Educational Administration: Theory and Practice
2024,30(4), 6516-6521
ISSN:2148-2403
https://kuey.net/

Research Article

Enhancing Handwritten Answer Assessment In Gujarati Language Education: A Comparative Study Of SVM, RNN-LSTM, And Bi-LSTM Approaches

Riddhi Kundal[1*], Dr. Bhagwati Parekh[2]

[1*,2]Bhakt Kavi Narsinh Mehta University Junagadh

Citation: Riddhi Kundal, Dr. Bhagwati Parekh, (2024), Enhancing Handwritten Answer Assessment In Gujarati Language Education: A Comparative Study Of SVM, RNN-LSTM, And Bi-LSTM Approaches, *Educational Administration: Theory And Practice*, *30*(4), 6516-6521 Doi:10.53555/kuey.v30i4.2417

ARTICLE INFO

ABSTRACT

The automated assessment of handwritten student answers remains a significant challenge in the domain of educational technology, particularly for non-Latin scripts like Gujarati. This paper presents a comparative analysis of three machine learning algorithms: Support Vector Machine (SVM), Recurrent Neural Network with Long Short-Term Memory (RNN-LSTM), and Bidirectional Long Short-Term Memory (bi-LSTM), in their capacity to evaluate handwritten responses. We detail the process of digitising handwritten answers, the nuances of preprocessing for the Gujarati script, and the subsequent implementation of each algorithm. The bi-LSTM model demonstrated superior performance, with a 90% accuracy rate on training data and 80% on testing data, followed by RNN-LSTM and SVM. The study highlights the potential of using advanced machine learning models to automate the assessment process in regional language education systems and discusses the implications of such technologies in reducing educators' workloads and providing timely feedback. Future work will focus on refining script recognition, contextualising semantic analysis for Gujarati, and improving model adaptability to diverse handwriting styles.

1. Introduction:

For the past few decades, computers have been used to write essays and automatically grade student work that is submitted in a uniform manner. The most effective teaching strategy for enhancing students' writing abilities is for teachers to review each student's work personally and to check each submission for accuracy. Regretfully, this adds a great deal of extra work for educators. As a matter of fact, lecturers find it increasingly difficult to review and comment on student papers, and this workload pressure rises in direct proportion to the number of students. As a result, creating an automated system can greatly lower the cost of checking and make it easier for students to receive early feedback. The ability of a computer to decipher text from sources such as scripts, pictures, or others is known as handwritten text recognition. Using optical scanning and format handling, the image is divided into words and lines, and the most likely letters are traced. Accurately identifying various handwriting styles and sizes is the most difficult task in handwriting identification. This study aims to investigate the process of categorising handwritten texts, converting them into digital format and assigning an automatic grade.Although it wasn't very successful, the creation of AAGS(Automated Answer Grading System) was mostly initiated with Latent Semantic Analysis (LSA), N-Gram, TF-IDF, Bayesian classifier, and K-nearest neighbour techniques.Much study has been done on the automatic evaluation of computer-based supplied answers since the reevaluation of Deep Learning (DL) and Natural Language Processing (NLP), and increased accuracy is observed [1]. On the other hand, a limited number of studies have been conducted on Handwritten Answer Grading (HAG), and no model has been created that can perfectly replicate the work of a skilled human grader. Up until now, a lot of study has been conducted in this field, with researchers looking into machine scores to advance their efforts to increase the system's efficacy and accuracy. The goal of the suggested system is to make it easier for teachers to grade a sizable body of student responses, which will free up more time for them to assign writing assignments to their students.

