Customisable Twitter Bots for Improved QoS in Product Marketing

Presentation By:

Anush Baskaran 13BCE0800
B Ramya 13BCE0613
Taruni Anand 13BCE0036

QUESTIONS TO BE ANSWERED IN THIS PRESENTATION

- Why Twitter?, How automation helps in easier product marketing? -> Problem Statement
- How our software extends its capabilities beyond computing to serve its purpose in marketing products -> Literature Survey
- Internal Design of the Software
  - System Design
  - Methodology Adapted
  - Flow of the model
- Hardware & Software Requirements
- Expected Outcome (Advantages/Disadvantages Expected/Future Work)
- Project Timeline
- Phases of the Application
- Result
- Conclusion
- References

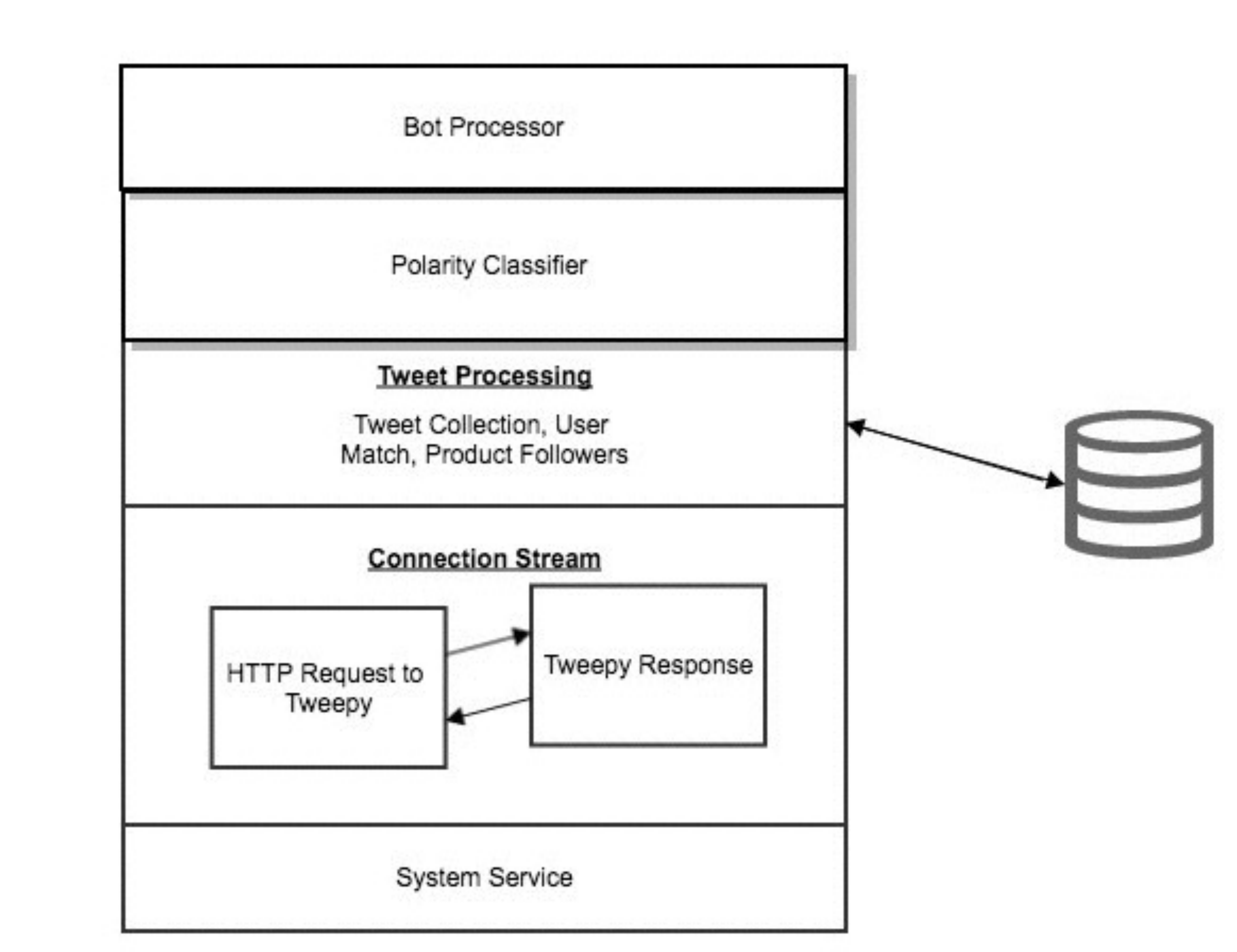
PROBLEM STATEMENT

- Increased popularity in using social media for not only social networking but also for microblogging.
- Twitter is a popular website that provides the above dual functionality, and hence is used for advertising products.
- Manual efforts of advertisement lead to inefficiencies in cost. time and energy.
- In order to avoid the above overhead, we analyse tweets on the basis of sentiment and keywords to identify target audience. (Sentiment Analysis and Opinion Mining)
- Twitter Bots use this information to tweet and tag potential customers who are likely to remember the product through these advertisements during purchase. (Automation)

