Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 230 (2023) 584–597

Procedia Computer Science

www.elsevier.com/locate/procedia

3rd International Conference on Evolutionary Computing and Mobile Sustainable Networks (ICECMSN 2023)

Enabling an On-demand Access to Community Sentiments using LSTM RNNs Web Service Architecture

Prabakaran.N [a], A.Anbarasi[b], N.Deepa[c], Pandiaraja P [d*]

[a] *School of Computer Science and Engineering , Vellore Institute of Technology,Vellore,Tamilnadu , India – 632014.*
[b] *Department of computing Technologies ,School of Computing , SRM Institute of Science and Technology, Kattankulathur, Chennai,603203.*
[c] *Department of Networking and Communications ,School of Computing , SRM Institute of Science and Technology, Kattankulathur, Chennai,603203.*
[*d] *Department of Computer Science and Engineering, M.Kumarasamy college of Engineering, Karur, India -639113*

dhoni.praba@gmail.com, anbarasa3@srmist.edu.in , deepan3@srmist.edu.in , sppandiaraja@gmail.com.

Abstract

Analyzing community response has always played an important role in marketing and development of various products ranging from services to manufacturing. Analyzing the responses and interpreting them helps improve the quality of the product. This task is traditionally done by community managers using surveys and focus groups. One other hands-off solution would be using the technology to analyze sentiments from users' digital interactions in various online communities. Sentiment Analysis is the use of Natural Language Processing to process text to identify and retrieve sentiments behind the textual data. Sentiment Analysis can be approached either by supervised classification where a machine learning model is trained using labeled data or lexicon-based unsupervised analysis where a sentiment dictionary is used to find the overall sentiment. The idea is to build a system which presents community sentiment results on discussion threads from reddit on demand with the end user not to concern about data processing. We propose a streamlined pipeline architecture for a web application with several closely connected components in the backend that allows users to simply send the reddit discussion thread's URL, which the backend receives and scrapes the comments using the Reddit API, creates a dataset, cleans it and performs sentiment analysis using a RNNs. We used a Long Short-Term Memory (LSTM) a variant of RNN like Bidirectional RNNs and Multi-layered RNNs are also used to get better results.

Peer-review under responsibility of the scientific committee of the 3rd International Conference on Evolutionary Computing and Mobile Sustainable Networks

Keywords: Sentiment Analysis, Community Response Analysis, Social Media Mining, Monitor sentiments, Sentiment analysis architecture, LSTM RNN, On demand architecture.

1877-0509
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the 3rd International Conference on Evolutionary Computing and Mobile Sustainable Networks
10.1016/j.procs.2023.12.114

1. Introduction

Sentiment describes the emotions behind the subjective or objective views extracted from the textual data. Sentiment analysis ensures the textual data to determine its polarity and determine how positive or negative or neutral the sentiment behind the data is. Marketing and branding are the largely benefiting area from understanding the community sentiments and response to news events and press releases. Most organizations monitor the community response triggered to the relevant news events to improve their products and services. Community response enables them to understand how to transform their outreach protocols and also where to direct their resources on. Data collected from such activities mostly have directly expressed sentiments for statistical analysis. One other way to obtain similar data would be to monitor and analyze the community response text data online which are related to discussions of a particular news event. The study is aimed at developing a system which monitors online community responses and analyses the overall sentiments from the discussions. It performs sentiment analysis on textual data triggered as responses to news events from discussion posts/threads. The sentiments of the event from discussion threads are compared and presented to draw useful conclusions.

Two main variants of sentiment analysis considered are mainly Supervised Sentiment Analysis and Unsupervised sentiment Analysis using sentiment dictionaries and other way categorized majorly into, Lexicon-based/sentiment dictionary based unsupervised classification and Supervised classification using Machine Learning. In the Lexicon-based analysis, a large sentiment dictionary is used to analyze the total polarity of the document by its context. In Supervised classification approach, the ML Models are trained with labelled data and are later used to classify the new input textual data. In Supervised sentiment analysis, a machine learning model is built and trained with labelled training data. This data would contain a large amount of community response texts which are related to the same field with their overall sentiment labels (positive, negative or neutral). The labelling of the training data is done manually so that the machine learning model could be trained to analyze and learn patterns from this give more accurate predictions. This dataset is divided into two parts, one is to train the machine learning model which is chosen to train the model based on its performance. The other part is used to test the trained model by comparing predicted results with the already available labelled results. In Unsupervised sentiment analysis however, the most popular method is a Dictionary based approach. This would be very useful in the cases where labelling large amounts of data or acquiring such data becomes a difficult task in the required field. VADER which stands for Valence Aware Dictionary for Sentiment Reasoning is a model that has been widely used for sentiment analysis on textual data. VADER is specifically tuned for social media texts and works with the help of a sentiment dictionary which links the features of a word to their emotion intensities. Words vectors in VADER dictionary have some hundreds of dimensions which are useful in calculating the overall sentiment scores of textual data. This sentiment dictionary approach doesn't need large amounts of labelled data and it is helpful when you consume huge amounts of time to label the training data. Instead, as it works based on a sentiment dictionary, which assigns sentiment scores to the textual data depending on the context in which the words were used.

In this model, use the Recurrent Neural Networks to build a model which consist of atomic units called neurons which take inputs and go through a series of hidden layers to produce output. Neural Networks are used to classify or predict outcomes further at every step, the output from the previous step is used to produce the next output. Normally, in NNs, outputs and inputs do not depend on each other and the next state can only be figured out if the previous state is known. RNNs have a hidden state where they remember the information from the previous states [30].

