Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

A deep learning model for automated marking of students' assessments in a learning management system (LMS)

Mohammed Altamimi [1, *], Yaser Altameemi [2], Adel Alkhalil [3], Romany F. Mansour [4], Magdy Abdelrhman [5], Ikhlaq Ahmed [6], Aakash Ahmad [7], Azizah Alogali [8, 9]

[1]*Department of Information and Computer Science, College of Computer Science and Engineering, University of Ha'il, Ha'il 81481, Saudi Arabia*
[2]*Department of English, College of Arts and Literature, University of Ha'il, Ha'il 81481, Saudi Arabia*
[3]*Department of Software Engineering, College of Computer Science and Engineering, University of Ha'il, Ha'il 81481, Saudi Arabia*
[4]*Department of Mathematics, Faculty of Science, New Valley University, El-Kharga 72511, Egypt*
[5]*Applied College, University of Ha'il, Ha'il, Saudi Arabia*
[6]*Department of Computer Software Engineering, National University of Sciences and Technology, Islamabad, Pakistan*
[7]*School of Computing and Communications, Lancaster University Leipzig, Leipzig 04109, Germany*
[8]*Department of Educational Leadership, University of Rochester, Rochester, NY 14627, USA*
[9]*Department of Educational Leadership, University of Akron, Akron, OH 44325, USA*

ARTICLE INFO

Article history:
Received 12 April 2025
Received in revised form
20 August 2025
Accepted 30 August 2025

Keywords:
Learning management systems
Automatic grading
Fill-in-the-gap questions
Spelling error correction
Deep learning

ABSTRACT

Learning Management Systems (LMSs) are widely used to support teaching and learning, with platforms such as Blackboard managing lectures, activities, assessments, and reports. Although LMSs provide useful tools and some automated feedback, the accuracy of evaluating students' typed responses has received little attention in prior research. A particular issue arises in fill-in-the-gap questions, where answers are marked only if they exactly match the instructor's input, often leading to unfair grading for minor spelling errors. To address this problem, we propose a model that integrates the Levenshtein edit distance with deep learning methods to identify and correct spelling errors, enabling fairer and more accurate automatic grading. The model demonstrated strong performance, achieving an average F1-measure of 0.938 on a dataset of misspelled words.

1. Introduction

Higher education institutions are eager to utilize emerging technologies in the learning process, particularly since the rapid development of artificial intelligence (AI) and the accessibility of various tools by higher education institutions. Learning management systems (LMSs) have been used as an effective application for the learning process, adopted by Blackboard, which is used to manage courses, including lecturing, activities, assessments, evaluations, and reports.

The Blackboard system has been applied in the learning process at many educational institutions since the start of the COVID-19 pandemic. During the pandemic-related lockdowns, all courses shifted to distance learning, and all assessments were performed online due to restrictions at the time. This shift had a positive impact on the teaching staff concerning their consideration of the importance of utilizing technology in the learning process, and this experience changed many staff attitudes toward the importance of enhancing education quality through the effective utilization of LMSs in the learning process (Aljaloud et al., 2022).

However, a central issue not yet addressed is the accuracy of marking students' assessments when students must type their answers to certain questions, as typos are common. This problem appears clearly in "filling the gap"-type questions, when an instructor writes questions about these gaps and their correct answers. As a result, the accuracy of automatic marking of students' answers is low, and it may appear unfair to mark certain answers wrong if only a single letter was missed. The evolution of this technical issue has raised concern among instructors about either avoiding "filling the gap" questions or marking them manually. Although LMSs have been developing rapidly in educational

* Corresponding Author.
Email Address: mh.altamimi@uoh.edu.sa (M. Altamimi)
https://doi.org/10.21833/ijaas.2025.10.001
Corresponding author's ORCID profile:
https://orcid.org/0000-0002-4170-6910

processes, issues regarding evaluation still require consideration (Deeva et al., 2021). Thus, the purpose of this study is to develop a method for correcting students' typing errors. This study applies to a multidisciplinary research approach that considers both linguistic and computational aspects. Thus, the research also aims to diagnose students' typing errors in their assignments/exams (i.e., linguistic goal) and mark students' assessments accurately by solving technical issues in the LMS (i.e., computational goal). The central contributions of this paper are as follows:

- Fairness: Develop a model that considers the types of potential spelling errors and evaluates students' answers based on their knowledge rather than spelling mistakes, particularly when the spelling errors do not affect the semantics of the answer.
- Methodological perspectives: Integrate the Levenshtein edit distance with deep learning models, which is effective at enhancing the accuracy of automatic marking of students' answers to identify and correct spelling errors.
- Practical solutions: The proposed model can be applied to any LMS to prevent students from receiving unfair marks due to typos. This will also assist instructors in avoiding manual correction of students' answers, whereas the role of the instructors will be primarily to analyze the marking.

This paper begins by providing the background of the issue and related research. In this section, the authors discuss reasons that students may make linguistic errors, including the types of errors. Then, previous works regarding models used to correct sentences automatically will be discussed. After, the paper highlights the methodology and proposed model of the study. Following this section, the results will be discussed in relation to answering the research question. Finally, the paper presents its conclusion, including the main findings and limitations

2. Background and related research

In this section, the authors discuss the findings of reviewing related studies regarding the errors students introduce in their answers or in conducting their assessments. The authors will highlight the linguistic aspects, considering the kinds of typos and issues students face in their expressions. Then, the authors will highlight how these types of errors can be computationally classified.

