalizing to unseen domains (e.g., unseen questions and unseen prompts) when equipped with very limited labeled data (from the unseen domains) for reference. In this line of research, Condor et al. [5] investigated how the combinations of different input content (e.g., responses, question text, scoring rubric) affect the generalizability of the deep learning-based ASAS model. META [33] goes further to include a set of support instances (i.e., labeled short answer responses from the target domain) as an additional source of input content. Specifically, the support instances were appended to the end of the original input, i.e., the target response to be scored as well as other contextual contents. This practice of explicitly appending support instances to the input is expected to reduce the scoring task to the relatively easier task of finding similar instances [33].

Although META [33] has demonstrated state-of-the-art performance for generalizable ASAS, it suffers from the following limitations: (i) It requires that the scoring range of the training domains must be consistent with the scoring range of the testing domains, limiting its ability to generalize to real-world data sources of various scoring ranges; (ii) The maximum number of support instances is limited by the structure of its input format, i.e., it packs all relevant elements including the target response and the set of support instances to form a single input string. Note that some modern large language models have limits with regard to their input length (e.g., the input length limit is 512 tokens for BERT [6]). The above limitations might hamper META's ability to apply to real-world short answer scoring scenarios commonly seen where scoring ranges and subject areas might vary from task to task. As an example, the short answer scoring datasets used in this study (see Table 1) consisted of 10 prompts across various subjects and various scoring ranges, where the average length of responses could be up to 50

tokens for some prompts. To better adapt to such real-world scenarios, it is necessary to develop generalizable ASAS approaches that are free from the above limitations.

Therefore, in this paper, we intend to add to the existing generalizable ASAS studies by proposing a framework based on Prototypical Neural Network (PNN) [22], considering its success in generalizable image classification problem in computer vision. Specifically, for a target short answer instance whose score needs to be determined, the proposed framework scores it following two steps: (i) Calculate the distance between the target instance and each cluster of support instances (support instances are a set of labeled short answer instances that are grouped to different clusters according to their labels, i.e., the ground-truth scores); and (ii) Rate the target instance as score S, where S is the ground-truth score of the cluster that has the closest distance to the target instance. With the proposed generalizable ASAS framework, in this paper, we intended to investigate the following research question:

- To what extent can the model trained with the proposed ASAS framework based on PNN generalize to new domains (prompts or questions unseen during training)?

To answer this question, we conducted extensive empirical experiments on an open-source short answer scoring dataset and summarized the main findings as follows: (i) The proposed approach consistently outperformed other baselines across settings concerning different numbers of support instances; (ii) The proposed approach performed better when with wider training data sources than when with restricted data sources for training, showing that the generalizability of the proposed framework could be boosted by including more data sources for training.

2 Related Work

Our generalizable ASAS study can be included in the broader automatic text scoring (ATS) studies that aim at learning reliable models with limited annotation efforts on specific domians [12, 5, 32, 8, 13, 21, 33]. These studies are mainly driven by the fact that in real-world settings, scoring tasks oftentimes involve unseen domains about which little knowledge is known and the cost of learning domain-specific models from scratch can be fairly high. Jiang et al. [12] investigated automated essay scoring in the one-shot setting where they proposed to augment the one-shot data with pseudo-labeled data to guarantee effective training. Similar attempts can also be found in [13, 32, 8], where authors tried to reduce annotation efforts by only annotating the centroids of clusters that consisted of similar answers [32] or sampling a subset of the most informative essays to annotate [8]. Jin et al. [13] proposed a two-stage method where a set of pseudo-labeled essays were extracted based on the prediction of a model trained on non-target domains, which would be then used for training a target domain-specific model. Ridley et al. [21] developed a zero-shot neural-based model that