The model is to be primarily conduct our analysis on comments extracted from Reddit. Reddit is an online discussion board/social media platform which has communities called "subreddits". Users can create their own subreddits or browse the existing ones to post content like textual content, news articles, pictures, videos, GIFs and other media. Other users can interact with these posts and post comments on them, others post replies to parent comments and so on. Reddit is mainly focused on textual discussion to connect the People who have similar interests all "subscribe" to subreddit that is focused on that specific interest. For example, there is a subreddit called "iOS" where users share all the latest news about iOS, their opinions, rants, helpful informative posts etc. The objective of the proposed model is to build a system that would get aggregated sentiments from these comments. These comments which are posted in response to a topic or a discussion posted on a subreddit are to be mined. Textual comments data can be scraped from different subreddits by retrieving data from the official API. Sentiment analysis is performed on this data by each community to analyze and contrast their biases and sentiments. The results are then presented showing the contrast of sentiment towards a particular news event between different communities. Users can quickly give input and retrieve sentiments from an abstracted pipelined architecture.

2. Related work

Several study methods [1], [2] compare the sentiment analysis methods but mostly the supervised ML algorithms to evaluate which gives the best predictions. Some research work[3], [4] has also done this comparison on unsupervised sentiment dictionary and supervised ML algorithms. While Supervised classification might yield better predictions in some discussion topics, it’s nearly impossible to find labelled data for every discussion as the topic being discussed might use words in the context that is only understood well in that specific community. The model [5] is aimed at research on text sentiment analysis about trending events on the social platform. They compared the classifications based on sentiment dictionary and Machine learning algorithms like Naive Bayes in order to achieve the best accuracy. The study [6] underline how training a model for sentiment analysis might be very challenging by trying to analyze the twitter tweets stream to predict the outcome of the 2016 U.S. Presidential election. Their preprocessing included excluding retweets, ignoring very similar tweets as they have tested the tweets for similarity. As the event was then active, it meant that the ever-changing dataset also posed a problem for which they had to regularly run the analysis.

The study [7] is to study sentiments of users and communities and its evolution over time. Their analysis revealed that the positive sentiment in communities is contagious as the members shared positive tweets over time. The results also shined a light on like mindedness and echo chambers within the community who have similar biases. Yun Xu et al.[8] researched on sentiment analysis on Yelp’s ratings based on text reviews. To anticipate the attitudes of the text, they used supervised classification ML systems. They have used and compared various ML algorithms like Naïve Bayes and Perceptron to compare and analyze the best performing configuration. [32] presented a machine learning summary of emotion analysis and classification techniques. Supervised, semi-supervised, and unsupervised machine learning methods each have their own set of advantages and disadvantages. A wide range of sentiment analysis techniques has been investigated, including Machine Learning Approaches, Rule-Based Approaches, and Lexical-Based approaches by Devika M D et al.[9]. The importance of a text's semantic analysis cannot be overstated. In this field, research is being done to improve analysis methods, including semantics by using n-gram evaluation instead of word-by-word analysis. [33] discussed supervised machine learning algorithms for sentiment analysis, highlighting that sentiment analysis is used to classify people's thoughts or sentiment regarding texts. Algorithms like nave bayes, support vector machine (SVM) and maximum entropy are employed in the supervised machine learning technique. To demonstrate how deep contextual language models like BERT can help with transfer learning in NLP, the authors of [10] employed the pretrained BERT model and finetuned it for the fine-grained sentiment classification task on the SST dataset. The model [11] proposed a method for processing the documents that used various machine learning algorithms to predict reader reaction to excerpts from the experience project, as well as predict human reactions to provide results as well as their pros and limitations. The study [12] suggested a sentiment classification method for text data based on LSTM which needs huge volume of data and a lot of manual review. When there is more training data, DL methods such as LSTM perform better with 85 percent accuracy in sentiment classification.

Using RNN and word2vec, the authors of [13]recommended a model for sentiment analysis. The Tensorflow system is used to implement RNN in this model and Word2Vec was developed by Google. Word2vec converts terms into vectors by examining their meaning in relation to other words in the sentence. The latter has a temporal component in RNN that differs from feedforward neural networks because of their temporal nature. The model [14]provided a brief overview of various feature selection processes, sentiment classification strategies and deep learning approaches for sentiment analysis including RNN, CNN, and LSTM. Sentiment analysis is used in a variety of decision-making, prediction and market applications. The author of [34] looked at why sentiment analysis from unstructured text is useful and what methods and techniques are available. A framework for mining public opinion for product-oriented tweets has also been introduced by the author. The device has been found to work well with a variety of tweets. The authors of [15]discussed the different sentiment classification strategies and two approaches are used such as machine learning-based approach and a lexicon-based approach which is less appropriate than a machine learning-based approach. Various deep learning models for sentiment analysis were summarised and the most common deep learning algorithms for sentiment analysis are CNN and RNN. The recurrent neural network’s performance is based on previous computations and sequential knowledge. According to this paper [15], the accuracy of RNN on the movie review dataset is 87.42 percent. Diederik P. Kingma et al.[16] developed a gradient-based optimization algorithm for stochastic objective functions that is both simple and computationally efficient. The authors' method was created for machine learning problems with large datasets and/or high-dimensional parameter spaces. The approach combines the benefits of two recently common optimization methods such as AdaGrad's ability to handle sparse gradients and RMSProp's ability to handle non-stationary objectives. The method is simple to implement and does not take much memory. The experiments back up the study of convex problem convergence rates and finally found Adam to be stable and well-suited to a variety of non-convex optimization problems. NWM-Adam has been designed by Yunlong Yu et al. [17] as a more flexible machine learning first-order gradient descent optimization algorithm to fix several optimization techniques' unfavourable convergence behaviour. The aim of the NWM-Adam algorithm is to save more information about previous gradients than information about recent gradients. The experimental results have demonstrated that NWM-Adam optimization algorithm performs better than other popular gradient descent optimization algorithms on some convex and non-convex problems in the field of machine learning.