As stated in the introduction, the central issue of this study is the lack of a tool to automatically correct students' answers to "filling the gap" questions or short answers in the Blackboard system. Usually, the answers to these types of questions are only a single word or two. Then, students' answers are marked automatically based on the inputs of the instructor. In doing so, students' answers will be marked according to whether they match the inputs exactly, without any misspellings of words (i.e., correct answers).

For example, if the correct answer is "write" and the student types "writ," the answer will be marked as wrong. Thus, it may appear unfair to mark students' answers in this way, as the purpose of the evaluation here is to match the words typed by the students to the correct words. Rather, the purpose of evaluation should be to check students' understanding of the subject, rather than their spelling skills.

The screenshot from the Blackboard system in Fig. 1a shows how omitting the "s" letter and replacing "I" with "u" from the word linguistics renders the answer wrong. Another example in Fig. 1b in the second answer shows how omitting the "l" from the name "Oller" renders the answer wrong. This method of evaluating students' assessments involves marking as 0 or full marks.

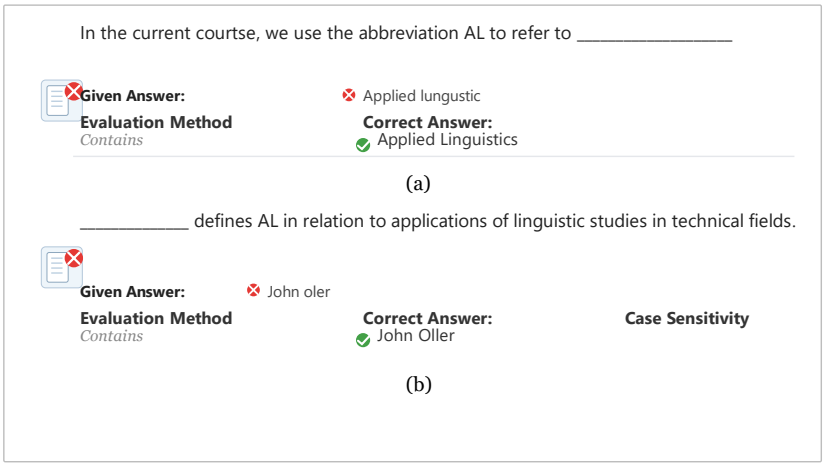

Fig. 1: Screenshot from the Blackboard system showing the marking of "fill-in-the-gap" questions (a) and (b)

2.1. Types of errors from a linguistic perspective

This research focuses on the analysis of errors more than mistakes, as in the study by Hourani (2008), who suggests that mistakes can be self-corrected while errors can be difficult to overcome. This perspective relates to the point that students submit their assignments or exams without

considering some of their errors, even when they revise their work.

Many researchers have studied types of errors from the perspective of the reasons for them, such as negative transfer of native language knowledge to the target language (Hasyim et al., 2022). The study in Cheng (2021), for instance, demonstrated several types of errors among non-native language students in general, without a clear focus on specific types of errors. For example, she discussed various linguistic levels, such as spelling, punctuation, prepositions, and article errors. The issue here is that among these general types, there are several subtypes, and this enhances the complexity of analyzing writing errors. However, for the purpose of this research, the authors focus on lexical errors and solving this issue for students when submitting their assessments. The authors agree with some studies that we must be more specific in analyzing the errors. Thus, the authors follow Cook's (1999) classification of typing errors, which suggests that there are four main types of spelling errors: omitting one or more letters, substituting by replacing one letter with another, inserting by adding one or more letters, and transposition by reversing the places of the letter(s).

Another perspective concerns the discussion of how several studies have dealt with the categorization and analysis of students' errors in their typing. Many studies have evaluated the errors qualitatively by manually correcting students' work. However, other studies applied automated marking to increase the proficiency of correcting the mistakes, as well as the efficiency of students' self-correction of their errors (Bridle, 2019; Cheng, 2021; Gilquin and Laporte, 2021). One of the tools used in teaching English is data-driven learning (DDL), adopted during the COVID-19 pandemic to help learners reflect on their work using DDL. This study investigates the effectiveness of using DDL to enhance students' level of writing. However, in the current study, the instructors faced problems in auto-correcting students' assignments and exams using Blackboard. The academic staff began applying auto-correction to students' answers in online activities and exams after the university switched to distance learning amid the COVID-19 pandemic. As stated above, Blackboard's auto correction of students' answers might be unfair when students omit a single letter. Also, students do not have a chance to correct their typing after the submission and are only evaluated according to their written answer, without full consideration of their knowledge in the subject.

2.2. Automatic spelling correction

The methods applied to the automatic correction of spelling mistakes generally involve dividing the process of correction into three phases: error detection, error correction, and error ranking (Kukich, 1992). Error detection is the process of detecting a word that is misspelled, error correction is the process where a system corrects a misspelled word, and error ranking is a process that sorts the suggested corrections and proposes the optimal correct word. Here, the authors will primarily focus on previous studies that highlight methods used to correct spelling errors.

