Customer Complain Detection in E-commerce Platforms using NLP

Fahmida Sabur Tasmia
Department of Computer Science and Engineering
Leading University

Arafat Habib Quraishi
Lecturer, Department of Computer Science and Engineering
Leading University

ABSTRACT
Online shopping has gained significant popularity in recent years. However, one major issue with online shopping is that buyers cannot physically inspect the products and so they have to rely on user reviews. In this paper, we construct a novel dataset for online reviews in the low-resourced Bengali language. Moreover, we conduct extensive experiments with strong deep learning-based baselines to benchmark the performance of such models in our dataset. We have applied RNN and Fast Text DNN to read customer's feedback and identify areas of dissatisfaction and satisfaction. We have found that Fast Text DNN performed better then RNN with an accuracy of 74%.

Keywords
Complain Detection, Online Product Review, E-commerce, NLP, Deep Learning.

1. INTRODUCTION
E-commerce has been incredibly popular in the commercial sector in recent years. Moreover, the COVID-19 pandemic has permanently altered online buying habits. People prefer to make purchases online rather than go to a physical store. That is why most businessmen are now focusing on online business. In the case of online shopping, customers typically enjoy the products after reading the reviews. So, to sustain this market of e-commerce, we must first ensure customer satisfaction. However, among the millions of reviews for thousands of products, it is difficult to identify where people complain or where they are satisfied. Even if it is achievable, it would need a large amount of manpower and time, both prohibitively expensive. Yet, if this problem could be handled by an automated model that analyzes customer reviews and reports on where the customers complain and where the customers are satisfied, that would be fantastic.

We attempted to use Natural Language Processing (NLP) to address the abovementioned concerns. For our study, we explored two deep learning-based models (RNN, and Fast Text DNN). We have created our own data sets based on real people's sentiments. To understand the types of sentiments expressed, we have categorized the text into three categories: "complain", 'Critical Complain', and "satisfied". We have prioritized the Bangla language in our study since, as we all know, Bengali is one of the world's most widely spoken languages, yet there has been relatively little research on it. On the contrary, while writing online reviews, we Bengalis prefer to use phonetic Bangla. This has motivated us to generate our data set by combining Bangla Phonetic Bangla and English text.

In this study, we used a variety of evaluation matrices, including accuracy, precision, recall, and f1-score, to measure the validity of our suggested models. Finally, we obtained 0.74% accuracy with the Fast Text (DNN) model.

2. RELATED WORK
Recent years have seen a proliferation of works devoted to natural language processing (NLP), with particular focus from some Bangladeshi researchers on resolving issues with Bangla text. A limited amount of research has been done on the topic of online retail. In this section, we take a look at some of the most recent and groundbreaking research in the field of Bangla natural language processing.

In [1], the authors deal with six individual emotion classes happy, sad, tender, excited, angry, and scared. Later, explored a Topical approach to extract the emotions from the Bangla text. To reach the goal they used Naive based classification algorithm which achieved above 90% accuracy. In [2], the authors created a dataset from Bengali news portal 'ProthomAlo'. While collecting data they focused on Bangladesh, its economy, opinion sports. For sentiment analysis, they work with fie category classes-(positive, strongly positive, negative, strongly negative, and neutral), applied three different ML algorithms (SVM, LSTM, CNN), and achieved their best (74.74%) accuracy by LSTM. In Another work [3], the authors initiate Natural Language (NLP) based way to deal with upgrading the sentiment classification by adding semantics in feature vectors and accordingly utilizing ensemble techniques for classification.

The authors in [4], try to show the taxonomy of various sentiment analysis methods. AND They additionally show that a support vector machine(SVM) gives high precision thought to Naive Bayes and most extreme entropy techniques. In [5], the authors intended to classify product reviews by analyzing the sentiments and building a model that can predict product quality. They made their own data set by collecting data samples from the Bangladeshi most popular online shopping platforms. To extract features from the text, they applied the FastText pretrained model. They built two deep neural network models and got 0.69 and 0.81 accuracy scores to classify the reviews. In [6], the authors attempt to present that the bilingual approach can play a crucial role in sentiment analysis. They used two Bengali data sets, distinct features sets (unigram, bigram), and multiple machine learning algorithms. In [7], the authors gathered a dataset (ABSA) about cricket comments in Bangla text of genuine individual opinions and categorized them into positive, negative, neutral. They utilized the word embedding method for vectorization of each word and used LSTM for long-term dependencies which reached 95% accuracy. In [8], the authors present an automated Bengali text sentiment analysis for restaurant reviews to classify an opinion into two classes: positive or negative sentiment. To achieve their goal, they used Three

