AI & SOCIETY
https://doi.org/10.1007/s00146-020-01011-0

STUDENT FORUM

Attention-based convolutional neural network for Bangla sentiment analysis

Sadia Sharmin[1] · Danial Chakma[1]

Received: 12 September 2019 / Accepted: 15 June 2020

Abstract
With the accelerated evolution of the internet in the form of web-sites, social networks, microblogs, and online portals, a large number of reviews, opinions, recommendations, ratings, and feedback are generated by writers or users. This user-generated sentiment content can be about books, people, hotels, products, research, events, etc. These sentiments become very beneficial for businesses, governments, and individuals. While this content is meant to be useful, a bulk of this writer-generated content requires using text mining techniques and sentiment analysis. However, there are several challenges facing the sentiment analysis and evaluation process. These challenges become obstacles in analyzing the accurate meaning of sentiments and detecting suitable sentiment polarity specifically in the Bangla language. Sentiment analysis is the practice of applying natural language processing and text analysis techniques to identify and extract subjective information from text. This paper presents how the attention mechanism could be incorporated effectively and efficiently in analyzing the Bangla sentiment or opinion.

Keywords Bangla sentiment analysis (BSA) · Polarity prediction · Attention · CNN · NLP · Deep learning (DL)

1 Introduction

Sentiment analysis or opinion mining is a natural language processing and information extraction task that aims to obtain writers feelings expressed in positive or negative comments, questions and requests by analyzing a large number of documents. Generally, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall tonality of a document. In recent years, the exponential increase in internet usage and the exchange of public opinion is the driving force behind sentiment analysis today. The web is a large repository of structured and unstructured data. The analysis of these data to extract latent public opinion and sentiment is a challenging task. Zhang and Liu (2017) defined a sentiment or opinion as a quintuple-

$<o_j, f_{jk}, so_{ijkl}, h_i, t_l>$, where o_j is a target object, f_{jk} is a feature of the object o_j, so_{ijkl} is the sentiment value of the opinion of the opinion holder h_i on feature f_{jk} of object o_j at time t_l, so_{ijkl} is +ve, -ve, or neutral, or a more granular rating, h_i is an opinion holder, and t_l is the time when the opinion is expressed.

The analysis of sentiments may be document-based where the sentiment in the entire document is summarized as positive, negative or objective. It can be sentence based where individual sentiment bearing sentences in the text are classified. It can also be phrase-based where the phrases in a sentence are classified according to polarity. Sentiment analysis identifies the phrases in a text that bears some sentiment. The author may express some objective facts or subjective opinions. It is necessary to distinguish between the two. SA finds the subject towards whom the sentiment is directed. A text may contain many entities, but it is necessary to find the entity towards which the sentiment is directed. It identifies the polarity and degree of sentiment. Sentiments are classified as objective (facts), positive (denotes a state of happiness, bliss or satisfaction on part of the writer) or negative (denotes a state of sorrow, dejection or disappointment on the part of the writer). The sentiments can further be

✉ Sadia Sharmin
sadiasharmin@cse.buet.ac.bd

Danial Chakma
danialcse08@gmail.com

[1] Department of CSE, Bangladesh University of Engineering and Technology, Dhaka, Bangladesh

given a score based on their degree of positivity, negativity or objectivity.

Novelty of our approach: To date, researchers have performed sentiment analysis using Bangla text in several methods and approaches, including SVM, MaxEnt, LSTM, and CNN which are discussed in Sect. 4. Attention techniques, which were introduced in early 2016, were developed and thoroughly investigated in NLP tasks involving the English language. However, to our knowledge, it has not yet been investigated or employed in analyzing Bangla sentiment. This is the main difference between the other approaches and our approach in the Bangla sentiment analysis task. This paper presents an empirical study about the effectiveness of the attention mechanism by incorporating it into a convolutional neural network.

Our contribution in this paper is threefold: (a) Investigate the effectiveness of the attention mechanism in detecting Bangla sentiment from Bangla social media data. (b) Minimize the false detection rate or enhancing prediction accuracy by empirical search of hyper-parameters and feature engineering. (c) Automate generation of synthetic data from existing real datasets using generative adversarial networks (GANs) and test the performance of our ne-tuned model against these synthetic data. In addition, this is the first application of the attention mechanism in Bangla sentiment analysis research papers, which deserves further investigation to determine whether the attention mechanism is actually capable of capturing feature dependency among long Bangla sentences.

The rest of the paper is organized as follows. Section 2 discusses some prominent applications of SA, Sect. 3 describes challenges in SA, Sect. 5 describes our model and approach, Sect. 6 describes the dataset and feature extraction process, Sect. 7 presents experimental results and performance analysis, and finally Sect. 8 provides concluding remarks and future works.

2 Applications of sentiment analysis

The process of conveying information from person to person is called word-of-mouth (WOM) and plays a major role in customer buying decisions. In commercial situations, WOM involves consumers sharing attitudes, opinions, or reactions about businesses, products, or services with other people. WOM communication functions based on social networking and trust. People rely on families, friends, and others in their social network. Research also indicates that people appear to trust seemingly disinterested opinions from people outside their immediate social networks, such as online reviews. This is where sentiment analysis comes into play. The growing availability of opinion rich resources such as online review sites, blogs, social networking sites have made this "decision-making process" easier for people. With the explosion of Web 2.0 platforms, consumers have a platform of unprecedented reach and power by which they can share opinions. Major companies have realized that these consumer voices affect shaping the voices of other consumers. Sentiment analysis thus finds its use in the consumer market for product reviews, marketing for knowing consumer attitudes and trends, social media for finding general opinion about recent popular topics locally, and to determine whether a recently released movie is a hit. Pang-Lee et al. (2002; Pang et al. 2002) broadly classifies the applications into the following categories.

1. Applications to review-related websites
   Movie reviews, product reviews, etc.
2. Applications as a subcomponent technology
   Detecting antagonistic, heated language in emails, spam detection, context-sensitive information detection, etc.
3. Applications in business and government intelligence
   Knowing consumer attitudes and trends.
4. Applications across different domains

Knowing public opinions for political leaders or their notions about rules and regulations in place, etc.

3 Challenges in sentiment analysis

Sentiment analysis approaches aim to extract positive and negative sentiment bearing words from a text and classify the text as positive, negative or objective if it cannot find any sentiment bearing words. In this respect, it can be thought of as a text categorization task. In text classification, there are many classes corresponding to different topics, whereas sentiment analysis has only 3 broad classes. Thus, it seems sentiment analysis is easier than text classification, which is not quite the case. The following general challenges apply not only to English languages but also to other languages, such as Bangla.

3.1 Implicit sentiment and sarcasm

A sentence may have an implicit sentiment even without the presence of any sentiment bearing words. Consider the following examples: (a) How can anyone sit through this movie? (b) One should question the stability of mind of the writer who wrote this book. Both sentences do not explicitly carry any negative sentiment bearing words although both are negative sentences. Thus, identifying semantics is more important in SA than syntax detection.

3.2 Domain dependency

There are many words whose polarity changes from domain to domain. Consider the following examples. (a) The story was unpredictable. (b) The steering of the car is unpredictable. (c) Go read the book. In the first ex-ample, the sentiment conveyed is positive, whereas the sentiment conveyed in the second is negative. The third example has a positive sentiment in the book domain but a negative sentiment in the movie domain (where the director is being asked to go and read the book).

3.3 Thwarted expectations

Sometimes the author deliberately sets up context only to refute it at the end. Consider the following example:

> This film should be brilliant. It sounds like a great plot, the actors are first grade, the supporting cast is good as well, and Stallone is attempting to deliver a good performance. However, it cannot hold up.

Despite the presence of words that are positive in orientation, the overall sentiment is negative because of the crucial last sentence, whereas in traditional text classification this would have been classified as positive as term frequency is more important there than term presence.

3.4 Pragmatics

It is important to detect the pragmatics of user opinion, which may change the sentiment thoroughly. Consider the following examples. (a) I just finished watching Barca DESTROY Ac Milan. (b) That ending completely destroyed me. Capitalization in words can be used with subtlety to denote sentiment. The first example denotes a positive sentiment, whereas the second denotes a negative sentiment. There are many other ways of expressing pragmatism.

3.5 Subjectivity detection

This differentiates between opinionated and non-opinionated text. It is used to enhance the performance of the system by including a subjectivity detection module to filter out objective facts. However, this is often difficult to do. Consider the following examples: (a) I hate love stories. (b) I do not like the movie "I hate stories". The first example presents an objective fact, whereas the second example depicts the opinion about a particular movie.

3.6 Entity identification

A text or sentence may have multiple entities. It is extremely important to determine the entity towards which the opinion is directed. Consider the following examples. (a) Samsung is better than Nokia. (b) Bangladesh defeated Sri Lanka in Cricket. The examples are positive for Samsung and Bangladesh but negative for Nokia and Sri Lanka.

3.7 Negation

Handling negation is a challenging task in SA. Negation can be expressed in subtle ways even without the explicit use of any negative word. A method often followed in handling negation explicitly in sentences such as "I do not like the movie", reverses the polarity of all the words appearing after the negation operator (such as not). However, this does not work for "I do not like the acting but I like the direction". Therefore, we need to consider the scope of negation as well, which extends only until "but". Therefore, the thing that can be done is to change the polarity of all words appearing after a negation word until another negation word appears. However, there can still be problems. For example, in the sentence "Not only did I like the acting but also the direction" the polarity is not reversed after "not" due to the presence of "only". Therefore, these types of combinations of "not" with other words such as "only" have to be considered while designing the algorithm.

4 Related works

Sentiment classification (Pang and Lee 2008) is a classical problem associated with determining the attitude and affective reaction of a person to an event, document and/or media item. A typical setting for training sentiment classification models involves feature extraction on a labeled corpus, followed by classifier training Pang et al. (2002). Such an approach has seen fair success in several sentiment classification tasks, such as Twitter sentiment classification (Jiang et al. 2011), movie reviews (Kennedy and Inkpen 2006) and product reviews (Cui et al. 2006). The opportunity to automatically capture the general public's sentiments about social events, political movements, marketing campaigns, and product preferences has raised interest in both the scientific community, for the exciting open challenges, and the business world, for the remarkable fallouts in marketing and financial market prediction (Cambria 2016). As the problem of sentiment classification expands to new areas, such as new modes of expressions on social media, different languages and even different cultures, large quantities of labeled data and technical resources may not be available, and the syntactical structure and semantic meaning of different languages may differ from language to language (Balahur and Turchi 2012).

