International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys.14, No.1 (Oct-23)

http://dx.doi.org/10.12785/ijcds/140191

Hybrid Model For Efficient Assamese Text Classification Using CNN-LSTM

Chayanika Talukdar[1] and Shikhar Kumar Sarma[2]

[1]*Information Technology,Gauhati University,Guwahati,India*

Received 4 Dec. 2022, Revised 11 Aug. 2023, Accepted 21 Sep. 2023, Published 1 Oct. 2023

Abstract: In modern times, there has been an exorbitant rise in unstructured digital text, owing to the ever-increasing use of internet. Therefore, to be able to extract knowledge out of it, a perceived need was felt to organize the enormous amount of digital text into different categories. This is why Text Classification is considered a critical task in NLP (Natural Language Processing). This research suggests a hybrid model(C-LSTM-ATC) that combines the benefits of two deep learning models, namely, the Convolutional Neural Network (CNN) and Long and Short Term Memory (LSTM), to categorize Assamese text, a topic that largely remains unexplored till now. Another hybrid model was also tried by combining LSTM with the Support Vector Machine (LSTM-SVM). The C-LSTM-ATC model performs splendidly with an accuracy of 97.2% while the LSTM-SVM model outputs an accuracy of 92.2% when tested on the dataset prepared. The model was trained using as many as 768 Assamese text documents and the test results showed that the proposed C-LSTM-ATC model produces more accurate classification and higher F1 scores, than the LSTM-SVM and also the CNN and LSTM models when used separately.

Keywords: Assamese, Text Classification, Long and Short Term Memory, Convolutional Neural Network , Support Vector Machine, Accuracy

1. INTRODUCTION

Thanks to the ever-increasing advancement of Internet and the influence of social networking, the volume of textual data on the internet is expanding quickly. Textual documents are being digitized instead of writing it in papers. Newspapers, books, journals and articles, are also digitized and accessible through the internet, thereby enabling users to read them online. As Internet platforms are highly dynamic in nature, vital information is mostly hidden in such textual data. However, currently there is high demand for effective organization and management of these texts. Text classification, an integral part in NLP (Natural Language Processing), is the process of categorizing texts into different groups based on their contents [1]. It is well-known for being an efficient method of categorizing and managing important information and is used extensively in sentiment analysis [2], information sorting [3], spam filtering, personalized news recommendation, user intention analysis, content moderation and so on.

With the rapid growth of Assamese textual content on digital platforms, developing Assamese text processing systems is becoming increasingly important. However, very little has been done as far as NLP-based research on Assamese text classification is concerned. Assamese is a morphologically rich inflectual language which belongs to the Indo-Aryan group. It is spoken by 23 million people approximately and is the official language of Assam. As English is the most used language across the world, therefore most of the research work in NLP is expectedly conducted in English. Research in India's other regional languages still has a lot of room to grow. The morphology of these languages is rich but they are low resourced. An important point to note here is that morphologically rich languages have greater complexity in sentence structure and grammar. This is the reason behind limited research in these languages.

Text classification may be branched into two phases. The first one uses text feature representation, while the second includes the use of a text classifier for classification. Usually, the Bag of Words (BoW) model is used for text representation, where, n-grams, bigrams and unigrams, are used as features representation. Moreover, several feature selection techniques, such as TF-IDF,LDA, pLSA, are also applied to extract more unique features. Lastly, models such as support vector machines (SVM), Naive Bayesian Classifier (NBC) and the K-Nearest Neighbor (KNN) are

E-mail address: ctalukdar@gmail.com,sks001@gmail.com

normally used for traditional text classification methods for classification purpose. Besides using a shallow structure, these methods offer a huge number of artificial features. However, their performance is primarily determined by the nature of hand-crafted features, which are handicapped by high latitude and data sparsity. To overcome this issue and enhance the accuracy rate, recently, text classification tasks have made substantial use of deep learning methods. Compared to the traditional methods, deep learning (DL) method, is rated as a highly effective method for the purpose of feature extraction. Moreover, the text classification task using DL has been substantially benefitted from the invention of word vectors. It was a significant milestone when Bengio et al. presented continuous word representation in 2006 [4]. In order to give an additional boost to the advancement of DL, Mikolov in 2013, put forward the word2vec model for training the word vectors [5].

Furthermore, a rising number of researchers of late have applied the CNNs and RNNs (Recurrent Neural Network) to the field of text classification. Between these two models, RNN has achieved better results while administering serialization tasks. As each word in a complete sentence has a nuanced semantic link, RNN treats a sentence as a collection of tokens, processing the tags in the left-to-right sequence. Equipped with recurrent network structure, RNN is capable of holding the overall sequence's internal state. Thus, it can blend information related to certain contexts in a better manner. In order to circumvent the gradient exploding/vanishing problem faced by standard RNN, LSTM [6] and GRU (Gated Recurrent Unit) [7] were designed. LSTM has lived up to expectations, and exhibited an exceptional ability in processing natural languages. LSTM is a model that processes sequential information , capable of storing language history information and extracting text context dependencies. LSTM can take word order into account, however the difficulty is that the later text has a higher impact on the final text representation than the earlier one, which is more noticeable in some text classification problems. On the other hand, CNN also has showcased amazing results in the fields of speech processing, computer vision [8] and NLP [9].This is because CNNs have the capability to capture the local association of spatial or temporal structures. When used in NLP tasks, CNN can capture n-gram features from different positions in a sentence using convolutional filters thus, making it easy for identifying the most commonly used words in a sentence and learning both long and short-range relations via the use of pooling operations.