The BRNN structure has been shown to produce better results than the other ANN structures in a series of comprehensive experiments. As a result, Schuster et al.[18]observed that the BRNN takes around the same amount of time to train as the other RNNs. Since the quest for an optimum delay (an additional search parameter during development) is not needed, BRNNs can provide faster development of real applications with better results than the other RNNs examined in this paper. It's worth noting that the modified BRNN structure is just a method for estimating the conditional likelihood of a given class sequence.

AfanGalihSalmana et al.[19], showed that intermediate variables can improve prediction capability of the LSTM model. The LSTM model is feasible and suggested to be implemented in predicting weather with the addition of intermediate data in order to improve the accuracy. In a time series data model, combining predicted and intermediate variables will improve forecasting accuracy. Pal Subarno et al.[20] investigated the impact of various architectures associated with LSTM models on sentiment analysis. First, the efficiency of a simple LSTM is assessed then layers are stacked on top of one another in subsequent steps, raising precision. Later, bidirectional LSTM layers were added to the network, allowing data to flow in both directions. The author discovered that a layered deep LSTM with bidirectional connections outperforms the simpler versions of LSTM used in this study in terms of accuracy.

Jenq-Haur Wang et al.[21] proposed a sentiment classification method focused on LSTM for short texts in social media. It is possible to train the contextual semantics of words in short texts using word embeddings such as the Word2Vec model. In addition, when there is more training data, deep learning methods like LSTM perform better at sentiment classification. Three RNN-based architectures were implemented by XipengQiu et al.[22]to model text sequence with multi-task learning. The mechanisms for exchanging information among the various tasks differ among them. Experiments show that by examining common features, the models can boost the performance of a group of similar tasks. Based on two contrasts, DoaaMohey El-Din Mohamed Hussein et al.[23]conducted a survey that addresses the significance and results of sentiment analysis problems in sentiment evaluation. The first distinction is based on the relationship between the framework of sentiment reviews and the difficulties of sentiment analysis. The second comparison is based on the accuracy rate's sentiment analysis challenges. The outcome of this comparison shows domain dependency as another important factor in recognizing sentiment challenges. Their findings demonstrate the value of sentiment challenges in assessing sentiments, as well as how to choose the most appropriate challenge to increase accuracy. Oscar B. Deho et al.[24]proposed using Word Embedding to assess the polarity of a text's emotions (positivity, neutrality, or negativity). The author used a Skip gramme version of the Word2Vec implementation and evaluated the findings using real-world Twitter datasets.

The experiments revealed that word embedding significantly improves sentiment classification accuracy. Ismoilov et al.[25]researched that regularisation approaches can effectively solve overfitting and underfitting problems, but even when regularisation algorithms were used, neural network models still had the same issues during training. This means that solving the problems is difficult and that a more advanced approach is needed to fully resolve the issues. A comparison of processing times for each regularisation scheme is the last thing to think about. It takes longer for stacked autoencoders to complete training and validation. Hancock et al.[27]looked at the latest methods for representing qualitative data for use as neural network input. The main contribution was that it was the first work to address all of these representation methods in one place. Another contribution was the provision of references for the source code of different techniques, allowing us to check the source code's validity. A new perspective on strategies for using categorical data in neural networks was the third contribution. The fourth contribution was to find some potential research opportunities.

Hyde et al.[11] showed that supervised machine learning is important and useful in ASD research. Due to the accessibility and cost of data gathering devices, machine learning can now be employed in ASD research. In the recent ten years, there has been an increase in the number of supervised machine learning research studies in the subject of ASD. PyTorch has become a widely used tool in the deep learning research field due to its focus on usability and thorough performance considerations. In addition to continuing to accept existing deep learning technologies and improvements, Adam Paszke et al.[28] worked to increase PyTorch's speed and scalability. Hyde et al.[11] showed that supervised machine learning is important and useful in ASD research. Due to the accessibility and cost of data gathering devices, machine learning can now be employed in ASD research. The PyTorch JIT, for example, is a suite of tools that allows PyTorch programmes to be run outside of the Python interpreter, where they can be optimised further. By offering powerful primitives for data parallelism as well as a Pythonic library for model parallelism based on remote procedure calls, the author sought to increase support for distributed computation.

Caffe, according to Jia et al.[29], provides a forum for cutting-edge deep learning algorithms as well as a range of reference models for multimedia scientists and practitioners. The framework is a BSD-licensed C++ library that includes Python and MATLAB bindings for efficiently training and deploying convolutional neural networks and other deep models on commodity architectures. Caffe uses CUDA GPU computation to meet business and internet-scale media needs, processing over 40 million images a day on a single K40 or Titan GPU (2.5 milliseconds per image). The Berkeley Vision and Learning Center (BVLC) maintains and develops Caffe with the aid of a large group of contributors on GitHub. The Crowd Counting Code Framework was developed by Gao et al.[31]with the aim of providing efficient and reliable crowd counting kits (C3F). C3F contributes in three different ways: 1) We present some solid baseline networks that have advanced to the state-of-the-art. 2) Some flexible parameter setting techniques are given to improve performance. 3) A effective log system is developed to record the experiment process, which can increase the reproducibility of each experiment. The major methods and studies are highlighted in Table1.