Despite challenges, sentiment analysis or opinion mining has become a major point of focus in natural language processing and has gained extensive attention and popularity in the research community. This is proven in the paper (Feldman 2013), with over 7000 articles written on the subject. In recent periods, deep learning applications have exhibited impressive performance across a wide range of NLP tasks, including machine translation (MT) and text to speech. Using deep learning methods, a large portion of works have involved learning word vector representations(Mikolov et al. 2013) and using convolutional neural network (CNN)-based feature extractors for classification(Kim 2014; dos Santos and Gatti 2014; Zhang and Wallace 2015). Shirnai explored the efficacy of different deep learning architectures for semantic analysis of movie reviews and achieved 46.4% accuracy in the Stanford Sentiment Treebank dataset using the CNN + word2vec model for five-label sentiment classification (Shirani-Mehr 2014). Currently, in addition to sentiment mining, the emotional aspects of texts have also attracted the attention of many research areas in NLP, and different researchers have focused on identifying emotions. Chaar and Inkpen adopted a supervised machine learning approach to recognize Ekmans (1992) six basic emotions using different feature sets. Bhowmick et al. (2009) used an ensemble-based multilabel classification technique called random k-label set (RAKEL) for emotion analysis. Although the problem of classifying sentiment in recent years has seen fair success in the English language using deep learning methods, it is not widely studied in Asian languages such as Bangla, Hindi, and Chinese. There are not many resources available when it comes to languages that are not commonly used in social communication or formal news reporting due to their informal and evolving nature. These languages often evolve from a main national language, such as English, and are broadly used by a local community in daily conversation both in the physical and online world. With the popularity of social media and the freedom of expression it affords, languages with localized expressions or variants of formal languages are be-coming widespread in the online environment (Lo et al. 2016). In addition, it is not uncommon to see a few languages being mixed to form a unique language in a multicultural society. One such example is Singlish, the colloquial Singaporean English that has incorporated elements of some Chinese dialects and the Malay language (Mathews and Abraham 2019). Sentiment Analysis on Twitter data for the Portuguese language has been done in (Souza and Vieira 2012). To fully understand the sentiments in this sort of languages, it is essential to analyse them alongside other formal languages. We have found limited resources about sentiment analysis in Bangla. Among these works, Amitava Das in (Das and Bandyopadhyay 2010) constructed a sentiment lexicon for the Bangla language from English SentiWordNet (Esuli and Sebastiani 2006), Shaika et al. (Chowdhury and Chowdhury 2014) performed sentiment analysis using SVM and MxtEnt for Bangla microblog posts, and Azharul et al. (Hasan et al. 2014) reviewed sentiment analysis among some Asian languages, including Bangla. Recently, researchers have expressed their interest in Bengali text, and there are many publications based on sentiment and emotion analysis, theme detection, and topwise opinion summarization with data resources from various Bengali corpora (Singh 2015). Different machine learning techniques, such as SVM with maximum entropy (Chowdhury and Chowdhury 2014), naive Bayes (NB) (Islam et al. 2016), and multinomial naive Bayes (MNB) with mutual information as feature representation (Paul and Shill 2016), have been used to classify Bangla sentences into positive or negative in various Bangla domain texts, such as microblog posts, comments from blogs, and translated review datasets. The word2vec word co-occurrence score combined with the sentiment polarity score of the words was proposed by Amin et al. (Al-Amin et al. 2017) and obtained 75.5% accuracy in each of two classes. Hasan et al. (Hassan et al. 2016) performed sentiment analysis on Bangla and Romanized Bangla text using long short-term memory (LSTM) with binary and categorical cross-entropy loss and achieved 70% accuracy for two classes. Much of the emotion analysis tasks in Bangla texts have been carried out by Das and Bandyopadhyay in words, phrases and sentence levels in Bangla corpora and blogs (Das and Bandyopadhyay 2012; Jack et al. 2014). Consequently, they developed WordNet Aect lists in Bangla from the affect wordlists already available in English. To identify emotions from sentences, they used a conditional random field (CRF)-based classier for recognizing six basic emotion tags for different words of a sentence.

5 Our proposed architecture and method

We adopted an attention-based convolutional neural network to analyze Bangla text sentiment. Our model utilizes the sparsity of the data matrix, which is the representation of all textual comments bearing positive, negative or neutral sentiment. In general, classification or prediction tasks in natural language processing represent a contextual document or comment in some numeric form based on word frequency or word presence. Representation of a document or a comment such as this usually results in a sparse dataset, which basically means most elements in a document or comment are zeros except some element that represent that document or comments. Additionally, working with a large volume of textual data tends to increase the vocabulary size, which presents a considerable challenge in containing the data matrix dimension. This problem is called the curse of dimensionality in the data mining field. Therefore, restricting the data dimension or feature dimension is also the utmost priority. Datasets constructed from large volumes of vocabulary or features are sparse in nature. In those cases, the application of attention in neural networks is typically useful for sparse datasets not only for faster training but also for obtaining good generalization of neural networks for future unknown data. With this in mind, our proposed architecture consists of a convolutional layer for feature extraction followed by one attention layer and then a couple of feedforward layers for the classification of sentiment bearing Bangla comments. Our proposed model is divided into four basic parts:

(a) Feature extraction layer
(b) Intermediate normalization layer
(c) Attention layer
(d) Feedforward classification layer.

Feature Extraction Layer: Feature extraction is one of the important aspects of machine learning-based detection or prediction systems. Convolutional neural networks have shown promising results in extracting image features in computer visions and many other fields. Inspired by that, we used three convolutional layers successively one after another for both feature extraction and input dimension and/or timestep reduction. In the first layer, four one-dimensional convolutions (Conv1D Keras layers) are used, each having 8 filters with kernel sizes of 2, 3, 4, and 5 and strides of 5. Then, these four Conv1D layers are concatenated. Having a stride size of 5 reduces the timestep (input feature dimension) to a factor of 5, in our case, to 1840 = 5 = 368. Next, the second layer is constructed with four Conv1D layers, each having 5 kernels of size 1, 2, 3, and 4 strides with 5 kernels. Then, these four Conv1D layers are concatenated again to form a single Keras tensor. Similarly, the following third layer is formed, with each having 3 filters and a stride of 2. In each convolution layer, we used the exponential linear unit (ELU) as the activation function, which is known to be fast and effective in training neural networks. Each convolutional layer consists of filters (also called the number of kernels) of a specific size (also called kernel size or sliding window size), and each kernel is slided or strided by a specific integer amount along the input dimension. The first convolutional layer consists of 78 filters or kernels each of size 32 (also called kernel size or sliding window size), and each kernel is slided or strided by 2 along the input dimension. With the defined kernel size and input size, the layer outputs a matrix of shape or size *x* by *y*, *x* is input-dim/stride and *y* is number of filters, each column of which holds the weight of a single filter. In our case, each learnable filter in the first layer contains 368 trainable weights. In general, each filter is sufficient to train one feature of the input data tuple.

Intermediate Normalization Layer Editor: Abbreviations and acronyms are typically defined the first time the term is used within the main text and then used throughout the remainder of the manuscript. Please consider adhering to this convention. The target journal may have a list of abbreviations that are considered common enough that they do not need to be defined. After the feature extraction layer, a maxpooling layer followed by one batch normalization layer and one ELU layer is introduced, which acts as an inter-mediate bridge between the feature extraction layer and the attention layer. Batch normalization (Ioe and Szegedy 2015) is crucial for deep neural networks not only for network stabilization but also to solve over fitting and under fitting problems during training deep networks. First, we were undecided whether to employ maxpooling before batch normalization + activation, but after some study, we decided to go with batch norm + activation + maxpool.

Attention Layer: Following the feature extraction layer, we introduced an attention layer, which is a critical component in our network architecture. An attention mechanism in a neural network was born to solve the long-distance dependency problem in machine translation (MT) from the source language to the target language. The idea behind attention came from how humans focus visually on different regions of objects, such as images or words, in one sentence. Human visual attention usually focuses on a certain part with "high resolution or attention" while perceiving the surrounding part in "low resolution or attention" and then adjusting the focal point or performing inference on the object accordingly. In other words, we do not need all aspects to recognize objects, and only important objects will suffice. Motivated by this, the attention mechanism was introduced in (Bahdanau et al. 2015; Xu et al. 2015; Luong et al. 2015; Vaswani et al. 2017) by applying some weight on the input data. We used an additive attention mechanism; our attention layer consists of a 32-dimensional attention vector, a local attention width of 16, and a softmax activation function. The formula of attention vector computation is given below:

$$h_t = \tanh\left(x_t^T W_t + x_t^T W_x + b_h\right) \tag{1}$$

$$e_t = h^T W_a + b_a \tag{2}$$

$$a_t = \frac{\exp\left(e_t\right)}{\sum_{i=1}^{N} \left\{\exp\left(e_{t(i)}\right) + \varepsilon\right\}} \tag{3}$$

where a_t is the attention vector, W_a, W_t, and W_x are trainable weight matrices, b_h and b_a are trainable biases, N is the input layer number in the attention layer, andϵ = exp(-7) is added to avoid dividing by zero runtime errors. Finally, the attention-focused value (output) v_t to the next layer is computed as the weighted sum of the input using the following equation:

$$v_t = \sum_{t'=1}^{n} a_{t,t'} x_{t'} \tag{4}$$

where $a_{t,t'}$ is the attention score indexed at *t* and *t'* computed using Eq. 3 and $x_{t'}$ is the input value indexed at *t'*.

Classification Layer: The classification layer consists of three consecutive dense layers after having flattened the previous layer output. With the exception just before the last dense layer, a batch normalization layer is added to normalize the previous two heavy dense layer outputs. Batch normalization proves to be useful for reducing over fitting during training. In each dense layer, the ELU activation function is employed except for the final layer, where softmax is employed to produce a probability distribution over the target class. The first two dense layers employ 64 and 32 neurons or units, whereas the last layer includes 3 neurons or units, as we have 3 different target classes: positive, negative and neutral sentiment bearing comments in the dataset.

The activation function used is as follows:

$$f(x) = \begin{cases} \alpha\left(e^{\{x\}} - 1\right), \text{ if } x \le 1 \\ x, \text{ if } x > 0 \end{cases} \tag{5}$$

where α is a scaler parameter and usually assumes values between [0, 1]. The softmax function is defined as

$$f_i(\vec{x}) = \frac{\exp\left(e^{x_i}\right)}{\sum_{j=1}^{J} \exp\left(e^{x_j}\right)}, \text{ for } i = 1, 2, \ldots J \tag{6}$$

where $\vec{x}$ is an input vector to the softmax function and J is the number of input layers.

