Improving Short Answer Grading Using Transformer-Based Pre-training

Chul Sung[1(✉)], Tejas Indulal Dhamecha[2], and Nirmal Mukhi[1]

[1] IBM Watson Education, Yorktown Heights, NY 10598, USA
{sungc,nmukhi}@us.ibm.com
[2] IBM Research, Bangalore, India
tidhamecha@in.ibm.com

Abstract. Dialogue-based tutoring platforms have shown great promise in helping individual students improve mastery. Short answer grading is a crucial component of such platforms. However, generative short answer grading using the same platform for diverse disciplines and titles is a crucial challenge due to data distribution variations across domains and a frequent occurrence of non-sentential answers. Recent NLP research has introduced novel deep learning architectures such as the Transformer, which merely uses self-attention mechanisms. Pre-trained models based on the Transformer architecture have been used to produce impressive results across a range of NLP tasks. In this work, we experiment with fine-tuning a pre-trained self-attention language model, namely Bidirectional Encoder Representations from Transformers (BERT) applying it to short answer grading, and show that it produces superior results across multiple domains. On the benchmarking dataset of SemEval-2013, we report up to 10% absolute improvement in macro-average-F1 over state-of-the-art results. On our two psychology domain datasets, the fine-tuned model yields classification almost up to the human-agreement levels. Moreover, we study the effectiveness of fine-tuning as a function of the size of the task-specific labeled data, the number of training epochs, and its generalizability to cross-domain and join-domain scenarios.

Keywords: Self-attention · Transfer learning · Student answer scoring

1 Introduction

Dialogue-based tutoring (DBT) platforms such as AutoTutor [6], Rimac [1], DeepTutor [24] and the Watson Tutor [28] have shown great promise in meeting individual student's needs. In such systems, the tutoring platform interacts with the student by asking questions and provides individual feedback based on all student answers. To provide appropriate feedback and rectify student mistakes, accurately understanding student answers is crucial. However, devising a generic short answer grading system that performs well across different questions and

S. Isotani et al. (Eds.): AIED 2019, LNAI 11625, pp. 469–481, 2019.
https://doi.org/10.1007/978-3-030-23204-7_39

domains of study is a challenge due to data distribution variations (differences in used language, length and depth of answers, use of non-sentential answers, among other issues).

Various Deep Learning (DL) based techniques have been explored for short answer grading [2,11,12,17,25]. However, availability of limited labeled data (reference and student answer pairs) often prohibits meaningful training; furthermore, due to domain discrepancy between the public corpora and short answer grading corpus, the utilization of the former by augmentation is not efficient. Lately, transfer learning has largely supplanted the use of the older DL techniques, and have had a substantial impact on the state of Natural Language Processing (NLP) [16]. The main concept within transfer learning is to apply the knowledge from one or more source tasks to a target task [18]. Broadly, a target task can use the knowledge of labeled data from other tasks or from unlabeled data called *self-taught learning* [21]. In NLP, word embedding is one of the most influential transfer models due to its capability of capturing semantic context of a word by producing vector representations of words from large unlabeled corpora such as Wikipedia and news articles [13].

As a transition of a robust transfer learning model, Peters *et al.* introduced contextualized word representations (called Embeddings from Language Models or ELMo) [19]. ELMo captured contextual information from word representations by combining the hidden states of multiple bidirectional LSTMs and initial embeddings. In 2018, diverse novel fine-tuning language models such as Universal Language Model Fine-tuning (ULMFiT) [9] and OpenAI′s Generative Pre-Training (GPT)[1] [20] were proposed followed by a robust transfer language model called Bidirectional Encoder Representations from Transformers, or BERT [5]. OpenAI′s GPT and BERT adapted the Transformer architecture to learn the text representations, a novel and efficient language model architecture based on a self-attention mechanism [27]. However, while OpenAI′s GPT used an unidirectional attention approach (the decoder in Transformer), BERT used a bidirectional one (the encoder in Transformer) to better understand the text context. BERT can be trained in two phases. In the *pre-training phase*, deep bidirectional representations inherited by the nature of the Transformer Encoder can use unlabeled huge corpora. In the *fine-tuning phase*, task-specific labeled data and parameter tuning is performed to optimize results for a specific problem, such as question answering or short answer grading.

