

# Exploring the potential of DistilBERT architecture for automatic essay scoring task

Soumia Ikiss<sup>1</sup>, Najima Daoudi<sup>2</sup>, Manar Abourezq<sup>2</sup>, Mostafa Bellafkih<sup>1</sup>

<sup>1</sup>National Institute of Posts and Telecommunications, Rabat, Morocco

<sup>2</sup>School of Information Sciences, Rabat, Morocco

## Article Info

### Article history:

Received Jan 17, 2024

Revised Aug 3, 2024

Accepted Aug 11, 2024

### Keywords:

Automatic essay scoring

DistilBERT

Fine-tuning

Natural language processing

Transformers

## ABSTRACT

Automatic assessment of writing essays, or the process of using computers to evaluate and assign grades to written text, is very needed in the education system as an alternative to reduce human burden and time consumption, especially for large-scale tests. This task has received more attention in the last few years, being one of the major uses for natural language processing (NLP). Traditional automatic scoring systems typically rely on handcrafted features, whereas recent studies have used deep neural networks. Since the advent of transformers, pre-trained language models have performed well in many downstream tasks. We utilize the Kaggle benchmarking automated student assessment prize dataset to fine-tune the pre-trained DistilBERT in three different scenarios, and we compare results with the existing neural network-based approaches to achieve improved performance in the automatic essay scoring task. We utilize quadratic weighted Kappa (QWK) as the main metric to evaluate the performance of our proposed method. Results show that fine-tuning DistilBERT gives good results, especially with the scenario of training all parameters, which achieve 0.90 of QWK and outperform neural network models.

This is an open access article under the [CC BY-SA](#) license.



## Corresponding Author:

Soumia Ikiss

National Institute of Posts and Telecommunications

Madinat El Irfane, Rabat, Morocco

Email: soumia.ikiss97@gmail.com

## 1. INTRODUCTION

In higher education, e-learning systems are becoming more and more common, as universities and institutions enlarge provisions and more students get enrolled. Assessment and evaluation of students' knowledge and capabilities is an integral part of educational systems; thus, the effectiveness of an e-assessment system is crucial and quite appealing to academic institutions to reduce time consumption and the workload of human raters. In the educational environment, assessment is generally done through exams and tests; hence, the purpose of such an e-assessment system is to automatically evaluate assignments and tests accomplished by the learners and determine an appropriate score. The assignments can be given in different types; the most common are either selection and fill-in-the-blank tasks, which are relatively easy to implement as the answers are usually predetermined, or they can be in the form of a short answer or essay, which examine higher-order abilities such as logical thinking and critical reasoning. In essay writing tests, a learner writes a piece of text in response to a given topic, called a prompt, and human raters evaluate and grade those essays. The process is expensive and time-consuming and is not always consistent among and within raters. Automatic assessment of writing essays aims to enable computers to analyze and assign grades to learners' essays automatically without human interference [1]. It is a highly challenging task that requires exploiting multiple aspects that influence the quality of essays, such as content, relevance, and coherence.

The essay assessment task has been the subject of extensive investigation. And is one of the most challenging tasks in natural language processing (NLP). The quality of such a model is generally related to its ability to extract useful features that represent the maximum possible characteristics of the essay, which leads to accurately evaluating and predicting the corresponding score. Such features might be extracted either manually (handcrafted features) or automatically using neural network-based techniques [2]. The early works mainly use features designed manually by human experts based on prior knowledge of linguistics, such as grammar, term frequency, and syntax [3], [4]. Project essay grade (PEG) [5] is among the first computer programs for grading. It used linear regression to predict the score of a given text based on the style of essays and linguistic features only. Over 40 years following page's initial work, many auto-scoring essay software programs have a striking resemblance to PEG. In 1999, the intelligent essay assessor (IEA) used latent semantic analysis (LSA) for semantic similarity. Many other commercial assessment systems have come out, namely the electronic essay rater (E-Rater) and C-Rater [1], which have been used for TOEFL and GRE examinations. Recently, automatic essay-scoring research has been based on neural networks and uses end-to-end models [6]. These models are mainly based on word embedding techniques to learn high-dimensional features from data and capture the most possible information. In 2012, Kaggle came up with ASAP. Wish is a large-scale dataset that comes with the increase of deep learning techniques that need huge data to achieve good performance. Since then, there has been a renewed interest in automatic assessment, and the most recent works use the ASAP dataset to measure their performance. More recent research has applied neural networks for automatic scoring [7]-[13]. Taghipour and Ng [10] use both recurrent neural networks (RNN) and convolutional neural networks (CNN), they use CNN to extract local information from text and long short-term memory (LSTM) to generate temporal context and long history. Later on, used CNN with LSTM and performed the output score with the attention mechanism [11].

