Product Review Sentiment Analysis with SmartBot

A Project Report

Submitted by:

Harsh Agola (AU1841106)

Prince Dalsaniya (AU1841124)

Akshay Bhimani (AU1841126)

at

Ahmedabad University

School of Computer Studies (SCS)

Ahmedabad, Gujarat

Table of Contents

1. INTRODUCTION

1.1 Problem Definition

Oftentimes we are not sure to buy any product. We are not sure that the product we are buying is worth the money or not . To solve this problem we ask someone who might have already purchased the product and get their reviews on the matter to get a proper decisive answer on whether or not we should buy the product. We are trying to solve this problem many times we might not have known the stakeholders who might have purchased the same product. So we tried to solve the problem with the power that the internet possesses , i.e to connect people who share the same interest .

The problem we are trying to solve is that, many times we ask some friends or relatives whether I should buy some product or not. They simply reply in yes or no. We have tried to achieve that by the reviews of that product over the internet. Finding sentiments of all of them and finding the final reply that You should buy this, or not. Using a chatbot it works seamlessly and automates the process of getting the answer of “Should I buy this product?”

1.2 Project Overview

By giving the overview of our project, we have divided this project into 3 parts. (ｉ) Getting product name by chatbot. (ｉｉ) Finding the reviews from the internet. (ｉｉｉ) Deriving the sentiments and giving final result. We have used some machine learning models to derive sentiments and some deep learning techniques in chatbot.

2. LITERATURE SURVEY

2.1 Existing Systems

There are no such systems existing which we are trying to build. But the parts of this are existing individually. Like there are systems of chatbots that exist. Sentiment analysis also exists using Machine Learning Algorithms [3][4]. But we are not aware of any system that gives all of this combined. “Building Chatbots with Python: Using Natural Language Processing and Machine Learning” [1] - Here it was described types of chatbots and how different chatbots work. Also there are different components involved with chatbots and what libraries and frameworks can be used to build one like there is pytorch , RASA NLU , wit.ai , chatterbot ,etc. Also there was an example of building an horoscope bot with RASA and how does it function when deployed on the “Ticketing Chatbot Service using Serverless NLP Technology,” [2] in this paper we have seen POS tagging with chatbot for booking tickets online .Get the general understanding of dividing messages into intent and Entity . Also how to set responses for particular types of request. Implementation using wit.ai for the combination of entity and intent recognition.

For sentiment analysis , " Improving the Sentiment classification " - [10], here we got the general working idea of SVM. It maps input vectors into some high dimensional feature space Z through some non-linear mapping chosen a priori. In this space, a linear decision surface is constructed with special properties that ensure high generalisation ability of the network. What we understood was the idea of the SVM is to convert data into a high-dimension map according to different features, and then find the best decision line, which leaves the largest margin to any data point in each feature category. This makes it to outperform other supervised learning sometimes in Sentiment Classification. Then divided the dataset in 2 parts containing Positive and Negative reviews with help of ratings given with the review. [4](Positive: 1,2, Negative: 4,5) . Then given labels 0 to Negatives and 1 to Positives.Divided the dataset into 2 parts , training and testing.Extracted features using countvectorizer.Applied SVM, Multinomial NB with different Kernels and tested our own reviews.[4]

2.2 Proposed System

The system we have built can be explained as, user give the product name to chatbot, Chatbot will give this to the next module that is a web scraper which scrapes the reviews of that product from the internet and passes those to the next module. Which finds the sentiments of all the reviews given and assigns the weight to it. After that we calculate the final weight of all reviews. According to that we give our final result. Which gives the user seamless experience as the user has to only use chatbot as the interface to get the results.

2.3 Feasibility Study

We have built a basic complexity system here. We have used the SVM algorithm with the rbf kernel and used TfIdf as a count vector for any review that is given. This was the part of the sentiment analysis. For data collection we have used a standard library for web scraping "beautifulsoup". This does not need some high end computing. So, it is very much feasible to use. Chatbot uses a feed forward neural network to classify the user query. It requires a GPU to get the desired results fast but also performs decently on a low end CPU. In case of high performance computers the pipeline works flawlessly.