concatenates the part-of-speech embedding and a set of prompt-irrelevant features to form representations of the textual data, which were then used as the input to a linear layer for regression to predict the essay scores. Condor et al. [5] examined the generalizability of a series of supervised models when coupled with different types of text representation approaches (e.g., *Sentence-Bert, Word2Vec, Bag-of-words*) and various sources of input content (e.g., *question text, question context, rubric text, and question bundle identifier*). They showed that generalizability could hardly be achieved even by using the state-of-the-art sentence-BERT for text representation or by including context information such as question text. META [33] introduced labeled instances as additional input information to augment the query instances whose scores are to be predicted. Then, META employed a BERT-based classifier to predict over the augmented query instances. Our generalizable ASAS framework distinguishes itself from the existing studies for the following: (i) We focus on the task of scoring short textual answers, which is different from existing essay scoring studies [12, 8, 13, 21]; (ii) Our framework requires no further training over the previously unseen target domains and can make predictions with merely a few support instances from the target domain (i.e., different from existing related works [12, 13]); (iii) As a generalizable ASAS approach, our framework is similar to META [33] but with more flexibility w.r.t. the scoring range of train/test data. Moreover, our framework generates a representation for each query/support instance instead of concatenating query instances and support instances to form a single and long input string, which might exceed the length limit of some text encoders (e.g., BERT has an input limit of 512 tokens).

3 Method

In the setting of ASAS, generalizability refers to the capacity of a model to predict over previously unseen domains [33] (e.g., questions, prompts, or subject areas). We identified the few-shot learning based on PNN [22] as a promising approach to address the generalizable ASAS problem, considering its success in the generalizable image classification problem in computer vision [22]. Specifically, for a query instance Q to be scored, as well as the set of labeled support instances of different scores, denoted as S (note that both Q and S are from the same domain, whether it be the training domain or the testing domain), the PNN-based model scores Q following four steps: (i) Compute a low-dimensional representation Q' for Q with a certain text encoder f_θ; (ii) Compute a low-dimensional representation for each of the support instances from S and then group them into different clusters according to their ground-truth scores; (iii) Average the representations within each cluster to form a representation p for each score, i.e., prototype; (iv) Calculate the distances between Q' and each prototype p and then predict the score of Q as s, where s is the ground-truth score of the prototype that has the shortest distance from Q'. Note that when the distance metric d is fixed, accurate scoring requires that the text encoder f_θ with learnable parameters θ should be able to encode the instances so that of

Input: Text encoder f_θ with learnable parameters θ; Training data $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_r\}$ where $\mathcal{D}_i = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ...\}$ denotes the dataset of the ith domain; Test data (i.e., target domain) $\mathcal{D}_{test}$ and note that $\mathcal{D}_{test} \notin \mathcal{D}$; Number of training episodes N; Distance function d where $d(v_1, v_2)$ meansures the distance between v_1 and v_2; Number of support instances per class (score) N_S; Number of query instances per class (score) N_Q.
Output: Trained text encoder f_θ for short answer scoring on **unseen** domains

for *episode* $\in \{1, 2, ..., N\}$ **do**
  $\mathcal{D}_j \leftarrow$ RandomSample($\mathcal{D}$, 1) ▷ Select a domain for episode
  $\mathcal{C}_j \leftarrow$ The set of distinct scores in $\mathcal{D}_j$ ▷ Obtain the scoring range
  for score $i \in \mathcal{C}_j$ **do**
    $\mathcal{D}_{j,i} \leftarrow$ The set of instances in $\mathcal{D}_j$ with score i
    $S_i \leftarrow$ RandomSample($\mathcal{D}_{j,i}$, N_S) ▷ Select support instances
    $Q_i \leftarrow$ RandomSample($\mathcal{D}_{j,i} \backslash S_i$, N_Q) ▷ Select query instances
    $p_i \leftarrow \frac{1}{|S_i|} \sum_{(\mathbf{x},y) \in S_i} f_\theta(\mathbf{x})$ ▷ Compute prototype for score i
  end for
  $Q \leftarrow \bigcup_{i \in \mathcal{C}_j} Q_i$ ▷ Obtain all query instances
  for $(\mathbf{x}, y) \in Q$ **do**
    $p_y \leftarrow$ The prototype with score y
    for $p \in \{p_i | i \in \mathcal{C}_j \text{ and } p_i \neq p_y\}$ **do**
      $\Delta_{Dist} \leftarrow d(f_\theta(x), p) - d(f_\theta(x), p_y)$
      $\mathcal{L}_\theta \leftarrow \frac{1}{N_Q |\mathcal{C}_j|}[-\log(\sigma(\Delta_{Dist}))]$ ▷ Obtain the loss
      Compute the gradients for θ with respect to $\mathcal{L}_\theta$ ▷ Backward pass
    end for
  end for
  Update θ based on the gradients accumulated in this episode ▷ Update
end for
Predict over unseen data $\mathcal{D}_{test}$ (i.e., target domain) using f_θ ▷ Testing