LITERATURE SURVEY

- Current Bot models are utilised for twitter updates at regular intervals of time.
- Sarcasm has been found to be a major problem, leading to futile bot tweets.
- Analysing parts of the tweet can give a better picture of what a customer thinks of a product, rather than capturing and processing the whole tweet.
- This result is used to direct the bot towards a more useful tweeting mechanism, and hence tagging the right people for the right posts.
- This can lead to a bigger and clearer “Available Market Size” for the product.

INTERNAL DESIGN (1) - SYSTEM DESIGN



INTERNAL DESIGN (1) - SYSTEM DESIGN

- Layer 1 - System Service: invokes all other components of the system, provides a command line for a search term, to start finding tweets.
- Layer 2 - Connection Stream: Connection to the twitter servers via the Twitter API.
- Layer 3 - Tweet Processing: Gathers tweets related to search term, as well as other tweet information like tweet_id and screen_name.
- Layer 4 - Polarity Classifier: Classify and categorise tweets based on whether they are positive, neutral or negative.
- Layer 5 - Bot Processor: Based on results from the polarity classifier, the bot processor processes the right users to tag during advertisement generation.

INTERNAL DESIGN(2) - METHODOLOGY ADAPTED

1. Consider a use case where a company sells “tennis balls”. The company starts searching for all tweets related to “tennis balls”.

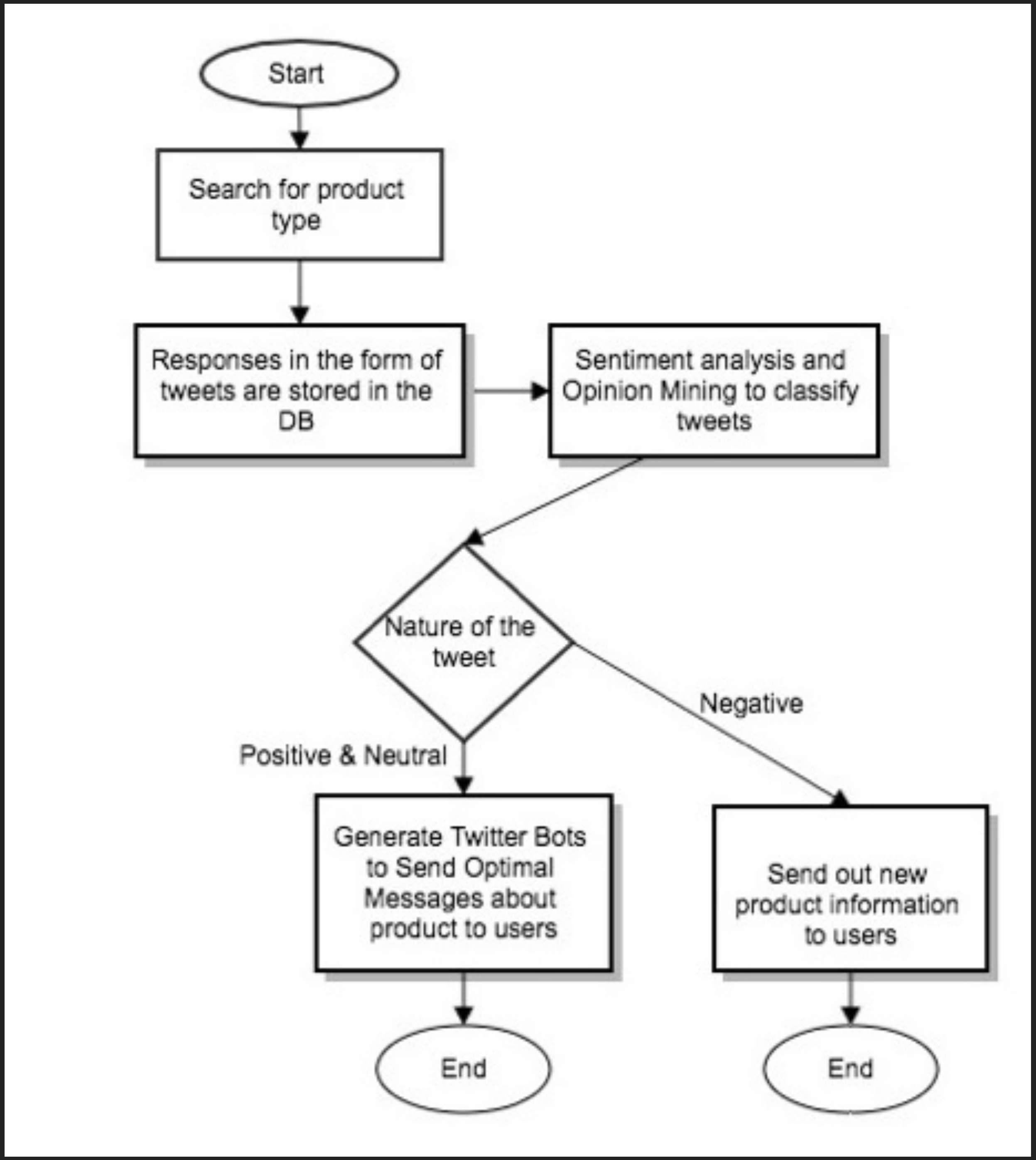
2. Tweets related to the search term as well as specific parts of the tweets like hashtags, user id and screen name are stored in the database.