The traditional systems that adapted handcrafted features achieved good results and revolutionized the field of automatic scoring. However, they typically use complicated, handcrafted features from a small amount of data [14]-[16]. These techniques of feature extraction can be modified, made simple to explain and adapt new features, yet they can be unperformed to understand some deep information in the text. Neural network models are mainly based on word embedding techniques to learn high-dimensional features from data and capture the most possible information. Nevertheless, they need an enormous amount of annotated data to retrieve profoundly semantic features from the text during training, as low data can result in poor performance. Vaswani *et al.* [17] came out with transformers, which are neural networks that are built on the attention mechanism [18]. The transformer architecture allows parallelism in data processing and avoids vanishing gradients. Numerous pre-trained language models have been developed using this architecture, and they have shown promising outcomes in a variety of downstream NLP applications by fine-tuning pre-trained models such as bidirectional encoder representations from transformers (BERT). Few works on automatic scoring have used this model; [19]-[21] have fine-tuned the BERT model for the task. Nevertheless, it has not been effectively applied to surpass other deep learning models like LSTM in the automated essay scoring domain. The BERT model suffers from fixed input length and size limitations, word piece embedding problems, and computational complexities. In this work, we investigate the potential of fine-tuning a pre-trained model, namely DistilBERT [22], a more compact, faster transformer-based architecture created by applying knowledge distillation [23] to the BERT architecture, which offers a more scalable and economical alternative to BERT.

The remaining sections of this paper are organized according to the following structure: In the next section 2, explain our method and research design. In this part, we detail step-by-step our proposed architecture and describe our numerical study, including data acquisition, evaluation metrics, and implementation. The experimental findings and analysis are shown in section 5. In conclusion, section 6 gives a summary and provides suggestions for further research and perspectives.

## 2. METHOD

In this section, we first describe the architecture and design of the proposed model. Then, we provide details on our numerical study, including data acquisition, experimental settings, and metrics used for performance evaluation.

### 2.1. Model description

The following sub-section describes the proposed approach, starting with an overview of the pertained DistilBERT model and then giving the detailed architecture of our method to adapt DistilBERT to our task.

**2.1.1. DistilBERT model**

DistilBERT is a pre-trained language model created by applying the mechanism of knowledge distillation [22] to the large model of BERT [24]. The general architecture is similar to BERT, where the base unit is the transformer, especially the encoder block, where the transformer is an encoder and decoder. Figure 1 depicts the general architecture of the BERT encoder, where the main block uses a self-attention mechanism to generate a contextual embedding representation. As with the original BERT, DistilBERT was also trained on data collected from the English Wikipedia and the Toronto Book Corpus. The main difference between them lies in the token-type embeddings and poolers that are omitted from DistilBERT. In addition, the distilled model uses huge batches with the help of gradient accumulation using dynamic masking in place of the masking used in the original BERT and without the next sentence prediction (NSP) objective during training. The number of transformer layers (encoders) in the BERT base (12 layers) has been reduced to six. Thus, the distilled version is approximately smaller with 66 million parameters, where the number is reduced to nearly half (40%) and faster with 60% while preserving more than 95% of BERT performance. This makes DistilBERT an ideal choice to perform our study.