In this work, we experiment with fine-tuning a pre-trained BERT language model and explore the following questions:

- How well do Transformer-based DL approaches (we use BERT as it is the latest iteration of such models) apply to short answer grading?
- How much does fine-tuning, involving the collection of domain-specific labeled answers, impact the results obtained?
- What is the amount of training (number of epochs) needed in order to produce an optimized model using this approach?
- How well does the same fine-tuned Transformer-based model work across different domains of study for the short answer scoring task?

[1] https://blog.openai.com/language-unsupervised/.

We begin with an overview of recent approaches in short answer grading, and an overview of BERT and the BERT model architecture, before presenting details on our experiments designed to answer these questions.

2 Related Work

Broadly speaking the literature pertaining to the problem of short answer grading can be categorized into two: (1) earlier approaches that relied heavily on hand-crafted features, and (2) recent deep learning approaches that require minimum, if not none at all, feature engineering.

2.1 Hand-Crafted Features

Mohler and Mihalcea [15] and Mohler *et al.* [14] are among the earliest research works towards automatic short answer grading. These approaches relied on various word similarity measures, corpus-based measures, and alignment of parses of reference and student answers. A benchmark in the field was established with the Student Response Analysis Challenge as part of SemEval-2013 [7]. Participating approaches relied on a range of hand-crafted features including corpus-based word similarities, WordNet based word similarities, part-of-speech tags, sentence parsing, and n-grams; one of the participants also explored domain adaptation. Broadly, the problem of Student Response Analysis is modeled as a special case of Textual Entailment or Semantic Textual Similarity. Ramachandran *et al.* [22] proposed to extract phrase patterns from reference answers to form basis of scoring approach. The approach improves over earlier approaches in that it explicitly extracts semantic information at sentence as opposed to earlier word similarity metrics. Ramachandran and Foltz [23] proposed a short answer grading based on text summarization.

2.2 Deep Learning Approaches

With the advances in deep learning approaches, various works leveraged these approaches. Sultan *et al.* [26] represented a sentence as sum of word embeddings [13] of its tokens in conjunction with other features. The approach uses word embeddings obtained by deep learning on large corpus; however, obtaining feature representations of a sentence as sum of word embeddings ignores the structural information. Thus, as a logical extension subsequent works have explored more sophisticated ways to obtain feature representations of answer sentences. Mueller and Thyagarajan [17] proposed a Long Short-Term Memory (LSTM) based Siamese network to compare student answer against reference answer. They observe that one of the major limitations in training LSTM networks is the lack of large amount of training data. They generate additional pairs of answers by replacing words in the original dataset. The extended dataset is used for training LSTM networks for short answer grading. The data intensive nature of deep learning approaches has emerged as an interesting issue for research, particularly in data-starved problems such as short answer grading.

Transfer Learning has evolved into a promising research direction to address this. It claims that a generic learning of natural language can be obtained from a data-rich generic task, which can be then *transferred* to downstream tasks which may have limited data. Research efforts to learn universal sentence embeddings for task-specific transfer have yielded impressive improvements on various benchmarks. Notable works include InferSent [4], ELMo [19], ULMFiT [9], GPT [20], and BERT [5]. Saha *et al.* [25] explored sentence embedding features from InferSent in conjunction with traditional token features. In another recent work, Marvaniya *et al.* [12] showed that short answer grading based on sentence embedding features can be further improved by leveraging their proposed scoring rubric approach. The current state of the short answer grading research has shown that transfer of sentence embeddings is useful, yet non-contextual approaches encounter their limitations at downstream tasks. In this study, we aim to demonstrate the ability and various characteristics of BERT (a latest and robust transfer language model) for short answer grading with limited domain-specific training data.