Fig. 1. The training algorithm for generalizable ASAS. Note that RandomSample(S, N) returns a subset of N elements randomly selected from set S.

all the prototypes $\mathcal{P} = \{p_1, p_2, ...\}$, Q' should have the minimum distance from prototype p_q ($p_q \in \mathcal{P}$), where p_q shares the same score with Q'. This can be achieved by training the text encoder f_θ following the algorithm in Figure 1. The algorithm follows an episodic training paradigm widely adopted in few-shot learning studies [22, 10, 27, 11] and formalizes the scoring task as the C-way-N_S-shot-N_Q-query problem, i.e., at each episode, it mimics the few-shot task by predicting over a batch of query instances (i.e., N_Q query instances per score) using a few labeled support instances (i.e., N_S support instances per score) and minimizes the loss accordingly. Note that C is the number of distinct scores.

Generally, for a query instance x, the PNN-based model [22, 11, 16] employs the softmax over x's distances to the prototypes to generate a predicted distribution over all classes (i.e., distinct scores). The probability of x being predicted as score y is as follows [22]:

$$p_\theta(score = y|x) = \frac{exp(-d(f_\theta(x), p_y))}{\sum_{k'} exp(-d(f_\theta(x), p_{k'}))}, \quad (1)$$

where p_y is the prototype with score y. Then the negative log-probability loss for true class (let y be the ground-truth score here) is formalized as [22]:

$$\mathcal{L}_\theta^{'} = -\log p_\theta(score = y|x), \tag{2}$$

which will be minimized via certain optimization algorithms (e.g., SGD). However, the cost of computing score probabilities with softmax grows linearly with the number of classes (the sum term in the denominator of Equation (1), making it too expensive in real-world applications when the number of classes (i.e., distinct scores in ASAS setting) is large [3]. Moreover, in the backward passing stage, the optimization algorithm needs to compute the gradients with respect to the loss, which also requires a RAM linearly correlated with the number of classes to store parameters' gradients. Considering that $\mathcal{L}_\theta^{'}$ in Equation (2) is a function of the learnable parameters θ, of which the size can be up to millions for large language models (e.g., BERT-base [6] model has 110M parameters), we argue that minimizing the negative log loss in Equation (2) can be memory-demanding, particularly with domains of wide scoring range, i.e., containing a large set of distinct scores.

To manage the memory required for model training to a controllable magnitude, we alternatively formalize the loss in a pairwise manner as follows [31, 34, 20]: (i) Given the query instance x (encoded as $f_\theta(x)$) and a pair of prototypes $< p_y, p >$, where p_y shares the same ground-truth score with x while p is another prototype with a different ground-truth score, we denote the distance between $f_\theta(x)$ and prototype p_y as $d(f_\theta(x), p_y)$ and $d(f_\theta(x), p)$ is defined similarly; (ii) Then, we subtract $d(f_\theta(x), p_y)$ from $d(f_\theta(x), p)$ and denote the difference as Δ_{Dist} (Line 14 in Figure 1); (iii) We follow [20, 31] to employ $\sigma(\Delta_{Dist})$ as the proxy for the probability of $d(f_\theta(x), p) > d(f_\theta(x), p_y)$, where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. For example, when $d(f_\theta(x), p) - d(f_\theta(x), p_y) = 0.2$, we have $\sigma(0.2) = 0.55$, meaning that we have a 55% probability of $d(f_\theta(x), p) > d(f_\theta(x), p_y)$, i.e., a 55% chance of successfully rating x as the ground-truth score against the incorrect score; (iv) Finally, we maximize the probability $\sigma(\Delta_{Dist})$ by minimizing the following loss over the pair of distances $< d(f_\theta(x), p), d(f_\theta(x), p_y) >$ [31, 20]:

$$\mathcal{L}_\theta = -log(\sigma(\Delta_{Dist})), \tag{3}$$

note that $\mathcal{L}_\theta$ always involves two prototypes, i.e., the pair of $< p_y, p >$, making the required memory (RAM) needed irrelevant to the number of classes (scores) when computing the gradients of θ w.r.t $\mathcal{L}_\theta$ when performing backward passing (Line 16 in Figure 1). This enables the proposed framework to be applied to domains with wide scoring ranges when merely modest RAM is available[3].

[3] All the experiments were completed on NVIDIA Tesla T4 GPU with 16GB RAM.

4 Experiment

4.1 Data

We based our generalizable ASAS study on the Short Answer Scoring datasets[4], which contain about 17,000 short textual answers from US students of Grade 10. The datasets consist of 10 prompts across four subject areas. The adopted datasets distinguish themselves from those used in existing generalizable ASAS studies [33, 5] in that their scoring ranges and subject areas vary from prompt to prompt. Consequently, the existing studies [26, 29, 30] based on these datasets tend to build one prompt-specific model for each prompt without considering the model generalizability[5] across domains.

Table 1. Statistics of the ASAS dataset. ELA is short for English Language Arts.

Prompt ID	Subject	#Answers	Score Range	Average Length	Prompt ID	Subject	#Answers	Score Range	Average Length
1	Science	1672	0-3	47 words	6	Biology	1797	0-3	23 words
2	Science	1278	0-3	59 words	7	English	1799	0-2	41 words
3	ELA	1808	0-2	48 words	8	English	1799	0-2	53 words
4	ELA	1657	0-2	40 words	9	English	1798	0-2	50 words
5	Biology	1795	0-3	25 words	10	Science	1640	0-2	41 words

4.2 Baseline Methods

- **PNN-BERT**: PNN [22] is widely adopted as a few-shot learning framework for image classification [22, 11]. To adapt to the generalizable ASAS setting, we equip it with BERT [6] as the encoder to deal with textual input, considering that BERT has been the core component for many ASAS studies [26, 24, 33, 30]. Details of this approach can be found in Figure 1. Specifically, PNN-BERT was further broken down into three versions of different ranges of training sources:
  - *All*. PNN-BERT (*All*) was trained on all prompts except for the target being tested, e.g., when prompt 1 was the target prompt that our model needed to generalize to, we trained PNN-BERT (*All*) over prompt 2-10.
  - *Restricted*. PNN-BERT (*Restricted*) was trained only over the prompts that shared the same scoring range with the target prompt, e.g., when prompt 1 was the target prompt, we exerted restrictions on PNN-BERT so that it could only be trained over prompts 2, 5 and 6 because prompts 1, 2, 5 and 6 shared the same scoring range of $[0, 3]$. This restricted version was meant to match META w.r.t. its training sources[6].

[4] https://www.kaggle.com/competitions/asap-sas/data

[5] We follow [33] to define generalizability as the capacity of a model to predict over previously unseen domains.

[6] This restriction also applies to META due to its own limitation (see Section 1).

  - *NoTuned*. PNN-BERT (*NoTuned*) means that we directly adopted the pre-trained BERT as the encoder without further fine-tuning it.
- **META** [33]: META is one of the strongest baseline approaches for generalizable ASAS. For a query instance q from the target prompt U, META appends a series of labeled support instances (from the same prompt, i.e., U) to q to form the input string (usually very long), which is then put to the BERT-based classifier to predict the score of q. Note that for this method, only the first 512 tokens of the input string could be encoded and the exceeding part will be truncated due to the input limit of BERT.
- **SBERT-C** [2]: SBERT-C encodes both support instances and query instances with SBERT [19]. Then it computes the Canberra distance [14] between the query instance q and each of the support instances. Then it predicts the query instance q using M's ground-truth score, where M has the minimum distance from q among all support instances.