3. Sentiment analysis is carried out by analysing these tweets to classify them as positive, neutral or negative.

4. When the company advertises for a product, based on the above classification, it tags the right users for the right post.

5. Positive and Neutral tweeters for timely adverts, Negative tweeters for NPD.

INTERNAL DESIGN(2) – FLOW OF THE MODEL


Start
Search for product type
Responses in the form of tweets are stored in the DB
Sentiment analysis and Opinion Mining to classify tweets
Nature of the tweet
Positive & Neutral
Negative
Generate Twitter Bots to Send Optimal Messages about product to users
Send out new product information to users
End
End

HARDWARE AND SOFTWARE REQUIREMENTS

- User Interfaces
  - Web Application created using HTML, CSS, Javascript, PHP.
  - Developer code for data extraction and processing using Python with Anaconda.
- Hardware Interfaces
  - No hardware dependent code will be written by the team in Phase 1 of the project.
  - Hardware Interface of the system to be handled by the Mac OS and Windows 10 operating systems.

EXPECTED OUTCOME

Expected Advantages

- Successful approximate classification of tweets on the basis of input tweets based on historic statistics.
- Bots tweeting to specific parts of the target audience based on results.
- Application designed for customers to easily customise their bots based on their product.
- Using negative tweets for the first time to produce a positive outcome for the company.
- Computing extended to business, through the use of bots for marketing.

Expected Disadvantages

- Complete classification cannot be made due to the varying nature of the tweet.
- There may be many millions of expressions not available for classification from the database.

EXPECTED OUTCOME

Future Works (outside the project's scope)

- Different parts of the tweet, other than specific may be used for sentiment analysis.
- Pre existing websites can be ignored to obtain sentiments into the database.
- An independent machine learning algorithm may be developed for efficient opinion mining and sentiment analysis.

PROJECT PLANNED TIMELINE

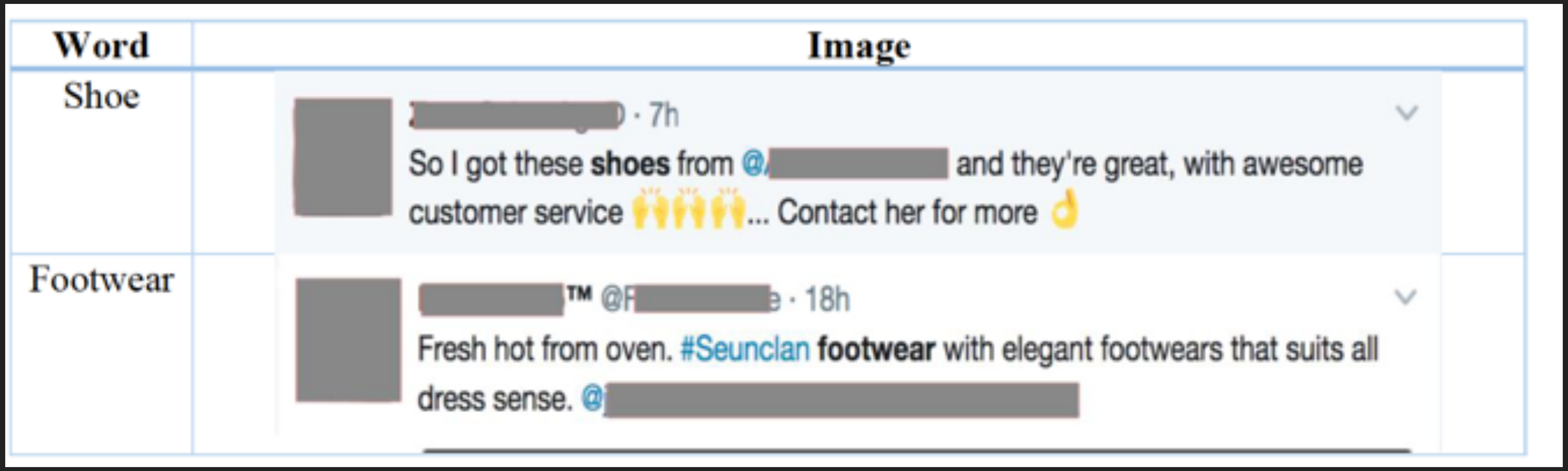
- Review 2 (65% Implementation)
  - Collection and Processing of Data
  - Sentiment Analysis of Tweets
- Review 3 (35% Implementation)
  - Creation of Automated Bots
  - Creation of Web Application