3 BERT for Short Answer Grading

The broad premise of BERT [5] is that there is a high-level language model that needs to be encoded into the network irrespective of the downstream task. The high-level language model is learned based on two semi-supervised objectives of (1) Masked Language Model (MLM) for a deep bidirectional representation and (2) Next Sentence Prediction (NSP) for understanding relationship between sentences; this training leverages multiple corpora. The resultant model, often called the pre-trained BERT model, forms the basis for downstream target tasks. For the task of short answer grading, we perform fine-tuning in the form of *Sentence Pair Classification*. This model allows to classify a pair of reference and student answers into desired categories of correct, incorrect, contradiction, and so on.

3.1 BERT Model Architecture

As described in Devlin *et al.* [5], BERT takes a single token sequence from a single text sentence for the MLM objective or from a pair of text sentences (adding `[SEP]` token between them as a separator) for the NSP objective. The special classification embedding `[CLS]` is added in front of each sequence and it is used as input to the classification-task layer. As shown in Fig. 1, the input representations are obtained by combining the token, segment, and *learned* position embeddings. The segment embeddings identify which sentence tokens are from and the position embeddings relative positioning of tokens. This is the input to the first Transformer Encoder layer and the output of this layer is fed into the next Transformer layer. BERT may have a stack of multiple Transformer layers. Each Transformer Encoder is composed of two major parts: a self-attention layer with multiple attention heads, followed by token-wise feed-forward layers.

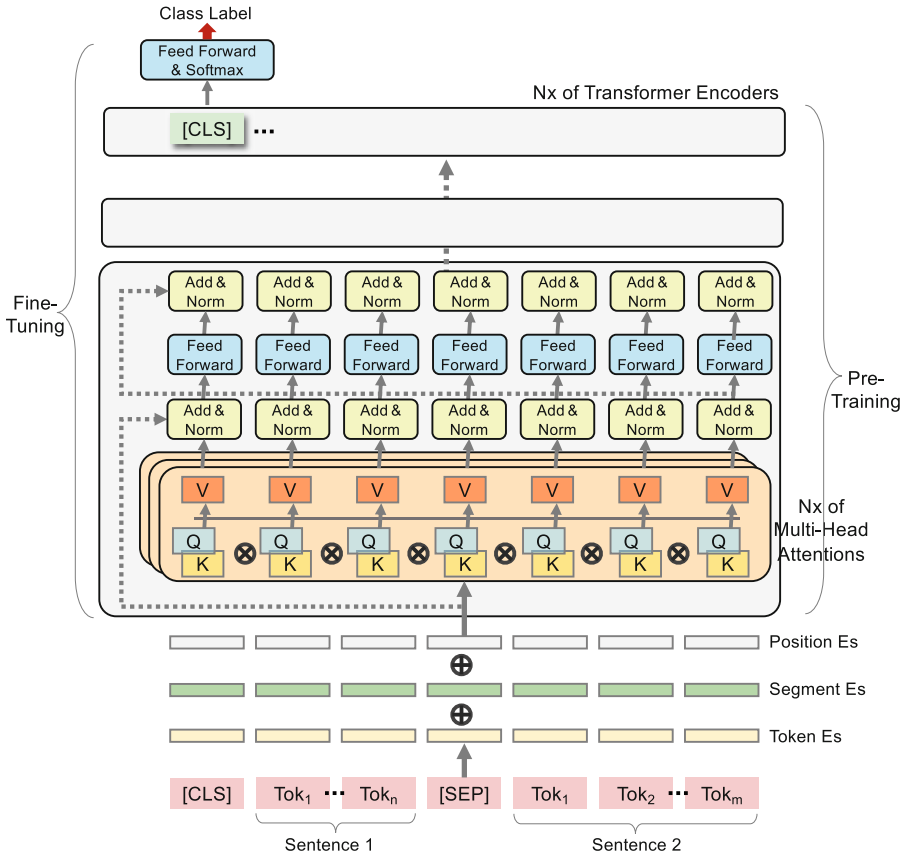

Fig. 1. BERT model architecture for short answer grading. We employed the *Sentence Pair Classification* task specific model using BERT. To describe the details of the model we used the same colors for the same representations as in [5,27].

