JKAU: Comp. IT. Sci., **Vol. 11** No. 2, pp: 25 – 31 (1444 A.H. / 2022 A.D.)
DOI: 10.4197/Comp.11-2.2

Automatic Short Answer Grading Using Paragraph Vectors and Transfer Learning Embeddings

Abrar S. Alreheli, Hanan S. Alghamdi
Faculty of Computing and Information Technology
King Abdulaziz University
Jeddah, Saudi Arabia
aalrehali0003@stu.kau.edu.sa, hsaalhamdi@kau.edu.sa

***Abstract*. automatic short answer grading (ASAG) is the process of assessing short answers by utilizing computational approaches. Recent research attempts to solve this problem based on semantic similarity and deep learning models. The objective of this paper is to evaluate the proposed models in grading short answers by computing the semantic similarity between the student and a key answer. We suggest training the paragraph vectors and transfer learning models on a domain-specific corpus instead of using the pre-trained models. Then, the trained models are used to generate embeddings that represent the student and reference answers as vectors. We computed the similarity between the vectors of the reference and student's answer. The similarity score will be used as a feature vector to train a regression model to predict the scores. We evaluated the models by comparing the actual score with the predicted score. The best accuracy achieved by fine-tuning the (Roberta-large) model on the domain-specific corpus is 0.620 for Pearson correlation, and 0.777 for Root Mean Square Error (RMSE). We conclude that pre-trained paragraph vectors achieve better semantic similarity than training paragraph vectors on a domain-specific corpus. On the contrary, fine-tuning transfer learning models on a domain-specific corpus improve the performance.**

Keywords*-- *Automatic Grading, Short Answer, Corpus, Paragraph Vectors, Transfer Learning, Similarity Masked Language Modeling.

I. INTRODUCTION

Grading is one of the most crucial activities for any instructor. The instructors usually spend much time and effort preparing and grading exams. There are several types of questions for assessment, such as multiple-choice, true or false, or essay questions. To recognize short answer questions from other types of questions, the length of the answer should approximately range from a single phrase to a single paragraph [1]. Indeed, the task of grading this type of question is time and effort-consuming, so there is an insistent need to adopt an approach for automatic grading for subjective questions. For this purpose, many researchers have attempted to automatically grade subjective answers since 1994 [2]. They continue the attempts to automatically grade subjective answers in their different types, trying to implement a model with a more accurate score. Most of the recent research uses machine learning and deep learning approaches [3]. This paper presents two experiments using two different approaches. First, using paragraph vectors to model the answers; an unsupervised algorithm that learns fixed-length feature representations from any piece of text. The doc2vec is a paragraph vector model presented by Thomas Mikolov [4]. Secondly, using a transformer-based approach to generate the embeddings that model the answer. The Transformer is a natural language processing approach that depends on the concept of self-attention. The self-attention process computes input and output vectors in parallel, which overcomes the problem of sequential processing in the case of "Recurrent Neural Network" (RNN), "convolutional neural network" (CNN), or "Long Short-Term Memory" (LSTM) approaches [5]. However, in this paper, we do not consider the idea of transfer learning, instead, we only use the embeddings of these models, supposing that they have extracted the semantic knowledge of each answer from the pre-trained corpus. The two approaches incorporate the context of the words individually and a paragraph as a whole since contextual representation improves a semantic similarity between student and reference answers. Moreover, one of the major limitations in the current research in ASAG is the non-availability of domain-specific training data. Hence, we hypothesize that the inferred vectors by the models trained on domain-specific corpora will result in the better semantic similarity between student's and reference answer. We aim to evaluate these two experiments by answering the following research questions:

Q1) Will training paragraph vectors on a domain-specific

corpus achieve better semantic similarity than pre-trained vectors?

Q2) Will fine-tuning transfer learning selected models on a domain-specific corpus improve the achieved results using the pre-trained masked language models in the task of ASAG?

The general workflow of the proposed models is demonstrated in Fig 1. The layout of this paper is as follows:

- Section II reviews the related works in the task.
- Section III describes the used dataset in this research.
- Section IV describes the prepared corpus.
- Section V explains the methodology of this work.
- Section VI presents the results of the conducted experiments.
- Section VII discusses the conclusion and recommendation.