3. SYSTEM ANALYSIS & DESIGN

3.1 General Idea

We divided the project into 3 important parts , i.e , chatbot , web scraper and sentiment analyzer. The first part that is chatbot is the main source that will act as the interface between user and main logic. Web scraper is the intermediate that will fetch results in real-time on the product that users have mentioned. And last but not the least sentiment analyzer is the main core of the system. It takes the reviews fetched from the web scraper and gives the final result. The result is returned to the user via chatbot.

For smartbot our primary feature was to tell the user whether he/she should buy the product or not. This was done by using sentiment analysis on the product reviews. Product reviews for the product user asked was gathered in real-time using web scraping techniques on the website like www.amazon.in . As mentioned earlier the reviews fetched will serve as the input for the sentiment analyzer. We then created an algorithm to derive the final result on the basis of the sentiment analysis done on each review. As of now we have kept 0.5 as the threshold for the algorithm . If the average of the reviews is > 0.5 then we believe that product should be bought. And the result we get here is then returned to the user as a message via chatbot.

3.2 System analysis

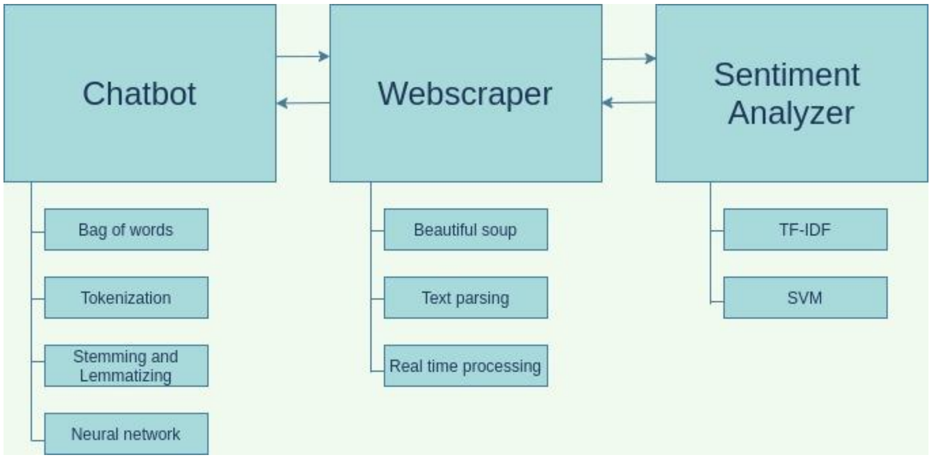


Fig 1 (a)

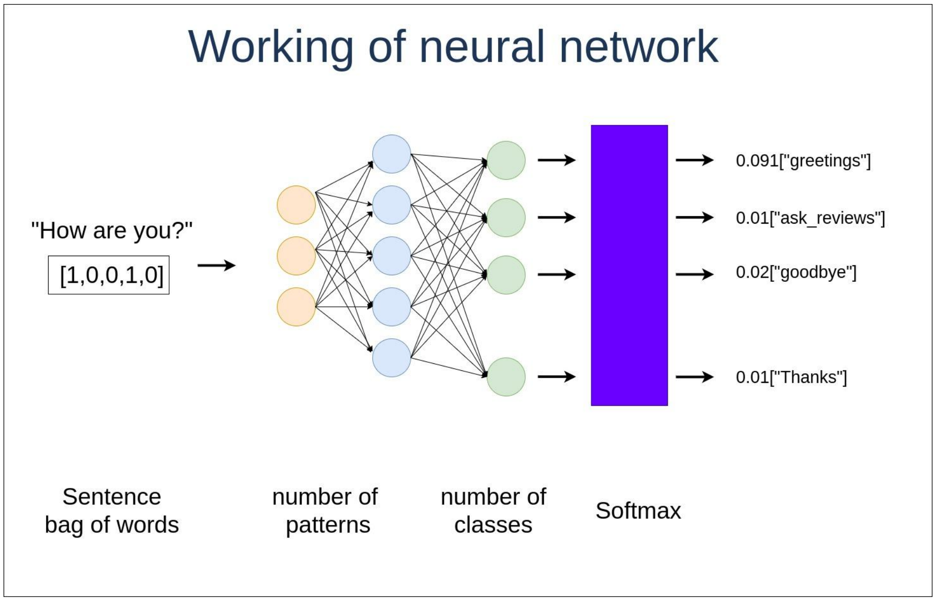


Fig 1 (b)