machine learning algorithms (decision tree, random forest, and multinomial na¨ ıve Bayes) and obtained their best accuracy of 80.48% by multinomial na¨ ıve Bayes. In [9], Here authors' fundamental goal is with a purpose to train the computer to read and understand an English sentence typed by a person to be able to classify it as a positive or negative feeling. To assemble their model, they use the NLTK dataset and text mining proce- dure to generate and process the variables. In [10], the authors collect Twitter data and preprocess their data using Tokenization, Lemmatization, removal of stop words, URL,@mention, punctuations, and special characters. They used supervised machine learning algorithms (Naive Bayes, Multinomial NB, Bernoulli NB, Logistic Regression, LinearSVC classifir) to classify their tweets into positive and negative compared them in terms of accuracies. The author obtained their best accuracy by the Naive Bayes Algorithm. Their work accuracy was 99.73% and article accuracy was 79.12%. In [11] In this paper, the author's applied eight classifirs namely, Naive Bayes, Decision Tree, Random Forest, Ripple Rule Learning, KNearest Neighbours, support vector classifier, Bayes Net, Stochastic Gradient Descent on their IMDB movie reviews dataset. To measure the performance of these eight classifiers they used fie different evaluation metrics, which are, precision, Recall, Accuracy, Area Under Curve(AUC), and F-1measure in their research, they gain their best results by Random Forest(RF) and found their worst result by Ripple Rule Learning(RRL).

3. DATASET DESCRIPTION

For this study, we have dealt with a dataset based on the real world. We gathered data from numerous well-known Bangladeshi e-commerce sites, including Daraz, BDShop, Picaboo, and AjkerDeal, as well as data from various electronics F-commerce group comments on Facebook, most notably from Samsung Bangladesh, Realme, Walton, Singer, and Vision. An interesting feature of this dataset is that it contains Bangla, phonetic Bangla, and English text. The inclusion of three distinct types of text in the dataset was done to make the results of the research more applicable to a wider audience. This dataset has been formed of 4089 valid, understandable reviews. This is a pretty balanced dataset that contains the sentiment of customers. To understand the types of sentiments expressed, we have categorized the text into three categories: "complain", "critical complain", and 'Satisfied'. The counting analysis of class distribution in our collected dataset is illustrated in fig -1:

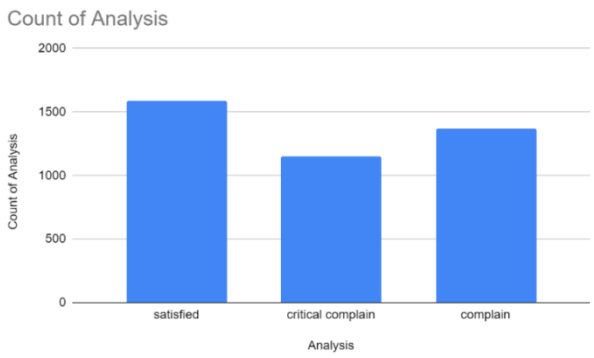

Fig 1: The counting analysis of class distribution in our collected dataset

4. DATA PREPROCESSING

Generally, real data is messy and often contains a large number of unwanted errors, unnecessary attributes, and duplications that directly affect the performance of machine learning and deep learning models. To ensure proper training of the model, we have deleted unwanted elements like stop words, punctuation, and unwanted characters from the data set. Next, to know the efficiency of our model, we have split our dataset into two parts: 1. Training dataset. 2. Testing Data set. For the training data set, we kept 90% of the data for training, and for the testing data set, we kept 10% of the data.

Since textual data cannot be processed by deep learning networks. In order to use the features in our model's training, we must first extract them from raw text. We've implemented a pre-trained model for the Bangla language, available through FastText. The Facebook Research Team [24] created FastText, a set of tools that provide shaped (1,300) word embeddings for each word. Furthermore, we utilized tokenization, text-to-sequence, and padding to extract features for the RNN model. In tokenization, we split the sentence into tokens and create a vocabulary of 6,000 most frequently used words. In text to the sequence, we first converted each sentence into a sequence of words, then we replaced each word in the sequence with its matching integer value. We added padding to equalize sentence length. We have a hard limit of 40 words for each sentence, so we'll never go beyond that. In order to compensate, we added 40 to the padding's length.

For the RNN model, we extracted features by using tokenization, text-to-sequence, and padding. In text-to-sequence, we have transformed each sentence into a sequence of words, then replaced every word with its corresponding integer value.

5. METHODOLOGY

We split the methodology section of the research study into two parts. At the outset, we mapped out the architecture for our suggested models. Next, we evaluated the accuracy of the model's predictions.