As observed, LSTM gives excellent results while handling serialization tasks. But it performs poorly while extracting features. On the other hand, CNN performs better in extracting or capturing features but, performs poorly when it comes to learning sequential correlations. Hence, in this paper a new architecture named C-LSTM-ATC (Assamese Text Classification) has been put forward by fusing CNN with LSTM. By furnishing the output of a two-layered CNN into an LSTM, a straightforward end-to-end, consolidated architecture has been created, which benefits from both the models simultaneously. The first layer of CNN is arranged on top of the trained word vectors of the text so as to learn more advanced n-gram representations. The extracted features of the two CNNs are then arranged as successive window features which acts as input to LSTM. This is done to ensure that it learns long-term sentence dependencies from higher-level sequential representations. In order to effectively separate the elements causing differences within sentences, each sentence is first transformed into a series of window (n-gram) features before building an LSTM from the input statements. This paper uses a sequence-based input rather than depending on the syntactic derivation trees, thus making the model independent of further knowledge of language structure. This model was tested by using it to classify Assamese text and the resulting effects are assessed.

The following points summarize the primary contributions of this paper:

- Preparation of an Assamese dataset which can be classified into four categories.
- A hybrid model C-LSTM-ATC is proposed for categorizing Assamese texts into any one of the preset classes. This model is built using two other popular deep learning models, the CNN and the LSTM.
- Assessing the effectiveness of the C-LSTM-ATC model on a prepared dataset.
- Assessing the performance of another hybrid model, LSTM-SVM ,built using LSTM and a traditional model called SVM.
- To compare these two models(C-LSTM-ATC and LSTM-SVM) with different deep learning algorithms.

The remaining part of this paper will cover the following sections. Section 2 reviews some of the notable contribution made in the area of text classification. In Section 3, an overview of the proposed models is discussed. In section 4, the model performance is evaluated and section 5 finally concludes this paper.

2. RELATED STUDY

Collobert et al. were the first to apply CNN to NLP tasks[10]. They used max-pooling to extract global features by using convolutional filters on a series of windows in a particular order. Kim[11] was the first to introduce CNN to text classification. He presented a CNN architecture with numerous filters (with variable window widths) and two word vector channels, as a subtle variant. With a view to identify the interrelation of words, Kalchbrenner [9] proposed a model based on CNN that used different K values for max pooling in different network layers. As a result, more feature sequences were used to extract more locally sensitive information. Instead of inputting word vectors to the input layer, Zhang et al. suggested a semi-supervised CNN model for text classification that used region embedding[12]. Another text embedding method, called Two View embedding, was used for classifying sentiments in sentence. To generate intermittent n-gram features, Lei et al. proposed a new operator for feature mapping[13]. With a view to boost performance, Wang et al. used a huge taxonomy knowledge base [14]. An input layer, a hidden layer, and finally an output layer made up the three-layer fastText model that Joulin et al. suggested. As inputs, their model added n-gram features (character level) of the word-to-word vectors of the entire document, to obtain the document vector [15]. They used the hierarchical softmax in the output with a view to reduce the model's training time. For text classification, three separate models were put forth by Liu et al. that communicated with RNN (Recurrent Neural Networks)[16]. The model aimed to enhance categorization through parallel task learning, by utilizing the correlation between related tasks. When applied on four benchmark text classification datasets, the model recorded better accuracy. Recurrent Convolutional Neural Networks (RCNNs) have been proposed by Lai et al. as a model for document categorization to address the biasness in RNNs where later words have a stronger influence than earlier ones [17]. Their model successfully utilized the advantages of recurrent structure and CNN, to both capture the contextual information as well as learn the feature representation of documents. Yao et al. proposed a novel text classification method called the Text Graph Convolutional Networks[18]. They constructed a graph from the entire corpus with words and documents being its nodes. They then modeled the graph using the Graph Convolutional network. The coincidence of words is represented by drawing an edge between the word nodes. The edge linking a word node to a document node is drawn with the help of word frequency and the document frequency of the word. Their experimental results, which were performed on four different datasets, show that Text GCN outperforms all the baseline models. Li et al. put forward a hybrid model by combining two Bidirectional LSTM layers and a CNN layer for the purpose of classifying of Chinese news [19][19].