The early stages of handling spelling are by identifying errors. The traditional approach to spelling error detection is based on finding the similarity between a misspelled word and the correct word presented in a dictionary. During dictionary lookup, every word of the input document is compared with words that exist in the dictionary. Such techniques as hashing (Mosavi Miangah, 2014) or search tree (Shang and Merrett, 1996) in the traditional approaches have been used to detect errors. However, the contribution of this approach is limited to the size of the dictionary. In other words, if the word does not exist in the dictionary, then it will be difficult to find the correct term, as the dictionary should be continuously updated. Thus, a robust dictionary would require having a good-sized vocabulary to perform well (Kukich, 1992).

These methods of spelling correction are effective for identifying non-word errors. However, another type of error to consider is real-world errors, which can be difficult to detect, as the word may exist in the dictionary (e.g., using "can" instead of "car"). Identifying these errors may require semantic analysis and a thorough review of the context surrounding the word (Pirinen and Lindén, 2014). Through developing the error correction phase, several methods have been proposed, such as minimum edit distance, rule-based methods, language models, neural networks, and deep learning.

A salient method of correcting errors is the edit distance method, which calculates the distance between the misspelled word and the correct word found in the dictionary. It is based on correcting errors using the Levenshtein edit distance (Levenshtein, 1965) and the Damerau–Levenshtein distance (Damerau, 1964), both of which correct basic types of errors, such as insertions, deletions, and substitutions. Furthermore, Damerau–Levenshtein considers correction of transportation errors that occur between two letters (Hagen et al., 2017).

Another method of error correction is the ruler-based method, wherein the entire word is examined in a dictionary that contains only roots. This method works by finding the best-matching correct word by comparing it with the original word, but it is highly dependent on language features and requires language experts to develop the rule (Fahda and Purwarianti, 2017). Despite this method taking time and effort to develop, it produces accurate results in most cases.

Another method involves using statistical language models that can achieve satisfactory results in automatic spelling correction (Ferrero et al., 2014; Mirzababaei and Faili, 2016). The study by Azmi et al. (2019) used an n-gram (n = 1–3) model of words and machine learning with support vector machines

(SVM) for detecting real-word errors. Then, correcting errors using the Damerau-Levenshtein distance achieved an accuracy of 98%. Furthermore, Flor et al. (2019) developed a minimally supervised model that uses both contextual and non-contextual features, including word frequency, phonetic similarity, orthographic similarity, n-grams, and word embeddings, and their best results achieved 87.63% accuracy using all features.

Another approach combines a statistical language model with an n-gram model based on word frequency. This method estimates the probability of a candidate word occurring, given its preceding context in the training corpus. If a particular n-gram sequence is absent from the corpus, its probability decreases and is ultimately disregarded. The main advantage of this approach is its applicability to any language. However, it requires a representative set of training and testing data, along with annotated errors, to be effective (Malema et al., 2019).

Deep learning has introduced new possibilities by allowing the prediction of the correct word based on the surrounding context in instances of spelling errors. This is performed by associating a word with a fixed-size vector, which helps to identify the correct word that is close in the embedding space. This method is utilized in spelling error correction to arrange a list of correct words, and it helps with identifying words that occur in the same context and are semantically similar (Hládek et al., 2020). Furthermore, Lee et al. (2020) applied various deep learning language models to correct context-sensitive spelling errors. Models used in this study include Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), the robustly optimized BERT approach (RoBERTa) (Liu et al., 2019), and the cross-lingual language RoBERTa (XLM-RoBERTa) (Conneau et al., 2019). They achieved their best results for the detection and correction of spelling errors, achieving a more than 96% F1-measure of errors using RoBERTa.

Etoori et al. (2018) applied automatic spelling correction to Indian languages using deep learning techniques. Their study showed that sequence-to-sequence (Seq2Seq) models, combining convolutional and recurrent encoder–decoder structures, outperformed other approaches. The method achieved 85.4% accuracy in spelling correction for Hindi and 89% accuracy in spelling identification for Telugu.

Salhab and Abu-Khzam (2024) proposed a model for Arabic spelling error correction and introduced different Seq2Seq models with an error injection schema. They achieved a character error rate (CER) of 1.11% and word error rate (WER) of 4.8%, resulting in 77.93% and 83.84% character and word error reduction rates, respectively. In the Malay language, Sooraj et al. (2018) proposed an error detection model, called character-based Long Short-Term Memory (LSTM), using an input entered with a sequence of characters. The trained network will be able to predict the next character given the previous letters using the SoftMax classifier.

Based on the above discussion of previous research, Table 1 presents a summary of the advantages and disadvantages of the different methods applied to the automatic correction of spelling mistakes.

Table 1: Comparison of methods for automatic spelling correction: advantages and limitations

Method	Advantages	Limitations
Traditional approach	Simplicity in comparison to other complex models Efficient and straightforward in its implementation	Limited in considering the textual context (Kukich, 1992) Lack in dealing with new vocabulary (Pirinen and Lindén, 2014)
Edit distance	Detect and correct typographical errors effectively using character-level edits (Levenshtein, 1965) Mathematical rigor for the similarities between words	Expensive and necessitates a large dataset (Navarro, 2001) Lack of consideration of textual context
Rule-based methods	Customizable to different languages' grammatical rules (Pedler, 2001) Transparency in the applied rules (Golding and Schabes, 1996)	Requires a linguist/expert in the target language rules (Pedler, 2001) The complexity of some linguistic rules may affect the efficiency of the model (Golding and Schabes, 1996)
Statistical language model	More accurate in probabilistic corrections than rule-based models Considers context in correcting errors (Chen and Goodman, 1999)	Requires a large dataset (Chen and Goodman, 1999) It can be computationally expensive (Katz, 1987)
Language model	Awareness of textual context Predicts the likelihood of word sequences (Kneser and Ney, 1995)	Requires a large amount of data (Chen and Goodman, 1999) Requires intensive computational resources
Neural networks	Apply complex patterns from texts Adaptable to various languages and text types (Gurney, 1997)	Difficult to interpret decision-making processes (Montavon et al., 2018) Requires intensive computational resources (LeCun et al., 2015)
Deep learning model	High accuracy and precision (Chollampatt and Ng, 2018) Considers context in corrections (Devlin et al., 2019) Handles different error types (Zhao et al., 2019)	Depending on the data, which requires high-quality labeled data for training (Zhang et al., 2021) Lack of transparency in explaining how specific corrections have been made (Rudin, 2019)