4.3 Training, Evaluation, Metrics and Implementation Details

Following Jin et al. [13], we adopted a prompt-wise cross validation for evaluation, i.e., ten-fold cross validation. In each fold, student answers of the target prompt were reserved for testing while answers from the remaining prompts were used for model training. In addition, 30% of the training data were reserved for validation and early-stopping would be triggered once no significant loss over the validation data was observed. Following existing few-shot learning studies [11, 22, 4], we adopted accuracy as the evaluation metric. We also reported quadratic weighted kappa (QWK) following existing ASAS studies [26, 33, 30].

For the text encoder, we adopted the pre-trained BERT-base-cased implemented in the Python package Transformers[7]. We followed its default settings to set dropout rate as 0.1 and activation function as GELU and adopt AdamW as the optimizer. For the episodic training, we adopt 5-shot, and 5-query setting as informed by previous studies [22, 7, 11]. Note that all methods were tested under different n-shot settings (i.e., 5-shot, 10-shot and 15-shot) to better understand the effects of n on their generalizability. We followed [22] to adopt Euclidean distance as the distance metric for PNN-BERT and searched the learning rate from $\{1e-5, 2e-5, 5e-5\}$ for all methods. As informed by previous studies [33, 5], we also adopted the prompt text as additional input content, which was by default appended to each query q for all methods in our implementations. It is noteworthy that none of the methods were fine-tuned on test (target) prompts due to the nature of our study being to investigate generalizability, which is also consistent with existing few-shot learning studies [22, 11]. All experiments were run on NVIDIA Tesla T4 GPU with 16GB RAM and the codes are available via `https://github.com/douglashiwo/GeneralizableASAS`.

5 Results

We summarized the results in Table 2 and organized our analysis as follows.

[7] https://github.com/huggingface/transformers

Overall Performance of the Proposed ASAS Framework. We observed that PNN-BERT (*All*) consistently outperformed all other approaches across different n-shot settings. Specifically, for different versions of the proposed PNN-BERT, we always had the observations w.r.t. the performance comparisons that *All* > *Restricted* > *NoTuned* (across different n-shot settings). Firstly, it is not surprising to see *NoTuned* being the worse because it had not been trained on any training data sources. Secondly, the observation of *All* > *Restricted* might suggest that the generalizability of the proposed PNN-BERT could benefit from including more data sources for training, even when the data sources have different scoring ranges with the target domain that the model is expected to generalize to. It should be pointed out that this observation could be of great significance to our proposed framework. Because one of the advantages of our proposed framework against META (the state-of-the-art baseline in generalizable ASAS) is the flexibility that our proposed framework requires no consistency between the scoring ranges of the train data and that of the test data. The observation of *All* > *Restricted* might suggest that the generalizability of the proposed framework can benefit from such flexibility.

Table 2. Experiment results (testing) of different generalizable ASAS approaches under the n-shot settings. Note that each entry of the table is the average over all the 10 prompts while the result for each prompt is the average over 500 episodes.

Method	Test Settings (n-shot)					
	Accuracy			QWK		
	5-shot	10-shot	15-shot	5-shot	10-shot	15-shot
PNN-BERT (*All*)	0.453	0.462	0.463	0.387	0.410	0.415
PNN-BERT (*Restricted*)	0.425	0.448	0.446	0.332	0.370	0.379
PNN-BERT (*NoTuned*)	0.365	0.376	0.380	0.183	0.206	0.216
META	0.430	0.432	0.430	0.364	0.362	0.363
SBERT-C	0.422	0.448	0.455	0.311	0.361	0.376