3. METHODS OF RESEARCH

A. *C-LSTM-ATC model*

The model's specification is discussed in great detail in this section. Word embeddings of the Assamese text are used as inputs in the model, where CNN learns to extract high-level features from the inputs. The LSTM receives the outcomes of the CNN, which is subsequently followed by a classifier layer. The proposed classification model C-LSTM-ATC's primary objective is to classify a given Assamese document into one of the four categories arts, children, history and sports. The model begins with cleaning the dataset followed by preprocessing of the text and finally applying the classification model on the dataset. Figure 1 presents the structure of the proposed model-

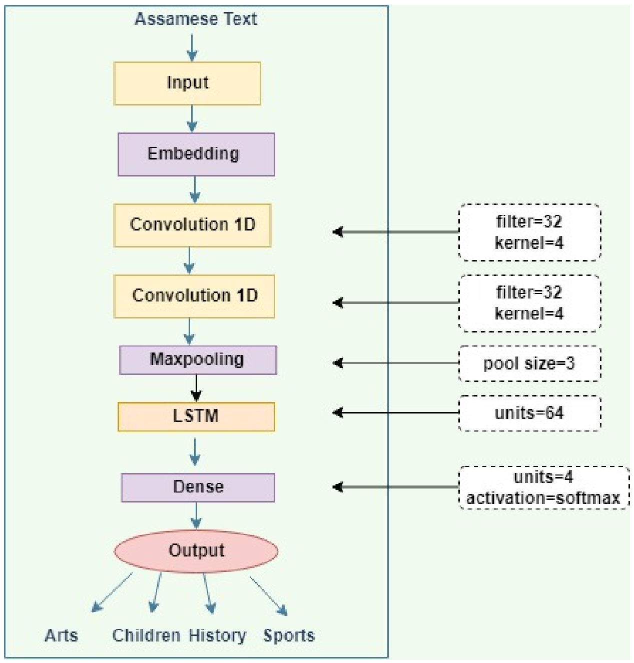

Figure 1. Model Architecture of C-LSTM-ATC

1) *Input Layer*

This layer accepts the raw Assamese text as inputs. But this text must be filtered first as they contain many content that may adversely affect the classification task. Hence the model's first stage involves cleansing the data set. The documents in the dataset contains many characters, namely the punctuation symbols, English letters and numerals as well as some extra white-spaces, which might adversely affect the performance of the model. Hence, it is required to remove these extra characters from the text as part of data cleaning. Secondly, the documents might contain stopwords, which are the most commonly appearing words in a sentence. This might misguide the classifier during classification. Hence the text needs to be free from these words. Some of the stopwords in Assamese are- সৈতে, কৰা, কৰে ,আৰু etc. Table I demonstrates the text before and after the cleaning.

2) *Embedding Layer*

This forms an important step in the classification process. The whole idea is to transform each text in the data set into numerical vectors to make it convenient for deep learning models to act on the data supplied. Machine learning or more precisely deep learning algorithms can interpret only numerical data. Hence this step is necessary. For this proposed work, Keras embedding layer was used for transforming the cleaned Assamese text into 100-dimensional real valued vectors. For every input token, the embedding layer learns its distributed representation. The way the tokens are represented reflects the hidden connections between words that are probably going to appear in the same context. Since each document is of varying length, therefore the maximum

TABLE I. Assamese text after cleaning

Before cleaning	After cleaning	Category
নিৰাময়ত নগ্নতাৰ পৰশ Issue Dated : মে' 15, 2011	নিৰাময়ত নগ্নতাৰ পৰশ মে '	Arts
নৃত্যকলা :ক্ষেত্ৰত অসমৰ মাৰ্গ নৃত্য আৰু লোকনৃত্য	নৃত্যকলা ক্ষেত্ৰত অসমৰ মাৰ্গ নৃত্য লোকনৃত্য	History
শচীনে এদিনীয়া ক্ৰিকেটত ১০,০০০ ৰান অতিক্ৰম	শচীনে এদিনীয়া ক্ৰিকেটত ১০ ০০০ ৰান অতিক্ৰম	Children
শুকুৰবাৰৰ পৰা টকিঅ'ত আৰম্ভ হ'ব "Tokyo Olympics"	শুকুৰবাৰৰ টকিঅ	Sport

length for each document was considered to be 1000.This helped in truncating longer documents and padding the shorter documents to the maximum size. Representation of the tokens reflected the latent relationships between words that are likely to appear in the same context.

3) *1D Convolutional Layers*

The primary goal of this layer is to retrieve the pertinent features from the given text in the documents. To achieve this, convolution operation was applied on those word vectors that were the outputs from the embedding layer. Two back to back Convolutional layers were used in this model .Let's assume that $Z_i \in D_d$ represent the d -dimensional vector that constitutes the i^{th} word of the Assamese document. Let $Z \in D^{n\times d}$ signifies the input text. Also, n denotes the text length. For every j^{th} position within the given text, let's assume a vector for the window denoted as w_j, along with k word vectors in a row, represented as:

$$w_j = [a_j, a_{j+1}, ...a_{j+k-1}] \quad (1)$$

A filter is required for every convolution process represented as $p \in R^{k\times d}$. This, when used on the window w, creates a fresh feature map $m \in R^{L-k+1}$. For window vector w_j, each feature element c_j is estimated in the following manner.

$$m_j = f(w_j \odot p + b) \quad (2)$$

where the operator $\odot$ denotes element-wise product, f being a nonlinear function, and $b \in D$ is the bias term. As an activation function, ReLU (Rectified Linear Unit) is employed in this work. It can be defined as:

$$f(a) = max(0, a) \quad (3)$$

For positive value ReLU returns a, otherwise it returns a zero value. The convolution layer was configured using 32 filters with a window size of 4 i.e the convolution looks a 4 consecutive tokens at a time for 32 steps. In this experiment the filter has been moved forward by 1 token at each of 32 steps. As such 32 features are extracted in one convolution.