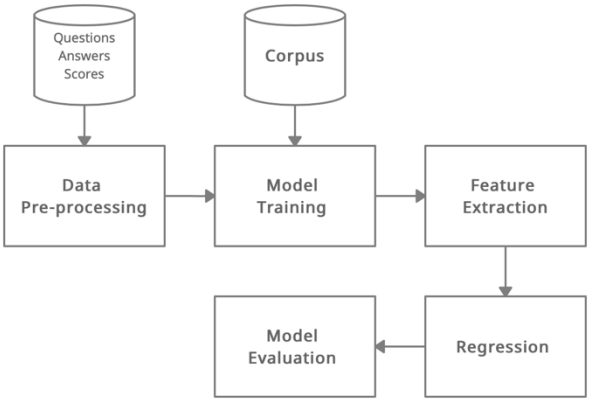

Fig. 1. Workflow of the Proposed Models

II. Related Works

Several approaches have been introduced to solve the problem of ASAG. An overall review of ASAG systems is summarized by the reference [1]. They present 35 automatic short answer grading systems conducted using different methods. Whereas [3], reviews only the studies conducted using the deep learning approach.

One approach uses unsupervised techniques that measure text similarities between student answers and a model answer to predict the grade based on the similarity. A study followed this approach; they compared two semantic similarity measures, the knowledge-based and corpus-based, including the explicit semantic analysis (ESA) and the latent semantic analysis (LSA). Their key idea is to incorporate the optimal student answer with the instructor's answer to enrich the vocabulary of the correct answers and enhance the model's performance. The best-achieved result for their model was by using latent semantic analysis that applied to a specified domain corpus. It achieved 0.509 for the correlation coefficient [6]. Similarly, another study combined String-based with Corpus-based similarity measures between student answers along with the key answers to predict the score. They achieved a 0.504 correlation coefficient [7].

The second approach followed by researchers for ASAG is machine learning using some known features to predict the score. Following this approach, a study discussed two machine learning models: regression and classification. Therefore, they used multiple features to train their models, such as text similarity between students' answers and reference answers and term weighting using Term Frequency and Inverse Document Frequency (TF-IDF) as a feature. Furthermore, they used the word count ratio in the student answer versus the ones in the reference answer as a feature. The first unsupervised model was training a ridge regression model to compute the value of student grades. They evaluated the model using the Texas dataset [8]. It achieved the result of a 0.887 RMSE and 0.592 for the Pearson correlation. The second machine learning model was to classify the student answers into one label, whether it is correct or incorrect. This model is evaluated on SemEval 2013 dataset [9]. It achieved the result of a 0.85 RMSE and 0.63 for the Pearson correlation [10].

The third approach, which recently became a state-of-art for most Natural Language Processing (NLP) tasks, is using deep learning architectures that allow automatic feature representations. Several advanced deep learning models are widely used to solve NLP tasks, such as CNN and RNN [11]. A study published in 2019 discussed using BERT for the ASAG task. They suggested two approaches to update the pre-trained BERT. The experiment uses labeled pairs of questions and answers. Furthermore, they fine-tune the BERT model on the SemEval-2013 [12]. Similarly, a study published in 2020 investigated several transformers and fine-tuned them on the SemEval2013 task. Furthermore, they found that training models with knowledge distillation improve the performance of ASAG task. They also experimented the ability of the multilingual transformers model to be generalized to other languages. The most important result of this paper was that large transfer learning models improve the result of the ASAG task more than base models [13]. Correspondingly, a study compared four different pre-trained transfer learning models for the task of ASAG. The first model is Embeddings from Language Models (ELMo) [14], compared with "Bidirectional Encoder Representations from Transformers" (BERT) [15], Generative Pre-training (GPT), and GPT-2. They apply their experiment to the Texas dataset. They found that ELMo outperformed other transformer models [16].