Effects of the Number of Shots. We observed that all methods, except META, performed better as the number of shots grew. For PNN-BERT, this could be explained as follows: the prototypes of the classes (i.e., distinct scores) could be more accurately represented as more support instances (shots) were provided, leading to better classification (scoring) performance. This is consistent with existing few-shot learning studies [10, 22]. Similar to PNN-BERT, SBERT-C also adopts a distance-based idea for classification but it directly assigns a query instance q to the support instance s that has the shortest distance to q, i.e., rating q with the score of s. When the number of support instances (i.e., n) increased, the query q became more likely to be correctly assigned to a support instance that shares the same score with q. Finally, we noticed that META performed quite well under the 5-shot setting (the second best and only worse than PNN-BERT (*All*)). However, its performance got no better (stayed stable)

as the number of shots (i.e., n) increased. This can be explained by its limitation mentioned in Section 1, i.e., it augments the query instance by appending to it all support instances to form a single input string, which would probably exceed the length limit of the adopted text encoder (they adopt BERT as the encoder and it accepted input of 512 tokens at most). Such practice is acceptable when the average length of the instances is short, which was actually the case for the ASAS study on mathematics domain [33] where META was first proposed. However, for datasets with relatively long average instance lengths, the disadvantage caused by this limitation becomes manifest. For example, when we were testing META on prompt 2 in the 5-shot setting, we would have a string of length l (estimated) when we packed all the support instances together to form a single input, where $l = 59 * (4 * 5 + 1) + c > 512$. Note that 59 is the average length of instances from prompt 2 (see Table 1), 4 is the number of distinct scores for prompt 2, and 5 is the number of shots while 1 means that we have 1 query and c denotes the length of other input contents (e.g., prompt text was used in our case). Note that only the first 512 tokens could be encoded and the exceeding part will be truncated, i.e., the additional shots beyond the limit were actually invalid and could unlikely bring any improvement. To summarize, due to the input length limit of the adopted text encoder (i.e., BERT) and the way it constructed its input, META eventually failed to benefit from increasing the number of shots. We could see the performance of META dropped from the second best (out of five methods) in the 5-shot setting to the second worse in the 15-shot settings.

6 Conclusion and Future Work

In this paper, we proposed a framework based on PNN [22] for the generalizable ASAS problem. Through extensive empirical studies on the open-source ASAS datasets, we summarized the following findings: (i) Across different n-shot settings (n denotes the number of support instances for each class, i.e., score), we observed that the proposed method consistently outperformed other baselines, including the state-of-the-art baseline META [33]; (ii) The generalizability of the proposed framework could benefit from including more data sources for training; (iii) The performance of the proposed approach improved as more and more support instances from the target domains were provided. For future work, we would like to extend our empirical experiments by including other text encoders (e.g., Sentence-BERT [19]). Another potential direction would be to evaluate the effectiveness of the proposed framework on automatic essay scoring (AES), which is considered a more challenging task.

References

1. Alikaniotis, D., Yannakoudakis, H., Rei, M.: Automatic text scoring using neural networks. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 715–725 (2016)