Each attention head acts akin to a convolution in a convolutional neural network (ConvNet), except for a weighted average. As part of self-attention mechanism, BERT computes three vectors from each token (called `query`, `key`, and `value`) by multiplying three trainable weight matrices (W^Q, W^K, W^V respectively). The weight matrices emphasize different location values of the input as the role of kernels in ConvNet and they are adjusted for every head.

$$\boldsymbol{q}_j^i = \boldsymbol{x}_j W_i^Q \qquad \boldsymbol{k}_j^i = \boldsymbol{x}_j W_i^K \qquad \boldsymbol{v}_j^i = \boldsymbol{x}_j W_i^V \tag{1}$$

where, $\boldsymbol{q}_j^i$, $\boldsymbol{k}_j^i$, and $\boldsymbol{v}_j^i$ are the `query`, `key`, and `value` vectors (projections) respectively for jth token x_j in ith head. Then, with the `query` and `key` vectors BERT calculates attention weights by: (1) the dot product of the `query` vector of a

particular token and all the `key` vectors ($\boldsymbol{k}_1^i ... \boldsymbol{k}_n^i$ in ith head where n is the number of tokens), (2) an adjustment of the dot products by $\frac{1}{\sqrt{d_k}}$ where d_k is the dimension of the key vectors, and (3) a softmax normalization sequentially. The scaling factor of $\frac{1}{\sqrt{d_k}}$ helps finely adjust larger vectors to avoid extremely small gradients from the softmax.

$$aw_{j1}^i, ..., aw_{jn}^i = \texttt{softmax}((\boldsymbol{q}_j^i \cdot \boldsymbol{k}_1^i, ..., \boldsymbol{q}_j^i \cdot \boldsymbol{k}_n^i)\frac{1}{\sqrt{d_k}})) \tag{2}$$

where aw_{jk}^i is kth normalized attention weight for jth token in ith head. The attention weights capture how much all tokens are related to a particular token in head_i. BERT multiplies each `value` vector by the corresponding attention weight and sums up the weighted results. The output vector contains the bi-directional attention information, the `value` vectors of related tokens contributing more than others.

$$\boldsymbol{z}_j^i = aw_{j1}^i \boldsymbol{v}_1^i + ... + aw_{jn}^i \boldsymbol{v}_n^i \tag{3}$$

where $\boldsymbol{z}_j^i$ is the output of a self-attention layer for jth token and $\boldsymbol{v}_k^i$ is kth `value` vector in ith head. There may be multiple $\boldsymbol{z}_j$ from multiple attention heads. To aggregate these results, BERT concatenates all $\boldsymbol{z}_j$ vectors, multiplying them by a weight matrix. The result vector having all attention information along all heads is summed with the original token representations, followed by layer normalization [3]. Each of the final vectors (representing a particular token) discretely goes to the corresponding fully connected feed-forward network. This full procedure repeats as many as the number of Transformer Encoders and at the last Transformer Encoder the final output for the `[CLS]` token is used as the sequence representation. Up to this point, this is the pre-training model and BERT can leverage an unlabeled huge corpus of text to construct a high-level language model. Then, BERT adapts the labeled data for short answer grading not only for fine tuning the pre-training model but also constructing a classification model through the feed-forward classification layer on the pre-training model.

4 Experiments

We evaluated our proposed approach on two datasets:

1. **SciEntsBank-3way dataset of SemEval-2013 [7]:** We used SciEntsBank dataset for the 3-way task in SemEval 2013 challenge. The data consists of questions, reference answers, student answers, and three-way labels (correct, incorrect, and contradictory or in short co, ic, and cd respectively) in the science domain. The SemEval 2013 challenge involves three classification subtasks on three given test sets: unseen answers (UA), unseen questions (UQ), and unseen domains (UD).

2. **Two psychology domain datasets:** The datasets contain a collection of questions, reference answers, student answers, and three-way labels (CORRECT, PARTIALLY-CORRECT, and INCORRECT or in short CO, PC, and IC respectively). These are based on student answers from two psychology-related textbooks (one is from behavioral physiology and has a lot of technical language and the other is from developmental psychology with mostly non-technical material). Each student response is manually annotated by three experts. Groundtruth is obtained as majority voting of the three annotations.