PHASES OF THE APPLICATION

- Data Collection - Tweepy, MongoDB
- Data Processing - TextBlob (Sentiment Analysis)
- Bot Processing - Python + Tweepy
- All the above phases uses Python 2.7 along with additional modules installed. For more information, refer to the README.txt file.

DATA COLLECTION

- We try to gather twitter data whose search terms are similar to the product being sold.
- Eg.if a company is selling shoes, we try to capture tweets about users review about a particular shoe. However, when a person is tweeting about a shoe, he may not always use the word shoe. Hence, we tried to use words that are synonymous to shoes like [boots, footwear].

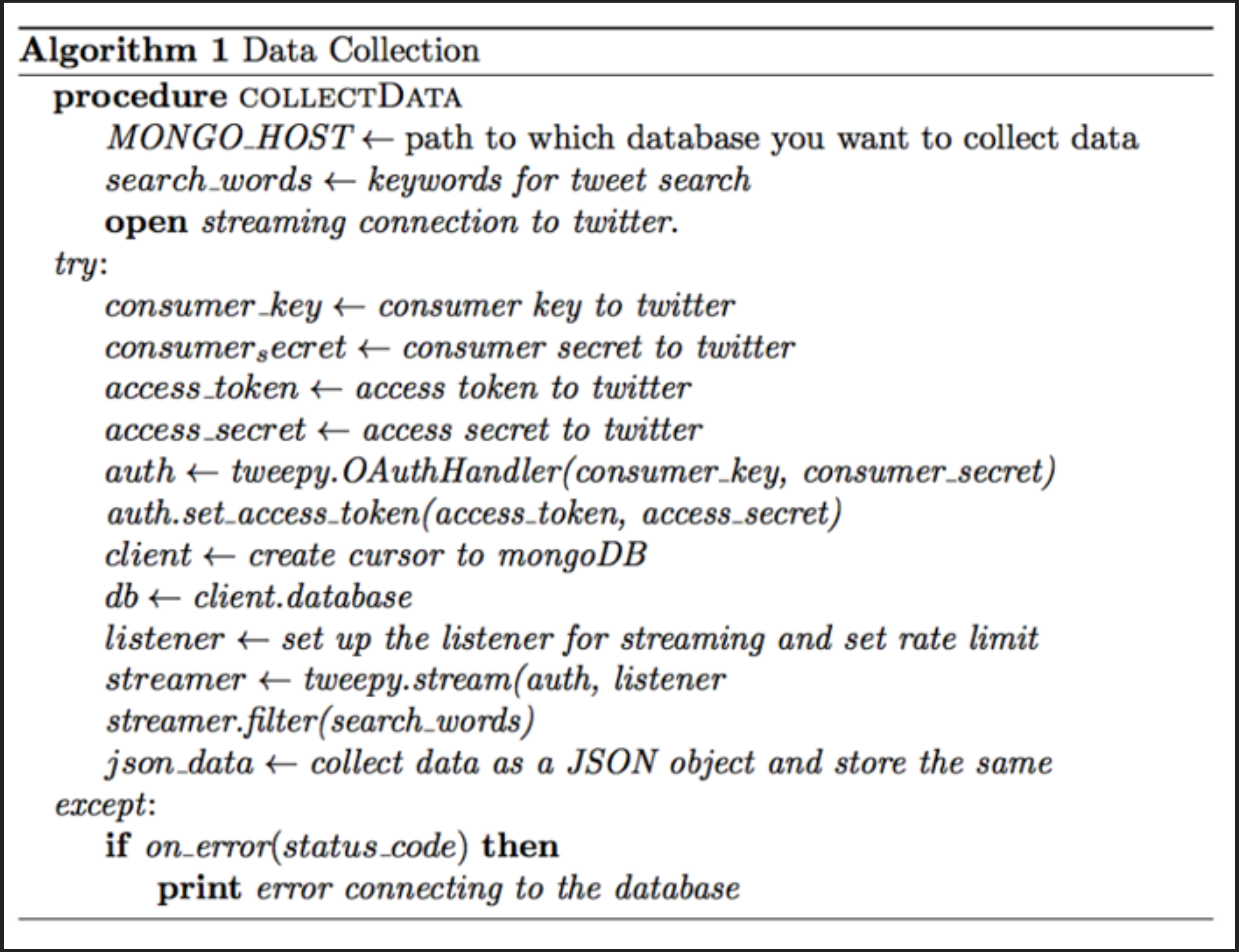


Word	Image
Shoe	[redacted] · 7h So I got these **shoes** from @[redacted] and they're great, with awesome customer service 🙌🙌🙌... Contact her for more 👌
Footwear	[redacted]™ @[redacted] · 18h Fresh hot from oven. #Seunclan **footwear** with elegant footwears that suits all dress sense. @[redacted]

DATA COLLECTION

- We used the tweepy API written in Python, due to the support and the features it provides.
- We also preferred to use the streaming API as compared to the REST API to access live stream data by establishing a continuous HTTP connection.
- We ran the streaming API and collected around 2000 tweets from users. In order to store these tweets, we used MongoDB, which in turn neatly stores this data in the form of a JSON object. This allows for easy parsing of the components of twitter data such as text, hashtags, user_id, screen_name.