3. Proposed model for spelling correction

In this study, we propose two spelling error correction models to detect spelling errors and correct these mistakes. The model adopted in the first part uses the Levenshtein edit distance to detect spelling mistakes, while in the second part, the authors use a deep learning model (LSTM) to correct spelling errors. The proposed methodology employs both models to address misspelled words effectively. As such, in this section, we outline the steps taken by the authors to perform this analysis.

The first part of the proposed model begins by using the Levenshtein edit distance (Levenshtein, 1965), as well as fuzzy matching based on a given threshold. The Levenshtein edit distance compares two words (correct and misspelled) to identify the percentage of differences between the words typed and the correct words in the reference corpus. The second part of the proposed method uses the LSTM model to predict the correct word. The model is trained on the brown corpus to learn a variety of words from various categories. The architecture of the LSTM unit contains three gates: a forget gate, an input gate, and an output gate, along with the memory cell, which is part of the LSTM unit. These gates are responsible for adding or removing information from the cell state, where the forget gate determines the relevant information to be retained or discarded during training, with values determined by the sigmoid function, usually ranging from 0 to 1. A value close to zero means the information is ignored, while a value close to one indicates the information is significant, meaning it must be kept. The input gate manages which information is stored in the memory, while the output gate decides which information is exposed to the memory cell. Both gates are controlled by the sigmoid activation function to determine whether information should be processed. In the experiment, the input will be a sequence of characters, and the trained network will be able to predict the next character in the misspelled word given the previous ones with the help of a SoftMax classifier (Sooraj et al., 2018).

As shown in Fig. 2, the methodology combines both Levenshtein edit distance matching for error detection and a trained deep learning LSTM model for more context-aware corrections, as well as provides a robust approach to detect and correct misspelled words. Both models will participate in enhancing the fairness of the automatic correction of students' answers in Blackboard. Levenshtein distance to determine how closely a given misspelled word matches the words in the reference word list in Peter Norvig's dataset, in our experiment. A threshold of 80% similarity is used for two purposes in our experiment.

- Correcting simple misspellings of words when the Levenshtein ratio exceeds the threshold, the misspelled word is deemed "correctable" by the fuzzy matcher, and the corresponding correct word is identified.
- Identification of potential correction candidates. If the fuzzy match ratio is high enough (above the threshold), it suggests that the misspelled word is close enough to a correct word in the list, and the word can be corrected directly.

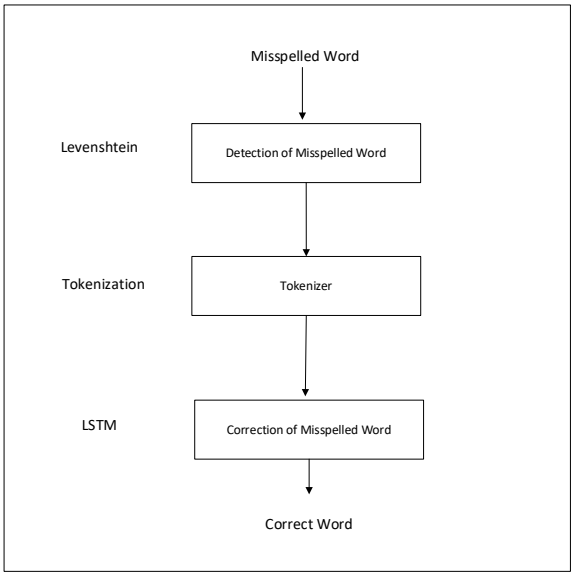

Fig. 2: Architecture of the proposed spelling error correction model combining Levenshtein distance and LSTM

In the second stage, words that don't have a sufficiently high percentage compared to those in the list are passed on to the LSTM model for more advanced predictions. The misspelled words that failed in the first phase are used as input for the LSTM to predict the correct word: The Tokenizer object transforms words into sequences of integers. For the input misspelled word, a sequence of tokenized integers is generated. The model then tries to predict the next word in the sequence (based on the training it has received from the brown corpus). The output is a sequence of probabilities for each possible word in the vocabulary. The predicted word corresponds to the word with the highest probability. The prediction is based on surrounding words in a sentence and uses the learned patterns from those sequences to make an informed prediction. Table 2 shows the settings of the LSTM models used in our experiment.