6 Dataset and features extraction

We used datasets collected from social media that were manually handcrafted and labeled (Fig. 1). Among the social media, we chose the BBC Bangla and Prothom Alo platforms for the source of the Bengali corpus. Some of the sample comments and reviews from the media are listed in Fig. 2. These are the user comments about specific topics, including politics and sports. In sports topics, we categorized those user comments about specific aspects of a sport. For example, for cricket, we included user comments about the bowling, bat-ting, and team management aspects of cricket.

Labeling of those reviews and comments was performed manually by examining the comment content. Our dataset includes 2979 reviews and comments; among these comments, 566, 2,152, and 261 were identified as positive, negative and neutral comments, respectively. The data distributions of these three classes are shown in Fig. 3.

The dataset consists of 5316 unique words or vocabulary sizes after preprocessing stop words and stemming. Stemming is a useful technique to find only root words after removing suffixes and/or prefixes from a specific word. In extracting features, we used bag-of-words (BOW) techniques called term frequency and inverse document frequency (TF-IDF) to extract features from filtered comments.

Although we preprocessed raw comments, we still had a large set of vocabulary or feature dimensions to work with. Therefore, we decided to apply the feature reduction technique using PCA (principal component analysis) and reduced dimensionality to 1536, which kept approximately 92.34% variance among the feature sets. To decide which feature to keep or which should be discarded, we performed an experiment (PCA) with the original dataset and plotted a line graph with the *Y*-axis as the cumulative variance ratio and the *X*-axis as the number of components, which is shown in Fig. 4. It is clear that if we want to ensure feature variance above 90, then we need to keep at least 1500 features (or vocabulary size) among 5316 features. In Fig. 5, some of the most frequently occurring negative words are listed from our dataset. Finally, we checked for missing or null values in any attributes and replaced those with zero values. Then, dimension-reduced datasets were transformed into the range of [−1,1] using the following min–max normalization formula for faster processing and/or training of our neural network.

$$x_{std} = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{7}$$

$$x_{\text{new}} = x_{\text{std}} \times (\max - \min) + \min \tag{8}$$

where min and max are the new feature ranges.

7 Experimental results and analysis

We performed experiments to test our model's rigidity with different settings. The following subsection discusses our model performance and sensitivity to different learning rates. In this experiment, we partitioned our dataset into training and validation subsets, which correspond to 75% and 25%, respectively, and trained our model with batch sizes of 32 and 140 epochs. For the optimization algorithm, we employed a stochastic gradient descent algorithm using the NADAM (Sutskever et al. 2013; Dozat 2015) optimizer that optimizes the loss function defined in Eqs. 17 and 18. We observed that our model generated the best performance with a learning rate of 0.002, and at the end of 140 epochs, it achieved 96.36% and 70% accuracy on the training and validation sets, respectively.

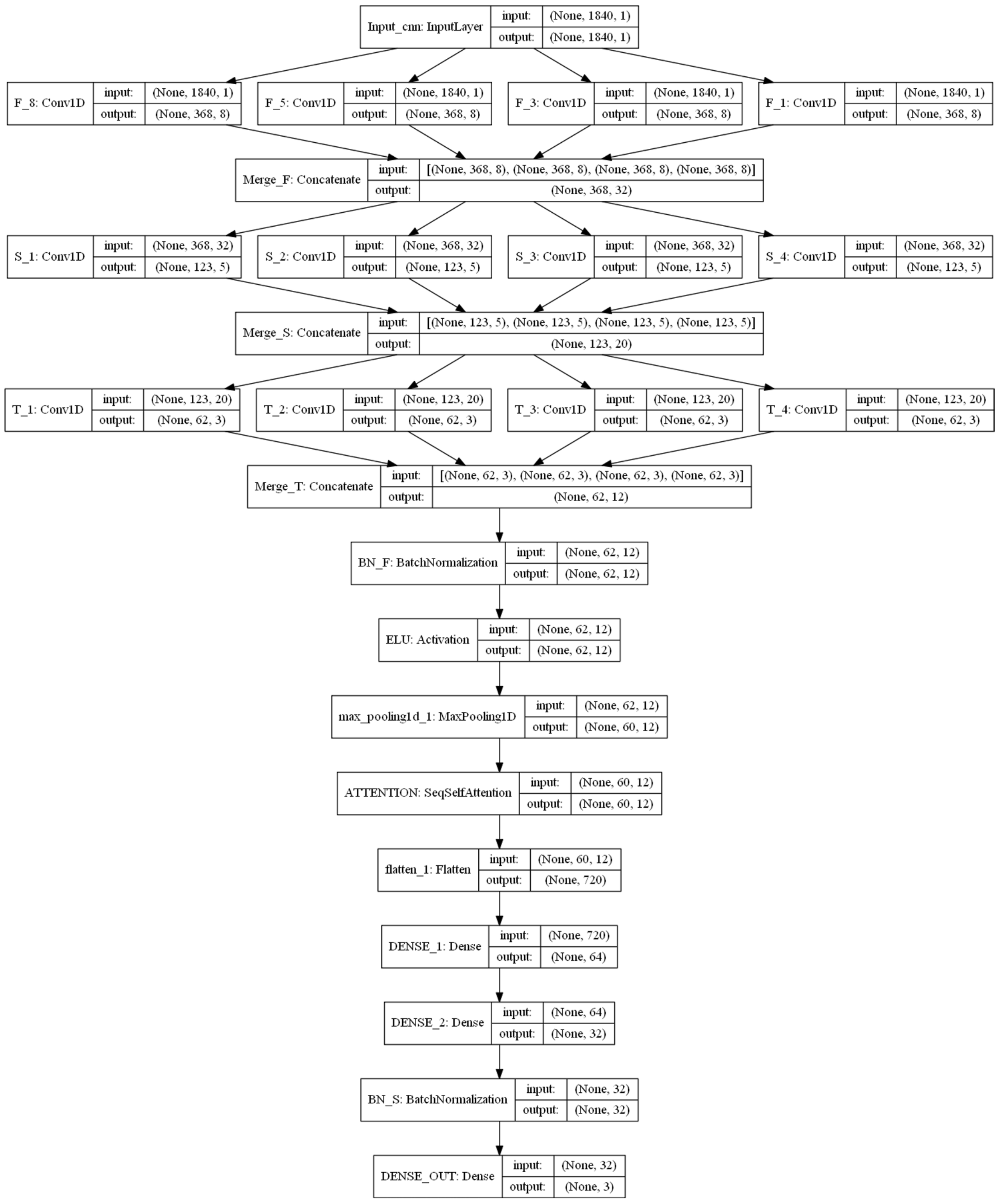
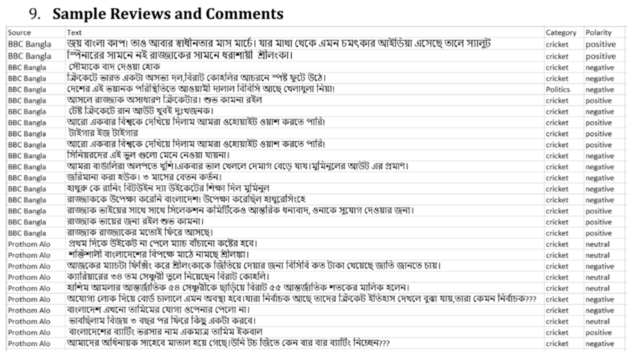

Input_cnn: InputLayer | input: (None, 1840, 1) | output: (None, 1840, 1)
F_8: Conv1D | input: (None, 1840, 1) | output: (None, 368, 8)
F_5: Conv1D | input: (None, 1840, 1) | output: (None, 368, 8)
F_3: Conv1D | input: (None, 1840, 1) | output: (None, 368, 8)
F_1: Conv1D | input: (None, 1840, 1) | output: (None, 368, 8)
Merge_F: Concatenate | input: [(None, 368, 8), (None, 368, 8), (None, 368, 8), (None, 368, 8)] | output: (None, 368, 32)
S_1: Conv1D | input: (None, 368, 32) | output: (None, 123, 5)
S_2: Conv1D | input: (None, 368, 32) | output: (None, 123, 5)
S_3: Conv1D | input: (None, 368, 32) | output: (None, 123, 5)
S_4: Conv1D | input: (None, 368, 32) | output: (None, 123, 5)
Merge_S: Concatenate | input: [(None, 123, 5), (None, 123, 5), (None, 123, 5), (None, 123, 5)] | output: (None, 123, 20)
T_1: Conv1D | input: (None, 123, 20) | output: (None, 62, 3)
T_2: Conv1D | input: (None, 123, 20) | output: (None, 62, 3)
T_3: Conv1D | input: (None, 123, 20) | output: (None, 62, 3)
T_4: Conv1D | input: (None, 123, 20) | output: (None, 62, 3)
Merge_T: Concatenate | input: [(None, 62, 3), (None, 62, 3), (None, 62, 3), (None, 62, 3)] | output: (None, 62, 12)
BN_F: BatchNormalization | input: (None, 62, 12) | output: (None, 62, 12)
ELU: Activation | input: (None, 62, 12) | output: (None, 62, 12)
max_pooling1d_1: MaxPooling1D | input: (None, 62, 12) | output: (None, 60, 12)
ATTENTION: SeqSelfAttention | input: (None, 60, 12) | output: (None, 60, 12)
flatten_1: Flatten | input: (None, 60, 12) | output: (None, 720)
DENSE_1: Dense | input: (None, 720) | output: (None, 64)
DENSE_2: Dense | input: (None, 64) | output: (None, 32)
BN_S: BatchNormalization | input: (None, 32) | output: (None, 32)
DENSE_OUT: Dense | input: (None, 32) | output: (None, 3)

Fig. 1 Our proposed neural network. Each layer is a Keras implemented layer except for the "attention" layer, which we implemented in Keras. The layers input and the output shape is shown where "none" indicates the batch dimension

7.1 Performance metrics

To measure model performance, some evaluation metrics were used whose definitions and formulas are given below.

Definition. True positive (tp) is the number of actual positive instances that are predicted as positive, false negative (fn) is the number of actual positive instances that are predicted as negative, false positive (fp) is the number of actual negative instances that are predicted as positive, true negative (tn) is the number of actual negative instances that are predicted as negative, N is the total number of instances, and C is the number of different classes.