4) *MaxPooling Layer*

The Convolution process produces feature maps that are identified by a vector representation of high-order. In order to decrease this, a MaxPooling layer was introduced, immediately following the second convolutional layer. This helped extract the salient information only, by eliminating non-prominent activation information. This assists in avoiding the overfitting problem caused because of the presence of noisy text. This layer consists of a pool size of 3X3 and a stride of 1.These features are directly passed onto the LSTM layer below.

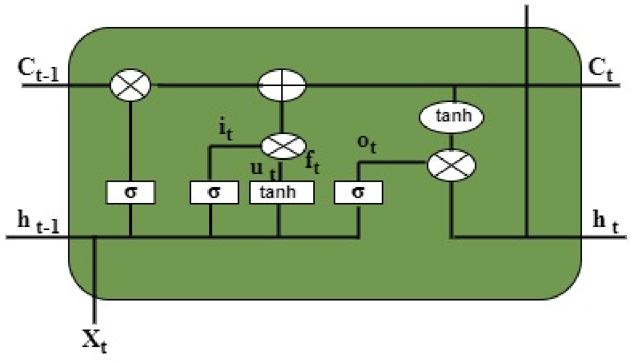

Figure 2. LSTM Model

5) *LSTM Layer*

CNN is extremely effective in capturing the relevant features from textual data. However, it fails to link the current information with the previous one. LSTM, being a form of RNN, has the capability to capture the long-term relationships that exist in sentences of unspecified length.Figure 2 depicts the basic architecture of an LSTM. The memory cell within an LSTM helps to retain data for longer period without loss/decay (losing anything). In order to act upon the current vector input, LSTM recursively executes the present memory cell with X_t as the present input and h_{t-1} the prior hidden state. Here t denotes the present time and t-1 denotes the preceding time. Besides these, LSTM include an input gate denoted by i_t , output gate o_t and a forgot gate f_t. c_t represents the memory cell at time t. These gates altogether decide the manner in which the present memory cell and hidden states are updated [20].

$$i_t = \sigma(W_i \cdot [h_{t-1} + b_i]) \quad (4)$$

$$q_t = \tanh(W_i \cdot [h_{t-1}, x_t] + b_q \quad (5)$$

q_t denotes the cell at time t. Equation (4) and (5) are used for computing the values of input gate and present memory cell state respectively. The forget gate value is estimated for time t using equation (6).

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (6)$$

Similarly, the new state of memory cell is determined, using the equation (7)

$$c_t = f_t \odot c_{t-1} + i_t \odot q_t \tag{7}$$

With the help of Equations (8) and (9) the values associated with the output gate is found out using

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{8}$$

$$h_t = o_t \odot \tanh c_t \tag{9}$$

In the above equation x_t indicates LSTM unit's input vector. Whereas, σ denotes the sigmoid function, the $\odot$ operator represents the element-wise product, *tanh* is the hyperbolic tangent function, and W and b denote the weight matrices and bias vector respectively .These are the parameters that needs to be learnt during training period. In this work, a single layer of LSTM with 64 units has been used. For the purpose of regulating the parameters a dropout rate of 0.2 has been used. This helps the model from preventing the overfitting problem.

6) *Dense Layer*

This layer, also known as fully connected layer, is fitted as the final layer of this model. Its main activity is to classify the text documents as per the LSTM layer outputs. Since multivalued classification model is considered for this work, a dense layer comprising of 4 neurons and the softmax activation function is also used. This helps predict a given Assamese text into one of the four categories i.e arts, children, history and sports. The softmax function performs as normalization technique for obtaining the group to which the document belongs. We specify the formula as:

$$(ys) = softmax(WT + b) \tag{10}$$

where (ys) is the probability for each class with softmax function. Also, s specifies the total number of considered target classifications.

B. *LSTM-SVM model*

The LSTM-SVM is the second hybrid model used for this research. This classification model is build up using an LSTM along with the traditional SVM at the output layer. SVM is considered to be an extremely efficient model for text classification as it learns the relations that exist within sentences. Therefore this hybrid model was built. Figure 3 depicts the abstract architecture of the model. The model accepts Assamese text as input which, after cleaning and preprocessing by the Embedding layer, is passed to the Spatialdropout. The dimension of the embedded vectors was set to 100. The purpose of placing this layer is that it has the ability to remove specific words with lesser importance and can boost the training time. Apart from this, it was added as a means for regularization. A dropout rate of 0.2 was used for this purpose. Next was the LSTM layer with 64 units, aimed to learn the correlation of sentences within a range in the documents. A recurrent dropout of 0.2 was used to update the values of LSTM cells. The LSTM layer is followed by a fully connected layer, the ouput of which is directed to the SVM layer. It consists of 10 neurons and ReLu being the activation function used. In the SVM layer, the l2-regulizer is used along with the softmax activation function in order to predict the class document to which the document belongs.