2. Baral, S., Botelho, A.F., Erickson, J.A., Benachamardi, P., Heffernan, N.T.: Improving automated scoring of student open responses in mathematics. International Educational Data Mining Society (2021)
3. Blanc, G., Rendle, S.: Adaptive sampled softmax with kernel based sampling. In: International Conference on Machine Learning. pp. 590–599. PMLR (2018)
4. Boney, R., Ilin, A., et al.: Active one-shot learning with prototypical networks. In: ESANN (2019)
5. Condor, A., Litster, M., Pardos, Z.: Automatic short answer grading with sbert on out-of-sample questions. In: Proceedings of the 14th International Conference on Educational Data Mining (2021)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: 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). pp. 4171–4186 (2019)
7. Dong, N., Xing, E.P.: Few-shot semantic segmentation with prototype learning. In: BMVC. vol. 3 (2018)
8. Dronen, N., Foltz, P.W., Habermehl, K.: Effective sampling for large-scale automated writing evaluation systems. In: Proceedings of the second (2015) ACM conference on learning@ scale. pp. 3–10 (2015)
9. Fazal, A., Dillon, T., Chang, E.: Noise reduction in essay datasets for automated essay grading. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". pp. 484–493. Springer (2011)
10. Geng, R., Li, B., Li, Y., Zhu, X., Jian, P., Sun, J.: Induction networks for few-shot text classification. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3904–3913 (2019)
11. Jakubik, J., Blumenstiel, B., Voessing, M., Hemmer, P.: Instance selection mechanisms for human-in-the-loop systems in few-shot learning **6** (2022)
12. Jiang, Z., Liu, M., Yin, Y., Yu, H., Cheng, Z., Gu, Q.: Learning from graph propagation via ordinal distillation for one-shot automated essay scoring. In: Proceedings of the Web Conference 2021. pp. 2347–2356 (2021)
13. Jin, C., He, B., Hui, K., Sun, L.: 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). pp. 1088–1097 (2018)
14. Jurman, G., Riccadonna, S., Visintainer, R., Furlanello, C.: Canberra distance on ranked lists. In: Advances in Ranking NIPS 09 Workshop (2009)
15. Leacock, C., Chodorow, M.: C-rater: Automated scoring of short-answer questions. Computers and the Humanities **37**(4), 389–405 (2003)
16. Li, O., Liu, H., Chen, C., Rudin, C.: Deep learning for case-based reasoning through prototypes: A neural network that explains its predictions. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
17. Nau, J., Haendchen Filho, A., Passero, G.: Evaluating semantic analysis methods for short answer grading using linear regression. Sciences **3**(2), 437–450 (2017)
18. Pappano, L.: The year of the mooc. The New York Times **2**(12), 2012 (2012)
19. Reimers, N., Gurevych, I.: Sentence-bert: Sentence embeddings using siamese bert-networks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3982–3992 (2019)

20. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence. pp. 452–461 (2009)
21. Ridley, R., He, L., Dai, X., Huang, S., Chen, J.: Prompt agnostic essay scorer: A domain generalization approach to cross-prompt automated essay scoring. arXiv preprint arXiv:2008.01441 (2020)
22. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems **30** (2017)
23. Sultan, M.A., Salazar, C., Sumner, T.: Fast and easy short answer grading with high accuracy. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1070–1075 (2016)
24. Sung, C., Dhamecha, T., Saha, S., Ma, T., Reddy, V., Arora, R.: Pre-training bert on domain resources for short answer grading. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 6071–6075 (2019)
25. Sung, C., Dhamecha, T.I., Mukhi, N.: Improving short answer grading using transformer-based pre-training. In: International Conference on Artificial Intelligence in Education. pp. 469–481. Springer (2019)
26. Surya, K., Gayakwad, E., Nallakaruppan, M.: Deep learning for short answer scoring. Int. J. Recent Technol. Eng **7**(6), 1712–1715 (2019)
27. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. Advances in neural information processing systems **29** (2016)
28. Wind, S.A., Peterson, M.E.: A systematic review of methods for evaluating rating quality in language assessment. Language Testing **35**(2), 161–192 (2018)
29. Xia, L., Guan, M., Liu, J., Cao, X., Luo, D.: Attention-based bidirectional long short-term memory neural network for short answer scoring. In: International Conference on Machine Learning and Intelligent Communications. pp. 104–112. Springer (2020)
30. Zeng, Z., Li, X., Gasevic, D., Chen, G.: Do deep neural nets display human-like attention in short answer scoring? In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 191–205 (2022)
31. Zeng, Z., Lin, J., Li, L., Pan, W., Ming, Z.: Next-item recommendation via collaborative filtering with bidirectional item similarity. ACM Transactions on Information Systems (TOIS) **38**(1), 1–22 (2019)
32. Zesch, T., Heilman, M., Cahill, A.: Reducing annotation efforts in supervised short answer scoring. In: Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications. pp. 124–132 (2015)
33. Zhang, M., Baral, S., Heffernan, N., Lan, A.: Automatic short math answer grading via in-context meta-learning. In: Proceedings of the 15th International Conference on Educational Data Mining (2022)
34. Zhu, Z., Wang, J., Caverlee, J.: Measuring and mitigating item under-recommendation bias in personalized ranking systems. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval. pp. 449–458 (2020)