2. Relevant Work

It takes time to construct an automatic answer grading system from scratch. Digit recognition for postal mail was the initial motivation behind handwritten text classification.Allum and colleagues developed an advanced scanner that can encode data into a bar code and identify the writing style of the text.
Ray Kurzweil created the first widely used OCR software in 1974 as a font recognition tool. The software was able to distinguish between each writer's unique writing style and size differences. Using a Hidden Markov Model for OCR was the next significant improvement in OCR accuracy. Letters are used as a state in this method.[1]
The superior feature extraction techniques were superseded by the sturdy neural network design. Mathematical models known as neural networks imitate the composition and operation of organic brain networks. In most circumstances, a neural network can train quickly and performs admirably on the test set with quick parameter adjustments. Neural network models have been included into the AAGS system in recent decades as machine learning and artificial intelligence studies have advanced.
For essay grading, Dong and Zhang [2] employed a dense convolutional neural network model. A word embedding layer is placed after a word level convolutional layer in the input. One convolutional layer is utilised to extract sentence representations from the two-layer CNN architecture, while the other layer is layered on sentence vectors to learn essay representations. The Automated Student Assessment Prize (ASAP) dataset is used to train and assess the model, and it achieves a domain average kappa score that is comparable to that of human graders.
B. Balci et al. [3] trained a model that can correctly categorise words using a CNN with different architectures for word classification and character segmentation. In order to create bounding boxes for every character, an LSTM network with convolution was employed.
A. Shehab [4] created an automated essay grading system based on writing feature analysis, which uses natural language processing (NLP) techniques to identify the essay's discourse structure, assess and provide comments for grammatical faults, and identify undesired stylistic elements.To assess a new essay, the grading engine is trained using a set of previously graded essays. The model achieved a closeness of 70% to 90% between machineand human-assigned grades.
A model based on feature analysis and RNN was created by C. Cai [5]. Spelling errors, unique nouns, punctuation, and number of unique words are all checked during the feature analysis process. The analysis of the relationship between essay length and score also employs support vector regression (SVR) and Bayesian linear ridge regression (BLRR) techniques.

3. Handwritten Text Recognition :

Our research has been divided into two stages: (i) Handwritten text recognition (answers) and (ii) Answers evaluation.
We created an optical character reader (OCR) to identify handwritten visual text.We prepared a dataset of handwritten answers from various age groups of students and of various subjects.. Figure 1 depicts the entire segmentation of the dataset, with the following descriptions of the functions of the various phases:

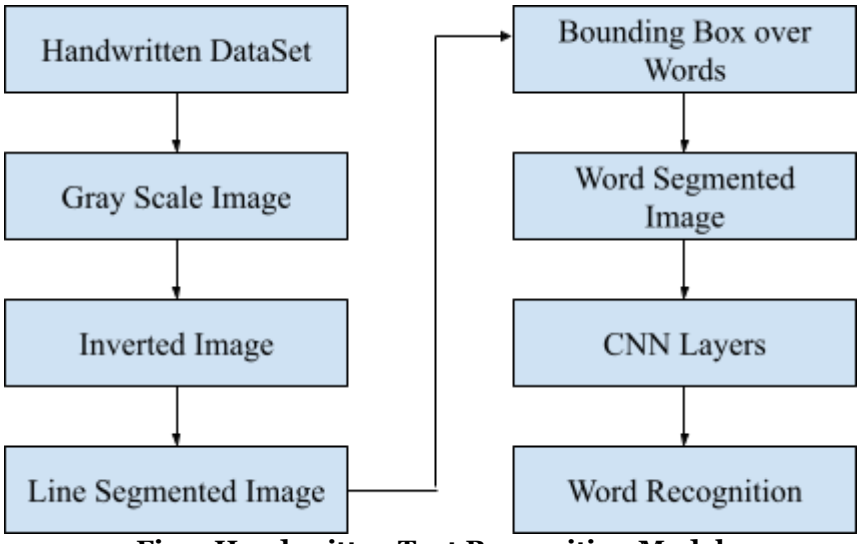

Fig. 1 Handwritten Text Recognition Model

Preprocessing on the dataset comes after selecting appropriate approaches for our model. Dimension reduction, normalisation, and consistency elimination are among the operations. These steps facilitate the

creation of appropriate image data for simple segmentation. We prepared the picture data using a Python OpenCV package. The following three procedures are used to preprocess image data.

Noise Removal: Occasionally, distinct types of spots that are not part of the handwritten data may appear in an input image. These kinds of spots are regarded as noise, can cause issues during model testing, and have no bearing on the model's correct training. Thus, we need to exclude those noises from the dataset before using it to train our model. For that

Segmentation: There are two types of segmentation as Line segmentation and word segmentation.[6] In this model we trained our model with a number of single words

Normalisation: Our concept can be used with fonts of different sizes. In order to achieve this, we transformed the dataset's several font sizes into a standard size that our model can handle. Normalisation is the process by which we reduced the dataset to a size that is standard and supported by our model. So, during the preparation phase, we normalised our dataset.