PSEUDO CODE FOR DATA COLLECTION



Algorithm 1 Data Collection

```
procedure COLLECTDATA
    MONGO_HOST ← path to which database you want to collect data
    search_words ← keywords for tweet search
    open streaming connection to twitter.
try:
    consumer_key ← consumer key to twitter
    consumer$_s$ecret ← consumer secret to twitter
    access_token ← access token to twitter
    access_secret ← access secret to twitter
    auth ← tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_secret)
    client ← create cursor to mongoDB
    db ← client.database
    listener ← set up the listener for streaming and set rate limit
    streamer ← tweepy.stream(auth, listener
    streamer.filter(search_words)
    json_data ← collect data as a JSON object and store the same
except:
    if on_error(status_code) then
        print error connecting to the database
```

DATA PROCESSING

- Extraction
  - Text from the JSON object is extracted into Python by using simple list methods, like list.append and list.insert.
  - This ensures that only the textual matter from the JSON object is removed and available for processing.
  - The JSON object format makes the text readable.
- Preprocessing and Tokenisation
  - In order to pre process the collected tweets, we use regular expressions to separate the hashtags, emoticons, tags, hyperlinks and normal text.

DATA PROCESSING

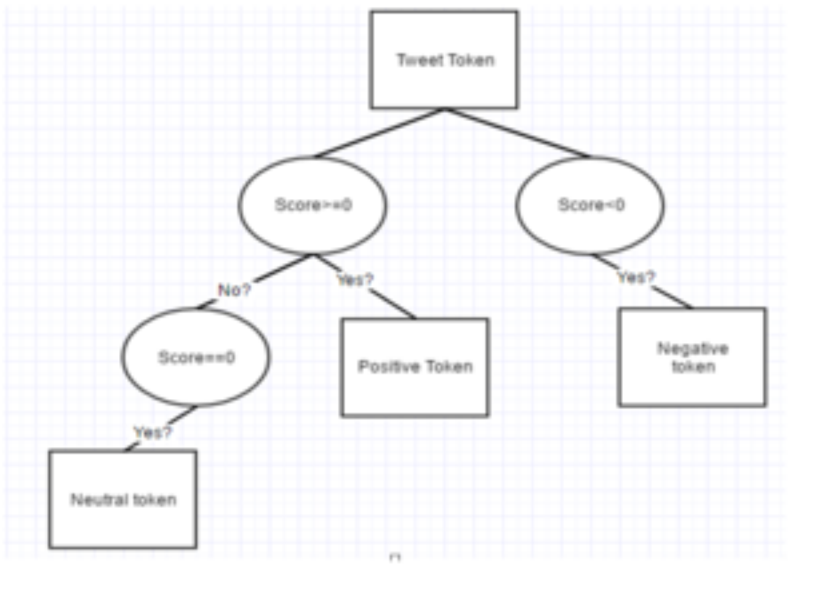
  - The text is then tokenised- separated into individual words, after removing stop words like prepositions and articles.