Two recent studies published in 2021 achieved good results. However, they have some cons to the applied methods. One study suggested utilizing Manhattan LSTM and the sense vectors provided by Semantic Space. They used Synset to represent student responses and reference responses. The similarity between these representations is measured using Manhattan Similarity. The proposed model is evaluated on the Texas dataset. They achieved a correlation of 0.949 and RSME of 0.040. The achieved result may appear to be state-of-the-art. However, they tested each answer separately instead of combining all the answers for testing [17]. The other study proposed three ASAG models to grade each student's answer individually. They used part-of-speech tagging, a Stanford dependency parser, and a two-dimensional matrix to represent the answers. Then Word2Vec and FastText are used to measure the similarity scores between the reference answer and students' answers. They evaluated the proposed models on the Texas

dataset, and they achieved a high correlation of 0.805. However, the applied techniques Word2Vec and FastText do not incorporate the context [18].

III. Dataset

For the task of ASAG, six common datasets are available. In this work, we utilized the Texas dataset by Mohler to evaluate the models [8]. The dataset is created from an introductory course on computer science provided by Texas University. It is collected from 10 assignments plus two exams; each has about 4 to 7 different questions, while each exam contains 10 questions. It contains 87 questions along with their reference answer. Around 28 different students have answered each question. The total number of all the answers is 2273. The answers are graded by two experienced evaluators in the computer science major. The average grade for both is provided. Each answer is graded from 0 to 5, in which grade 0 refers to (wrong), and grade 5 refers to (correct). We used the average grade following the original research in this work [8]. Table 1 illustrates an example for one question from the dataset, along with the reference answer, and students' answers with the corresponding score for each.

TABLE I. Sample Question and its Answers from the Dataset

Question	"What is a variable?"	
Model Answer	"A location in memory that can store a value."	
Student Answer and assigned score		
Student Answer 1	"it is a location in memory where value can be stored."	5
Student Answer 2	"a placeholder to hold information used in the program."	3
Student Answer 3	"Variable can be a integer or a string in a program."	1

IV. Corpus

We will explain how we prepared the corpus used for training paragraph vectors in the first experiment and fine-tuning the language model in the second experiment. We consider the domain of computer science. First, we collect computer science textbooks in PDF format from different recourses. To make the corpus more specific, we only consider the introductory books in the domain. We convert the textbooks from PDF files to text files to be processed. We ignore the cover pages, table of contents, and references. In addition, we used a regular expression to find any hyperlinks, dates, and coding parts and remove them from the corpus. After that, we tokenized the text into sentences. Since some sentences are too short such as titles or captions, we printed a histogram for sentence length to check for valid sentences. Thus, we only keep sentences that are more than five words and less than 50 words. Finally, we write each sentence in a single line, thus each line represents a document.

V. Methodology

The input for the ASAG model is the vector that represents the student answer (SA), along with the vector that represents the reference answer (RA). We use different models to obtain the vectors in each experiment. The vectors are inferred using two models; the paragraph vector (PV) model and the transfer learning model. Then, the similarity between SA and RA is measured using the cosine similarity. After that, the computed cosine similarity is used as a feature for a regression model to predict a particular answer score. We evaluate the models by comparing the actual score provided in the dataset along with the predicted score using two evaluation metrics. We use the Pearson correlation coefficient and RMSE to evaluate the models. Fig (2) illustrates the architecture for the proposed methodology. Next, we will explain the methods and implementation in further detail.

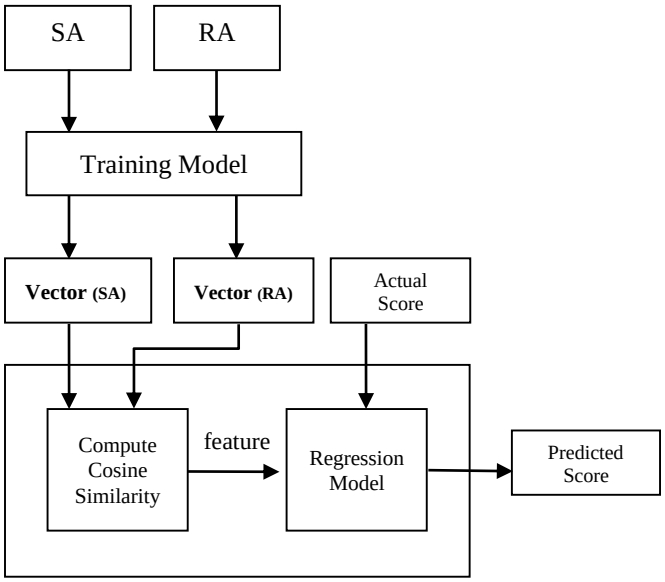

Fig. 2. Answer Grading Architecture.