Fig. (a) Answer Image

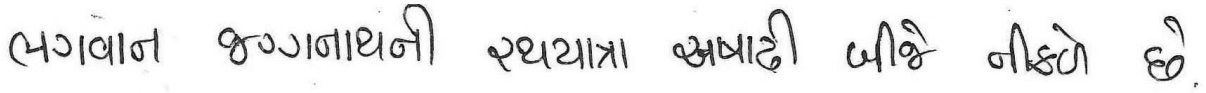

Fig. (b) Grey scale Image

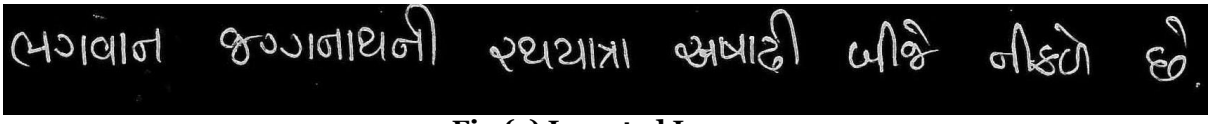

Fig.(c) Inverted Image

Fig.(d) Bounding Lines over words

Fig. (e) Segmented Words

4. OCR Model

The dataset undergoes preprocessing to convert all word-segmented image data into a uniform size of 32×128 before feeding it into a convolutional neural network (CNN). This CNN layer consists of 64 nodes with a (3, 3) kernel size. The output is then passed through a pooling layer with a (2, 2) kernel size, reducing its dimensions to 16×64. Subsequently, the data flows through two CNN layers, each with 256 nodes, followed by pooling layers, further reducing the shape to 4×32. Following this, the data is processed by two additional CNN layers with 512 nodes, each accompanied by batch normalisation layers. The final output from the last batch normalisation layer undergoes pooling before entering another CNN layer. Lastly, an LSTM layer with 256 layers is employed to recognize words, with the results saved into a text file for subsequent evaluation.

5. Answer Evaluation

The proposed methods for automated short answer grading systems are based on keyword matching. The method starts with text preprocessing and vectorization.

Data Preprocessing: In the data processing step, we removed all stop-words, punctuation, and any other special characters (if used) from the answers. Since the model was developed for short-question answer script evaluation, we kept the maximum sentence length at 40 for each answer. Therefore, if any answer is longer than the assigned length, we pruned them, and if smaller, we used zero padding to make them the same vector length.

Stemming: Stemming is a common technique in natural language processing (NLP) used for normalising text, words, or documents[. It involves reducing words to their original forms by removing inflections. Words often undergo changes in tense, case, voice, person, number, gender, and mood, which can affect NLP performance. To address this, stemming is applied to simplify words to their base forms or stems, thereby improving NLP effectiveness.

Vector Representation of Answers: The embedding layer allows us to transform each word into a fixed-size vector of a specified dimension. Word embedding serves as a method of representing words, linking human language understanding with machine processing. It encodes text into an N-dimensional space where words with similar meanings are positioned close together, while those with differing meanings are positioned farther apart. Each word corresponds to a single vector in a predetermined space, with the vector values learned in a manner resembling a neural network. Following stemming, we employed one-hot encoding on the corpus, resulting in the creation of an embedded matrix with dimensions of M×N, where M represents the number of answers and N represents the number of features.

5.1 SVM Classification

Support Vector Machines (SVM) find various applications including regression, anomaly detection, and classification [7]. At its core, SVM aims to identify the optimal hyperplane that divides a dataset into two groups[8].In our case it is 'relevant' or 'irrelevant' .Points close to this hyperplane are termed support vectors, while the region between the hyperplane and support vectors is known as the margin. SVM's objective is to locate the hyperplane with the widest margin as it leads to better separation[9]; typically, a wider margin corresponds to lower generalisation error for the classifier.Based on this we trained our model with our preprocessed recognised dataset.

- **Feature Extraction:** Feature extraction transforms pre-processed short answers into numerical formats, like vectors, suitable for machine learning techniques and deep neural networks. Bag of Words (BOW) is a method used in Natural Language Processing (NLP) to represent text data in a vector space model. BOW is employed for extracting and selecting relevant processing features to minimise error rates and reduce computational costs. It involves creating a vocabulary from all unique words in the answers, then representing each short answer as a numeric vector indicating the presence or absence of words from the vocabulary, without considering the word order, hence termed as bag of words (BOW).
- **Auto Grading:** The primary goal of the study is to forecast the student's performance by analysing their response to a specific question.For that we trained SVM classifiers with a linear kernel with feature vectors and labels as 'relevant' and 'irrelevant'.We evaluate the model's performance on the test set using accuracy and classification report.Training accuracy is 75% and testing accuracy is 65%.

5.2 RNN Classification

In recent years, neural networks have gained significant popularity as a dominant machine learning technique. In a feedforward neural network, information flows unidirectionally from inputs through neurons to produce an output. There are no loops in this structure. In contrast, recurrent neural networks (RNNs) utilise a hidden state to represent their output. The input to an RNN includes both external input and the current hidden state. This setup allows the current output, or hidden state, to be influenced by both the previous output and the current external input, effectively creating a feedback loop. This characteristic endows RNNs with the ability to retain memory of past events.