Table 1 Summary of various studies

Methodology	Key features/Limitations
Supervised ML algorithms	sentiment analysis predictions
unsupervised sentiment dictionary ML algorithms	impossible to find labelled data for every discussion
text sentiment analysis	classifications based on sentiment dictionary
training a model for sentiment analysis	preprocessing included excluding retweets, ignoring very similar tweets as they have tested the tweets for similarity
sentiments of users and communities	analysis revealed that the positive sentiment in communities.
supervised classification ML algorithms	Predicts text reviews
Lexical-Based Approaches	n-gram evaluation instead of word-by-word analysis
supervised machine learning methods	classify the sentiment about the text or people's opinions.
BERT	fine-grained sentiment classification
various machine learning algorithms	predict human reactions
LSTM	sentiment classification method for text data
RNN	sentiment analysis from word2vec

3. Proposed service Architecture

The proposed pipelined architecture as shown in figure 2, would consist of several processes which assist the free flow of data and streamline the workflow of analyzing sentiments. There are multiple steps that take place starting from taking user input and all the way to presenting the output to the user. The flow of the processes can be seen in Figure 1 and it shows the information in the proposed architecture. It has various stages where data is processed, transported, analyzed and presented. The sentiment analysis component has a pre-trained model saved on to the disk. The model is loaded and used to evaluate the sentiment of new comments extracted from reddit discussion post.

A. User input on the frontend

The aim was to streamline the workflow hence the proposed architecture would have a frontend with which the user can interact on the client. In the first step, the user inputs the data containing the links for a specific news event community discussion. The client on the frontend sends a HTTP - POST request to the backend server with the user input as the payload.

B. Backend Server generates tasks

Backend server receives the input data from the client in the previously received POST request. A new web socket connection is established so that the server could send messages to the client at a later time. The server then creates a task for the workers to further process the data. A new task is created and is stored in a FIFO data structure, which is a Queue.

C. Worker processes spawned to process the task

A worker threads constantly polls the previously mentioned queue for new tasks which found is assigned to a worker to retrieve the data and process sentiments. First, the task is examined and the scraper is initialized to retrieve responses from the discussion. The scraped data is then processed to reduce inconsistencies and make it more useful for sentiment analysis. And finally, the workers also invoke and perform sentiment analysis and store the data in the database and send the obtained results data to the client via previously opened web socket connection.

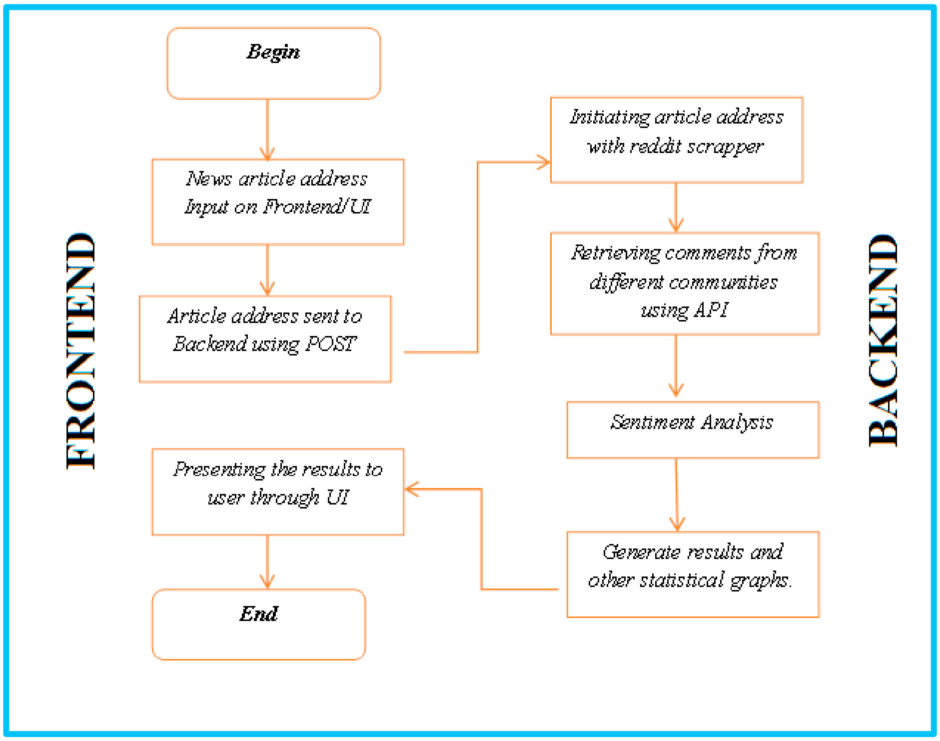

Fig 1. Sentiment retrieval process

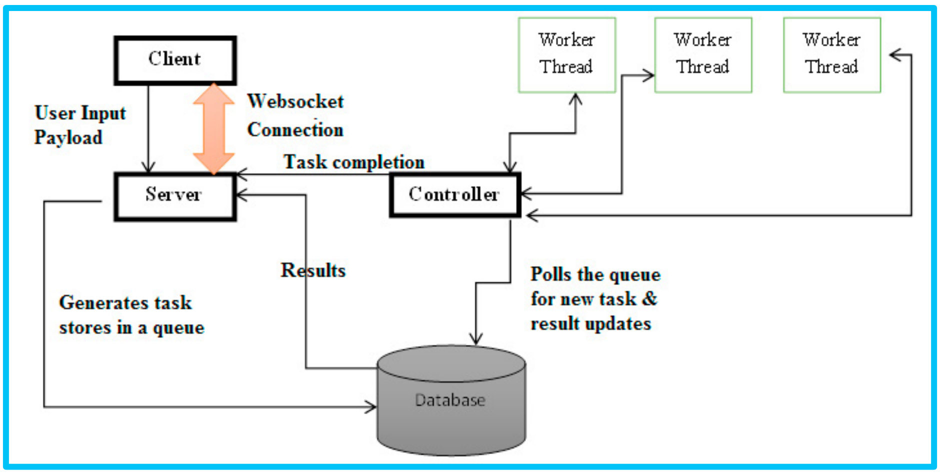

Fig 2. Web service Architecture visualized

D. *Scraper to scrape community responses*

The worker invokes an instance of the scraper to scrape the responses from the discussion. Reddit provides an easy and simple RESTful API. A python library, PRAW, which is a python reddit API wrapper module is used to retrieve data from the Reddit servers. This data is then iterated and saved in a structured form in the database.