9. **Sample Reviews and Comments**

Source	Text	Category	Polarity
BBC Bangla	জয় বাংলা কাপ! তাও আবার স্বাধীনতার মাস মার্চে। যার মাথা থেকে এমন চমৎকার আইডিয়া এসেছে তালে স্যালুট	cricket	positive
BBC Bangla	স্পিনারের সামনে নই রাজ্জাকের সামনে ধরাশায়ী শ্রীলংকা।	cricket	positive
BBC Bangla	সৌম্যকে বাদ দেওয়া হোক	cricket	negative
BBC Bangla	ক্রিকেটে ভারত একটা অসভ্য দল,বিরাট কোহলির আচরনে স্পষ্ট ফুটে উঠে।	cricket	negative
BBC Bangla	দেশের এই ভয়ানক পরিস্থিতিতে আওয়ামী দালাল বিবিসি আছে খেলাধুলা নিয়া!	Politics	negative
BBC Bangla	আসলে রাজ্জাক অসাধারণ ক্রিকেটার। শুভ কামনা রইল	cricket	positive
BBC Bangla	টেষ্ট ক্রিকেটে রান আউট খুবই দুঃখজনক।	cricket	negative
BBC Bangla	আরো একবার বিশ্বকে দেখিয়ে দিলাম আমরা ওহোয়াইট ওয়াশ করতে পারি!	cricket	positive
BBC Bangla	টাইগার ইজ টাইগার	cricket	positive
BBC Bangla	আরো একবার বিশ্বকে দেখিয়ে দিলাম আমরা ওহোয়াইট ওয়াশ করতে পারি!	cricket	positive
BBC Bangla	সিনিয়রদের এই ভুল গুলো মেনে নেওয়া যায়না।	cricket	negative
BBC Bangla	আমরা বাঙালিরা অলপতে খুশি।একবার ভাল খেললে দেমাগ বেড়ে যায।মুমিনুলের আউট এর প্রমাণ।	cricket	negative
BBC Bangla	জরিমানা করা হউক। ৩ মাসের বেতন কর্তন।	cricket	negative
BBC Bangla	হাথুরু কে রানিং বিটউইন দ্যা উইকেটের শিক্ষা দিল মুমিনুল	cricket	negative
BBC Bangla	রাজ্জাককে উপেক্ষা করেনি বাংলাদেশ! উপেক্ষা করেছিল হাথুরেসিংহে	cricket	negative
BBC Bangla	রাজ্জাক ভাইয়ের সাথে সাথে সিলেকশন কমিটিকেও আন্তরিক ধন্যবাদ, ওনাকে সুযোগ দেওয়ার জন্য।	cricket	positive
BBC Bangla	রাজ্জাক ভায়ের জন্য রইল শুভ কামনা।	cricket	positive
BBC Bangla	রাজ্জাক রাজ্জাকের মতোই ফিরে আসছে।	cricket	positive
Prothom Alo	প্রথম দিকে উইকেট না পেলে ম্যাচ বাঁচানো কষ্টের হবে।	cricket	neutral
Prothom Alo	শক্তিশালী বাংলাদেশের বিপক্ষে মাঠে নামছে শ্রীলঙ্কা।	cricket	neutral
Prothom Alo	আজকের ম্যাচটা ফিক্সিং করে শ্রীলংকাকে জিতিয়ে দেয়ার জন্য বিসিবি কত টাকা খেয়েছে জাতি জানতে চায়।	cricket	negative
Prothom Alo	ক্যারিয়ারের ৩৪ তম সেঞ্চুরী তুলে নিয়েছেন বিরাট কোহলি।	cricket	neutral
Prothom Alo	হাশিম আমলার আন্তর্জাতিক ৫৪ সেঞ্চুরীকে ছাড়িয়ে বিরাট ৫৫ আন্তর্জাতিক শতকের মালিক হলেন।	cricket	neutral
Prothom Alo	অযোগ্য লোক দিয়ে বোর্ড চালালে এমন অবস্থা হবে।যারা নির্বাচক আছে তাদের ক্রিকেট ইতিহাস দেখলে বুঝা যায়,তারা কেমন নির্বাচক???	cricket	negative
Prothom Alo	বাংলাদেশ এখনো তামিমের যোগ্য ওপেনার পেলো না।	cricket	negative
Prothom Alo	ভাবছিলাম বিজয় ৩ বছর পর ফিরে কিছু একটা করবে।	cricket	neutral
Prothom Alo	বাংলাদেশের ব্যাটিং ভরসার নাম একমাত্র তামিম ইকবাল	cricket	positive
Prothom Alo	আমাদের অধিনায়ক সাহেবে মাতাল হয়ে গেছে।উনি টচ জিতে কেন বার বার ব্যাটিং নিচ্ছেন???	cricket	negative

Fig. 2 Samples reviews and comments

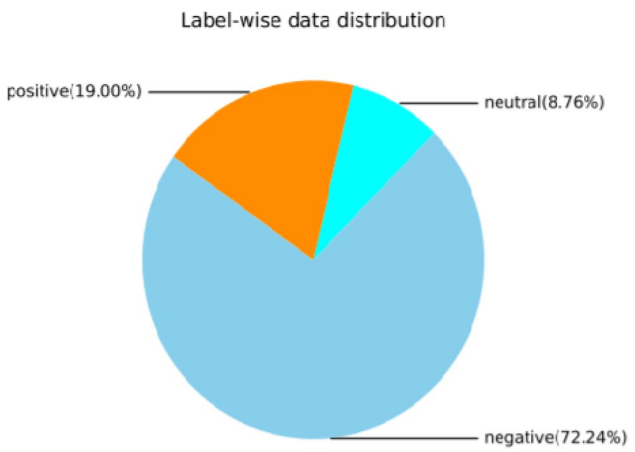


Fig. 3 Proportion of class labels in the dataset

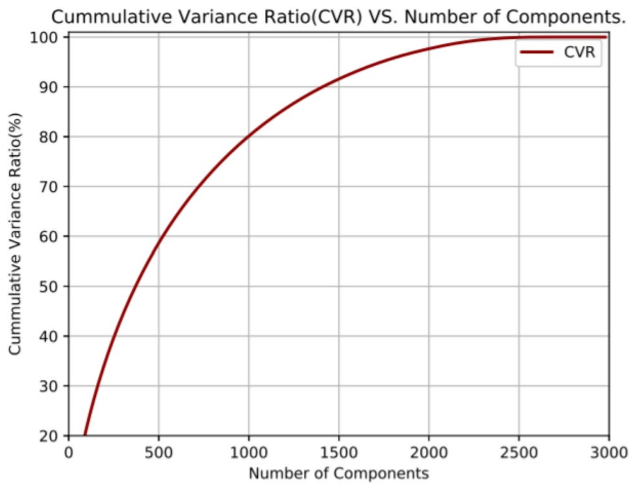
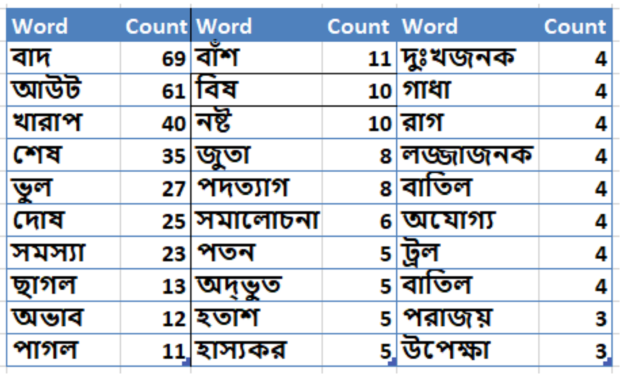


Fig. 4 Cumulative variance ratio vs component numbers



Word	Count	Word	Count	Word	Count
বাদ	69	বাঁশ	11	দুঃখজনক	4
আউট	61	বিষ	10	গাধা	4
খারাপ	40	নষ্ট	10	রাগ	4
শেষ	35	জুতা	8	লজ্জাজনক	4
ভুল	27	পদত্যাগ	8	বাতিল	4
দোষ	25	সমালোচনা	6	অযোগ্য	4
সমস্যা	23	পতন	5	ট্রল	4
ছাগল	13	অদ্ভুত	5	বাতিল	4
অভাব	12	হতাশ	5	পরাজয়	3
পাগল	11	হাস্যকর	5	উপেক্ষা	3

Fig. 5 Some most common negative words in dataset

Having defined the above definition, we can now dene accuracy (AC), precision (PR), recall (RC), F-measure, micro and macro average measures.

Definition. Accuracy (AC) is the fraction of correctly classified instances.

$$AC = \frac{\sum_c^C tp_c}{N}, \text{ in general} \tag{9}$$

$$AC = \frac{tp + tn}{N}, \text{for binary cases} \tag{10}$$

Definition Error(ER) is the fraction of misclassified instances.

$$ER = \frac{\sum_c^C fp_c}{N} = 1 - AC \tag{11}$$

Definition. Precision (PR) or positive predictive value (PPV) is the ratio of correctly classified instances (tp) in front of all the positively predicted instances (tp + *fp).*

$$PR = tp + fp \tag{12}$$

Definition. Recall (RC) OR Sensitivity (OR true positive rate) is the ratio of correctly classified instances (tp) in front of all true positive instances (tp + *fn).*

$$RC = tp + fn \tag{13}$$

Definition. The F-measure (F1), which takes into account both precision (PR) and recall (RC), is the single measure of the combination of two and is defined as the harmonic mean of two measures precision and recall.

$$F1 = \frac{2}{\frac{1}{PR} + \frac{1}{RC}} \tag{14}$$

The macro averaged results can be computed as indicated by $L = \{\lambda_j: j = 1 \ldots q\}$, which is the set of all labels. Consider a binary evaluation measure B(tp, tn, fp, fn), which is calculated based on the number of true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn). Let $tp_\lambda, fp_\lambda, tn_\lambda andfn_\lambda$ is the number of true positives, false positives, true negatives and false negatives after binary evaluation for a label.

$$B_{\text{macro}} = \frac{1}{q}\sum_{\lambda=1}^{q} B\left(tp_\lambda, tn_\lambda, fp_\lambda, fn_\lambda\right) \tag{15}$$

A microaverage can be computed as follows:

$$B_{\text{micro}} = B\left(\sum_{\lambda=1}^{q} tp_\lambda, \sum_{\lambda=1}^{q} tn_\lambda, \sum_{\lambda=1}^{q} fp_\lambda, \sum_{\lambda=1}^{q} fn_\lambda\right) \tag{16}$$

For the loss function, we used categorical cross-entropy to compute loss at the end of each epoch because our target label was one-hot encoded. The equation for computation of loss is given below:

$$\hat{y}_{\text{norm}} = \frac{\hat{y}}{\sum_{x\in\hat{y}} x} \tag{17}$$

$$\mathcal{L}\left(\hat{y}_{\text{norm}}, y_{\text{target}}\right) = -\sum_{x} y_{\text{target}}(x) \times \log\left(\hat{y}_{\text{norm}}(x)\right) \tag{18}$$

where $\hat{y}$ is the output vector of the last layer, $\hat{y}_{\text{norm}}$ is the normalized vector of $\hat{y}$, y_{target} is the true class label vector, x is the index of both equal dimensional vectors y_{target} and $\hat{y}_{\text{norm}}$, and finally, $\mathscr{L}$ is the scalar loss value that is minimized by the optimizer.