As shown in the Table 1, the class distribution of both datasets is highly skewed. Due to the class imbalance we select a macro-average-F1 method to observe how our proposed approach preforms overall across the latest other approaches. The macro-average-F1 computes the F1 score independently for each class and then takes the average of all F1 scores. Moreover, we report results in terms of accuracy and weighted-average-F1, but due to the class-imbalance in the datasets, these two metrics may provide biased evidences.

Table 1. Details of class distribution and train-test split protocols for SCIENTSBANK 3-way dataset of SemEval 2013 challenge and our psychology domain 1 and 2 datasets. The test set of SCIENTSBANK is divided into three different test sets for the three subtasks: unseen answers (UA), unseen questions (UQ), and unseen domains (UD).

Dataset	Class distribution			Train-test split	
				Training	Test
SemEval-2013 [7]	4,459 (CO)	5,307 (IC)	1,038 (CD)	4,969	540 (UA) 733 (UQ) 4,562 (UD)
Psychology domain 1	14,460 (CO)	3,845 (PC)	1,790 (IC)	16,076	4,019
Psychology domain 2	12,295 (CO)	2,495 (PC)	1,090 (IC)	12,704	3,176

4.1 Pre-training Setup

We chose **BERT$_{\text{BASE}}$**, **Uncased**[2] pre-trained model, which used the concatenation of BooksCorpus (800M words) and English Wikipedia (2,500M words) for pre-training. **Uncased** means that the text has been converted to lower-case before tokenization, dropping any accent markers. BERT uses WordPiece embeddings [29] using a 30,000 token vocabulary and up to 512 tokens are supported for the input sequence. The details of the BERT$_{\text{BASE}}$ model can be found in [5].

4.2 Fine-Tuning Setup

For fine-tuning the pre-trained BERT$_{\text{BASE}}$ model and a classification layer, we generated the two datasets in tab-separated values (TSV) files. We changed the learning rate of Adam optimizer to 2e–5 for SemEval-2013 and 3e–5 for two

[2] https://github.com/google-research/bert.

psychology domain datasets with the same batch size 32. We have also gradually reduced the training size up to 20% of the entire set to observe how many labeled data are required for fine-tuning. We changed the number of epochs from 4 to 12 to observe how many epochs the BERT and classifier are required to complete fine-tuning. For the fine-tuning process, we used two NVIDIA Tesla P100 GPUs (Graphics Card RAM 16 GB) and 120-GB memory.

Table 2. Performance on SciEntsBank Dataset of SemEval-2013 [7]. All results of ‡ are as reported in [25]. MEAD [23], Graph [23] and Marvaniya *et al.* [12] reported results on unseen answer protocol only as their approaches are designed for this scenario. Accuracy (Acc), macro-average-F1 (M-F1), and weighted-average-F1 (W-F1) are reported in percentage.

	Unseen answer			Unseen question			Unseen domain		
	Acc	M-F1	W-F1	Acc	M-F1	W-F1	Acc	M-F1	W-F1
Baseline [7]	55.6	40.5	52.3	54.0	39.0	52.0	57.7	41.6	55.4
ETS [8]	72.0	64.7	70.8	58.3	39.3	53.7	54.3	33.3	46.1
SOFTCAR [10]	65.9	55.5	64.7	65.2	46.9	63.4	63.7	48.6	62.0
MEAD [23]	-	42.9	55.4	-					
Graph [23]	-	43.8	56.7	-					
Sultan *et al.* [26]‡	60.4	44.4	57.0	64.3	45.5	61.5	62.7	45.2	60.3
Saha *et al.* [25]	71.8	66.6	71.4	61.4	49.1	62.8	63.2	47.9	61.2
Marvaniya *et al.* [12]	-	63.6	71.9	-					
Proposed **$\text{BERT}_{\text{BASE}}$**	**75.9**	**72.0**	**75.8**	**65.3**	**57.5**	**64.8**	**63.8**	**57.9**	**63.4**

4.3 Results and Analysis

We performed a set of experiments to study various aspects of the proposed $\text{BERT}_{\text{BASE}}$ model for the problem of short answer grading, including (1) performance comparison with published literature and human agreements, (2) sufficiency of fine-tuning in terms of supervised data requirement and the number of training epochs, (3) applicability of fine-tuned model on different domain, and (4) ability to jointly fine-tune for multiple domains. Based on the various experiments and their results presented on benchmark SciEntsBank dataset and our two psychology domain datasets, we make following key observations:

Table 3. Performance comparison of human agreements and the proposed method on our two psychology (psych.) domain datasets. Accuracy (Acc), macro-average-F1 (M-F1), and weighted-average-F1 (W-F1) are reported in percentage.