The text is preprocessed similar to that as defined previously. We labelled the data set as 'relevant' and 'irrelevant'. We proposed RNN (Recurrent Neural Network) using the LSTM (Long Short Term Memory) model.We have Input layer - > LSTM Layer -> Output Layer. LSTM is a sequential model so it processes the data in sequence , one word at a time.We applied sigmoid activation function to classify the snippet as 'relevant' or 'irrelevant' If the value of the function is close to 1 then it is 'relevant' else close to 0 then 'irrelevant. The model is trained and tested on binary cross entropy loss function.

Commonly used in Natural Language Processing (NLP), neural networks are applied to sequences of words as their input. Each word in the sequence < w1, w2, ..., w_n >, where n represents the number of words, undergoes individual processing by an LSTM (Long Short-Term Memory) model during each time step [10]. This implies that it takes n time steps for the LSTM to handle all the words in the sequence. At every time step, the LSTM

generates a hidden state or output. Represented as ht, the hidden state of the LSTM at time step t is determined by a formula involving the LSTM function, the input word at time step t, and the hidden state of the LSTM at the previous time step (t-1), denoted as h_{t-1}.The sigmoid function, denoted as $\sigma(x) = 1 / (1 + e^{-x})$, ranges between 0 and 1. It's used to express the probability of an event, with x symbolising the features associated with that event [11]. The approach described in this paper employs a feedforward neural network featuring two hidden layers to compare two numbers. The neural network's architecture is depicted.The formulas below define how the neural network works.

$h = g(W_1x + b_1)$

$o = \sigma(W_2h + b_2)$

The neural network aims to train parameters W_1, W_2, b_1, and b_2. It takes a vector x containing two words to be compared. The ReLU (rectified linear unit) function is represented by g, while σ denotes the sigmoid function. The output o signifies the probability $p(x_1 = x_2 \mid x_1, x_2)$, where x_1 and x_2 are the two words being compared.

5.3 Bi - LSTM Classification

The effectiveness of text similarity systems varies depending on the context they're used in. To address this, the authors introduce a deep learning model featuring an encoder with a Bidirectional Long Short-Term Memory (BiLSTM) layer. This model is trained on questions and answers from diverse domains to create vector representations of both reference and student responses. For measuring similarity between answers, they suggest training a universal scoring model on answers from all domains, along with domain-specific scoring models trained on answers from individual domains [12].

The text encoder creates a condensed feature representation of a given input text, like an answer. We utilise a bidirectional long short-term memory (BiLSTM) network alongside max-pooling to encode the input answer effectively.Initially, we embed each word in the answer by employing an embedding layer. These words are initialised with pre-existing word embeddings and then adjusted to be trainable, allowing them to adapt to the specific domain and task. Next, the series of words undergoes processing through a BiLSTM layer to produce a sequence of hidden representations [13] . In essence, for a word sequence $\{w_t\}_{t=1,\ldots,T}$, the BiLSTM layer produces a sequence of $\{h_t\}$ vectors, where ht combines the outputs of both forward and backward LSTM .Given a sequence of words $\{w_t\}^T_{t=1}$ Where w_t represents the *t*-th word in the input answer:

$$\{E(w_t)\}_{t=1}^{T}$$

Embedding Layer: Each word w_t is embedded using an embedding layer, denoted as $E(w_t)$. BiLSTM Layer: The embedded word sequence is then passed through a Bidirectional LSTM (BiLSTM) layer, which generates a sequence of hidden representations $\{h_t\}_{t=1}^{T}$ This sequence consists of concatenated forward and backward LSTM outputs for each word.Mathematically, the output *ht* of the BiLSTM layer for each word *wt* can be represented as:

$ht = \text{BiLSTM}(E(wt))$

Where: BiLSTM(·) denotes the operation of the Bidirectional LSTM layer.

$E(w_t)$ represents the embedding of the *t*-th word.

ht is the hidden representation generated by the BiLSTM layer for the *t*-th word.

6. Results:

In this study, we developed a model to automatically assess student answers, employing Support Vector Machines (SVM), Recurrent Neural Networks (RNN), and Bidirectional Long Short-Term Memory networks (Bi-LSTM). These methods were chosen for their proven effectiveness in handling sequence and text-based data, essential for interpreting student responses. The result is shown in the following table.