5.1 DNN Model

The layers in a DNN are as follows: input, hidden, and output. There is a direct correlation between the number of neurons and the number of hidden layers in a DNN model's performance. In our work, we have defined 300 neurons in the input layer. The number of neurons in the output layer for classification is based on the number of distinct classes. Since our data set is not particularly large, we chose three hidden layers for our model to obtain the best result. Our model made use of a three-layer neural network, with the first hidden layer consisting of 1024 neurons, the second of 512, and the third of 256. Our views for the DNN Architecture are shown in figure (2):

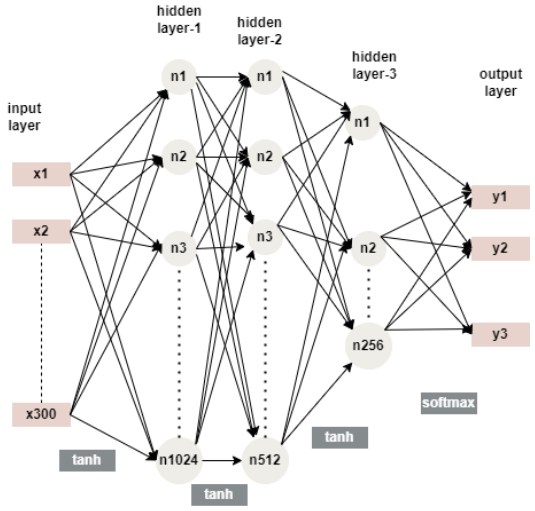

Fig 2: DNN Architecture

5.2 RNN Model

In the field of Natural Language Processing, Recurrent Neural Networks (RNNs) have become increasingly prevalent. They have hidden states that permit prior output to be repurposed as inputs. The RNN's output depends on the computations that came before it, but the goal of each part of the sequence is the same. An input layer, embedding layer, flatten layer, connected layers, and output layers are typical parts of an RNN network. Figure (3) depicts our proposed RNN architecture.

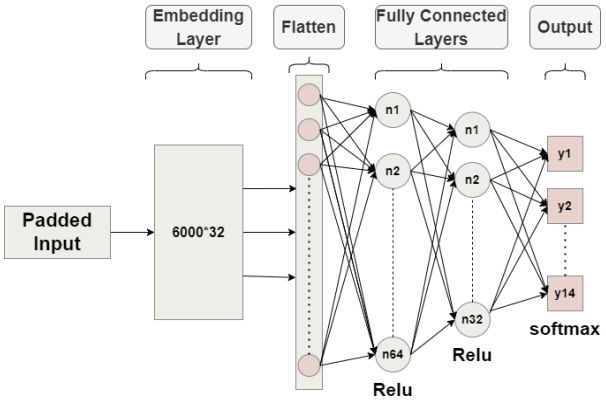

Fig 3: RNN Architecture

5.3 Embedding Layer

To generate word embeddings from padded input, we mapped 32 characteristics to each word. It produced a 6000*32-dimensional matrix. 32 features make each word important for sentiment analysis. We have implemented flattening layers to reduce this 2D matrix to 1D.

5.4 Hidden Layers

For our RNN model, we decided to use two hidden layers. Between the input and output layers is a hidden set of layers. We settled on a size of 64 neurons for the primary hidden layer and a size of 32 neurons for the secondary. Instead, the model's hidden layers were purposefully selected to avoid adding unnecessary complexity.

5.5 Activation function

The capabilities and results of a neural network are greatly affected by the chosen activation functions. Various activation functions are required for use with the various model layers. Here is a quick rundown of how we put those three activation functions to use

5.6 Relu

One of the most used activation functions for hidden layers is relu, which stands for rectified linear activation function. Since Relu activation functions are not affected by vanishing gradients, we decided to employ them in our hidden layer to help us get over the restrictions imposed by other activation functions (such sigmoid, tanh, erf, and so on). To express Relu mathematically, we have:

$$f(x) = \max\{0, x\}$$

5.7 Softmax function

In the hidden layers of multi-class taxonomy-managing neural network models, the softmax activation function is used to generate classification decisions. For this purpose, we have used this feature of our product review classification model to estimate the likelihood of occurrence for each of the three categories. The softmax function accepts scores of any kind and returns the probability that corresponds to them. A formal definition of softmax is:

$$softmax\ (x_i) = \frac{exp(\varkappa_i)}{\Sigma_j\ exp(\varkappa_j)}$$

5.8 Tanh

It is well known that Tanh is a non-linear activation function. The sigmoid range of the tanh activation function is [-1, 1]. Since the vectorized version of our training dataset contained both negative and positive values in the range of -1 to 1, we figured to use the tanh activation function in both the input and hidden layers. Expressed mathematically, the tanh function is:

$$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$$

5.9 Optimization Algorithm

Optimization algorithms are crucial for deep learning since they connect the model by updating the weights in the most effective way. Adam, RMSprop, SGD, and many other optimization methods exist. In our approach, we apply Adam optimization with a learning rate of 0.1. Adam's symbolic representation in mathematics is:

$$\theta_{n+1} = \theta_n - \frac{\alpha}{\sqrt{\vartheta_{\hat{n}}} - \varrho} m\hat{n}$$