	Psych. domain 1			Psych. domain 2		
	Acc	M-F1	W-F1	Acc	M-F1	W-F1
Majority-vote vs. Human1	86.0	77.4	86.5	91.2	81.8	91.0
Majority-vote vs. Human2	89.4	81.1	89.6	88.9	80.9	89.1
Majority-vote vs. Human3	85.7	78.0	86.0	87.6	79.8	88.4
Proposed **$\text{BERT}_{\text{BASE}}$**	**91.8**	**85.7**	**91.8**	**91.0**	**82.2**	**91.0**

Effectiveness of Transfer Learning: As shown in Tables 2 and 3, on all the datasets the fine-tuned model yields impressive results. On SciEntsBank dataset, we establish state-of-the-art results. Compared to state-of-the-art, Saha *et al.* [25], which includes sentence embeddings of InferSent [4] along with token features, we report improvements ranging from 6% up to 10% in macro-average-F1. Note that, unsupervised pre-training of BERT helps to leverage a huge amount of existing natural language material. This puts the approach at an advantage over techniques such as InferSent [4] that requires large supervised (and therefore expensive and limited) corpus for pre-training.

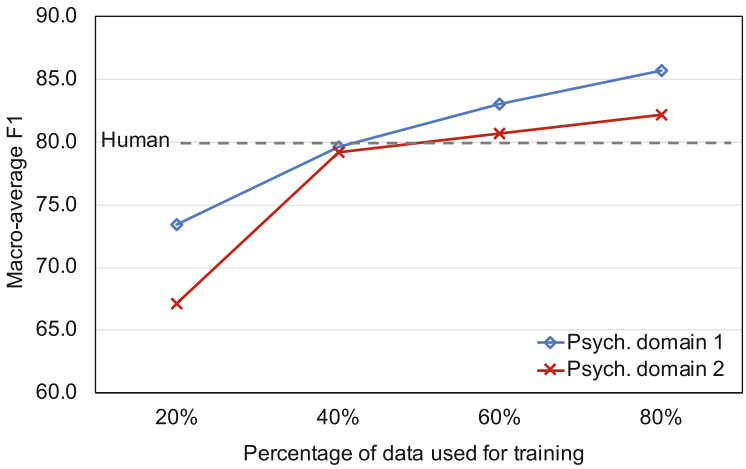

Fig. 2. Macro-average-F1 scores with different size of training sets of two domains, overlaid human performance. Evaluations are done on a held-out test set of 20%.

On our datasets, we obtain impressive macro-average-F1 of from 80% up to 85%, indicating the robustness of the model's transferability to the target task of short answer grading. On our datasets, we report human performance as a baseline against which the model can be compared. As outlined earlier, each student response is annotated by three experts. The variability in the annotation enables us to establish a human performance baseline. Table 3 lists each human annotation's comparison against the majority vote (MV) in terms of accuracy (Acc), macro-averaged-F1 (M-F1), and weighted-average-F1 (W-F1).

Effectiveness for Data-Starved Problems: Task-specific supervised fine-tuning is possible with small number of samples. On SciEntsBanks dataset, the training set includes ∼5K samples; which yields results better than task-specific learning. To further study this property of the model, we design an experiment to train the model with small portions of training data. Figure 2 shows the performance in terms of macro-average-F1, when the training data is reduced from 80% of the whole set to mere 20%. Evaluation is done on a constant held-out test

set consisting of 20% samples. Note the decrease in the slope as the training set expands, suggesting diminishing returns as training data is added. The increase in M-F1 is about 10% as the training set increases from 20% to 80%. For data-starved problems, a rather generous trade-off can be made to obtain a reasonably good performance with limited task-specific fine-tuning data. Interestingly, the M-F1 with 40% training data is in same range as human performance (shown in Table 3).