7.2 Model accuracy evaluation

In Fig. 6, we observe that over epoch training accuracy increases for the noted learning rate [0.3, 0.1, 0.03, 0.01, 0.002, 0.001, 0.0003, 0.0001]. However, in all the depicted learning rate (lr) settings, the model does not perform equally. For instance, the model achieves accuracy slightly above 0.7 with lr(0.3) while achieving accuracy slightly below 0.92 with lr(0.1, 0.03, 0.0001) because the other learning rate models achieve approximately the same training accuracy at the end of 140 epochs. Notably, the model performs best in accuracy when the learning rate is set to 0.002 and degrades performance if set either below or above this learning rate, which confirms our earlier claim. The validation curves in Fig. 7 follow similar trends. Note that for the best learning rate [0.001, 0.002, 0.003] within 20 epochs, the model learns approximately 80% about the distribution of the training set, while the validation set achieves approximately 70%. In Fig. 8, we notice that the validation accuracy curve follows the training accuracy curve, and on average, the validation accuracy is always below the training accuracy, although we witness some overlapping and spike for both curves at the initial epoch, namely, between [0, 21]. This is perhaps due to the random initialization of weights and biases of the model in all layers. Overall, the model achieves a satisfactory level of accuracy with limited model capacity (few layers and parameters) if not up to the mark.

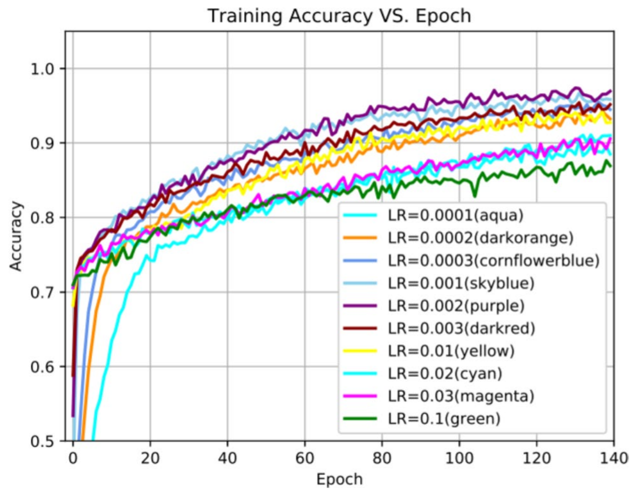


Fig. 6 Training accuracy with varying numbers of learning rates (LRs) of the optimizer. The *Y*-axis corresponds to accuracy [0, 1], while the *X*-axis corresponds to the epoch number. One epoch means training the whole training dataset once. The higher the accuracy with a lower number of epochs, the better the model's performance

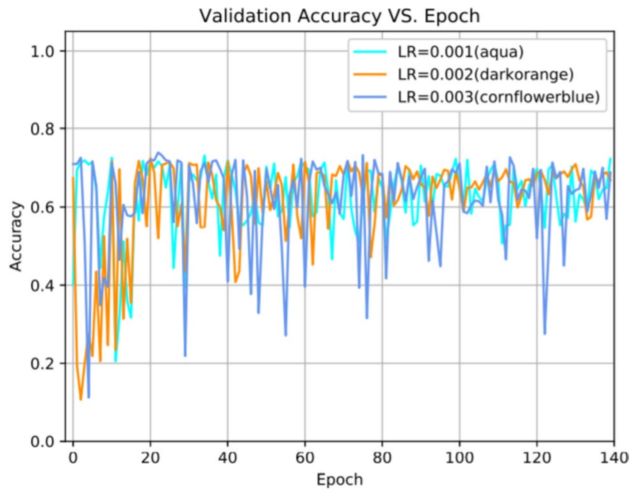


Fig. 7 Validation accuracy with varying numbers of learning rates (LRs) of the optimizer. The *X*-axis corresponds to the accuracy of the validation dataset measured at the end of every training epoch. Notes, the validation dataset is independent of the training dataset and is not used for training the model. The accuracy of the validation dataset measures how well the model will perform for the unseen data

In Fig. 9, for learning rate [0.001, 0.002], the model does quite well to learn or distinguish target classes from the feature set or data distribution, but performance degrades or

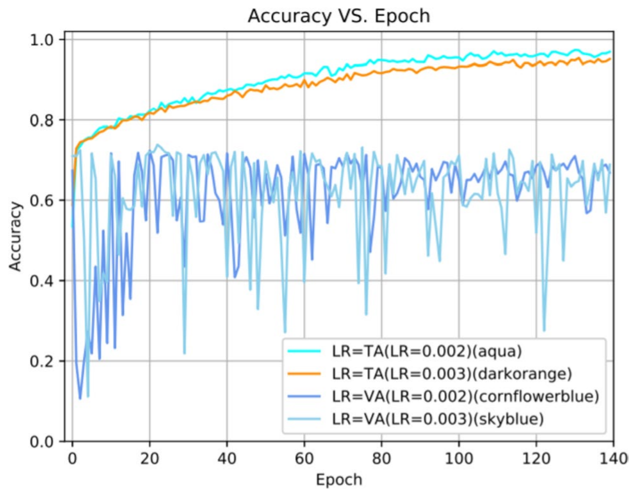


Fig. 8 The training and validation accuracies were plotted together to determine whether the model was overfitting. TA represents training accuracy and VA represents validation accuracy

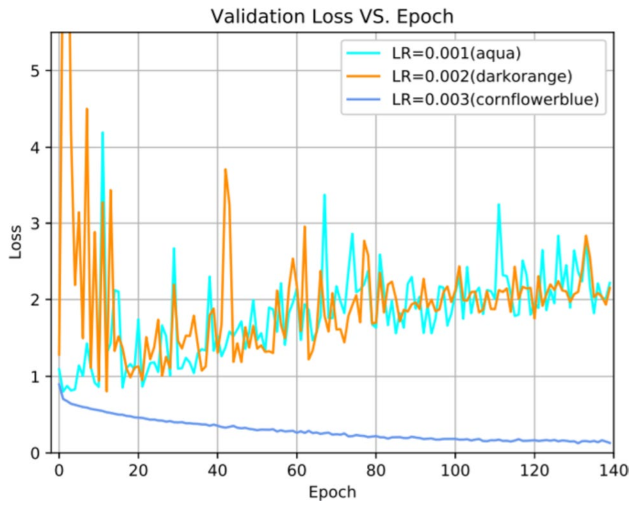


Fig. 10 Validation loss similar to the training loss except the quantity is measured using the validation dataset. It indicates how well a model will perform for the unseen future data

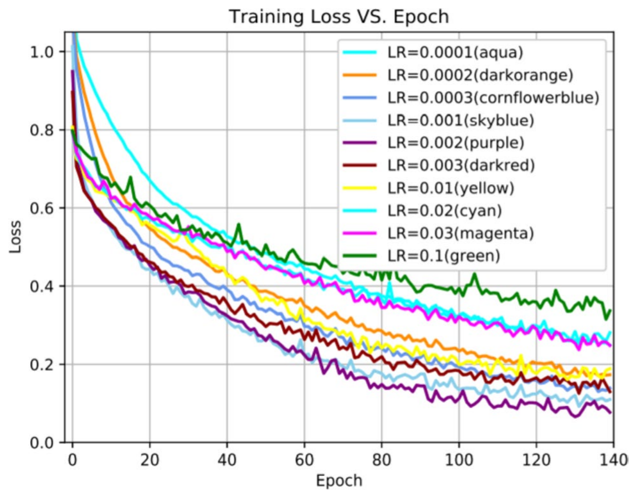


Fig. 9 Training loss with varying learning rate (LR). The *X*-axis corresponds to the epoch number, while the *Y*-axis corresponds to the cross-entropy loss. The lower the cross-entropy loss with a lower number of epochs is, the better the model's performance. Cross-entropy measures the similarity between two distributions; a higher similarity between two distributions corresponds to a lower cross-entropy quantity and becomes zero when two distributions are identical

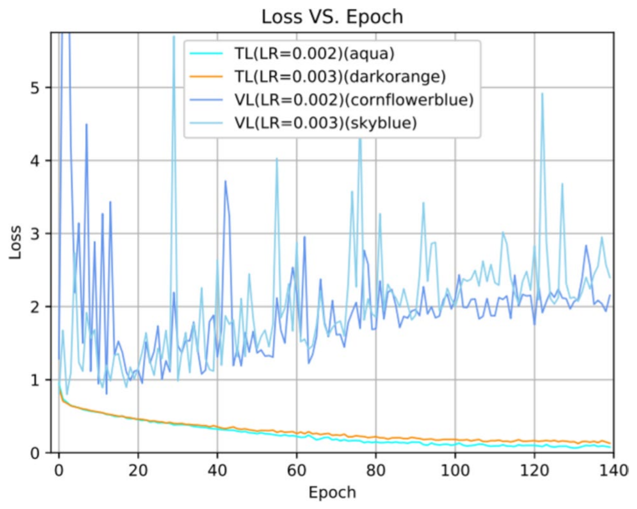


Fig. 11 Training and validation loss were plotted together to determine whether the model was overtting or undertting. TL represents training loss and VL represents validation loss

slowly learns for other learning rates. For the best learning rate group, the model achieves an overall training loss of ap-proximately 0.1 at the end of epoch 140. For the second-best learning rate group, the model gains an overall training loss of between 0.15–0.2, and for the third-best group, it achieves a training loss of approximately 0.25. Note that for lr(=0.3), the model does not learn at all, and the loss curve goes upward instead of downward for almost all epochs. Likewise, almost all loss curves in Fig. 10 follow a similar pattern for the validation set. To observe whether the model is overfitting or underfitting, we plotted training loss and validation loss together over epoch in Fig. 11. We witnessed that our model does not significantly overfit or underfit the future dataset, which confirms the effectiveness of batch normalization to reduce underfitting or overfitting. Therefore, our model does, in fact, generalize towards ground truth quite efficiently for future unseen data.