Table 2: Details of the LSTM model

Layer	Output shape	Number of parameters
Embedding	(None, 180, 50)	2,490,800
LSTM	(None, 50)	20,200
Dense	(None, 49816)	2,540,616
Total parameters	15,154,850	
Trainable parameters	5,051,616	
Non-trainable parameters	0	
Optimizer params	10,103,234	

4. Experiments and results

In this section, we first provide the training and testing datasets used in the model to demonstrate the comprehensive evaluation capability of the proposed model using various datasets. Then, the evaluation metrics used in the experiment are provided. Finally, the results of the model are

discussed in relation to answering the research question.

4.1. Training dataset

For training the detection models, we used the brown corpus, which has been selected for two main reasons. First, it has a good range of data to be used in the training, as it includes 1,014,312 words. Second, it offers distinct text types collected from 15 text categories, including reportage, editorial, reviews, religion, skill and hobbies, popular lore, biography, U.S. government, learning, general knowledge, mystery and detective, science, adventure and western, romance, and humor. This variety will help the model to consider the different contexts of lexical words in the training process, and the data are tokenized and processed using text to sequences. The sequences are then fed to the LSTM model for training purposes.

4.2. Testing data

For the testing dataset, we used the spelling corrector dataset from Peter Norvig's classic spelling corrector, which contains a variety of misspelled words and correct words from various topics, including medical, business, teaching, and sports. In total, 9,017 words are used to test the model. Table 3 details the statistics of the dataset. The baseline is to achieve 75% of 270 correct words on Spell-testset1 and achieving 68% of 400 correct words on Spell-testset2.

Table 3: Composition of testing datasets used in the experiments (Birkbeck, Wikipedia, Spell-testsets, Aspell)

Name	Total number of words	Total number of correct words
Birkbeck	6,137	0
Wikipedia	1,923	61
Spell-testset1	142	75
Spell-testset2	364	68
Aspell	451	43

4.3. Evaluation metrics

Evaluation metrics were employed to assess the system performance, and the F1-measure score was used to calculate the harmonic mean of recall and precision (Davis and Houck, 1992). It is defined as:

$$\mathrm{F-measure} = 2 * \frac{Recall \times Precision}{Recall + Precision},$$

Alongside the average variation metric. The average variation metric quantifies differences between misspelled words and their correct counterparts using the Levenshtein edit distance, facilitating a comparison of differences between the two words.

4.4. Results

The proposed model is evaluated with different datasets. Table 4 shows the results in terms of F1-measure score metrics and the average word variation using our proposed method and Bidirectional Encoder Representations from Transformers (BERT). Our proposed method consistently outperforms the BERT-based approach across all test datasets in terms of F1-measure, indicating higher accuracy in detecting spelling errors and dealing with these errors to avoid marking them as mistakes in Blackboard and to correct these mistakes. The initial experiments using the LSTM architecture for training showed that the proposed model has a higher success rate in the detection of both correct and misspelled words. As seen in Table 4, the F1-measure scores for Spell-testset1 and Spell-testset2 are 0.964 and 0.972, respectively, compared to BERT's 0.750 and 0.570. Even on other datasets such as Wikipedia and Aspell, the proposed method still shows a clear advantage with F1 scores of 0.944 and 0.937, outperforming BERT's 0.430 and 0.710.

In addition, on larger datasets with a higher frequency of spelling errors, such as Birkbeck, the proposed method achieved its lowest F1-measure of 0.927, which still outperformed BERT's corresponding score of 0.901. It is worth noting that although the number of misspelled words in the Birkbeck datasets is high, the model achieved good results.

Table 4: Performance comparison of the proposed model and BERT across testing datasets

Name	F1-measur	Average variation	F1-measure	Average variation
	Our method		BERT	
Birkbeck	0.927	79.4	0.901	84.2
Wikipedia	0.944	81.2	0.430	85.0
Spell-testset1	0.964	82.7	0.750	85.2
Spell-testset2	0.972	83.0	0.570	84.9
Aspell	0.937	83.2	0.710	85.1

In terms of average variation, the proposed method consistently yields slightly lower values across datasets compared to the BERT, indicating the average differences between misspelled words and their correct counterparts. The authors set a threshold of 80% similarity between the misspelled word and the correct word before the input answer is marked as a grade with the correction. This

approach considers giving each letter a weight depending on the word length. For example, if the word "about" is written as "abbot," the answer will be marked as incorrect because the variation ratio is approximately 79%, falling below the threshold. conversely, if the word "consisit" is corrected to "consist," and the variation Ratio of the two words is 88%. Thus, in this case, the misspelled word has been marked as correct, and the word was corrected. Overall, these findings show that the applied model not only targets the fairness of marking students' answers but also provides corrections to spelling errors.

The authors classified the types of errors based on the variation ratio to discuss the effectiveness of the model, as shown in Table 5.

Table 5: Analysis of misspellings by variation ratio and error type with examples

Variation ratio range	Type of error	Example
0.90 – 1.00	Very minor edits	conciousness→ consciousness sufer→ suffer summer_time→ summertime
0.85 – 0.89	Minor phonetic/typos	conect→ connect compossed→ compose consisit→ consist
0.80 – 0.84	Moderate phonetic	comfer→ comforted exsenive→ extensive egnore→ ignore
0.50 – 0.79	Heavier edits	imagin→ magi incessant→ Vincent encluding→ ending
0.00 – 0.49	Severe error	comaritialy→ was a_rone→ man voat→ was

It might be seen that the first type has been corrected accurately, specifically when students missed some letters that do not affect the overall percentage of accuracy of the written words, such as "conciousness" that was corrected to "consciousness." In this case, the student's answer will not be marked as wrong, while it will be calculated as a mark, so this process will enhance the fairness of students' marking as they will not lose the whole mark due to a little spelling mistake. This scenario is also like the second type that might be linked to minor phonetic/typo mistakes. In this type, the students may misspell a word that has little difference with its written form, such as in the word "conect" that was corrected to "connect."