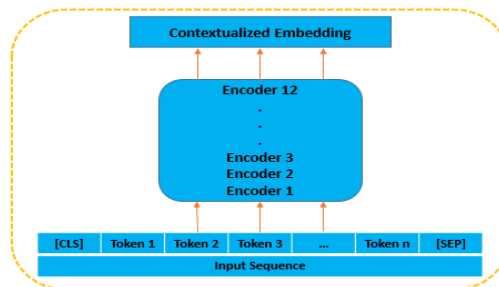


Figure 1. The general architecture of BERT

**2.1.2. Proposed architecture**

In this subsection, we introduce the overall architecture of the proposed model. Figure 2 presents a schematic representation of the entire architecture. The model performs fine-tuning on the DistilBERT model to tackle the problem of essay scoring. The model receives as an input X, which represents an essay (a sequence of words). The input sequences are tokenized using DistilBERT tokenize and converted to a set of embedding vectors. Then apply the transformer encoder and use a self-attention mechanism to learn the contextualized embeddings. Later, the contextualized embeddings are concatenated into a single vector and used as input to a regression layer added on top to fine-tune the pre-trained DistilBERT on the essay scoring task and predict a score for each essay. In what follows, we proceed with the main components of the model in detail.

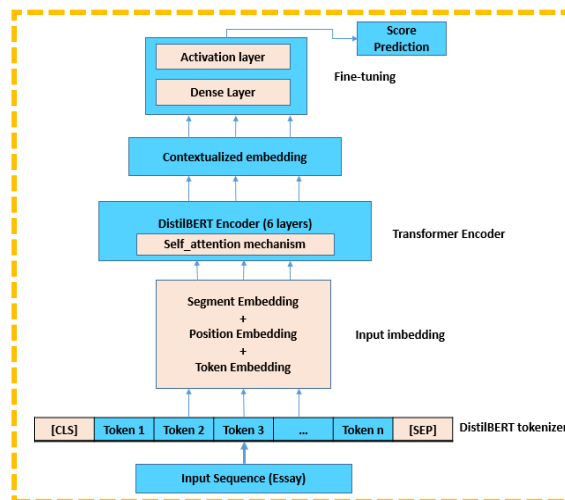


Figure 2. The proposed essay scoring model

- a) Input embeddings: the tokenizer involves breaking down the input sequence into unique tokens to enable the model to understand the meaning of the text. We use the DistilBERT tokenizer to split the input essay into tokens and add the special tokens [CLS] at the first position and [SEP] at the end of the sequence. The tokens are used to generate vectors of word, segment, and positional embeddings. The latter are then summed up to one embedding vector for each token to be passed to the transformer encoder.
- b) Transformer encoder: the contextual information is extracted via pre-training using the pre-trained DistilBERT encoder to convert input tokens into contextual embeddings. The encoder comprises a stack of six layers, each of which contains a feed-forward neural network and a multi-head self-attention mechanism that aims to recognize contextual connections between tokens. Figure 3 describes a single layer of the encoder component.

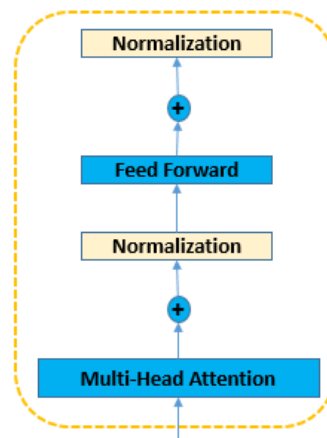


Figure 3. A single layer of the encoder component

- c) Fine-tuning on essay scoring: to handle our task of essay scoring, we fine-tuned the last layer of the encoder block, which outputs 768 dimensional hidden units. The classification layer is replaced with a dense layer, which consists of a fully connected network and is then fed to the regression layer to predict essay scores.

**2.2. Experiments**

In this sub-section, we introduce a description of the experiment’s procedure, including the used dataset, performance metrics, and experimental setup.