A. Dataset Preprocessing

We pre-process the dataset by removing punctuation marks and stopwords. In addition, we extract the tokens from each RA and SA using the NLTK tokenizer. However, we ignore applying the spell checker, assuming that misspelled words might affect the assigned score to such an answer.

B. Inferring Vectors

We present two experiments to generate the vectors that represent the RA and SA. These vectors are used to obtain semantic knowledge from the words depending on their context. Each experiment represents a single model that is used separately to infer the vectors.

For both experiments, we conduct a baseline model and a proposed model to compare the performance. For the baseline,

we use a pre-trained model on a general domain corpus such as Wikipedia. For the proposed model, we use the specific-domain corpus (Section IV) to fine-tune the pre-trained models. We train the model to learn and infer the corresponding vectors in each experiment.

1) Using the paragraph vector (PV) model to infer the vector of a given answer:

In this experiment, we obtain the vector that represents the answer directly by using the PV model. We use Doc2vec provided by the Gensim library in python. It is a natural language processing library used for unsupervised topic modeling. It provides the implementation for many tasks such as generating word/ paragraph vectors and corpus handling tasks [19]. We trained the doc2vec model with 300 dimensions' vector size on the tokenized answers. We obtained the learned PV vectors for SA and RA.

For the baseline, we used pre-trained PV. We selected pre-trained Doc2Vec on English Wikipedia DBOW [20]. For the proposed model, we train paragraph vectors on the prepared domain-specific corpus. First, we used the (smart_open) library to read the corpus. It is a python library used for reading large files [19]. It reads the corpus line-by-line and pre-processes each line by tokenizing the text, removing the punctuation, and converting the text into lowercase. Each line of the corpus represents a document. Each document in the training corpus will be tagged with a number (tagged documents) for model training. We create a Doc2vec model with a 300-dimensional vector size and 40 epochs. Then we infer the vectors that represent a given answer using the trained model on the domain-specific corpus. Fig (3) illustrates the methodology for inferring vectors in this experiment.

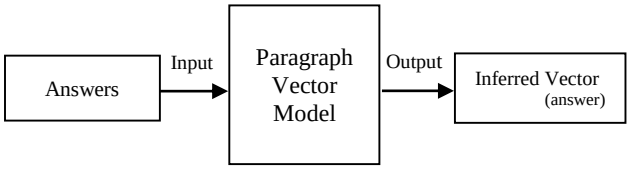

Fig. 3. Inferring Vectors from the PV Model.

2) Using the transfer learning model to infer the vector of a given answer:

In this experiment, we obtain the vector that represents the answer by generating the answer's embeddings using the transfer learning model. The victor size of the answer's embeddings will be 768. Fig (4) illustrates the methodology for inferring vectors in this experiment. The transfer-learning models can be used for fine-tuning and feature-based approaches [15].