The third type is heavier edits, and this occurs when there is a need for a deep consideration of mistakes, as is the case in Table 5 with the word "encluding" that was corrected to "ending." These corrections might be helpful to students to provide feedback in their assessments, excluding exams, because answers in the exam are marked for the purpose of grading rather than providing students with feedback.

When analyzing incorrect corrections with severe errors and high variation, several factors can be identified. First, the words that have not been corrected are uncommon in the corpus itself. For example, "comaritialy" and "cemfurmation" do not have clear matches with words in the dataset, so they are not categorized as either correct or incorrect. Another plausible reason is that when a word has three letters or fewer, such as "agg," the word may be used as an acronym because such a word might be corrected to either egg or age. A similar reason for missing an error in the model is that some words appear compounded, such as "evil_looking" and "event_bold." These three possible reasons affect the accuracy of the automatic marking of students' mistakes; while they do not present a serious issue in the model, they might develop later, as mentioned above, for the purpose of providing students with feedback rather than grading students' answers. Fig. 3 shows the flowchart to clarify how the system would integrate into Blackboard.

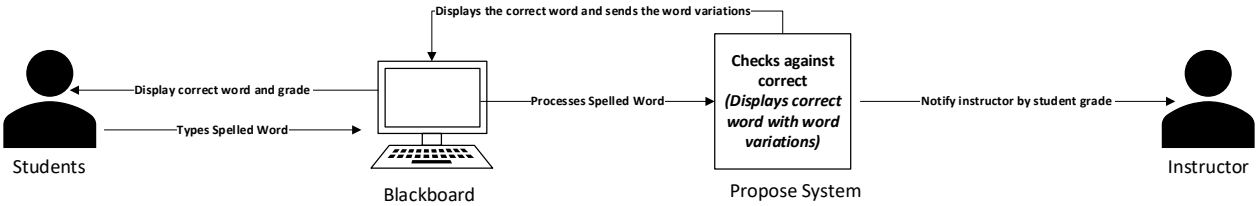

Fig. 3: Flowchart of the integration between Blackboard and our proposed system

5. Discussion

After assessing the results, we can see how the developed model can enhance the fairness of students' marks, as well as decrease the time wasted on marking students' answers to "filling the gap" questions. Instructors need only upload the correct answers to the questions to Blackboard; then, the model will detect the percentage of differences between the correct and incorrect spellings, and the mark will be dependent on the percentage of the difference between the two. The model has been

developed to identify when the difference between the right and wrong spelling is more than 20%, in which case the answer will be marked as correct. Meanwhile, if the difference is less than 20%, the answer will be marked as incorrect.

Even though the accuracy of marking student work using the model has not reached 100%, the proposed model will facilitate the instructor's role in scanning the marking and double-checking the marking process, rather than marking students' sheets from scratch. As stated, the angle of the research is taken from the perspective of facilitating the work of instructors in the marking process. Additionally, the model is designed to provide feedback to students and to ensure students' answers are marked fairly and accurately.

Our current approach is based on deep learning, which generally excels at handling spelling errors. As a result, the F1-measure scores of our model (ranging from 0.927 to 0.972) indicate that it may outperform other methods in the discussion in the literature.

For instance, the work by Azmi et al. (2019) used n-grams and SVMs to achieve an F1-measure of 90.7 using Damerau-Levenshtein distance. Thus, n-gram models still seem to have a slight edge in accuracy, especially when statistical distance measures like Damerau-Levenshtein are used. Other methods using statistical language models and n-grams also report good results, such as Flor et al. (2019), which achieved 87.63% accuracy using a supervised logistic regression model.

Additionally, Seq2Seq models applied in other languages (e.g., Hindi, Telugu) reported accuracies in Salhab and Abu-Khzam (2024) of around 85.4% to 89%. In contrast, Studies using advanced transformer models like BERT, RoBERTa, and XLM-RoBERTa, such as Devlin et al. (2019), Liu et al. (2019), and Conneau et al. (2019), have shown impressive results with 96% of F1-measure.

However, the proposed method, which is based on a combination of LSTM and Levenshtein edit distance, demonstrates impressive performance. It achieves an average F1-score of 94% across the entire dataset, outperforming BERT in direct comparisons.

6. Conclusions and future work

The study applied the model of increasing the fairness of the automatic correction of students' spelling errors in their answers on exams. Although previous studies discussed issues related to providing feedback to learners, the current study developed a model to facilitate automatic correction for instructors. Therefore, the main contribution of this study is centered on recognizing how students' answers should be marked automatically in LMSs.

The study fills the gap in the research on marking students' spelling errors and recognizing these mistakes as still correct, specifically when they do not affect the evaluation of the student's knowledge. This gap has been filled by combining the Levenshtein edit distance with deep learning models, achieving 94% of F1 measure score in detecting, correcting, and evaluating the answers. The proposed model will increase the efficiency of the automatic marking of students' answers, as well as the fairness of automatic marking.