**2.2.1. Data acquisition**

To implement the essay scoring task and validate the proposed model, the experiments were carried out using the automated student assessment prize (ASAP) dataset (<https://www.kaggle.com/c/asap-aes/data>), a frequently used corpus as a reference dataset for tasks involving automatic essay grading. It is a large-scale dataset introduced and supported by the Hewlett, William, and Flora Hewlett Foundation as part of a Kaggle competition in 2012. It is composed of 12,976 labeled essays with eight prompts. Table 1 shows more details about the ASAP dataset.

Table 1. Statistic details of Kaggle’s ASAP dataset

Poemts set	Number of essays	The mean length of essays	Score range
1	1,783	350	1-12
2	1,800	350	1-6
3	1,726	150	0-3
4	1,772	150	0-3
5	1,805	150	0-4
6	1,800	150	0-4
7	1,569	250	0-30
8	723	650	0-30

### 2.2.2. Evaluation metric

To measure the performance, we use quadratic weighted Kappa (QWK), which is the standard evaluation metric for the ASAP competition, officially designated by Kaggle. The metric is highly sensitive to incorrect predictions, which can help to evaluate the consistency of such model predictions. It measures the agreement between two raters, specifically between the human raters' score and the system's predicted score. Typically, the QWK score falls between 0 and 1, which indicates no consistency between raters and complete consistency, respectively. QWK is calculated using in the (1):

$$K = 1 - \frac{\sum_{i,j} w_{i,j} O_{i,j}}{\sum_{i,j} w_{i,j} E_{i,j}} \quad (1)$$

where the weights, observed scores, and predicted scores are represented by the matrices  $w$ ,  $O$ , and  $E$ , respectively. The number of essays that earn a score of  $i$  from the first rater and a score of  $j$  from the second rater is represented by the value of  $O_{i,j}$ . The weights are:

$$w_{i,j} = \frac{(i-j)^2}{(N-1)^2} \quad (2)$$

where  $N$  represents the total potential score. The outer product of two score histogram vectors is used to compute matrix  $E$ . Matrix  $O$  and matrix  $E$  are then normalized so that their sums are equal. And after that, the QWK score is calculated.

### 2.2.3. Implementation settings

We used the GPU hardware accelerator platform from Google Colab and the Tensorflow environment to build our model. For our experiments, we instantiated the transformer model, namely "Distilbert-base-uncased" from the hugging face library, and modified the architecture for the essay scoring task in order to fine-tune the weights using the labeled ASAP dataset. As input to the model, we perform tokenization for each essay from the ASAP dataset using the DistilBERTTokenizer class, which converts each sentence into a sequence of tokens and then a sequence of numbers. [CLS] and [SEP] tokens were inserted before and after each sequence, respectively. Our tokenization function ends up returning two arrays: an array of IDs that represent the sequence of words in an essay encoded as sequences of numbers, and an array of attention-mask  $A$  which is a binary sequence that helps the model either pay attention to the corresponding ID or not. The two arrays are then fed to our model as inputs. To build our model architecture, we begin by importing "Distilbert-base-uncased" from the transformer package and making some changes to arguments in the "DistilBertConfiguration" class by increasing dropout and attention-dropout from 0.1 to 0.2. then we initialize the base DistilBERT using the "TFDistilBertModel" class with no added classification head on top. We temporarily freeze Distilbert's pre-trained weights to prevent them from updating during the training of our added layers. We try several fine-tuning scenarios, as shown in Table 2. First, we keep the weights of the pre-trained DistilBERT frozen, and we train only the added output layers. We train with one dense layer for six epochs in the first scenario, then with three layers of 100, 32, and 1 for five epochs in the second. Another case of fine-tuning that we experimented with was to train all layers. To do that, we made the parameters trainable and trained with a small learning rate of  $1e-5$  for 2 epochs. As we built our model architecture, for training purposes, we considered 60% of the data for training, 20% for validation, and 20% for testing to measure the performance after training. We use Adam optimizer, rectified linear unit (ReLU) for the activation function, and 32 batch sizes.