3.3 Algorithms and pseudo code

3.3.1 Pseudocode for chatbot:

```
Step 1 : Get the query from the user
Step 2 : Apply tokenization to the query
Step 3 : Transform the tokens into lower case and apply stemming
Step 4 : Apply bag of words model to the processed query
Step 5 : Feed the bag of word models into the feed forward neural network.
Step 6 : Get the tags from the neural network and give random responses from the intents.json file
Step 7 : If the tag is for “ask_review” get the product name from user
Step 8 :  forward the product name query to web scraper
```

3.3.2 Pseudo code for sentiment analysis:

```
Step 1 : Get the data and make it according to the requirement.
Step 2 : Given weights as 0 (Negative), 1 (Positive). According to the rating given in the dataset. Such as ratings 4,5 are Positive, 1,2 are negative.
Step 3 : Splitted the dataset using the ScikitLearn Library function.
Step 4 : Use TfidfVectorizer from ScikitLearn Library to Convert a collection of raw documents to a matrix of TF-IDF features.
Step 5 : Then trained the SVM model with the rbf kernel using the data we split previously.
Step 6 : Then just predicted the review and printed the Positive or Negative according to the returned value.
```

3.3.3 Pseudo Code for the bunch of reviews and giving the final output :

```
Step 1 : Got the reviews from the scraper and converted into 2 arrays containing reviews and Ratings.
Step 2 : Then Predicted the sentiments and stored the weights 0 (for Negative) and 1 (for Positive)
Step 3 : Take the average of all weights and if that is greater than 0.5 then it can be bought. Or else vice-versa.
Step 4 : Then we also took the average of all ratings we got. If it is less than 3 then product shouldn’t be bought. This is to just check if our sentiment analysis model is predicting the sentiments correctly or  not.
```

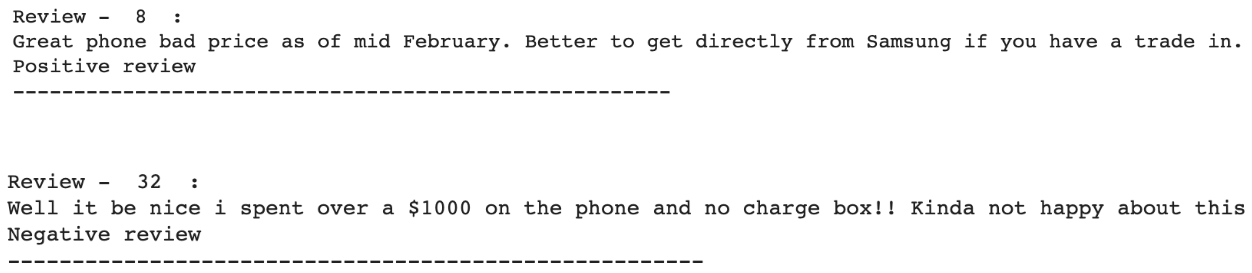
3.4 Testing Process

First we have trained model using a dataset of reviews containing electronic products. Then we have used the TF-IDF vectorizer to convert the reviews into the matrix containing the TF-IDF features. Then split the data and train the SVM model using the train data. We then checked the accuracy of the model by predicting the outputs on the test data. We have tried various models for this. SVM, Naive Bayesian both with countvectorizer and TF-IDF vectorizer. Also SVM with linear and rbf kernel. We got the highest accuracy on SVM(using rbf kernel) with TF-IDF vectorizer. So we have used that for our final part. After that we have scraped the reviews of any product from the internet. We have predicted the sentiments and assigned the weights according to that. Then we have taken the average of all weights and returned with the final output that contains the details of how much percent this product has positive reviews and should I buy it or not, with the purchase link.