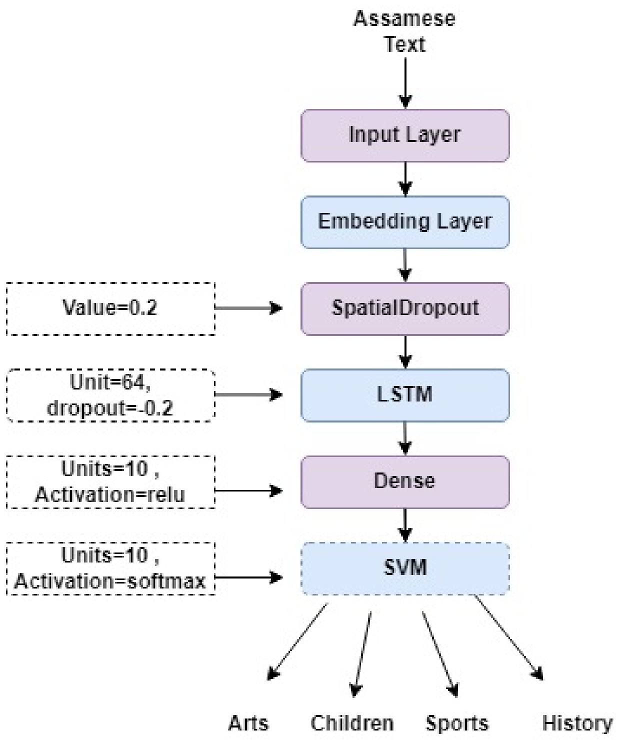

Figure 3. Model architecture of LSTM-SVM

C. *CNN*

This classification model is built using three one dimensional convolution layers, the first two of which are stacked one after the other. In all the convolution layers, 32 filters were applied with varying sizes. The first filter was configured to look at 3 consecutive tokens at a time whereas the second and the third were to look at 5 consecutive tokens and 6 consecutive tokens respectively. The stride was fixed at 1 step at each glance. The first layer received preprocessed word vectors of 100 dimensions from the embedding layer. The tanh function was used as the activation function in this whole experiment as against ReLu because better performance rate was recorded. Following the second and the third convolution layers, a 1d maxpooling was used so as to retain only the prominent features from the feature map. Next to it, a flatten layer was applied which converts the 2X2 matrices into one dimensional vectors. The output of this layer is passed to a dense layer that has 128 neurons, followed by another dense layer with 4 neurons and reflecting the number of classes. The last layer had softmax as its activation function. It helped in transforming numeric vectors into vectors of probability. Adam was used as the optimizing function.

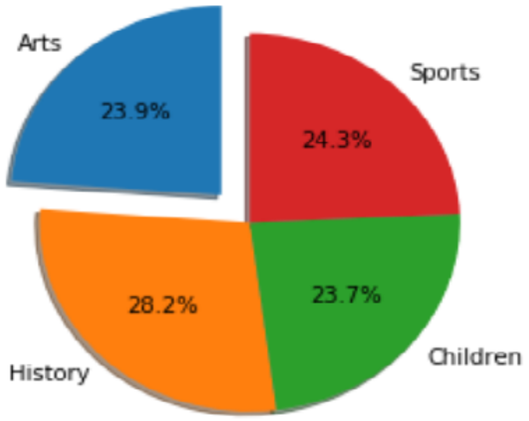

Figure 4. Distribution of documents under each class

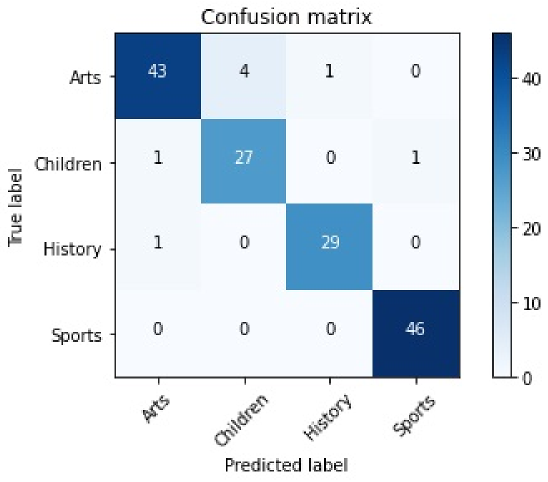

Figure 5. Confusion Matrix of C-LSTM-ATC

D. LSTM

This model accepts its inputs from the embedding layer that converted the Assamese words into 100 dimensional real valued vectors. The first layer fitted into the model is a spatialdropout layer with a dropout value of 0.2. Next to it is a 64 units LSTM layer. In order to update the LSTM memory cells, a recurrent dropout of 0.2 is used in this layer. The output of this layer flows to a dense layer with 4 neurons corresponding to the four classes. Sparsecrossentropy is used as the loss function and Adam is used for optimizing the output.