Effectiveness of Training Epoch on Fine-Tuning: We also performed experiments for fine-tuning BERT with varying number of epochs. We observed that fine-tuning for 4 and 12 epochs does not yield significantly different results on macro-average-F1 (85.7 and 85.4 on domain 1, 82.2 and 83.7 on domain 2 respectively), indicating that task-specific transfer takes place within initial few epochs only.

Table 4. Cross- and joint- domain fine-tuning. Accuracy (Acc), macro-average-F1 (M-F1), and weighted-average-F1 (W-F1) are reported in percentage.

Training set	Test set					
	Psych. domain 1			Psych. domain 2		
	Acc	M-F1	W-F1	Acc	M-F1	W-F1
40% of Psych. domain 1	**88.0**	**79.7**	**88.1**	76.4	51.1	75.5
40% of Psych. domain 2	72.4	48.2	70.0	**90.1**	**79.1**	**90.0**
40% each of domain 1 & 2	86.7	79.1	87.5	88.9	77.0	88.7

Effectiveness in Cross- and Joint- Domain Fine-Tuning: We further evaluated the fine-tuned model's ability to generalize to unseen domains. Table 4 reports the performance of fine-tuned models on both domains. It shows that the model fine-tuned using domain 1 yields very poor results on domain 2, and vice versa. This suggests that domain specific supervised data is indeed required for efficient fine-tuning. As a follow-up, we fine-tuned a model using a combined set of both domain data; which yields results relatively similar to domain specific tuning. It provides evidence that the model can be jointly fine-tuned for both models.

5 Conclusion

This paper conclusively demonstrates that Transformer-based pre-trained models push the state-of-the-art in short answer grading to a level that may be approaching the ceiling of what is possible. In comparison with human scorers, the model learns the "wisdom of the crowd", surpassing the performance of any individual human scorer on our datasets. The amount of fine-tuning needed is reasonable; even with just a few thousand labeled samples, we are able to get

superior results. We also show that while applying a model fine-tuned on data associated with one domain cannot directly apply to grading other domains, it is possible to create a single model fine-tuned using data from multiple domains that works for each of them. Going forward, we expect to investigate whether adding an additional domain-specific text corpus to a pre-trained model improves the ability to process language for that domain. We will continue to experiment with ways to minimize the amount of fine-tuning (e.g., through characterization of what types of labeled samples yield the highest marginal improvement during fine-tuning, thus allowing for more efficient data collection for automated grading). Finally, work on model management, reuse of models, and devising efficient methods to add new labeled samples to existing fine-tuned methods will be of interest so that a model adapts over time.

Acknowledgements. We would like to thank Yoonsuck Choe (Texas A&M University) for helpful comments on an earlier version of this paper.

References

1. Albacete, P., Jordan, P., Katz, S.: Is a dialogue-based tutoring system that emulates helpful co-constructed relations during human tutoring effective? In: Conati, C., Heffernan, N., Mitrovic, A., Verdejo, M.F. (eds.) AIED 2015. LNCS (LNAI), vol. 9112, pp. 3–12. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-19773-9_1
2. Alikaniotis, D., Yannakoudakis, H., Rei, M.: Automatic Text Scoring Using Neural Networks, June 2016. https://doi.org/10.18653/v1/p16-1068, https://arxiv.org/abs/1606.04289
3. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer Normalization, July 2016. http://arxiv.org/abs/1607.06450
4. Conneau, A., Kiela, D., Schwenk, H., Barrault, L., Bordes, A.: Supervised learning of universal sentence representations from natural language inference data. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pp. 670–680 (2017)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, October 2018. http://arxiv.org/abs/1810.04805
6. D'mello, S., Graesser, A.: Autotutor and affective autotutor: learning by talking with cognitively and emotionally intelligent computers that talkback. ACM Trans. Interact. Intell. Syst. **2**(4), 23:1–23:39 (2013). https://doi.org/10.1145/2395123.2395128, http://doi.acm.org/10.1145/2395123.2395128
7. Dzikovska, M., et al.: Semeval-2013 task 7: the joint student response analysis and 8th recognizing textual entailment challenge. In: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), vol. 2, pp. 263–274 (2013)
8. Heilman, M., Madnani, N.: ETS: domain adaptation and stacking for short answer scoring. In: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), vol. 2, pp. 275–279 (2013)