- Analysis

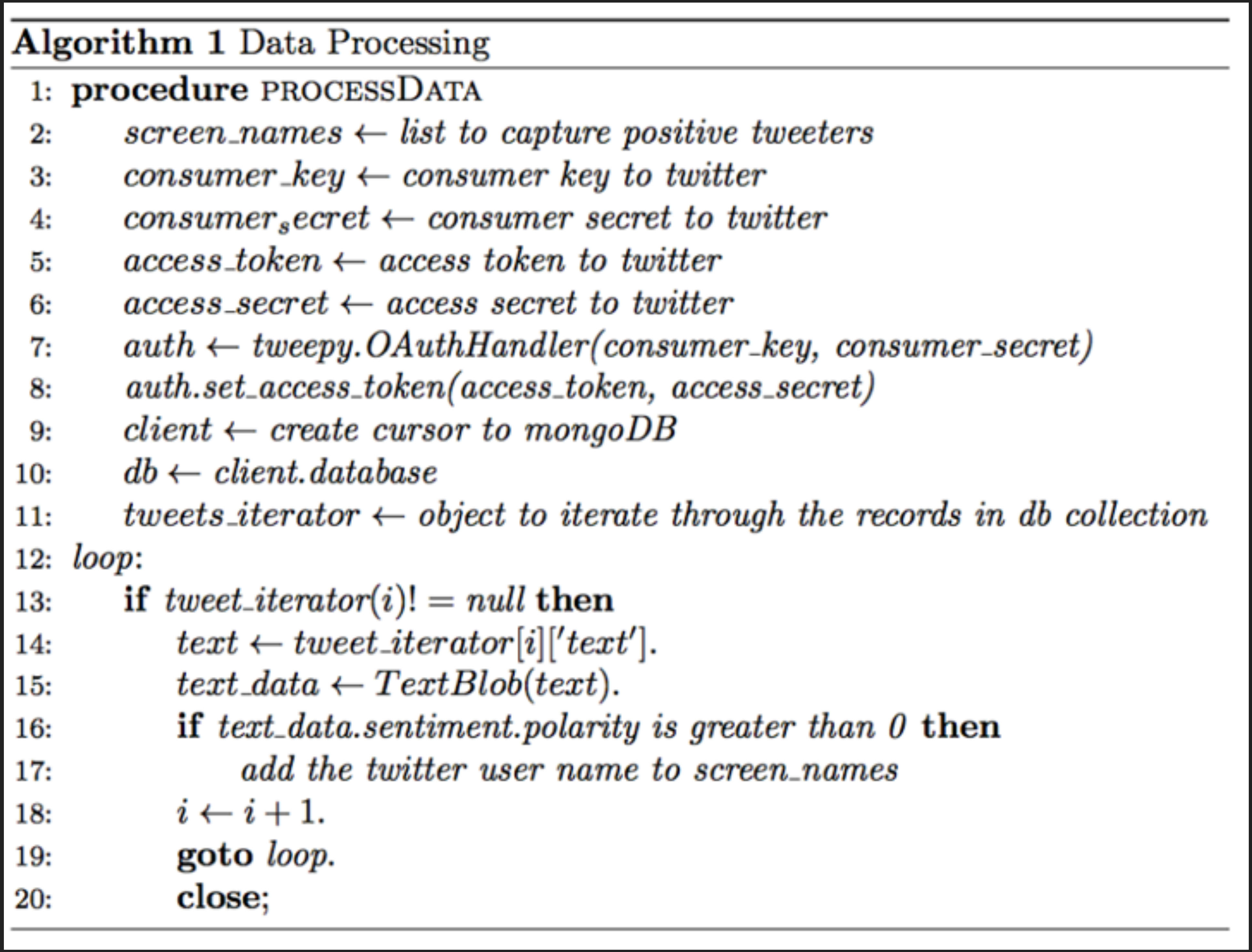
  - Analysis of text has been done using the textblob API.
  - Lexicon analysis, the next step, basically utilises a pre existing lexicon (or bag) of words to compare every word in a given tweet. Assuming the tweet contains the word “awesome”, the lexicon helps to classify this as a positive word, and vice versa for a negative term.
  - The next step involves, this involves checking every word in the tweet, and calculating its positivity or negativity by means of scoring.

DATA PROCESSING

- The bag of words method otherwise follows the Naïve Bayes Classification algorithm. This means that it uses the common probabilistic method to classify data as positive and negative.
- P is the polarity, P(C) is the probability of d occurring in class C, and count (d,C) is the number of times d has occurred in class C.