4. Experimental Settings

A. Dataset Description

The data set utilized for this research has been built from the Assamese corpus developed by the NLP Lab of Gauhati University. This dataset contains articles gathered from some popular Assamese history books, newspapers and magazines. It comprises of numerous text files covering a variety of subjects. From this collection, some files have been chosen for forming the dataset. It contains 768 documents altogether. The dataset is manually categorized into 4 classes viz ’Arts’, ‘Sports’, ‘Children’ and ’History’. Figure 4 displays the distribution of articles (in terms of percentage) and Table II displays the distribution (in terms of numbers) with respect to each of the four classes.

B. Experimental Environment

Four Experiments were carried out using CNN, LSTM, C-LSTM-ATC and LSTM-SVM on the dataset that contained 768 Assamese documents using Python, with Keras and TensorFlow as backend. 80% of the documents were randomly selected for training while the rest 20% were left for the testing. Therefore, a set consisting of 615 documents from all four categories was used as training set and the rest 153 documents formed the testing set. Table III displays the experimental environment for all the four experiments.Table IV gives the parameter settings for C-LSTM-ATC model.The model’s testing and training accuracy rate reached a saturation level after 40 epochs. Table V shows the parameter settings used in LSTM-SVM hybrid model. After exposing this model to 45 epochs, it was observed that the same pattern was repeated. The CNN model’s accuracy and loss rate got saturated after 30 epochs.

C. Results

In order to measure the effectiveness of the four proposed models, each was separately tested on the dataset with documents categorized into four classes. The model C-LSTM-ATC‘s testing and training accuracy rate reached a saturation level at 40 epochs .It outputted an accuracy of 97.2% with a loss rate of 0.11. Figure 5 displays the confusion matrix(CM), which is applied to evaluate the performance of a classifier , for the proposed C-LSTM-ATC classification model.

The LSTM-SVM model, on being exposed to 45 epochs, repeated the same pattern of loss and accuracy. It outputted an accuracy of 92.2% and loss of 0.28. Figure 6 displays LSTM-SVM model’s CM.

The CNN model displayed training saturation at an early stage. However, during testing, it displayed an accuracy of 95.4% at 30 epochs with a loss rate of 0.2. Figure 7 displays the CM for CNN.

The fourth model of this research that employed LSTM model,outputted an accuracy of 94% at 40 epochs and a loss of 14%. Figure 8 presents the CM for this model.

Furthermore, 4 separate evaluation metrics namely accuracy, recall, precision and F1-score were applied to compare the correctness of the models. Table VI shows that the proposed C-LSTM-ATC model not only registered high accuracy but also outperformed the other models in terms of all the other metrices too. The comparative testing accuracy

TABLE II. Distribution of documents under each class in the dataset

Category	Number of Documents
Arts	184
Children	216
History	182
Sports	186

TABLE III. Experimental environment

Length	Number
Vocabulary	27497
Average	201
Max	14827

TABLE IV. Parameter settings for C-LSTM-ATC

Parametrs	Values
No. of epochs	40
Batch size	128
No. of filters	32
Filter sizes	4
Embedding dimension	100

TABLE V. Parameter settings for LSTM-SVM

Parameters	Values
No. of epochs	45
Batch size	128
Embedding dimension	100

TABLE VI. Performance of the models in terms of different parameters

Model	Accuracy	Precision	Recall	F1-Score
CNN	95.4	90.3	90.3	90
LSTM	94	87	87	86
LSTM-SVM	92.2	86	85	84
C-LSTM-ATC	97.2	94	95	95

TABLE VII. Comparative analysis of the Proposed Models with other works

Reference	Model Used	Language	Purpose	Accuracy Rate(in percentage)
[21]	LSTM	Bangla	Topic Classification	95.42
[19]	Bi-LSTM with CNN	Chinese	News Classification	96.45
[22]	CNN-RNN	English	News Classification	93.38
[23]	Attention GRU	Arabic	News Classification	96.94
[24]	SVM	Turkish	News Classification	91.65(F1-score)
[25]	Wordnet	Assamese	word Classification	90.27
[26]	Naive Bayes	Assamese	Document Classification	94.4(F1-Score)
Proposed work	CNN	Assamese	Document classification	95.4
Proposed work	LSTM	Assamese	Document classification	94
Proposed work	C-LSTM-ATC	Assamese	Document classification	97.2
Proposed work	LSTM-SVM	Assamese	Document classification	92.2