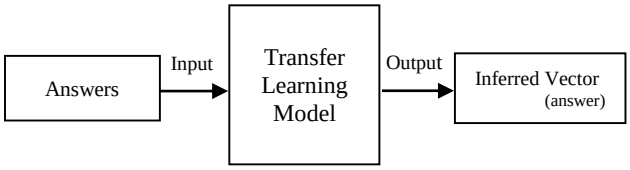

Fig. 4. Inferring Vectors from the Transfer Learning Model

We specify two pre-trained transformer models to apply to this experiment; (Roberta-large) [21] and (Scibert) [22]. This is because the Roberta model can generalize better than other models in term of short answer grading task [13]. We also use (Scibert) model since it is trained on scientific data; we hypothesized that it would achieve good results since the domain of the corpus is computer science. For a baseline, we use the pre-trained model of (Roberta-large) and (Scibert); we use them to obtain the embeddings directly without any further training. For the proposed model, we use the prepared domain-specific corpus to fine-tune a previously trained transformer model. The objective of fine-tuning the models on the corpus is to learn the language and context of the domain. The transfer learning model can learn the language of the domain by two approaches in parallel. They are "Masked Language Modeling" (MLM), and "Next Sentence Prediction" (NSP). In MLM, the model takes in a sentence and masks random words. The objective is to output these masked tokens; it helps the model understands the bi-directional context within a sentence. In NSP, a pair of sentences are input to the model, and it learns to predict whether the second sentence follows the first in the original corpus. Fig. (5) illustrates the mechanism of MLM and NSP tasks from the paper of BERT. This approach can be generalized to all transformed models [15].

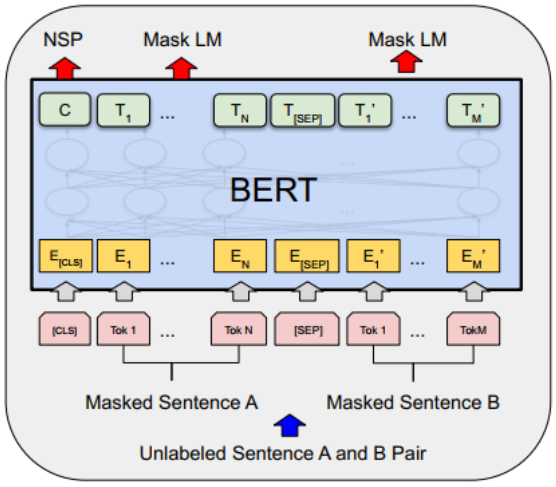

Fig. 5. The Mechanism of MLM and NSP tasks [15]

First, we load the corpus as raw text, labeled data are not required for this task. The model takes in two sentences randomly. Each sentence is tokenized by a pre-trained tokenizer to several tokens. Then, 15% of random tokens are masked by substituting each one with [MASK]. For instance, consider the following sentence from the corpus as the input sequence:

"Variables that are declared inside a function are local variables"

MLM masks some random tokens within the input and replaces that token with a special token called [MASK] as follows:

"Variables that are declared inside a [MASK] are local variables".

The model is supposed to predict the same input sequences as output. Moreover, it is supposed to predict whether (Masked sentence A) follows (Masked sentence B) in the original corpus.

The process of feeding the pair of sentences is done automatically using the *Autotokenizer* class provided by the (*HuggingFace*) library [23]. For training, we tested some hyperparameters that are compatible with our limited computational power. After several experiments with multiple tests and failures, we set the hyperparameters as follows; learning rate (2e-5), weight decay (0.01), and batch size: 16. By completing the fine-tuning process on the specific-domain corpus, the trained language model is supposed to adapt its vocabulary from the general corpus that it was originally pre-trained on, to the specified terminologies in the corpus domain, is in our case, computer science. Then, the fine-tuned transform is used to generate the embeddings that represent a given answer as a single vector.

C. Feature Extraction:

We calculate the cosine similarity between each SA vector and RA vector. We use Eq.1 to calculate the similarity score, where, $\vec{s}$ represents the SA vector and $\vec{r}$ represents the RA vector [24]. The computed similarity score will be used as a feature to train the ridge regression.

$$similarity(\vec{s},\vec{r}) = \cos(\theta) = \frac{\vec{s}.\vec{r}}{|\vec{s}|.|\vec{r}|} \quad (1)$$

D. Training and Testing:

Training: We split the data of questions and answers into 80% for training and 20% for testing. We train the ridge regression to predict the score using the similarity score as a feature along with the given score. We train each model in each experiment separately for 1000 iterations. For each iteration, we train and test different data randomly.

Testing: We train the model on the test data. In this phase, the regression model uses unseen data. The similarity score calculated between RA and SA of the test data is input to the trained regression model. As a result, the predicted score is obtained.