PSEUDO CODE FOR DATA PROCESSING



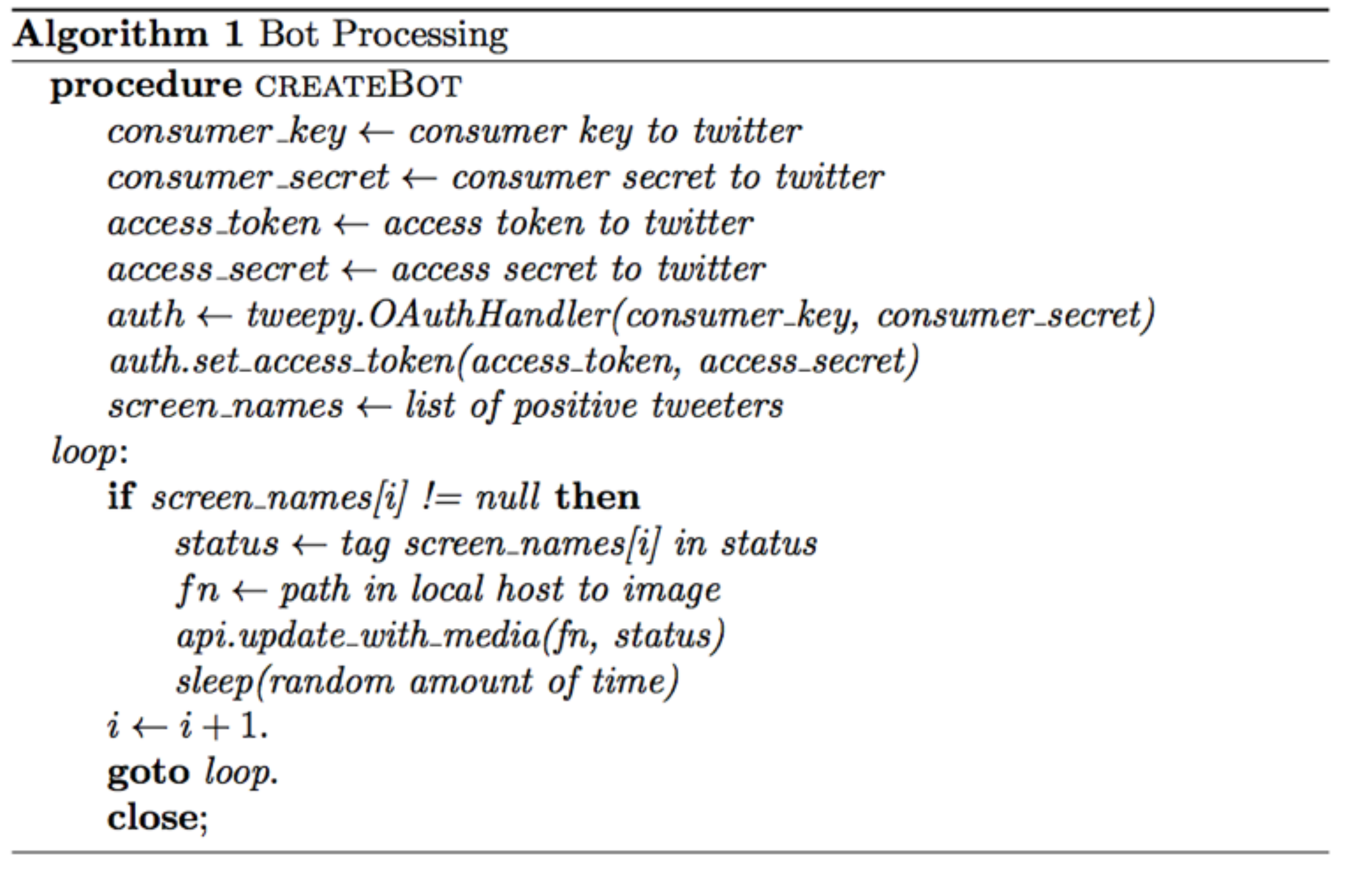
Algorithm 1 Data Processing

```
procedure ProcessData
    screen_names ← list to capture positive tweeters
    consumer_key ← consumer key to twitter
    $consumer_secret$ ← consumer secret to twitter
    access_token ← access token to twitter
    access_secret ← access secret to twitter
    auth ← tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_secret)
    client ← create cursor to mongoDB
    db ← client.database
    tweets_iterator ← object to iterate through the records in db collection
loop:
    if tweet_iterator(i)! = null then
        text ← tweet_iterator[i]['text'].
        text_data ← TextBlob(text).
        if text_data.sentiment.polarity is greater than 0 then
            add the twitter user name to screen_names
        $i \leftarrow i + 1$.
        goto loop.
        close;
```

BOT PROCESSING

- We use twitter bots (which were traditionally used for spamming) to our advantage.
- These twitter bots tag screen names to promote a product of the company by sending each user a tweet with an image about the product.
- In order to keep to the tweet limit, these bots send tweets after a random time interval to each of the twitter users from the screen name data set.
- In doing this, we make sure that the users remember the product name and eventually as their sent posts, they might tend to buy the product.