7.3 Performance comparison with LSTM

To compare the performance of our base model (A-CNN) with other RNN-based models, we chose to replace the third convolution layer with 8 units of LSTM layer into our base model, keeping all other layer parameters exactly the same. In fact, the model capacity of A-LSTM is increased by 15,924 in terms of training parameters. Our base model has a total of 51,740 parameters, of which 51,652 are trainable and 88 are non-trainable. In contrast, the A-LSTM model has 67,576 trainable and 80 non-trainable summing 67,656 parameters in total.

We encompass the classic LSTM cell with components of exterior information. Since many existing approaches have either ignored the problem of multiple target instances or simply used an averaging vector over target expressions (Tang et al. 2015, 2016; Wang et al. 2017). We have employed an affective extension of LSTM (Sentic LSTM) (Ma et al. 2018a, b) to achieve extra information complementary to the textual word sequence usually absent from text. Sentic LSTM assists with the filtering of information owing from one time step to the next and provides complementary information to the memory cell. The hidden output regarding the target does not suffice to represent the sentiment of the whole sentence. To represent a given target by its most informative components our Sentic method weights each target word with an attention weight. After comparing with LSTM, the target-level attention achieves some improvement (not significant), as the target attention is capable of identifying the part of target expressions.

The increased model capacity of the latter, however, is never reflected in the performance in terms of training and validation accuracy, and training and validation loss, as clearly evident from Figs. 12, 13, 14, 15.

7.4 Performance evaluation with CM (Confusion Matrix) and ROC (Receiver Operating Curves) curves

To further investigate and evaluate the performance, we plotted the confusion matrix and receiver operating curves (ROC) with a validation dataset. ROC curves are a very useful technique or tool for evaluating the performance of models trained with unbalanced datasets and examining the performance of specific classes of data. For instance, medical diagnosis detects cancer, which is rare in nature from signatures or symptoms. In our case, measuring the performance of the model by evaluating the validation dataset from each class label (positive, negative and neutral) perspective is justified. From Fig. 16, we observe that our model achieved 90%, 35%, and 7% in predicting negative, positive and neutral sentiment containing comments, respectively. There-fore, our model convincingly

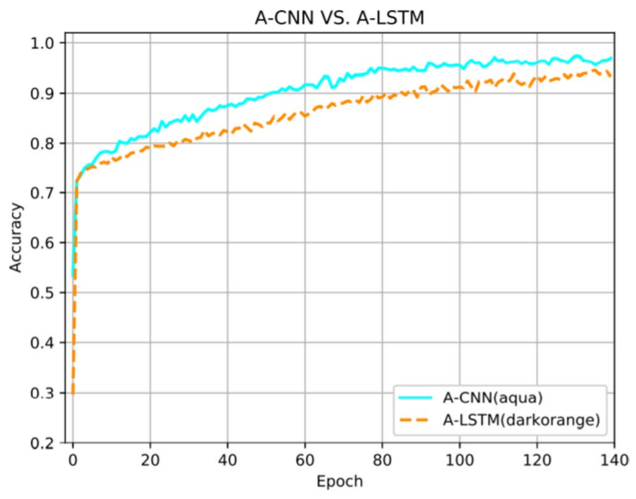


Fig. 12 Training accuracy comparison, **a** our model A-CNN (aqua), **b** A-LSTM (darkorange). For all, an adaptive gradient (NADAM) optimizer with a learning rate of 0.002 was used

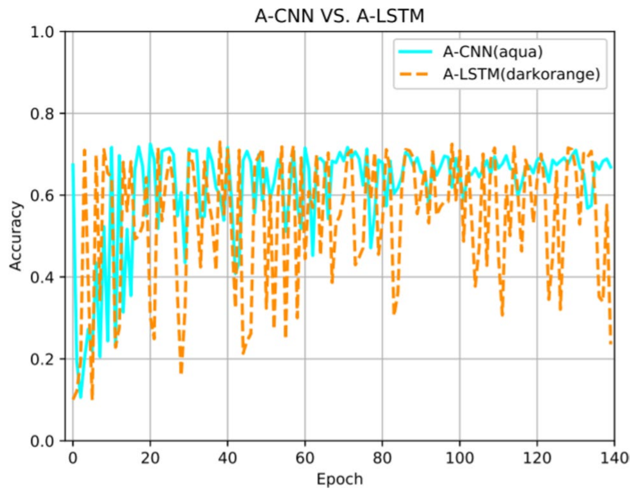


Fig. 13 Validation accuracy comparison, **a** our model A-CNN (aqua), **b** A-LSTM (darkorange). For all, an adaptive gradient (NADAM) optimizer with a learning rate of 0.002 was used

detects negative sentiment bearing comments but struggles to predict or correctly classify positive and neutral sentiment in comments. This is caused by the words in sentences having overlapping connotations and challenges mentioned in Sect. 3. The confusion matrix (CM) in Fig. 17 not only confirms these findings but also tells us that a considerable portion of positive and neutral sentiment bearing comments (61% and 76%, respectively) are predicted as negative and 17% of neutral oriented comments are predicted as positive, which prohibits our model from achieving overall greater performance (nearly 100%). Here, some negative comments are predicted as neutral

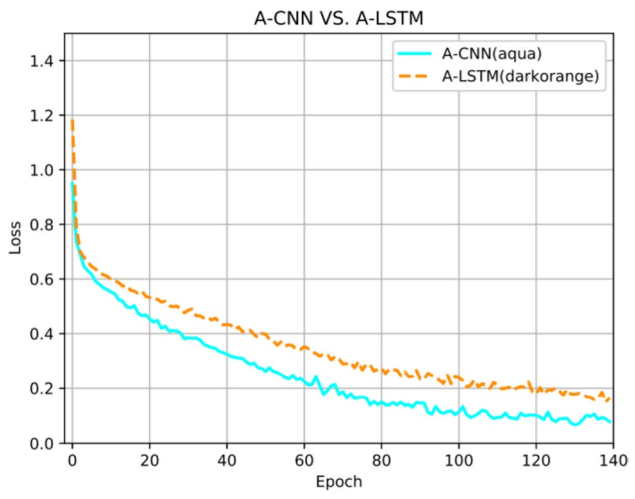


Fig. 14 Training loss comparison, **a** our model A-CNN (aqua), **b** A-LSTM (darkorange). For all, an adaptive gradient (NADAM) optimizer with a learning rate of 0.002 was used

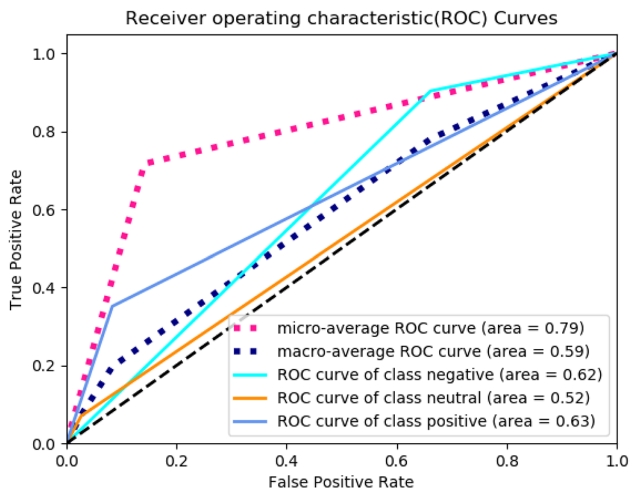


Fig. 16 Receiver operating characteristic curves

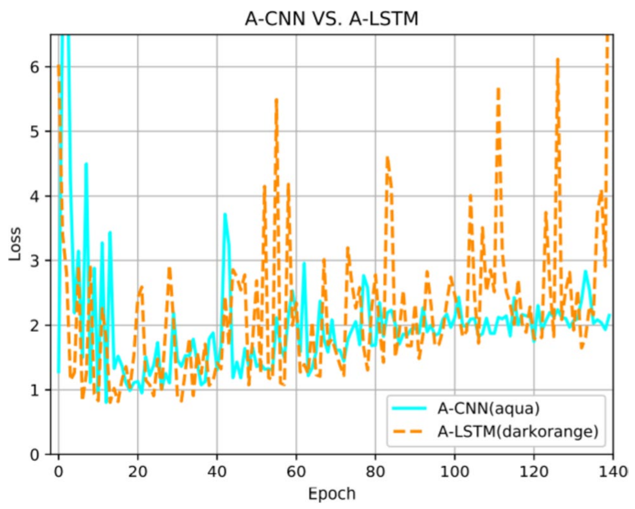


Fig. 15 Validation loss comparison, **a** our model A-CNN (aqua), **b** A-LSTM (darkorange). For all, an adaptive gradient (NADAM) optimizer with a learning rate of 0.002 was used

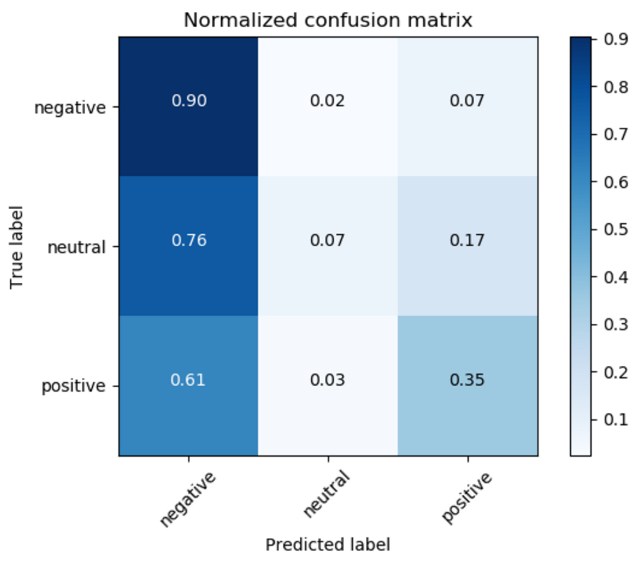


Fig. 17 Normalized Confusion Matrix

(2%) and positive (7%), which is quite disappointing since we are mostly concerned with predicting negative sentiment correctly.