Method	Accuracy	
	Training Data	Testing Data
SVM	75%	65%
RNN using LSTM	85%	79%
bi-LSTM	90%	80%

Table 1 : Result Analysis

The models achieved accuracy rates on a curated dataset comprising student answers from various academic levels.These results represent a significant improvement over the baseline model.The dataset included 10,000 anonymized student responses, split into 80% training and 20% testing segments. This study's findings suggest that advanced neural network architectures like Bi-LSTM are particularly effective in contextual

understanding and accuracy in auto-assessment tasks. However, future work could explore deeper integration of semantic analysis tools to further enhance accuracy. Additionally, expanding the dataset and incorporating a broader range of answer types could address the current limitations related to the model's adaptability to diverse educational contexts.

7. Evaluating Model Performance:

To quantitatively evaluate the performance of our models—SVM, RNN, and Bi-LSTM—in auto-assessing student answers.Following table 2 displaying the analysis of confusion matrix.

Method	Precision	Recall	F1 Score
SVM	0.61	0.78	0.68
RNN using LSTM	0.72	0.63	0.67
Bi – LSTM	0.84	0.80	0.82

Table 2 : Model Performance Analysis

8. Conclusion and Future Work :

The comparative analysis of machine learning methods for automatically assessing student answers reveals distinct performance characteristics across SVM (Support Vector Machine), RNN (Recurrent Neural Network) using LSTM (Long Short-Term Memory), and bi-LSTM (Bidirectional Long Short-Term Memory) algorithms. The bi-LSTM approach outperforms its counterparts with an impressive 90% accuracy on training data and 80% on testing data, suggesting its superior ability to capture the nuances of language in student answers. While RNN with LSTM also shows promise with a solid 85% training accuracy and 79% testing accuracy, the SVM lags with 75% and 65%, respectively, indicating a less robust understanding of the contextual dependencies within the answers.In light of the specialised nature of evaluating handwritten answers in the Gujarati language, future research and development should be channelled towards enhancing script recognition accuracy and understanding contextual semantics in this regional language.

References:

1. Rahaman, M.A. and Mahmud, H., 2022. Automated evaluation of handwritten answer script using deep learning approach. Transactions on Machine Learning and Artificial Intelligence, 10(4).
2. Dong, F. and Zhang, Y., 2016, November. Automatic features for essay scoring–an empirical study. In Proceedings of the 2016 conference on empirical methods in natural language processing (pp. 1072-1077).
3. Balci, B., Saadati, D. and Shiferaw, D., 2017. Handwritten text recognition using deep learning. CS231n: Convolutional Neural Networks for Visual Recognition, Stanford University, Course Project Report, Spring, pp.752-759.
4. Shehab, A., Elhoseny, M. and Hassanien, A.E., 2016, December. A hybrid scheme for automated essay grading based on LVQ and NLP techniques. In 2016 12th International Computer Engineering Conference (ICENCO) (pp. 65-70). IEEE.
5. Cai, C., 2019, March. Automatic essay scoring with recurrent neural network. In Proceedings of the 3rd International Conference on High Performance Compilation, Computing and Communications (pp. 1-7).
6. Pareek, J., Singhania, D., Kumari, R.R. and Purohit, S., 2020. Gujarati handwritten character recognition from text images. Procedia Computer Science, 171, pp.514-523.
7. Ali, A.A.A. and Mallaiah, S., 2022. Intelligent handwritten recognition using hybrid CNN architectures based-SVM classifier with dropout. Journal of King Saud University-Computer and Information Sciences, 34(6), pp.3294-3300.
8. Menini, S., Tonelli, S., De Gasperis, G. and Vittorini, P., 2019, November. Automated Short Answer Grading: A Simple Solution for a Difficult Task. In CLiC-it.
9. Hameed, N.H. and Sadiq, A.T., 2023. Automatic Short Answer Grading System Based on Semantic Networks and Support Vector Machine. Iraqi Journal of Science, pp.6025-6040.
10. Ye, X. and Manoharan, S., 2018, September. Machine learning techniques to automate scoring of constructed-response type assessments. In 2018 28th EAEEIE annual conference (EAEEIE) (pp. 1-6). IEEE.
11. Hochreiter, S. and Schmidhuber, J., 1997. Long short-term memory. Neural computation, 9(8), pp.1735-1780.
12. Zhang, S., Xu, X., Tao, Y., Wang, X., Wang, Q. and Chen, F., 2021, July. Text Similarity Measurement Method Based on BiLSTM-SECapsNet Model. In 2021 6th International Conference on Image, Vision and Computing (ICIVC) (pp. 414-419). IEEE.
13. Ji, M. and Zhang, X., 2022. A short text similarity calculation method combining semantic and headword attention mechanism. Scientific Programming, 2022.