E. Evaluation:

To evaluate the models, we measure the performance of the regression model. We compute Pearson Correlation Coefficient (ρ), and "Root Mean Square Error" (RMSE) for the given score provided in the dataset and the corresponding predicted score.

VI. Experiments Results

Experiment 1: For the baseline, we extract the vectors that represent the answers directly using pre-trained PV on a general domain corpus. Then we used the similarity score between RA and SA to train the regression model. The value of the Pearson Correlation ρ and RMSE are 0.569 and 0.797. For the proposed model, we train the PV vectors on the specific-domain corpus. It achieved a 0.401 correlation and 0.893 RMSE. The result of both models is shown in Table (2). By comparing the achieved results of both models, we notice that training the paragraph vectors model on the specific-domain corpus doesn't improve the achieved result by the baseline, which suggests that this approach might not be well suited for paragraph vectors. We believe this is because paragraph vector models depend on the availability of extensive corpora such as Wikipedia or Google news to train these models.

TABLE II. Results of PV models trained on a general and specific-domain corpus

Paragraph Vector Model	ρ	RMSE
Pre-trained PV (baseline)	0.569	0.797
Trained PV on specific-domain corpus	0.401	0.893

Experiment 2: For the baseline, we extract the vectors that represent the answers by obtaining the embedding of the answer using the transfer learning models without further training. Then we used the similarity score between RA and SA to train the regression model. For the proposed model, we fine-tuned (roberta-large) and (Scibert) models on the specific-domain corpus. The result of both models is shown in Table (3). By comparing the achieved results of both models, we notice that fine-tuning the transformer models on the domain-specific corpus improves the achieved result of the baseline for both tested models.

Table III. Result of transfer learning models trained on a general and specific-domain corpus

Transfer Learning Model	ρ	RMSE
scibert (baseline)	0.568	0.803
scibert (fine-tuned)	0.596	0.787
roberta-large (baseline)	0.587	0.799
roberta-large (fine-tuned)	**0.620**	**0.777**

We also compare the achieved results to different conducted approaches on the Mohler dataset. We also include old methods such as Bag-of-Words (BOW) or TF-IDF. By looking at the result in Table (4), fine-tuning the transformer models on the domain-specific corpus is achieving the best result with higher correlation and lower RMSE. Fig (6) illustrates the accuracy of our proposed model compared to other models.

Table IV. Comparison of the achieved results with former studies

Model		ρ	RMSE
Former studies results	BOW [8]	0.480	1.042
	TF-IDF [10]	0.592	0.887
	Word2Ve [25]	0.488	1.016
	GloVe [25]	0.507	0.838
	FastText [25]	0.519	0.831
	SkipThoughts [25]	0.468	0.861
	ELMo [16]	0.485	0.978
	GPT [16]	0.248	1.082
	BERT [16]	0.318	1.057
	GPT2 [16]	0.311	1.065
Pr op	Pre-trained PV	0.569	0.797
	Trained PV on corpus	0.401	0.893

	Roberta–large	0.587	0.799
	Roberta–large (Fine-tuned)	0.620	0.777
	Scibert	0.568	0.803
	Scibert (Fine-tuned)	0.596	0.787

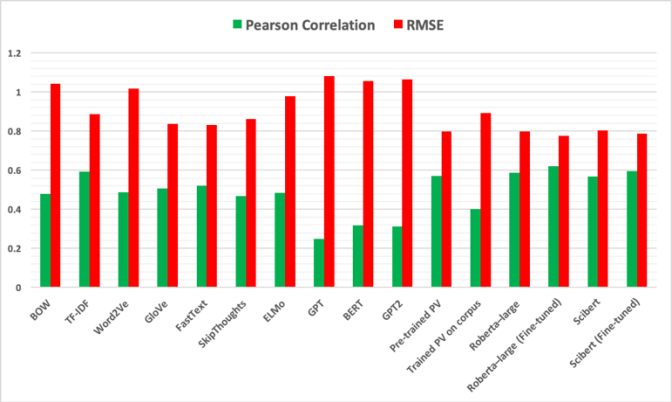

Fig. 5. The accuracy of our proposed model compared to other models.