E. *Pre-processing data to reduce inconsistencies*

The response data is then subjected to various transformations to reduce the inconsistencies. Hyperlinks in the responses are removed and responses with null data are removed. In the same way, depending on the community, there would be responses which may be deemed not useful for classification. These data elements are removed to make sure that they do not affect the sentiment scores. The following steps for pre-processing are:

i. Comments are pre-processed by normalizing extra letters such as (“worlld” will become “world”) and misspelled words like (“bcoz” to “because”).

ii. Removing all hashtags like “# Hashtags” and hyperlinks will take us to unwanted page “(url:/http:)” that are in comments.

iii. Avoiding unwanted spaces and special characters or symbols such as “ <> , ‘ ? ; : ! @ # $ % & * ~ / To seek an entry-level position that will enable me to apply my academic knowledge, hone my technical skills, and collaborate within a dynamic team. Eager to contribute to the growth and success of the organization while continuously expanding my expertise in the ever-evolving field, leveraging my technical skills effectively to innovative projects while continuously expanding my knowledge in a professional environment. [ ] { } + = _ - ( ) | ^ .

iv. Removing stop words such as “all, is, was, that, this, of, so, such, etc.” will not add more meaning to the sentence.

v. Avoiding duplicated comments and emojis.

vi. Removing case conversions.

vii. Separating sentences into phrases or words are stated to be tokens by which some characters are discarded (Tokenization).

F. ***Sentiment analysis on community response datasets***

The main function of the sentiment analysis component is to analyze the sentiments of the textual data sent to it. The textual data which are to be analysed would be the community responses from a reddit discussion thread. As such, this component only focuses on getting the sentiments of the givendata.

4. Communication Model

A. ***Preparing Data***

The first step in preparing the scraped comment data is performing tokenization. It’s the process of dividing the given text into tokens, which are just small words. Tokenizing doesn’t strictly split the tokens based on spaces. Using a “spaCy” tokenizer, It first splits the string based on whitespaces and then parses each token from left to right and checks if any of the tokens match exception rules like “don’t” will be broken down into “do” and “n’t” etc. After tokenizing, Split the sentiment dataset into train and test in 70/30 ratio. Now vocabulary would have to build using the train data which is a lookup table with every word in the dataset. The length of this vocabulary would be equal to the number of unique words in our dataset. Each index in this vocabulary is a “One-hot” vector with the length same as the total unique words with 0 in all indexes except that of the word. An example vocabulary with 4 words and their one-hot vectors are given below in figure 3.

$Word$	$Index$	$One - hot\ vector$
I	0	[1,0 0,0]
$hate$	1	[0,1,0,0]
$this$	2	[0,0,1,0]
$film$	3	[0,0,0,1]

Fig 3. Matrix of Vocabulary - One-hot vectors visualized

As including all the unique words would yield a very high number of dimensions, chose to include the top most common 25,000 words which will lead the vocabulary would effectively have 25,000 one-hot vectors. The other words will be replaced with the <unk> (unknown) token. Also, when the sentences are given as input to the model, they are fed as fixed size batches at a time. When the batch isn't full, padding tokens <pad> are added to make the batch complete.

B. ***Building the Model using Long Short-Term Memory RNN***

A general Recurrent Neural Network (RNN) as shown in figure 4 takes in batch of words, one word at a time and outputs a hidden state h for each word. This RNN will be fed a word, xt and the previous hidden state, ht-1, to produce the next hidden state, ht where $\boldsymbol{h_t = RNN(x_t, h_{t-1})}$.

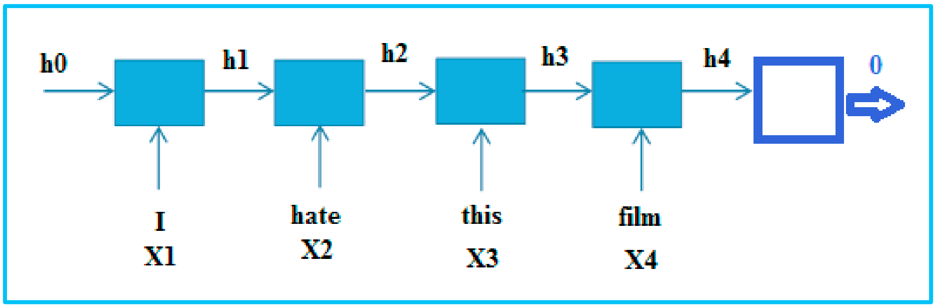

Fig 4. A Simple RNN Architecture

The model would have 3 layers. An embedding layer, RNN layer and a linear layer. The embedding layer actually tries to convert a "one-hot" vector which is sparse with lots of zeros and just 1 for the word into a dense vector which the algorithms could infer more information from. For words with similar sentiments, the dense vectors would have similar values. The figure 5 shows a representation of a sparse one-hot vector and the dense vector of the same.