Although 100% accuracy has not yet been achieved, the authors highlight the importance of limiting the role of instructors to check corrections to the model rather than marking from scratch. The solutions provided in this study open the door to the development of models for not only correcting spelling errors at the lexical level but also considering the semantics and contexts of the answers to essay questions. Specifically, we now highlight the potential for extending the current approach to support the semantic evaluation of open-ended and essay-type responses.

The solutions proposed in this study provide a foundation for developing models that go beyond surface-level error detections, such as spelling or lexical inaccuracies, to include deeper semantic and contextual analysis. Future research could explore the integration of these models into educational assessment tools capable of evaluating student writing on multiple dimensions, including grammatical accuracy, coherence, cohesion, argumentation structure, and relevance to the prompt. This would be particularly valuable for assessing complex writing tasks in educational settings, enabling more nuanced feedback and support for language development. Therefore, the authors suggest that this model be developed further to mark essay questions and long answers.

List of abbreviations

AI	Artificial intelligence
BERT	Bidirectional encoder representations from transformers
CER	Character error rate
CNN	Convolutional neural network
DDL	Data-driven learning
LMS	Learning management system
LSTM	Long short-term memory
RoBERTa	Robustly optimized BERT approach
Seq2Seq	Sequence-to-sequence
SVM	Support vector machines
WER	Word error rate
XLM-RoBERTa	Cross-lingual language RoBERTa

Acknowledgment

This research was funded by the Scientific Research Deanship at the University of Ha'il, Saudi Arabia, through project number RG-21 149.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Aljaloud AS, Uliyan DM, Alkhalil A, Elrhman MA, Alogali AFM, Altameemi YM, Altamimi M, and Kwan P (2022). A deep learning model to predict student learning outcomes in LMS using CNN and LSTM. IEEE Access, 10: 85255–85265. https://doi.org/10.1109/ACCESS.2022.3196784

Azmi AM, Almutery MN, and Aboalsamh HA (2019). Real-word errors in Arabic texts: A better algorithm for detection and correction. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 27(8): 1308–1320. https://doi.org/10.1109/TASLP.2019.2918404

Bridle M (2019). Learner use of a corpus as a reference tool in error correction: Factors influencing consultation and success. Journal of English for Academic Purposes, 37: 52–69. https://doi.org/10.1016/j.jeap.2018.11.003

Chen SF and Goodman J (1999). An empirical study of smoothing techniques for language modeling. Computer Speech & Language, 13(4): 359–394. https://doi.org/10.1006/csla.1999.0128

Cheng YH (2021). EFL college students' concordancing for error correction. English Teaching & Learning, 45: 431–460. https://doi.org/10.1007/s42321-021-00075-5

Chollampatt S and Ng HT (2018). Neural quality estimation of grammatical error correction. In the Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium: 2528–2539. https://doi.org/10.18653/v1/D18-1274

Conneau A, Khandelwal K, Goyal N, Chaudhary V, Wenzek G, Guzmán F, Grave E, Ott M, Zettlemoyer L, and Stoyanov V (2019). Unsupervised cross-lingual representation learning at scale. Arxiv Preprint Arxiv:1911.02116. https://doi.org/10.48550/arXiv.1911.02116

Cook S (1999). Nonsymmetric error correction revisited. Applied Economics Letters, 6(7): 467–470. https://doi.org/10.1080/135048599353014

Damerau FJ (1964). A technique for computer detection and correction of spelling errors. Communications of the ACM, 7(3): 171–176. https://doi.org/10.1145/363958.363994

Davis LM and Houck CL (1992). Is there a midland dialect area?—again. American Speech, 67(1): 61–70. https://doi.org/10.2307/455758

Deeva G, Bogdanova D, Serral E, Snoeck M, and De Weerdt J (2021). A review of automated feedback systems for learners: Classification framework, challenges and opportunities. Computers & Education, 162: 104094. https://doi.org/10.1016/j.compedu.2020.104094

Devlin J, Chang MW, Lee K, and Toutanova K (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In the Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Minneapolis, USA, 1: 4171–4186. https://doi.org/10.18653/v1/N19-1423

Etoori P, Chinnakotla MK, and Mamidi R (2018). Automatic spelling correction for resource-scarce languages using deep learning. In the Proceedings of the ACL 2018, Student Research Workshop, Association for Computational Linguistics, Melbourne, Australia: 146–152. https://doi.org/10.18653/v1/P18-3021

Fahda A and Purwarianti A (2017). A statistical and rule-based spelling and grammar checker for Indonesian text. In the International Conference on Data and Software Engineering (ICoDSE), IEEE, Bandung, Indonesia: 1–6. https://doi.org/10.1109/ICODSE.2017.8285846

Ferrero CL, Renau I, Nazar R, and Torner S (2014). Computer-assisted revision in Spanish academic texts: Peer-assessment. Procedia-Social and Behavioral Sciences, 141: 470–483. https://doi.org/10.1016/j.sbspro.2014.05.083

Flor M, Fried M, and Rozovskaya A (2019). A benchmark corpus of English misspellings and a minimally supervised model for spelling correction. In the Proceedings of the 14th Workshop on Innovative Use of NLP for Building Educational Applications, Association for Computational Linguistics, Florence, Italy: 76–86. https://doi.org/10.18653/v1/W19-4407

Gilquin G and Laporte S (2021). The use of online writing tools by learners of English: Evidence from a process corpus. International Journal of Lexicography, 34(4): 472–492. https://doi.org/10.1093/ijl/ecab012