VII. Conclusion and Recommendation

In this paper, we address the problem of ASAG. We analyzed the effect of training two different models on the domain-specific corpus. The best accuracy was achieved by fine-tuning the (Roberta-Large) on the domain-specific corpus. These findings answer the research questions and conclude that fine-tuning transfer learning models on a domain-specific corpus improved the results more than the pre-trained models. This superiority is reasonable because transformers can learn the context of the words from both directions. On the contrary, the pre-trained paragraph vectors perform better than the trained paragraph vectors on a domain-specific corpus. This indicates that paragraph vectors increase the model's generalizability. In future work, we will intend to apply the same methods to other different domains of dataset and corpus for short answer grading to see if the same result will be achieved. Furthermore, the regression models can be trained with various features along with the similarity

VIII. References

[1] S. Burrows, I. Gurevych, and B. Stein, "The eras and trends of automatic short answer grading," *International Journal of Artificial Intelligence in Education*, vol. 25, no. 1, pp. 60–117, 2015.

[2] E. B. Page, "Computer grading of student prose, using modern concepts and software," *The Journal of experimental education*, vol. 62, no. 2, pp. 127–142, 1994.

[3] S. Bonthu, S. Rama Sree, and M. H. M. Krishna Prasad, "Automated Short Answer Grading Using Deep Learning: A Survey," in *Machine Learning and Knowledge Extraction*, vol. 12844, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, Eds. Cham: Springer International Publishing, 2021, pp. 61–78. doi: 10.1007/978-3-030-84060-0_5.

[4] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188–1196.

[5] A. Vaswani *et al.*, "Attention Is All You Need," *arXiv:1706.03762 [cs]*, Dec. 2017.

[6] M. Mohler and R. Mihalcea, "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*, 2009, pp. 567–575.

[7] W. H. Gomaa and A. A. Fahmy, "Short answer grading using string similarity and corpus-based similarity," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 3, no. 11, 2012.

[8] M. Mohler, R. Bunescu, and R. Mihalcea, "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*, 2011, pp. 752–762.

[9] M. O. Dzikovska, R. D. Nielsen, and C. Leacock, "The joint student response analysis and recognizing textual entailment challenge: making sense of student responses in educational applications," *Lang Resources & Evaluation*, vol. 50, no. 1, pp. 67–93, Mar. 2016, doi: 10.1007/s10579-015-9313-8.

[10] M. A. Sultan, C. Salazar, and T. Sumner, "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*, San Diego, California, 2016, pp. 1070–1075. doi: 10.18653/v1/N16-1123.

[11] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," *arXiv:1708.02709 [cs]*, Nov. 2018.

[12] C. Sung, T. I. Dhamecha, and N. Mukhi, "Improving Short Answer Grading Using Transformer-Based Pre-training," in *Artificial Intelligence in Education*, vol. 11625, S. Isotani, E. Millán, A. Ogan, P. Hastings, B. McLaren, and R. Luckin, Eds. Cham: Springer International Publishing, 2019, pp. 469–481. doi: 10.1007/978-3-030-23204-7_39.

[13] L. Camus and A. Filighera, "Investigating Transformers for Automatic Short Answer Grading," in *Artificial Intelligence in Education*, vol. 12164, I. I. Bittencourt, M. Cukurova, K. Muldner, R. Luckin, and E. Millán, Eds. Cham: Springer International Publishing, 2020, pp. 43–48. doi: 10.1007/978-3-030-52240-7_8.

[14] M. E. Peters *et al.*, "Deep contextualized word representations," *arXiv:1802.05365 [cs]*, Mar. 2018.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv:1810.04805 [cs]*, May 2019.

[16] S. K. Gaddipati, D. Nair, and P. G. Plöger, "Comparative Evaluation of Pretrained Transfer Learning Models on Automatic Short Answer Grading," *arXiv:2009.01303 [cs]*, Sep. 2020.

[17] C. N. Tulu, O. Ozkaya, and U. Orhan, "Automatic Short Answer Grading With SemSpace Sense Vectors and MaLSTM," *IEEE Access*, vol. 9, pp. 19270–19280, 2021, doi: 10.1109/ACCESS.2021.3054346.