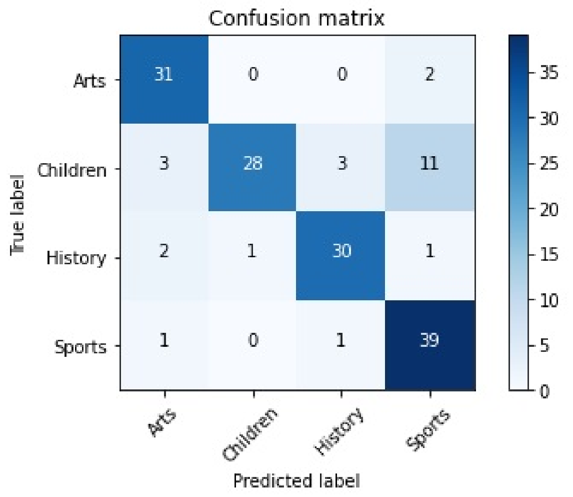

Figure 6. Confusion Matrix of LSTM-SVM

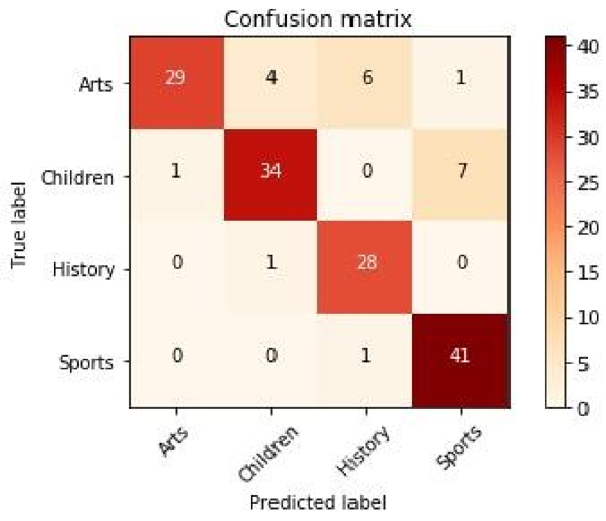

Figure 8. Confusion Matrix for LSTM

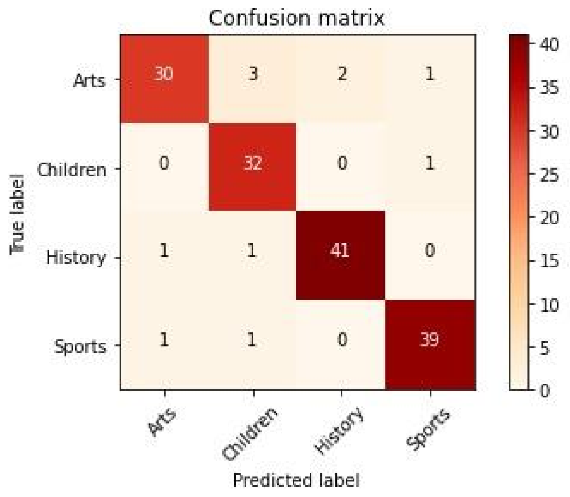

Figure 7. Confusion Matrix of CNN

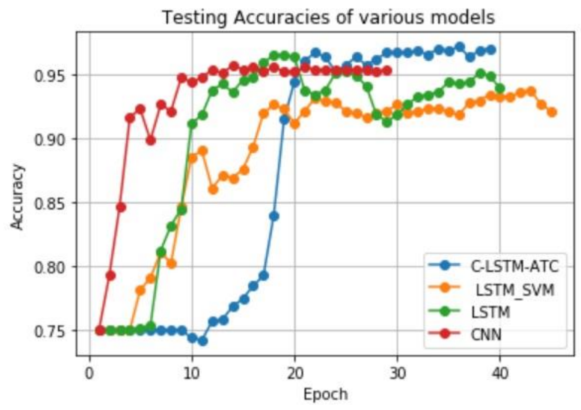

Figure 9. Accuracy graph of the four models

and losses of the different models can be displayed in the Figure 9 and Figure 10 respectively.

Table VII shows works on text classification in different languages across the world, using different models.

Additionally, the performance of the models were also assessed by applying them to three different sizes of datasets. The first one comprised of 456 documents ,while the second set contained 634 documents and finally the third dataset had 768 documents in it. It can be observed from Figure 11 that all the models improved their accuracy rate with the increase in the size of the datasets.

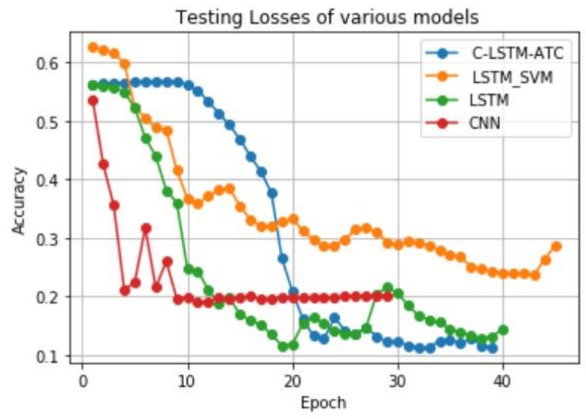

Figure 10. Loss Graph of the four models

D. Analysis of the performance of C-LSTM-ATC model

The proposed C-LSTM-ATC model for Assamese text classification achieved an accuracy of 97.2%, yet, it was not free from erroneous results. The 2.8% errors observed here