For testing of the final product as a chatbot we have provided this chatbot to some of the potential stakeholders to test it out. Out of 5 , 3 of the stakeholders tested the chatbot and get satisfaction and desired output with the products they have already purchased and products that they are willing to purchase. Other 2 were facing ambiguity due to the requirement of a properly named model of the product. If the product is not properly named it is difficult to scrape and the system breaks.

4. RESULTS / OUTPUTS

In the results we can see in this below image, we give the review and it returns the sentiment of that review These are the examples of “Samsung S21 Ultra”.

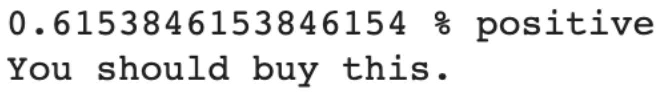


```
Review -  8  :
Great phone bad price as of mid February. Better to get directly from Samsung if you have a trade in.
Positive review
------------------------------------------------------


Review -  32  :
Well it be nice i spent over a $1000 on the phone and no charge box!! Kinda not happy about this
Negative review
------------------------------------------------------
```

Fig 2 - Sentiment analysis of product reviews

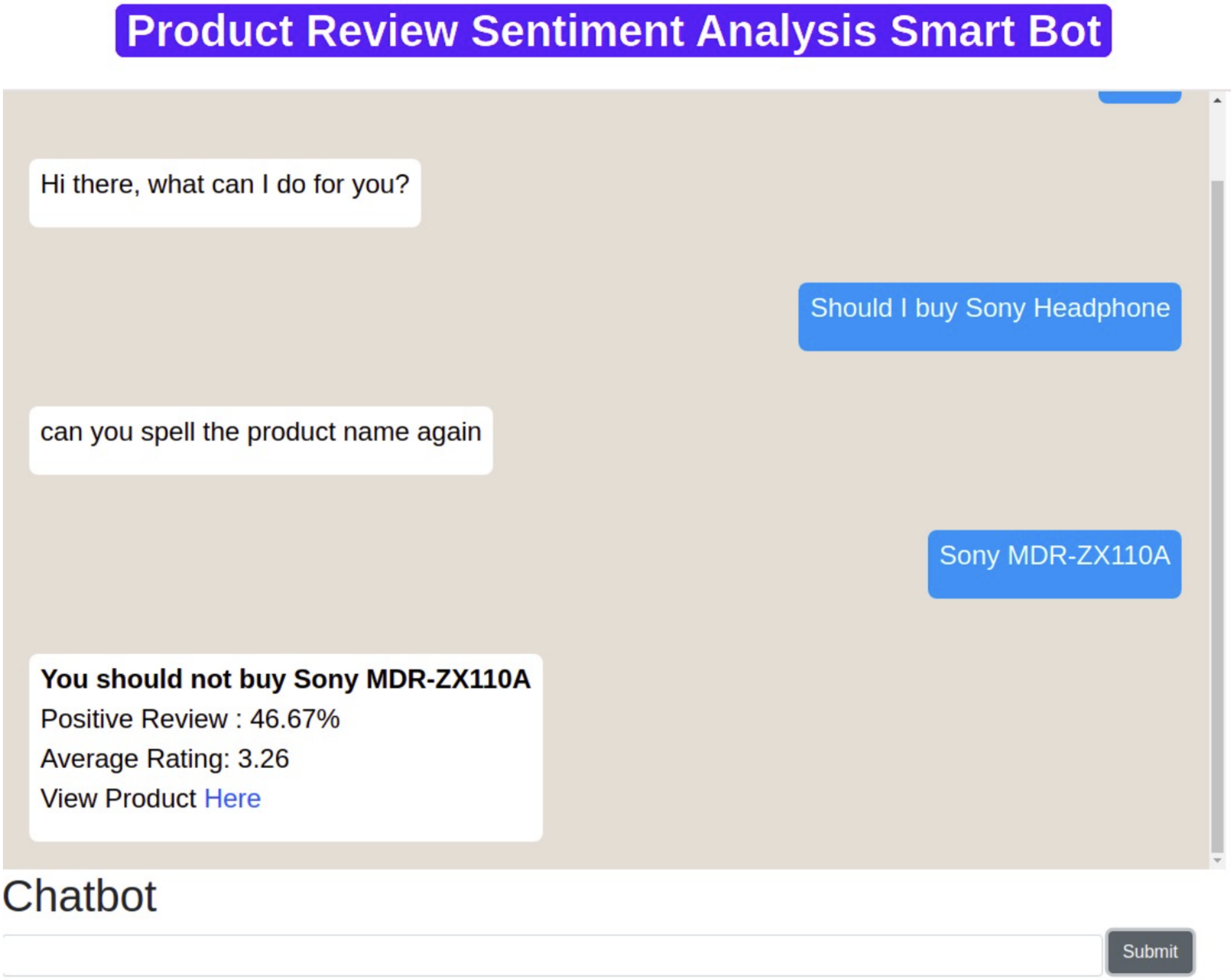
We have scraped over 300 reviews and given the final result. That gives the final answer.



```
0.6153846153846154 % positive
You should buy this.
```