9. Howard, J., Ruder, S.: Universal Language Model Fine-tuning for Text Classification (2018). http://arxiv.org/abs/1801.06146
10. Jimenez, S., Becerra, C., Gelbukh, A.: Softcardinality: hierarchical text overlap for student response analysis. In: Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), vol. 2, pp. 280–284 (2013)
11. Kumar, S., Chakrabarti, S., Roy, S.: Earth mover's distance pooling over siamese lstms for automatic short answer grading. In: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, pp. 2046–2052 (2017). https://doi.org/10.24963/ijcai.2017/284
12. Marvaniya, S., Saha, S., Dhamecha, T.I., Foltz, P., Sindhgatta, R., Sengupta, B.: Creating scoring rubric from representative student answers for improved short answer grading. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 993–1002. ACM (2018)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed Representations of Words and Phrases and Their Compositionality, October 2013. http://arxiv.org/abs/1310.4546
14. Mohler, M., Bunescu, R., Mihalcea, R.: Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, pp. 752–762. Association for Computational Linguistics (2011)
15. Mohler, M., Mihalcea, R.: Text-to-text semantic similarity for automatic short answer grading. In: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, pp. 567–575. Association for Computational Linguistics (2009)
16. Mou, L., et al.: How Transferable are Neural Networks in NLP Applications? March 2016. http://arxiv.org/abs/1603.06111
17. Mueller, J., Thyagarajan, A.: Siamese recurrent architectures for learning sentence similarity. In: AAAI, vol. 16, pp. 2786–2792 (2016)
18. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Trans. Knowl. Data Eng. **22**(10), 1345–1359 (2010). https://doi.org/10.1109/TKDE.2009.191, http://dx.doi.org/10.1109/TKDE.2009.191
19. Peters, M.E., et al.: Deep contextualized word representations, February 2018. http://arxiv.org/abs/1802.05365
20. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018)
21. Raina, R., Battle, A., Lee, H., Packer, B., Ng, A.Y.: Self-taught learning: transfer learning from unlabeled data. In: Proceedings of the 24th International Conference on Machine Learning ICML 2007, pp. 759–766. ACM, New York (2007). https://doi.org/10.1145/1273496.1273592, http://doi.acm.org/10.1145/1273496.1273592
22. Ramachandran, L., Cheng, J., Foltz, P.: Identifying patterns for short answer scoring using graph-based lexico-semantic text matching. In: Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 97–106 (2015)
23. Ramachandran, L., Foltz, P.: Generating reference texts for short answer scoring using graph-based summarization. In: Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications, pp. 207–212 (2015)

24. Rus, V., Stefanescu, D., Niraula, N., Graesser, A.C.: Deeptutor: towards macro- and micro-adaptive conversational intelligent tutoring at scale. In: Proceedings of the First ACM Conference on Learning @ Scale Conference L@S 2014, pp. 209–210. ACM, New York (2014). https://doi.org/10.1145/2556325.2567885, https://doi.acm.org/10.1145/2556325.2567885
25. Saha, S., Dhamecha, T.I., Marvaniya, S., Sindhgatta, R., Sengupta, B.: Sentence level or token level features for automatic short answer grading?: use both. In: Penstein Rosé, C., et al. (eds.) AIED 2018. LNCS (LNAI), vol. 10947, pp. 503–517. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93843-1_37
26. 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)
27. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems, vol. 30, pp. 5998–6008. Curran Associates, Inc. (2017). http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf
28. Ventura, M., et al.: Preliminary evaluations of a dialogue-based digital tutor. In: Penstein Rosé, C., et al. (eds.) AIED 2018. LNCS (LNAI), vol. 10948, pp. 480–483. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93846-2_90
29. Wu, Y., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. CoRR abs/1609.08144 (2016). http://arxiv.org/abs/1609.08144