5.10 Evaluation Metrix

We have utilized multiple evaluation matrices including accuracy, precision, recall, and F1 Score to verify the efficacy of our models' categorization abilities. By adopting the following formulas, we have produced the confusion matrix for both models and determined its value.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = 2 * \frac{Recall * Precision}{Recall + Precision}$$

6. RESULT AND ANALYSIS

At this point, we've finished analyzing the data to see how well our proposed models work. To better comprehend the results of our models in each class, we have prepared a classification report and confusion matrix. There are two digits of precision in our record of the result. Afterward, we analyzed the model's actual performance by making a prediction on a test dataset.

RNN model: We trained our model for 40 epochs to obtain the training and validation curve. The results of the training and validation rounds are shown graphically in figure (4.1).

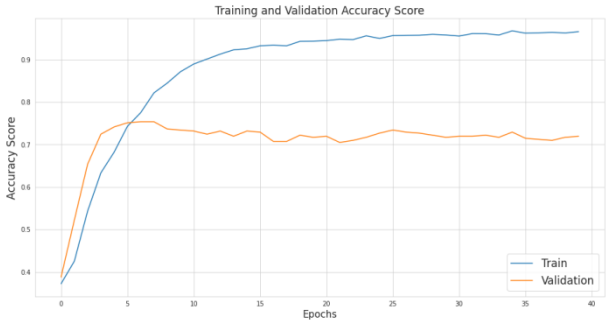

Fig 4: The results of the training and validation rounds

Table 1. The accuracy, training, and validation scores are displayed in table.

Dataset	Accuracy
Training	0.96
Test	0.71

Table 2. Classification report of OUR RNN model

Class Level	F1-score
Satisfied	0.84
Complain	0.45
Critical Complain	0.44

Similarly, we have built the confusion matrix, which is displayed in the figure, to help us comprehend the true and false classification behavior of our model (5). In figure (5), we can see that our model correctly predicted 167 out of 158 instances of satisfied data, 118 out of 115 instances of critical complain data, and 125 out of 137 instances of complain data.

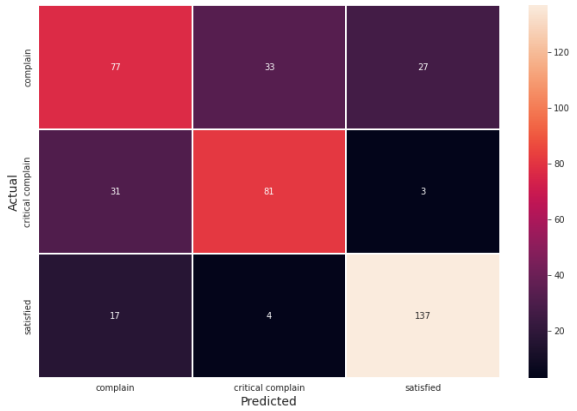

Fig 5: Confusion matrix of OUR RNN model

DNN model: In order to obtain the Training and Validation Curve used to create the Learning Graph, we trained our model for 60 epochs. Loss versus epoch is shown in Fig. (5), and the losses in both training and validation are displayed in Fig. (5.1). Detailed results for the complain detection model's accuracy during training, validation, and testing are provided in table (4.3). The fact that the validation accuracy (=0.74) is greater than the training accuracy (=0.69) shows that our model is not over-fit and was trained correctly.