7.5 Cross validation with stratified K-fold

We performed cross-validation to verify how our model performs with the whole dataset into consideration. In the previous experiment, we used two fixed independent set training and validation sets for training and testing, respectively. In this case, we perform stratified tenfold cross validation, which is similar to k-fold cross validation with one difference.

K-fold does not consider class or group distribution while producing k number of folds, but StratifiedKFold maintains an equal percentage of samples for each class in each of the k-fold. In each of the 10 iterations, the training set contains 9 folds, and the test set contains onefold; with 9 folds, we train our model, and at the end of training, we perform evaluation with the remaining onefold. Therefore, we are testing all the datasets each time testing a unique fold against all the other 9 folds. Finally, the average of all 10 round scores is taken as the final score. Our model achieved an accuracy of 66.06% (± 3.44%), precision

Table 1 Cross-validation result with original dataset

Round	Accuracy	Precision	Recall	*F* measure
1	67.66	68.26	67.00	67.62
2	67.89	68.07	67.89	67.98
3	66.10	66.84	65.77	66.28
4	62.08	62.23	61.07	61.63
5	69.46	69.69	69.46	69.57
6	64.76	65.11	64.42	64.76
7	65.65	65.76	65.31	65.53
8	71.71	71.77	71.04	71.40
9	66.32	67.09	65.99	66.53
10	58.92	59.12	58.58	58.85
Mean (μ)	66.06	66.40	65.66	66.02
SD (σ)	3.44	3.45	3.51	3.48

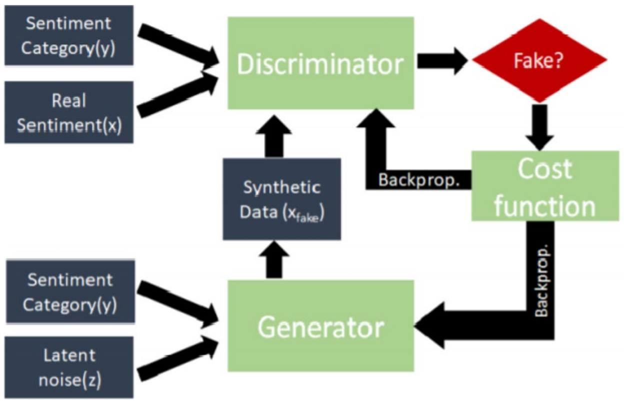


Fig. 18 Overview of synthetic data generation network, conditional generative adversarial network (CGAN)

of 66.40% (±3.45%), recall of 65.66% (±3.51%), and F-measure of 66.02% (±3.48%), as shown in Table 1 in 7.5.

7.6 Boosting performance using synthetic data

Imbalanced datasets are a major obstacle in achieving state-of-the-art performance in any classification task. For example, our dataset contains 72.24% negatives, 19.0% positives, and 8.76% neutral comments. Henceforth, we decided to produce a synthetic dataset using a conditional generative adversarial network (CGAN) (Mirza and Osindero 2014) from a family of GAN networks. We implemented the CGAN network using the Keras Dense and BatchNorm layers with the LeakyRelu, sigmoid and tanh activation functions and trained both the generator and discriminator networks using our real dataset to optimize KL divergence as an objective function before generating the synthetic dataset. An overview of the synthetic data generation network is shown in Fig. 18. We generated a total of 2979 instances that consist of 50% negative, 25% neutral, and 25% positive comments. Using this generated dataset, we trained our model (A-CNN) with a 75% training set and validated it with a 25% validation set. We performed experiments several times and noticed that every time, the performance of our model was enhanced by a considerable margin using the synthetic dataset.

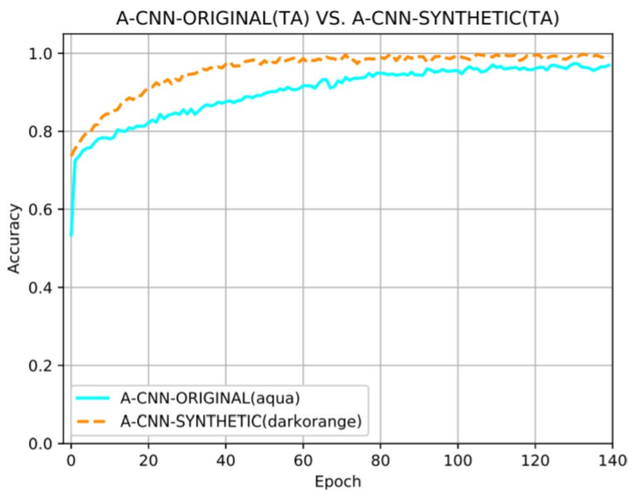


Fig. 19 A-CNN model's training accuracy comparison, using **a** the original dataset (aqua), **b** the synthetic dataset (darkorange). For two, the NADAM optimizer with a learning rate of 0.002 was used

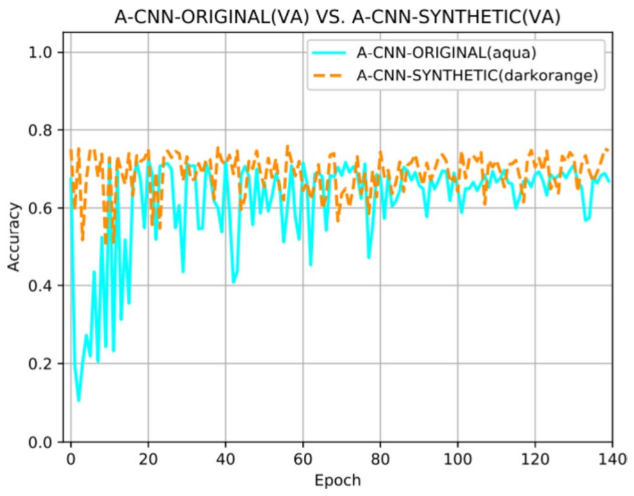


Fig. 20 A-CNN model's validation accuracy comparison, using **a** the original dataset (aqua), **b** the synthetic dataset (darkorange). For two, the NADAM optimizer with a learning rate of 0.002 was used

The performances of our model under the original and synthetic datasets are given in Figs. 19, 20, 21 and 22 in terms of the accuracy and cross-entropy loss of both the training and validation datasets.

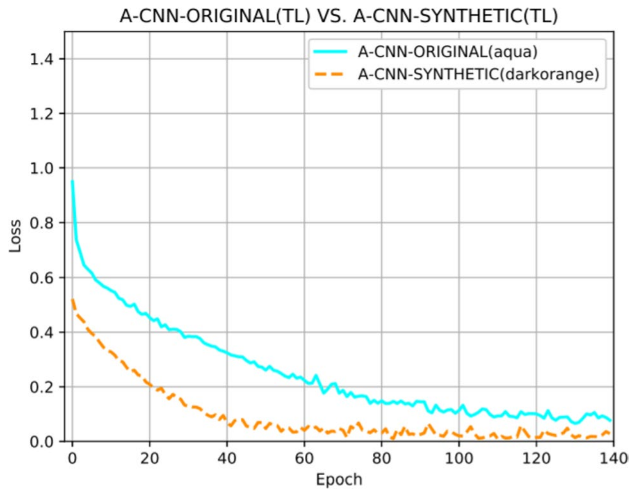


Fig. 21 A-CNN model's training loss comparison, using **a** the original dataset (aqua), **b** the synthetic dataset (darkorange). For two, the NADAM optimizer with a learning rate of 0.002 was used

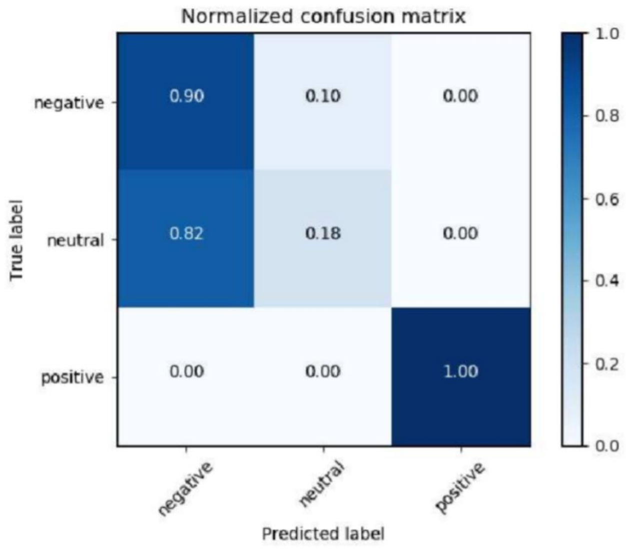


Fig. 23 Samples reviews and comments

Even though our A-CNN model struggles to correctly predict neutral sentiment, as seen from the confusion matrix in Fig. 23, the overall correct identification or prediction rate has in-creased by approximately 6.06% compared with the original real dataset, as seen from the cross-validation result in Table 2 in 7.6.

7.7 Summary of experimental results

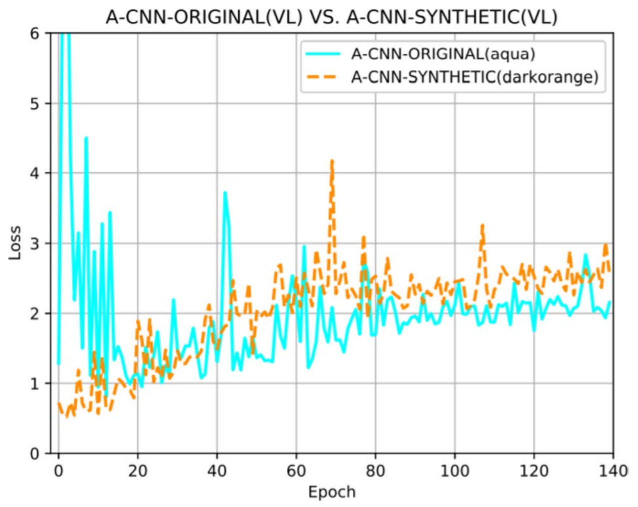


Fig. 22 A-CNN model's validation loss comparison, using **a** the original dataset (aqua), **b** the synthetic dataset (darkorange). For two, the NADAM optimizer with a learning rate of 0.002 was used

In summary, the experimental results show that our model (A-CNN) achieves the best performance using the NADAM optimizer with a learning rate of 0.002. In addition, the attention mechanism in CNN (A-CNN) performs better than A-LSTM. Although training accuracy achievement at the end of 140 epoch is quite impressive at approximately 96.55% but fails to maintain that performance in terms of overall performance matrices: accuracy (66.06%), precision (66.04%), recall (65.66%), and F-measure (66.02%). This is due to the noise and data imbalance problems associated with the dataset that prevent us from achieving satisfactory performance. Hence, we tried to reduce noise and data imbalance problems by adopting the CGAN network for synthetic data generation, which proved to be effective, as shown by the performance boost of the A-CNN network by approximately 6.06% under the synthetic balanced dataset, resulting in a training accuracy at the end of 140 epochs of approximately 99.32% and an overall validation accuracy of 72.06%.