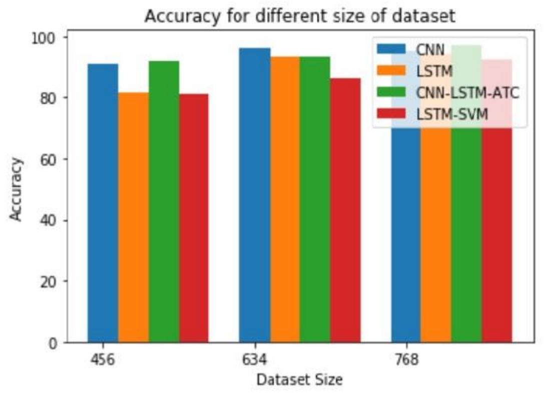

Figure 11. Accuracy Rate of the four models under different dataset sizes

were mainly due to the ambiguous nature of text itself. As for instance, an article from the sports category mainly gave details about the historical aspect of a sport itself. Since its content was mainly historical, therefore the classifier categorized it as an article from the history class.

5. Conclusion

During the course of this research, an attempt was made to introduce a hybrid model using the CNN and LSTM for classifying Assamese text. Apart from this hybrid model, the CNN, LSTM and a hybrid model consisting of LSTM and SVM were also tested on the Assamese dataset consisting of 768 documents which were categorized into four classes- arts, children, history and sports. The experimental findings using this dataset exhibit that over the other models, the C-LSTM-ATC model outperforms by registering an accuracy of 97.2%, F1-score 95%, precision 94% and recall 95%. This model can significantly contribute towards the classification of Assamese documents into specific categories, which was a field that has remained unexplored for long. In the future, efforts should be given to classify Assamese documents using document vectors with higher precision.

References

[1] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.

[2] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the 2013 conference on empirical methods in natural language processing*, 2013, pp. 1631–1642.

[3] S. I. Wang and C. D. Manning, "Baselines and bigrams: Simple, good sentiment and topic classification," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2012, pp. 90–94.

[4] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," *Advances in neural information processing systems*, vol. 13, 2000.

[5] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[7] K. Cho and B. Van Merriënboer, "C. g ülçehre, d. bahdanau, f. bougares, h. schwenk, and y. bengio. learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proc. of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

[8] A. Krizhevsky, "Advances in neural information processing systems," *(No Title)*, p. 1097, 2012.

[9] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[10] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. ARTICLE, pp. 2493–2537, 2011.

[11] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1746–1751.

[12] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," *Advances in neural information processing systems*, vol. 28, 2015.

[13] T. Lei, R. Barzilay, and T. Jaakkola, "Molding cnns for text: non-linear, non-consecutive convolutions," *arXiv preprint arXiv:1508.04112*, 2015.

[14] J. Wang, Z. Wang, D. Zhang, and J. Yan, "Combining knowledge with deep convolutional neural networks for short text classification." in *IJCAI*, vol. 350, 2017, pp. 3 172 077–3 172 295.

[15] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[16] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv preprint arXiv:1605.05101*, 2016.

[17] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 29, no. 1, 2015.

[18] L. Yao, C. Mao, and Y. Luo, "Graph convolutional networks for text classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 7370–7377.

[19] Y. Li, X. Wang, and P. Xu, "Chinese text classification model based on deep learning," *Future Internet*, vol. 10, no. 11, p. 113, 2018.

[20] C. Zhou, C. Sun, Z. Liu, and F. Lau, "A c-lstm neural network for text classification," *arXiv preprint arXiv:1511.08630*, 2015.

[21] M. M. Rahman, R. Sadik, and A. A. Biswas, "Bangla document classification using character level deep learning," in *2020 4th In-*

ternational Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). IEEE, 2020, pp. 1–6.

[22] R. Wang, Z. Li, J. Cao, T. Chen, and L. Wang, "Convolutional recurrent neural networks for text classification," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–6.

[23] A. Elnagar, R. Al-Debsi, and O. Einea, "Arabic text classification using deep learning models," *Information Processing Management*, vol. 57, no. 1, p. 102121, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457319303413

[24] Ö. Köksal, "Tuning the turkish text classification process using supervised machine learning-based algorithms," in *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 2020, pp. 1–7.

[25] J. Sarmah, N. Saharia, and K. Shikhar, "A novel approach for document classification using assamese wordnet," in *6th International Global Wordnet Conference*, 2012, pp. 324–329.

[26] M. Gogoi and S. K. Sarma, "Document classification of assamese text using naïve bayes approach," *International Journal of Computer Trends and Technology*, vol. 30, pp. 182–186, 2015.

Chayanika Talukdar is an Assistant Professor in the Department Of Computer Science, NERIM, Guwahati (Assam). She is currently pursuing her Ph.D from the Department of Information Technology, Gauhati University. She completed her Master of Computer Application (MCA) from Gauhati University. Her research interests are NLP and AI.

Shikhar Kumar Sharma is a Professor and Head of the Department of Information Technology, Gauhati University. He is also the Director of Brahmaputra Studies, Gauhati University. He completed his Ph.D on Computer Communication from Gauhati University. He has over 100 research publications to his credit. His research interests are NLP, Language Technology and AI.