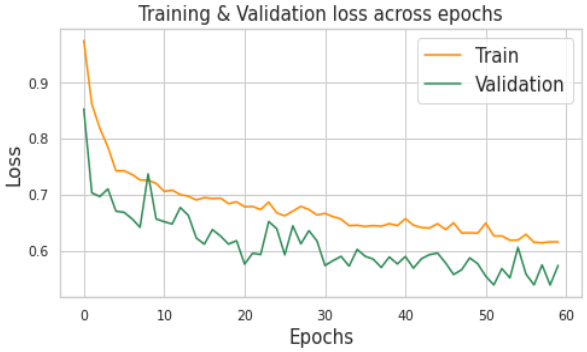

Fig 6: Loss Vs Epochs graph for complain detection

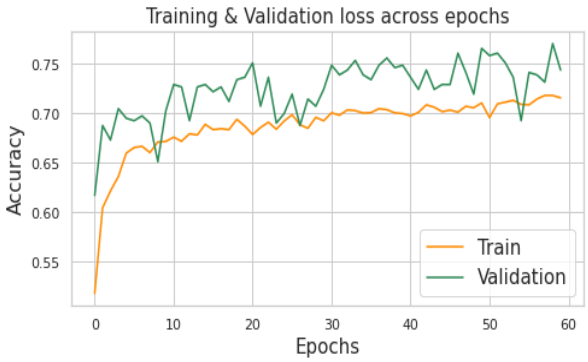

Fig 7: Accuracy Vs Epochs graph for complain detection

Table 3. The accuracy score of DNN model:

Dataset	Accuracy
Training	0.72
Validation	0.74
Test	0.74

Table 4. Classification report of DNN Model:

Class Level	F1-score
Satisfied	0.86
Complain	0.66
Critical Complain	0.69

Confusion matrix:

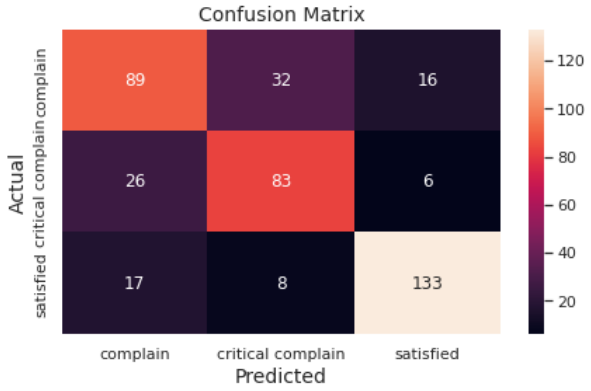

Fig 8: Confusion matrix of DNN Model

Our model did a great job of predicting outcomes, as shown in figure (6), with a satisfaction rate of 155 out of 158, a critical complaint rate of 123 out of 115, and a complaint rate of 132 out of 137.

7. CONCLUSION

For the purposes of this thesis, we employ two different types of deep learning models (RNN and Fast Text DNN) to identify customer complaints on online stores. We have collected the data from a variety of sources, including the most popular Bangladeshi online marketplaces and forums dedicated to the f-commerce industry. First, we used techniques for data preparation to clean and organize the data. Then, we used feature extraction to derive numerical features from the textual information. In an attempt to develop our model, we have adopted the tried and reliable methods of several carefully designed studies. As evaluation criteria, we used the accuracy matrix, precision, recall, and f1-score. In order to verify that the training phase was successful, we plotted the loss against the number of training epochs. As a result of our efforts, we were able to achieve an impressive accuracy of 0.74. Our long-term goal is to add even more data and compare it to other NLP systems in other provinces.

8. REFERENCES

[1] M. H. a. S. B. F. Jemai, "Sentiment analysis using machine learning algorithms," IEEE, p. 775–779, 2021.

[2] M. K. a. R. M. R. Guddeti, "Nlp based sentiment analysis on twitter data using ensemble classifirs," IEEE, pp. 1-5, 2015.

[3] B. K. P. F. N. M. A. a. A. K. D. R. A. Tuhin, "An automated system of sentiment analysis from bangla text using supervised learning techniques," IEEE, p. 360–364, 2019.

[4] S. S. a. S. H. M. A.-U.-Z. Ashik, "Data set for sentiment analysis on bengali news comments and its baseline evaluation," IEEE, pp. 1-5, 2019.

[5] T. S. a. J. Shetty, "Sentiment analysis of product reviews: a review," IEEE, pp. 298-301, 2017.

[6] N. I. T. a. M. E. Ali, "Detecting multilabel sentiment and emotions from bangla," IEEE, pp. 1-6, 2018.

[7] S. S. a. S. Jayarathna, "A sentiment classifiation in bengali and machine translated english corpus," IEEE, p. 107–114, 2019.

[8] M. J. H. a. M. S. A. M. F. Wahid, "Cricket sentiment analysis from bangla text," IEEE, pp. 1-4, 2019.

[9] M. M. H. a. E. H. O. Sharif, "Sentiment analysis of bengali texts on online," IEEE, p. 1–6, 2019.

[10] M. H. a. S. B. F. Jemai, "Sentiment analysis using machine learning algorithms," IEEE, p. 775–779, 2021.

[11] M. Y. a. S. Tedmori, "Movies reviews sentiment analysis and classifiation," IEEE, p. 860–865, 2019.