8 Conclusion and future works

Sentiment analysis has played a crucial role for IT-enabled business and service owners due to the automatic analysis of user reviews and opinions about goods and services. Furthermore, with current sentiment analyzers, it is now possible to understand user activities and choices. Many works have already been performed for English languages. In contrast, work performed in Bangla is not significant enough and has shown little success in predicting textual sentiment. This empirical study about Bangla sentiment analysis is a small step forward to fill the void. Bengali lacks both benchmark datasets and a well-furnished model for sentiment analysis

Table 2 Cross-validation result with synthetic dataset

Round	Accuracy	Precision	Recall	*F* measure
1	70.90	70.90	70.90	70.90
2	76.25	76.25	76.25	76.25
3	74.24	74.24	74.24	74.24
4	68.90	68.90	68.90	68.90
5	74.16	74.16	74.16	74.16
6	73.40	73.40	73.40	73.40
7	70.71	70.71	70.71	70.71
8	71.04	71.04	71.04	71.04
9	71.71	71.71	71.71	71.71
10	69.93	69.93	69.93	69.93
Mean (μ)	72.12	72.12	72.12	72.12
SD (σ)	0.021	0.021	0.021	0.021

despite being one of the most used languages in the world. Moreover, researchers usually do not publish their datasets. The dataset that was made for this research is a step ahead since it can be further enriched and published for research purposes. Al-though our model (A-CNN) achieves satisfactory performance in terms of performance metrics compared to other works, it can still be improved to a higher performance than now if word sense ambiguity can be removed, and word sense semantics can be embedded in the scoring of words rather than term presence or term frequency. We will try to employ semantic meaning for each word and avoid word sense ambiguity as much as possible in the future. Therefore, using the attention mechanism in CNN-based deep learning models, it is possible to achieve relatively better performance in Bangla sentiment analysis. In the future, it will be interesting to see the business applications or sentiment analyzers for Bengali text using attention in deep CNN models. In conjunction with attention-based CNN (A-CNN), a mixture of other high-performing techniques can also be applied to the task of sentiment classification and opinion mining.

References

Al-Amin MT, Islam MS, Uzzal SD (2017) Sentiment analysis of bengali comments with word2vec and sentiment information of words. In: 2017 international conference on electrical, computer and communication engineering (ECCE), 186–190

Bahdanau D, Cho K, Bengio Y (2015) Neural machine translation by jointly learning to align and translate. In: 3rd international conference on learning representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, conference track proceedings. URL https://arxiv.org/abs/1409. 0473

Balahur A, Turchi M (2012) Multilingual sentiment analysis using machine translation? In: Proceedings of the 3rd Workshop in Computational Approaches to Subjectivity and Sentiment Analysis, p 52–60

Bhowmick PK (2009) Reader perspective emotion analysis in text through ensemble based multi-label classification framework. Comput Inf Sci 2:64–74

Cambria E (2016) Affective computing and sentiment analysis. IEEE Intell Syst 31:102–107. https://doi.org/10.1109/MIS.2016.31

Chowdhury S, Chowdhury W (2014) Performing sentiment analysis in bangla microblog posts. In: 2014 international conference on informatics, electronics & vision (ICIEV). IEEE, May 2014. 10.1109/ICIEV.2014.6850712

Cui H, Mittal V, Datar M (2006) Comparative experiments on sentiment classification for online product reviews. In: Proceedings of the 21st National Conference on Artificial Intelligence, Vol 2, AAAI'06, pp 1265–1270. AAAI Press, 2006. ISBN 978-1-57735-281-5. URL http://dl.acm.org/citation.cfm?id=1597348.1597389

Das A, Bandyopadhyay S (2010) Sentiwordnet for bangla sentiwordnet for bangla

Das SRD, Bandyopadhyay S (2012) Emotion tracking on blogs-a case study for bengali. In: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems., page 447456, Berlin, Heidelberg, 2012. Springer. https://doi.org/10.1007/978-3-642-31087-4_47

dos Santos C, Gatti M (2014) Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 69{78, Dublin, Ireland. Dublin City University and Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C14–1008

Dozat T (2015) Incorporating nesterov momentum into Adam

Ekman P (1992) An argument for basic emotions. Cognition and Emotion, pages 169–200

Esuli A, Sebastiani F (2006) Sentiwordnet: A publicly available lexical re-source for opinion mining. In: Proceedings of the 5th conference on language resources and evaluation (LREC06), pp 417–422

Feldman R (2013) Techniques and applications for sentiment analysis. Commun. ACM, 56(4):82–89. ISSN 0001–0782. doi:10.1145/2436256. 2436274. http://doi.acm.org/10.1145/2436256.2436274

Hasan KMA, Rahman MMU, Badiuzzaman (2014) Sentiment detection from bangla text using contextual valency analysis. In: 2014 17th international conference on computer and information technology (ICCIT), pp 292–295

Hassan A, Mohammed N, Azad AK et al (2016) Sentiment analysis on bangla and romanized bangla text (BRBT) using deep recurrent models. CoRR, abs/1610.00369, 2016. URL http://arxiv.org/abs/1610.00369

Ioe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd international conference on machine learning, ICML 2015, Lille, France, 6–11 July 2015, pages 448–456, 2015. URL: https://jmlr.org/proceedings/papers/v37/ioffe15.html

Islam MS, Islam MA, Hossain MA, Dey JJ (2016) Supervised approach of sentimentality extraction from bengali facebook status. In: 2016 19th international conference on computer and information technology (ICCIT), pp 383–387, 2016

Jack RE, Garrod OGB, Schyns PG (2014) Dynamic facial expressions of emotion transmit an evolving hierarchy of signals over time. Curr Biol 24:187–192

Jiang L, Yu M, Zhou M, Liu X, Zhao T (2011) Target-dependent twitter sentiment classification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Volume 1, HLT '11, pp 151–160, Stroudsburg, PA, USA, 2011. Association for Computational

Linguistics. ISBN 978-1-932432-87-9. https://dl.acm.org/citation.cfm?id=2002472.2002492

Kennedy A, Inkpen D (2006) Sentiment classification of movie reviews using contextual valence shifters. Comput Intell 22:110–125

Kim Y (2014) Convolutional neural networks for sentence classification. CoRR, abs/1408.5882. URL http://arxiv.org/abs/1408.5882

Lo S, Cambria E, Chiong R, Cornforth D (2016) Multilingual sentiment analysis: from formal to informal and scarce resource languages. Artif Intell Rev. https://doi.org/10.1007/s10462-016-9508-4

Luong M, Pham H, Manning CD (2015) Effective approaches to attention-based neural machine translation. CoRR, abs/1508.04025. URL https://arxiv.org/abs/1508.04025

Ma Y, Peng H, Cambria E (2018a) Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In: The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)

Ma Y, Peng H, Khan T, Cambria E, Hussain A (2018) Sentic lstm: a hybrid network for targeted aspect-based sentiment analysis. Cogn Comput. https://doi.org/10.1007/s12559-018-9549-x

Mathews D, Abraham S (2019) Social data sentiment analysis of a multilingual dataset: a case study with malayalam and english, p 70:78. 09 2019. ISBN 978-981-15-0107-4. doi: 10.1007/978-981-15-0108-18

Mikolov T, Chen K, Corrado GS, Dean J (2013) Efficient estimation of word representations in vector space. CoRR, abs/1301.3781

Mirza M, Osindero S (2014) Conditional generative adversarial nets. CoRR, abs/1411.1784. URL https://arxiv.org/abs/1411.1784

Pang B, Lee L (2008) Opinion mining and sentiment analysis. Found Trends Inf Retr, 2(1–2):1–135, ISSN 1554–0669. 10.1561/1500000011

Pang B, Lee L, Vaithyanathan S (2002) Thumbs up?: Sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing, Volume 10, EMNLP '02, pages 79–86, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics. 10.3115/1118693.1118704

Paul AK, Shill PC (2016) Sentiment mining from bangla data using mutual information. In: 2016 2nd international conference on electrical, computer & telecommunication engineering (ICECTE), pp 1–4

Shirani-Mehr H (2014) Applications of deep learning to sentiment analysis of movie reviews. Technical report, Stanford University

Singh VK (2015) Sentiment analysis research on Bengali language texts. Int J Adv Sci Res Dev 02:122127

Souza M, Vieira R (2012) Sentiment analysis on twitter data for portuguese language. In Proceedings of the 10th International Conference on Computational Processing of the Portuguese Language, PROPOR12, page 241247, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 9783642288845. doi: 10.1007/978-3-642-28885-2 28

Sutskever J, Martens J, Dahl GE, Hinton GE (2013) On the importance of initialization and momentum in deep learning. In Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16–21 June 2013, pages 1139–1147. URL http://jmlr.org/proceedings/papers/v28/sutskever13.html

Tang D, Qin B, Feng X, Liu T (2015) Target-dependent sentiment classification with long short term memory. 12

Tang D, Qin B, Liu T (2016) Aspect level sentiment classification with deep memory network. CoRR, abs/1605.08900. URL http://arxiv.org/abs/1605.08900

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. CoRR, abs/1706.03762, 2017. URL https://arxiv.org/abs/1706.03762

Wang B, Liakata M, Zubiaga A, Procter R (2017) TDParse: Multi-target-specific sentiment recognition on twitter. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pp 483–493, Valencia, Spain. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/E17-1046

Xu K, Ba J, Kiros R, Cho K, Courville AC, Salakhutdinov R, Zemel RS, Bengio Y (2015) Show, attend and tell: Neural image caption generation with visual attention. In: Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6–11 July 2015, pages 2048–2057, 2015. URL http://jmlr.org/proceedings/papers/v37/xuc15.html

Zhang L, Liu B (2017) Sentiment Analysis and Opinion Mining, pages 1152–1161. Springer US, Boston, MA, 2017. ISBN 978-1-4899-7687-1. doi: 10.1007/978-1-4899-7687-1907

Zhang Y, Wallace BC (2015) A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. CoRR, abs/1510.03820. URL https://arxiv.org/abs/1510.03820