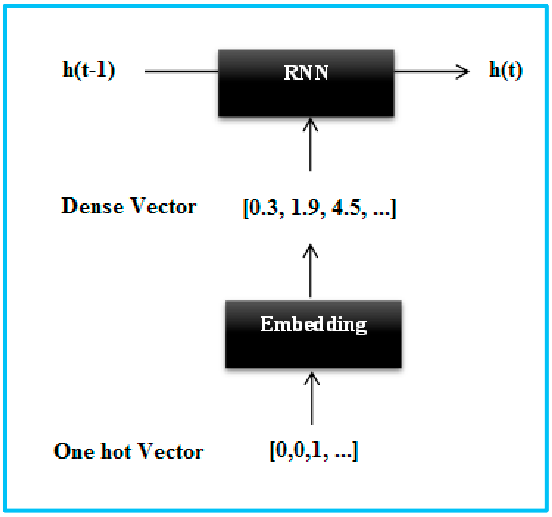

Fig 5. Spare to dense vector transformation

Instead of randomly initializing word embeddings, we propose using pre-trained vectors using models like Word2Vector, Glove (Global Vectors) etc. In this specific case, use "glove.6B.100d" model. The algorithm used to calculate vectors is called glove and 6B indicates that vectors were trained on 6 Billion tokens and the 100d represents that the dimensions of the vectors would be 100. If these word embeddings used, then would have vector initializations such that words with similar already start with being close to each other.

Long Short-Term Memory (LSTM) Architecture (Figure 6) is used in the place of a basic RNN architecture in order to achieve better results. LSTM is basically focused on text semantics and positioning, for which this model is most suitable because we trim the written comments in to words and words to sentences which are then trimmed to letters, hence the positioning of text obtained matters the most. Not to mention, this paper even improves the recent discoveries of LSTM which solves the vanishing gradient problem by using a more refined approach of LSTM model. They do this by including extra state which is recurrent called cell, c, which is like memory and rewrite the previous equation for the current hidden state as a function with ct, ht, xt in the following way [26],

$$(h_t, c_t) = RNN(x_t, h_{t-1}, c_{t-1})$$

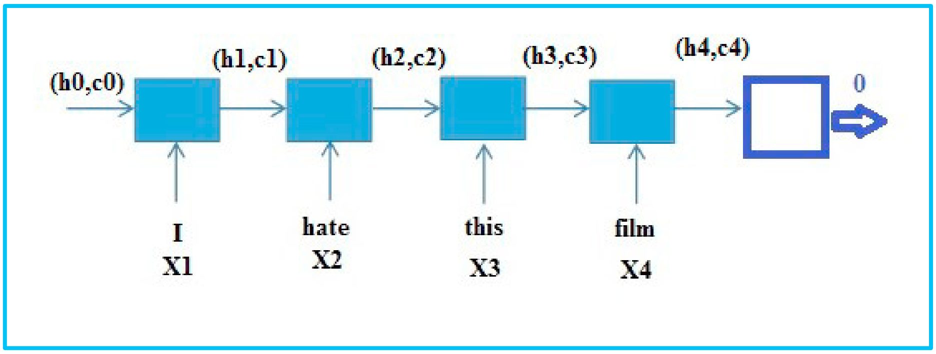

Fig 6. LSTM RNN Architecture

To attain better outcomes, also apply the approaches that are minor versions of RNN, such as Bidirectional RNN and Multilayer RNN. In a Bidirectional RNN, there's an RNN which processes words from first to last and another which does the same from last to first. Concatenate the last state of the forward RNN with that of the backward RNN yields the sentiment forecast. A Multilayered RNN, on the other hand, is one in which additional RNNs are piled on top of the original RNN. The input to the RNN above it will be the original RNN's hidden output and so on. The hidden state of the final RNN is used to make the prediction. The following are the variables used in Table 2 to set the configuration for the LSTM Model discussed till now.

Table 2 Variables configuration

INPUT DIMENSIONS = length of vocabulary
(25000 most common words)
EMBEDDING DIMENSIONS = 100
(same as the dimension in glove vectors)
HIDDEN DIMENSIONS = 256
OUTPUT DIMENSIONS = 1
NUMBER OF LAYERS = 2

C. *Training*

The model parameters are updated using the Adam optimizer method, it adapts the learning rate for each parameter. Suppose a parameter is more frequently being updated, Adam gives it a lower learning rate and the opposite for infrequently updated parameters. We set number of epochs = 5 in our tests to train our model.

5. Results & Observations

A. Presentation of the sentiment results

Sentiment analysis results are then presented in a meaningful way and conclusions can be drawn and interpreted from the displayed data. The aforementioned architecture could be implemented using many different technologies. In implementation, the model built the frontend which the user interacts with, using React framework. Building on React would have advantages like unidirectional data flow and declarative states which can be managed easily compared to the usual DOM manipulation approach using vanilla JavaScript. A simple backend server was spawned using Express.js (a Node.js framework) as it is quick and easy to work with. A LSTM model constructed with the pyTorch framework was used to do the sentiment analysis. Pandas is used to process the data. PRAW python library is used to retrieve data from Reddit's servers.

B. Experimental Data Set

The experiment used the Reddit Sentiment dataset from Kaggle which had scraped reddit comments and the associated sentiments whether the comment was positive(pos) or negative(neg) as shown in Table 3. The model cleaned the dataset to reduce inconsistencies. The dataset consists of a total 50,000 records and 25,000 of them are positive and 25,000 are negative. The sentiment classification is evaluated and discussed at Table 6.

Table 3 Examples from the reddit dataset

Positive	Negative
This is my favorite phone I've ever had, I love the developer support, loud speakers and price. Ram and CPU is good for the $.	Wifi reception has become s**t on Miui12....Starts dropping to 2 bars just going next room to the router
Love you AMD Ryzen Kudos	I do hope that for the high end they add an extra CPU chiplet, and not useless igpu.