Golding AR and Schabes Y (1996). Combining trigram-based and feature-based methods for context-sensitive spelling correction. Arxiv Preprint cmp-lg/9605037. https://doi.org/10.48550/arXiv.cmp-lg/9605037

Gurney K (1997). An introduction to neural networks. First Edition, CRC Press, London, UK. https://doi.org/10.1201/9781315273570

Hagen M, Potthast M, Göhsen M, Rathgeber A, and Stein B (2017). A large-scale query spelling correction corpus. In the SIGIR '17: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, Tokyo, Japan: 1261–1264. https://doi.org/10.1145/3077136.3080749

Hasyim J, Roza Y, and Maimunah M (2022). Students' error analysis on linear program based on the KIAT model and students' learning interest. Kalamatika: Jurnal Pendidikan Matematika, 7(1): 43–56. https://doi.org/10.22236/KALAMATIKA.vol7no1.2022pp43-56

Hládek D, Staš J, and Pleva M (2020). Survey of automatic spelling correction. Electronics, 9(10): 1670. https://doi.org/10.3390/electronics9101670

Hourani TMY (2008). An analysis of the common grammatical errors in the English writing made by 3rd secondary male students in the eastern coast of the UAE. M.Sc. Thesis, The British University in Dubai, Dubai, UAE.

Katz SM (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Transactions on Acoustics, Speech, and Signal Processing, 35(3): 400–401. https://doi.org/10.1109/TASSP.1987.1165125

Kneser R and Ney H (1995). Improved backing-off for m-gram language modeling. In the International Conference on Acoustics, Speech, and Signal Processing, IEEE, Detroit, USA: 181–184. https://doi.org/10.1109/ICASSP.1995.479394

Kukich K (1992). Techniques for automatically correcting words in text. ACM Computing Surveys (CSUR), 24(4): 377–439. https://doi.org/10.1145/146370.146380

LeCun Y, Bengio Y, and Hinton G (2015). Deep learning. Nature, 521: 436–444. https://doi.org/10.1038/nature14539 **PMid:26017442**

Lee JH, Kim M, and Kwon HC (2020). Deep learning-based context-sensitive spelling typing error correction. IEEE Access, 8: 152565–152578. https://doi.org/10.1109/ACCESS.2020.3014779

Levenshtein VI (1965). Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics—Doklady, 10(8): 707–710.

Liu Y, Ott M, Goyal N, Du J, Joshi M, Chen D, Levy O, Lewis M, Zettlemoyer L, and Stoyanov V (2019). RoBERTa: A robustly optimized BERT pretraining approach. Arxiv Preprint Arxiv:1907.11692. https://doi.org/10.48550/arXiv.1907.11692

Malema G, Okgetheng B, Motlhanka M, and Rammidi G (2019). Auto correction of Setswana real-word errors. International

Journal on Natural Language Computing, 8(5): 61–66. https://doi.org/10.5121/ijnlc.2019.8405

Mirzababaei B and Faili H (2016). Discriminative reranking for context-sensitive spell-checker. Digital Scholarship in the Humanities, 31(2): 411–427. https://doi.org/10.1093/llc/fqu062

Montavon G, Samek W, and Müller KR (2018). Methods for interpreting and understanding deep neural networks. Digital Signal Processing, 73: 1–15. https://doi.org/10.1016/j.dsp.2017.10.011

Mosavi Miangah T (2014). FarsiSpell: A spell-checking system for Persian using a large monolingual corpus. Literary and Linguistic Computing, 29(1): 56–73. https://doi.org/10.1093/llc/fqt008

Navarro G (2001). A guided tour to approximate string matching. ACM Computing Surveys (CSUR), 33(1): 31–88. https://doi.org/10.1145/375360.375365

Pedler J (2001). Computer spellcheckers and dyslexics—A performance survey. British Journal of Educational Technology, 32: 23–37. https://doi.org/10.1111/1467-8535.00174

Pirinen TA and Lindén K (2014). State-of-the-art in weighted finite-state spell-checking. In: Gelbukh A (Ed.), Computational linguistics and intelligent text processing. CICLing 2014. Lecture Notes in Computer Science, 8404: 519–532. Springer, Berlin, Germany. https://doi.org/10.1007/978-3-642-54903-8_43

Rudin C (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence, 1: 206–215. https://doi.org/10.1038/s42256-019-0048-x **PMid:35603010 PMCid:PMC9122117**

Salhab M and Abu-Khzam FN (2024). AraSpell: A deep learning approach for Arabic spelling correction. Arxiv Preprint Arxiv:2405.06981. https://doi.org/10.48550/arXiv.2405.06981

Shang H and Merrett TH (1996). Tries for approximate string matching. IEEE Transactions on Knowledge and Data Engineering, 8(4): 540–547. https://doi.org/10.1109/69.536247

Sooraj S, Manjusha K, Anand Kumar M, and Soman KP (2018). Deep learning based spell checker for Malayalam language. Journal of Intelligent & Fuzzy Systems, 34(3): 1427–1434. https://doi.org/10.3233/JIFS-169438

Zhang C, Bengio S, Hardt M, Recht B, and Vinyals O (2021). Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, 64(3): 107–115. https://doi.org/10.1145/3446776

Zhao ZQ, Zheng P, Xu S, and Wu X (2019). Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems, 30(11): 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865 **PMid:30703038**