Table 2. Experiment scenarios

DistilBert experiments	Parameters
Finetuning 1	Freeze base = true, epochs = 6, learning rate = $5e-5$ , added layers = dense 1
Finetuning 2	Freeze base = true, epochs = 5, learning rate = $5e-5$ , added layers = dense 100 and dense 32 and dense 1
Finetuning 3	Freeze base = false, epochs = 2, learning rate = $1e-5$ , added layers = dense 1

## 3. RESULTS AND DISCUSSION

In this section, we evaluate the performance of each of the scenarios we tried in our experiments. Figure 4 shows the results of QWK for the three fine-tuning models. In this part, we evaluate the performance of each of the scenarios we tried in our experiments. Figure 4 shows the results of QWK for the three fine-tuning models. It is clear that fine-tuning 3, which consists of updating all layers of the pre-trained

Distilbert, gives better results, especially in terms of QWK, which achieves 90%. Fine-tuning by training all the parameters of a pretrained large language model to perform better on new tasks the model wasn't trained on. For finetuning 1 and 2, which consist of freezing the pre-trained layers and only training the newly added layers. We can see that finetuning 2 with two layers on top with 100 and 32, respectively, and a regression layer achieves 84%. QWK, which is higher than 79% achieved by finetuning 1, which is similar to finetuning 1, but the difference here is to freeze the parameters of Distilbert and train only the output regression layer. These results show that fine-tuning with more layers gives better performance; however, it is worth mentioning that training without freezing increases time consumption.

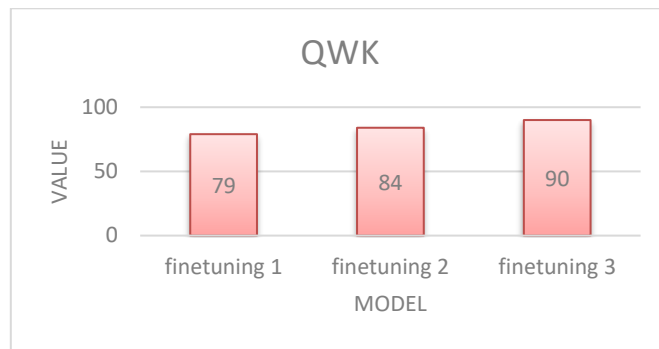


Figure 4. Performance in terms of QWK and accuracy

Comparing our findings with baseline models, the results are listed in Table 3. From the table, we can see that EASE (SVR) [4], which applies regression techniques to a set of hand-engineered features, has low performance in comparison to neural network-based methods. This explains the potential of deep learning to extract useful features that help assign a score that is clearly consistent with a human one. Combining Word2Vec with neural networks improves performance over methods that employ directly CNN and/or LSTM [10], [25]. R2BERT [21] proposes a fine-tuning of the pre-trained language model BERT. It combines regression along with ranking using multiple losses of the same task. The model gets good results but fails to overcome all neural network-based models. The observed QWK gotten by DistilBERT can outperform other baseline models, especially for finetuning 3, indicating the effectiveness of pre-trained models for automatic essay scoring tasks regarding their capacity to capture deeply semantic information from the text using the attention mechanism. Overall, results showed that the pre-trained distillation of BERT is feasible for the automatic assessment task and can correctly assess essays.

Table 3. Performance results achieved by different models based on QWK

Models	QWK
EASE (SVR) [4]	0.69
LSTM_CNN [10]	0.76
R2BERT [21]	0.79
SKIPFLOW [25]	0.76
Finetuning 1	0.79
Finetuning 2	0.84
Finetuning 3	0.90