We trained the RNN and LSTM Model for 50 epochs. The validation accuracy peaked at 89.29%. The following are the Train Loss, Train Accuracy, Validation Loss and Validation Accuracy data for each epoch as shown in Table 5 and standard RNN model in Table 4.

Table 4 Standard RNN Model

Epoch	Train		Validation	
	Loss	Accuracy	Loss	Accuracy
10	0.694	50.12	0.696	50.17
20	0.693	49.72	0.696	51.01
30	0.693	50.22	0.696	50.87
40	0.693	49.94	0.969	49.91
50	0.693	50.07	0.696	51.00

TEST LOSS – 0.708
TEST ACCURACY – 47.87%

Table 5 LSTM RNN Model with bidirectional and Multilayer RNNs

Epoch	Train		Validation	
	Loss	Accuracy	Loss	Accuracy
10	0.649	61.45	0.685	55.95
20	0.639	62.15	0.516	75.98
30	0.485	82.27	0.460	86.20
40	0.255	89.88	0.311	87.32
50	0.219	91.67	0.291	89.29

TEST LOSS – 0.293
TEST ACCURACY – 87.82%

Table 6 Accuracy & Loss of the sentiment Classifier with actual data

Method	Loss		Accuracy	
	Min	Max	Min	Max
sentiment Negative	0.04533	0.4297	0.79372	0.98651
sentiment Positive	0.05541	0.422053	0.80563	0.98312
Neutral Classifier accuracy	0.023200	0.29040	0.88267	0.97323

C. On-demand web service

The proposed model able to design a streamlined pipeline architecture to get access to the reddit community discussion sentiments on-demand. The overall process contains several closely connected components. The results of an unsupervised classification can't be evaluated as opposed to a supervised classification model, but it provides many other benefits over a supervised classification. Supervised classification requires a large amount of correctly labelled data to give accurate predictions. But labelled data isn't always available when considering analysis on textual data which might contain discussion on a wide range of topics. Mis-labelled data on the data skews the predictions and results. A large chunk of labelled data online comes from unsupervised and it's not effective and might even affect the predictions in a wrong way. After the discussion-response data is retrieved from the internet, it has to be processed to remove any quantities that might affect the analysis in a bad way. Sarcasm is a large problem in sentiment analysis as its almost impossible to detect sarcasm in a document. Methods like training models with known sarcastic textual data to predict and discard them from the datasets can be employed. There are some discussions in which the reactions might not necessarily have polarizing sentiments. Like for example, a discussion which starts with an objective question and has objective answers in response. Finding Sentiments is most effecting when it is known that the discussions would have polarizing opinions like: Launch of a new product, Roster moves for a professional sports club, Press conferences etc. Hence it is in the hands of the user to select discussions that would potentially reveal useful information and also to interpret the results. The interface is illustrated in figure 7a, 7b and 7c.

Fig. 7a) User input on the frontend

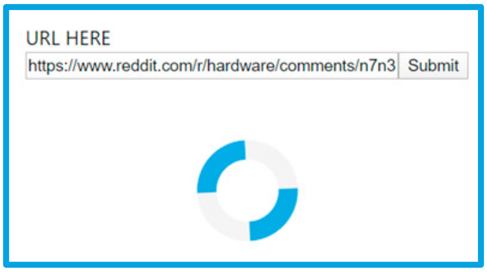

Fig. 7 b) User input on the frontend

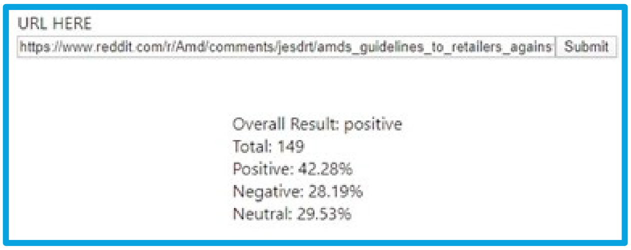

Fig. 7 c) Result on the frontend

6. Conclusion

In our proposed model, the Adam algorithm has been used to train the weights and observed that the LSTM model performed approximately 1.8 times better than a standard RNN model. The LSTM model includes Bidirectional RNNs and Multilayer RNNs which used an Adam method to train the model parameters as opposed to Stochastic Gradient Descent (SGD) in case of the standard RNN architecture. In sentiment analysis, attention-based neural networks are becoming increasingly prominent. It's an RNN version that tells the network where to look when the task is completed. Finally, BERT (Bidirectional Encoder Representations from Transformers), a novel language model, is touted to attain the highest accuracy for a variety of applications, including sentiment analysis.

References

[1] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan.(2002) " Thumbs up? Sentiment Classification using Machine Learning Techniques" In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002) .79–86.

[2] P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha, (2013) "Comparing and combining sentiment analysis methods," 2013, Proceedings of the first ACM conference on Online social networks. 27-38.

.[3] W. Maharani(2013)" "Microblogging sentiment analysis with lexical based and machine learning approaches, International Conference of Information and Communication Technology (ICoICT), Bandung, Indonesia. 439-443

[4] S Padmaja, Prof. S Sameen Fatima and Sasidhar Bandu (2014) "Evaluating Sentiment Analysis Methods and Identifying Scope of Negation in Newspaper Articles" International Journal of Advanced Research in Artificial Intelligence(ijarai), **3(11)** 1-6.