Fig - Final answer of the product reviews.

Chatbot GUI



Product Review Sentiment Analysis Smart Bot

Hi there, what can I do for you?

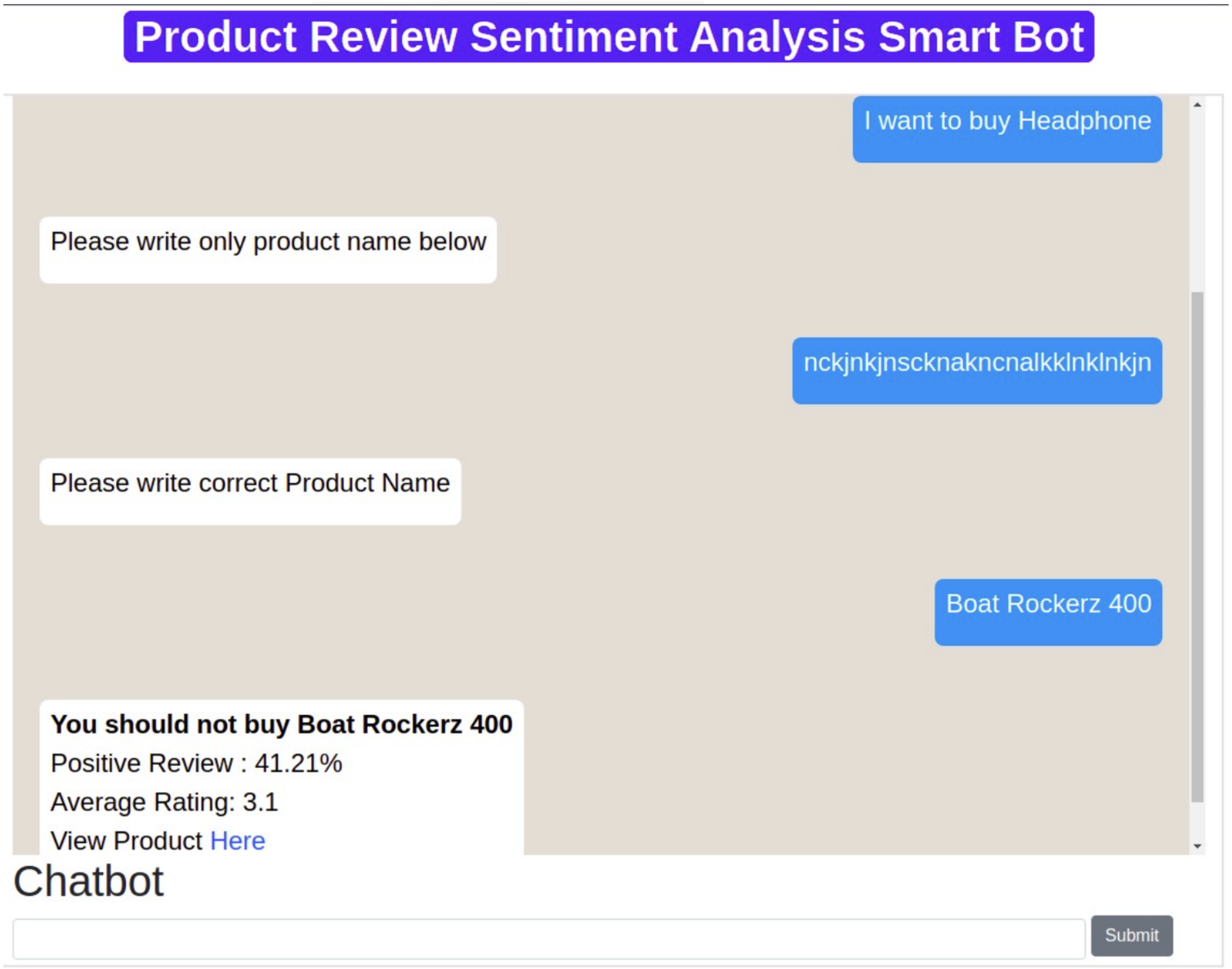
Should I buy Sony Headphone

can you spell the product name again

Sony MDR-ZX110A

You should not buy Sony MDR-ZX110A
Positive Review : 46.67%
Average Rating: 3.26
View Product Here

Fig 4 - Conversation with Smart Bot fetching product reviews



Product Review Sentiment Analysis Smart Bot

I want to buy Headphone

Please write only product name below

nckjnkjnscknakncnalkklnklnkjn

Please write correct Product Name

Boat Rockerz 400

You should not buy Boat Rockerz 400
Positive Review : 41.21%
Average Rating: 3.1
View Product Here

Chatbot

Fig 5 - Smartbot detecting wrong product name and asking to correct it.

5. CONCLUSIONS / RECOMMENDATIONS

In the final conclusion we can say that this is the project which helps you to get an idea of the product, is it the product to buy or not. It automates the task of manually getting reviews from other persons and getting a decisive answer to the ultimate question of whether or not we should buy this product. Using this you can shortlist from the list of products you are confused about. Ex. We want to buy a phone, we have a list of phones, we check them here. We can shortlist 2-3 phones. After that we can confirm the phone from those 2-3.