#### 4. CONCLUSION AND FUTURE WORK





Evaluating essays automatically helps to overcome the bias of human evaluation and makes the process more consistent. Thus, a model that overcomes the shortcomings and improves the grading performance is critical. Above, we address the potential of applying the transformer architecture to the essay-scoring task. Specifically, we fine-tuned DistilBERT, which is a compressed and faster variant of the BERT model for essay scoring tasks. We tried different implementation scenarios of finetuning using the Kaggle ASAP dataset, and results show that training DistilBERT by updating all parameters gives better results than finetuning with a freezing base but comes with an increased cost in terms of time. To show the benefit of DistilBERT over the previous works, we compare it with different baseline models from the state of the art in terms of the QWK metric, and our method shows good results over them, especially in comparison to neural network-based methods. Our method works well for prompt-specific essay scoring tasks, as labeled data is

available while training. Our target for the next work is to generalize the model for unseen data as a cross-prompt task, which is a real-world case where labeled data are not always available and require huge effort and time to prepare.





## REFERENCES

- [1] S. Valenti, F. Neri, and A. Cucchiarelli, "An overview of current research on automated essay grading," *Journal of Information Technology Education: Research*, vol. 2, pp. 319–330, 2003, doi: 10.28945/331.
- [2] M. A. Hussein, H. Hassan, and M. Nassef, "Automated language essay scoring systems: a literature review," *PeerJ Computer Science*, vol. 2019, no. 8, p. e208, Aug. 2019, doi: 10.7717/peerj-cs.208.
- [3] A. K. . Maya, J. Nazura, and B. L. Muralidhara, "Recent trends in answer script evaluation – a literature survey," in *Proceedings of the 3rd International Conference on Integrated Intelligent Computing Communication & Security (ICIIC 2021)*, 2021, vol. 4, doi: 10.2991/ahis.k.210913.014.
- [4] P. Phandi, K. M. A. Chai, and H. T. Ng, "Flexible domain adaptation for automated essay scoring using correlated linear regression," in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 431–439, doi: 10.18653/v1/d15-1049.
- [5] E. B. Page, "Computer grading of student prose, using modern concepts and software," *Journal of Experimental Education*, vol. 62, no. 2, pp. 127–142, Jan. 1994, doi: 10.1080/00220973.1994.9943835.
- [6] M. Uto, "A review of deep-neural automated essay scoring models," *Behaviormetrika*, vol. 48, no. 2, pp. 459–484, Jul. 2021, doi: 10.1007/s41237-021-00142-y.
- [7] W. Song, D. Wang, R. Fu, L. Liu, T. Liu, and G. Hu, "Discourse mode identification in essays," in *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 2017, vol. 1, pp. 112–122, doi: 10.18653/v1/P17-1011.
- [8] F. Dong and Y. Zhang, "Automatic features for essay scoring - an empirical study," in *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2016, pp. 1072–1077, doi: 10.18653/v1/d16-1115.
- [9] D. Alikaniotis, H. Yannakoudakis, and M. Rei, "Automatic text scoring using neural networks," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 715–725, doi: 10.18653/v1/P16-1068.
- [10] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring," in *EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings*, 2016, pp. 1882–1891, doi: 10.18653/v1/d16-1193.
- [11] F. Dong, Y. Zhang, and J. Yang, "Attention-based recurrent convolutional neural network for automatic essay scoring," in *CoNLL 2017 - 21st Conference on Computational Natural Language Learning, Proceedings*, 2017, pp. 153–162, doi: 10.18653/v1/k17-1017.
- [12] Y. Wang, Z. Wei, Y. Zhou, and X. Huang, "Automatic essay scoring incorporating rating schema via reinforcement learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, 2018, pp. 791–797, doi: 10.18653/v1/d18-1090.
- [13] R. Kumar, S. Mathias, S. Saha, and P. Bhattacharyya, "Many hands make light work: using essay traits to automatically score essays," in *NAACL 2022 - 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, 2022, pp. 1485–1495, doi: 10.18653/v1/2022.naacl-main.106.
- [14] I. Persing and V. Ng, "Modeling argument strength in student essays," in *ACL-IJCNLP 2015 - 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*, 2015, vol. 1, pp. 543–552, doi: 10.3115/v1/p15-1053.
- [15] H. V. Nguyen and D. J. Litman, "Argument mining for improving the automated scoring of persuasive essays," *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, vol. 32, no. 1, pp. 5892–5899, Apr. 2018, doi: 10.1609/aaai.v32i1.12046.
- [16] A. Louis and A. Nenkova, "Automatically assessing machine summary content without a gold standard," *Computational Linguistics*, vol. 39, no. 2, pp. 267–300, Jun. 2013, doi: 10.1162/COLL\_a\_00123.
- [17] A. Vaswani *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.
- [18] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [19] P. U. Rodriguez, A. Jafari, and C. M. Ormerod, "Language models and automated essay scoring," *arXiv*, 2019, [Online]. Available: <http://arxiv.org/abs/1909.09482>.
- [20] E. Mayfield and A. W. Black, "Should you fine-tune BERT for automated essay scoring?," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 151–162, doi: 10.18653/v1/2020.bea-1.15.
- [21] R. Yang, J. Cao, Z. Wen, Y. Wu, and X. He, "Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking," in *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, 2020, pp. 1560–1569, doi: 10.18653/v1/2020.findings-emnlp.141.
- [22] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv*, 2019, [Online]. Available: <http://arxiv.org/abs/1910.01108>.
- [23] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: a survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, Jun. 2021, doi: 10.1007/s11263-021-01453-z.
- [24] J. Devlin, M.-W. Chang, K. Lee, K. T. Google, and A. I. Language, "BERT: pre-training of deep bidirectional transformers for language understanding," *Naacl-Hlt 2019*, no. Mlm, pp. 4171–4186, 2018, [Online]. Available: <https://aclanthology.org/N19-1423.pdf>.
- [25] Y. Tay, M. Phan, L. A. Tuan, and S. C. Hui, "SkipFlow: incorporating neural coherence features for end-to-end automatic text scoring," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, pp. 5948–5955, Apr. 2018, doi: 10.1609/aaai.v32i1.12045.