[5] Fulian Yin, Beibei Zhang, Pei Su and Juanfang Chai(2016), "Research on the text sentiment classification about the social hot events on Weibo," 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, China. 1537-1541

[6] M. Ebrahimi, A. H. Yazdavar and A. Sheth (2017) "Challenges of Sentiment Analysis for Dynamic Events," in IEEE Intelligent Systems **32(5)** . 70-75

[7] N. Alduaiji and A. Datta (2018) "An Empirical Study on Sentiments in Twitter Communities," 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore.1166-1172

[8] Y. Xu, X. Wu, and Q. Wang (2015) "Sentiment Analysis of Yelp's Ratings Based on Text Reviews", 2015 17th Int. Symp. Symb. Numer. Algorithms Sci. Comput., 2015.

[9] M. D. Devika, C. Sunitha, and A. Ganesh (2016) "Sentiment Analysis: A Comparative Study on Different Approaches.Procedia Computer Science **87.** 44 – 49

[10] M. Munikar, S. Shakya and A. Shrestha (2019) "Fine-grained Sentiment Classification using BERT," 2019 Artificial Intelligence for Transforming Business and Society (AITB), Kathmandu, Nepal, 1-5

[11] Hyde, K., Novack, M.N., LaHaye, N. et al(2019)." Applications of Supervised Machine Learning in Autism Spectrum Disorder Research: a Review" . Rev J Autism Dev Disord 6, 128–146 .

[12] G. S. N. Murthy, Shanmukha Rao Allu, Bhargavi Andhavarapu, and Mounika Bagadi, Mounika Belusonti, "Text based Sentiment Analysis using LSTM," *Int. J. Eng. Res.***9(5)** . 299-303.

[13] L. Kurniasari and A. Setyanto (2020)"Sentiment Analysis using Recurrent Neural Network, Journal of Physics: Conf. Series **1471** .1-5

[14] A. Patel and A. K. Tiwari (2019) , "Sentiment Analysis by using Recurrent Neural Network," 2 nd International Conference on Advanced Computing and Software Engineering (ICACSE-2019) .1-4.

[15] S. Ruder, "Overview Optimization Gradients," *arXiv Prepr. arXiv1609.04747*, 2017.

[16] D. P. Kingma and J. L. Ba (2015) , "Adam: A method for stochastic optimization, arXiv:1412.6980

[17] Y. Yu and F. Liu (2019) , "Effective Neural Network Training with a New Weighting Mechanism-Based Optimization Algorithm," *IEEE Access*,**7(2019)** 72403-72410

[18] M. Schuster and K. K. Paliwal (1997) , "Bidirectional recurrent neural networks," in *IEEE Transactions on Signal Processing*, **45(11),** 2673-2681

[19] A. G. Salman, Y. Heryadi, E. Abdurahman, and W. Suparta(2018) , "Single Layer & Multi-layer Long Short-Term Memory (LSTM) Model with Intermediate Variables for Weather Forecasting **135(2018).**89-98.

[20] S. Pal, S. Ghosh, and A. Nag(2018 "Sentiment Analysis in the Light of LSTM Recurrent Neural Networks," *Int. J. Synth. Emot.***9(1).**33-39.

[21] J. H. Wang, T. W. Liu, X. Luo, and L. Wang(2018) "An LSTM approach to short text sentiment classification with word embeddings," In Proceedings of the 30th Conference on Computational Linguistics and Speech Processing (ROCLING 2018) 214–223 .

[22] P. Liu, X. Qiu, and H. Xuanjing(2016) ,"Recurrent neural network for text classification with multi-task learning , Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI-16).2873-2879.

[23] D. M. E. D. M. Hussein (2018) "A survey on sentiment analysis challenges," Journal of King Saud University - Engineering Sciences, **30(4):** 330-338.

[24] O. B. Deho, W. A. Agangiba, F. L. Aryeh, and J. A. Ansah (2018) , "Sentiment analysis with word embedding," AIP Conf. Proc. 2290, 1-7

[25] Nusrat, Ismoilov, and Sung-Bong Jang. 2018. "A Comparison of Regularization Techniques in Deep Neural Networks" *Symmetry* **10(11)** :648.

[26] Prabakaran, N., Rajasekaran Palaniappan, R. Kannadasan, Satya Vinay Dudi, and V. Sasidhar.(2021) "Forecasting the momentum using customised loss function for financial series." International Journal of Intelligent Computing and Cybernetics **14(4)** : 702-713

[27] J. T. Hancock and T. M. Khoshgoftaar(2020) "Survey on categorical data for neural networks," *J. Big Data*.**7(28)** : 1-41.

[28] A. Paszke *et al.(2019)* "PyTorch: An imperative style, high-performance deep learning library," *arXiv*. 2019.

[29] Y. Jia et al.(2014) Caffe: Convolutional architecture for fast feature embedding proceedings of the 22nd ACM international conference on Multimedia .November 2014: 675–678

[30] Prabakaran, N., Suvarna Sree Sai Kumar, P. Kranthi Kiran, and Pabbidi Supriya(2022) "A deep learning based social distance analyzer with person detection and Tracking Using Region based convolutional neural networks for novel coronavirus." *Journal of Mobile Multimedia* (2022): 541-560.

[31] J. Gao, W. Lin, B. Zhao, D. Wang, C. Gao, and J. Wen (2019) "C3framework: An open-source PyTorch code for crowd counting," *arXiv*. 2019.

[32] Usharani, B (2018) "Online Product Reviews Based on Sentiment Analysis." *Asian Journal of Computer Science Engineering* 3.1 (2018): 14-21.

[33] Kamble, Sangharshjit S., and A. R. Itkikar (2018) "Study of supervised machine learning approaches for sentiment analysis." *International Research Journal of Engineering and Technology (IRJET)*.**5(4):**3045-3047.

[34] Unnamalai, K(2012) . "Sentiment analysis of products using web." *Procedia engineering* **38 (2012):** 2257-2262.