There are some shortcomings of using this method also , as we are doing this whole process in real time , it takes some time to fetch the results. And there is also the processing of the reviews which also takes some time to give the desired result. Sometimes bot also recognizes the wrong product name and there is also a possibility of getting wrong product results if the model name of the product name is not correctly mentioned.

6. REFERENCES

[1] S. Raj, Building chatbots with Python: using natural language processing and machine learning. Berkeley, CA: Apress, 2019.

[2] E. Handoyo, M. Arfan, Y. A. Soetrisno, M. Somantri, A. Sofwan, and E. W. Sinuraya, "Ticketing Chatbot Service using Serverless NLP Technology," 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), 2018.

[3] "A Complete Sentiment Analysis Algorithm in Python with Amazon Product Review Data: Step by Step", Medium, 2021. [Online]. Available: - link

[4] SENTIMENT ANALYSIS OF AMAZON REVIEWS. - link

[5]Jiao, A., 2020. An Intelligent Chatbot System Based on Entity Extraction Using RASA NLU and Neural Network. Journal of Physics: Conference Series, 1487, p.012014.

[6] Design and implementation of a chatbot for e-commerce , Amir-reza Asadi, Reza Hemadi

[7] Scraping Search Results from Google Search - link

[8] Shelar, H., Kaur, G., Heda, N. and Agrawal, P., 2020. Named Entity Recognition Approaches and Their Comparison for Custom NER Model. Science & Technology Libraries, 39(3),

pp.324-337.

[9] Xiang Cheng, Mitchell Bowden , “An End-to-End Solution for Named Entity Recognition in eCommerce Search “, 2020

[10] S. Li, "IMPROVING THE SENTIMENT CLASSIFICATION OF STOCK TWEETS", Etheses.bham.ac.uk, 2014. [Online]. - link