**BIOGRAPHIES OF AUTHORS**

**Soumia Ikiss**     is a Ph.D. student at the National Institute of Posts and Telecommunications, Rabat, Morocco since 2021. Graduated from the University of Sultane Moulay Slimane, Beni Mellal, Morocco in 2020 with a master's degree in 'Business Intelligence'. Her research areas are artificial intelligence, deep learning, and natural language processing and their application to real-world problems. She is currently working on the application of artificial intelligence in the education industry, mainly on the evaluation part. She can be contacted at email: [soumia.ikiss97@gmail.com](mailto:soumia.ikiss97@gmail.com).







**Najima Daoudi**     is a full professor at the School of Information Sciences, Rabat, Morocco. She has an engineering degree from the National Institute of Statistics and Applied Economics (INSEA), Rabat, Morocco, and a Ph.D. in Computer Science from ENSIAS, Rabat, Morocco. Her research interests include artificial intelligence, e-learning, recommendation systems, and natural language processing. She can be contacted at email: [ndaoudi@esi.ac.ma](mailto:ndaoudi@esi.ac.ma).



**Manar Abourezq**     is a professor at the School of Information Sciences, Morocco. She is an Engineer of the National School for Computer Science and Systems Analysis and holds a Ph.D. in Computer Science from Mohammed V University in Rabat, Morocco. Her research interests include artificial intelligence, Arabic natural language processing, and cloud computing. She can be contacted at email: [manar.abourezq@gmail.com](mailto:manar.abourezq@gmail.com).



**Mostafa Bellafkih**     is a professor at The National Institute of Posts and Telecommunications (INPT) in Rabat, Morocco. received a Ph.D. thesis in Computer Science from the University of Paris 6, France, in June 1994 and a doctorate Science in Computer Science (option networks) from the University of Mohammed V in Rabat, Morocco. His research interests include network management, knowledge management, AI, data mining, and databases. He can be contacted at: [mbellafkih@yahoo.com](mailto:mbellafkih@yahoo.com).