[18] B. Chaturvedi and R. Basak, "Automatic Short Answer Grading Using Corpus-Based Semantic Similarity Measurements," in *Progress in Advanced Computing and Intelligent Engineering*, vol. 1199, C. R. Panigrahi, B. Pati, P. Mohapatra, R. Buyya, and K.-C. Li, Eds. Singapore: Springer Singapore, 2021, pp. 266–281. doi: 10.1007/978-981-15-6353-9_24.

[19] "Gensim: topic modelling for humans." https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html#doc2vec-model (accessed Feb. 02, 2022).

[20] P. Karvelis, D. Gavrilis, G. Georgoulas, and C. Stylios, "Topic recommendation using Doc2Vec," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–6.

[21] Y. Liu *et al.*, "RoBERTa: A Robustly Optimized BERT Pretraining Approach." arXiv, Jul. 26, 2019.

[22] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A Pretrained Language Model for Scientific Text," *arXiv:1903.10676 [cs]*, Sep. 2019.

[23] "Fine-tuning a masked language model." https://huggingface.co/course/chapter7/3?fw=tf

[24] V. Piuri, S. Raj, A. Genovese, and R. Srivastava, Eds., *Trends in deep learning methodologies: algorithms, applications, and systems*. London, United Kingdom ; San Diego, CA, United States: Academic Press, 2021.

[25] S. Hassan, A. A. Fahmy, and M. El-Ramly, "Automatic Short Answer Scoring based on Paragraph Embeddings," *INTERNATIONAL JOURNAL OF ADVANCED COMPUTER SCIENCE AND APPLICATIONS*, vol. 9, no. 10, pp. 397–402, 2018.

التقدير التلقائي للإجابات القصيرة باستخدام متجهات الفقرة ونقل التضمينات التعليمية

أبرار الرحيلي [١] ، حنان الغامدي [١]

[١] قسم نظم المعلومات ، كلية الحاسبات وتقنية المعلومات

جامعة الملك عبد العزيز، جدة ، المملكة العربية السعودية

مستخلص. التقدير التلقائي لعلامات الإجابات القصيرة (ASAG) هو عملية تقييم الإجابات القصيرة باستخدام الأساليب الحسابية. حاول عدد من الباحثين مؤخرا حل هذه المشكلة بناءً على التشابه الدلالي ونماذج التعلم العميق. نهدف في هذه الورقة إلى تقييم عدد من النماذج المقترحة لتقييم الإجابات القصيرة عن طريق حساب التشابه الدلالي بين إجابة الطالب والإجابة المرجعية. اقترحنا تدريب متجهات الفقرة ونقل نماذج التعلم على مجموعة محددة المجال بدلاً من استخدام النماذج المدربة مسبقًا واستخدامنا النماذج المدربة لإنشاء المتجهات التي تمثل الطالب والإجابات المرجعية كنواقل. حسبنا التشابه بين متجهات المرجع وإجابة الطالب واستخدمنا درجة التشابه كمتجه لتدريب نموذج الانحدار للتنبؤ بالدرجات. قمنا بتقييم النماذج من خلال مقارنة النتيجة الفعلية مع النتيجة المتوقعة. حصلنا على أفضل دقة من خلال الضبط الدقيق لنموذج (Roberta–large) على الجسم الخاص بالمجال: ٠,٦٢٠ لعلاقة بيرسون، و ٠,٧٧ لخطأ مربع متوسط الجذر (RMSE). نستنتج أن متجهات الفقرة المدربة مسبقًا تحقق تشابهًا دلاليًا أفضل من متجهات الفقرة التدريبية على مجموعة خاصة بالمجال. وكذلك عمل الضبط الدقيق لنماذج التعلم المنقولة على مجموعة محددة المجال على تحسين الأداء.

الكلمات المفتاحية: التقدير التلقائي، الإجابة القصيرة، مجموعة النصوص، متجهات الفقرة، نقل التعلم، نمذجة اللغة المقنعة للتشابه.