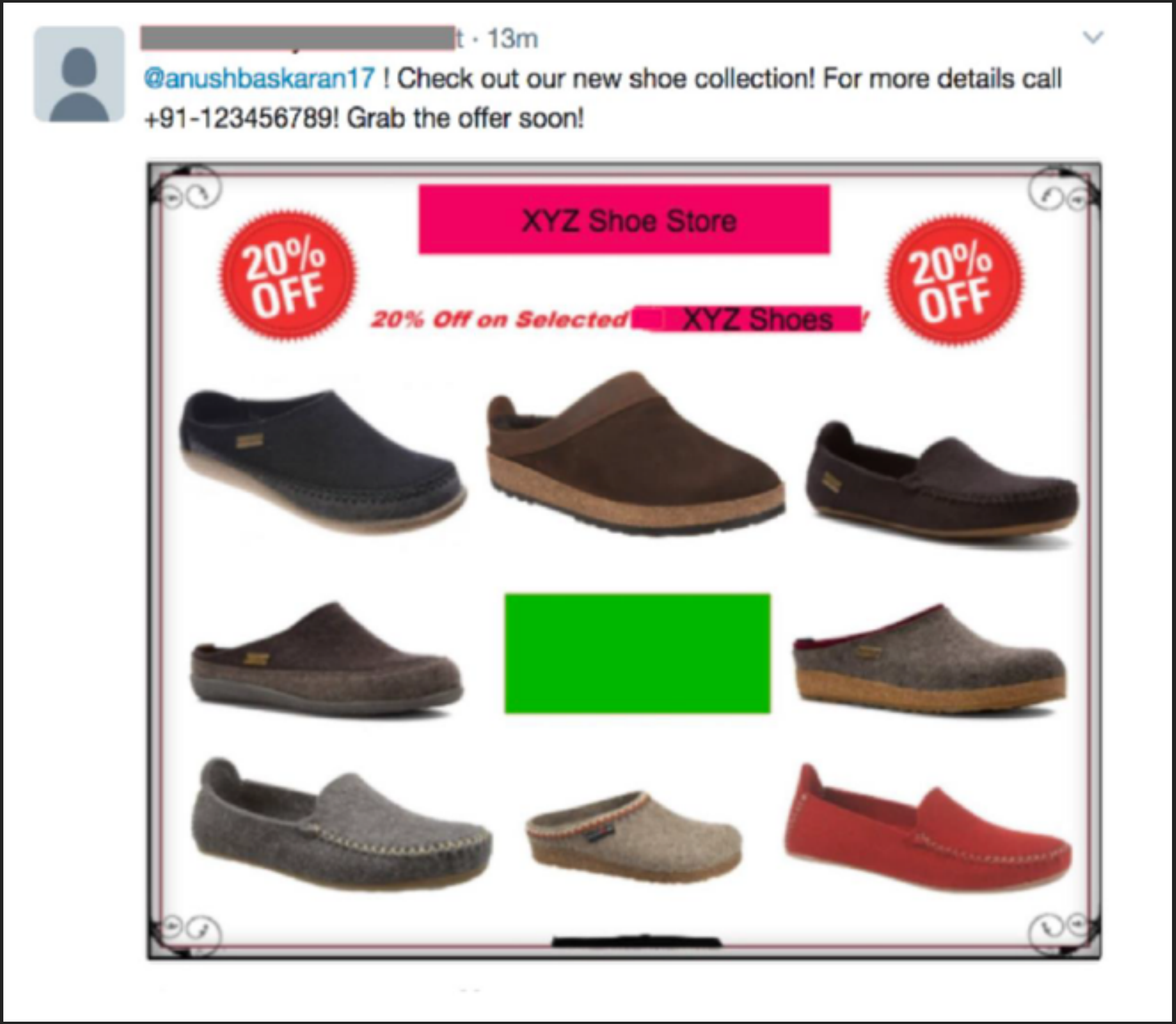
PSEUDO CODE FOR BOT PROCESSING



Algorithm 1 Bot Processing

```
procedure CREATEBOT
    consumer_key ← consumer key to twitter
    consumer_secret ← consumer secret to twitter
    access_token ← access token to twitter
    access_secret ← access secret to twitter
    auth ← tweepy.OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_secret)
    screen_names ← list of positive tweeters
loop:
    if screen_names[i] != null then
        status ← tag screen_names[i] in status
        fn ← path in local host to image
        api.update_with_media(fn, status)
        sleep(random amount of time)
    i ← i + 1.
    goto loop.
    close;
```

RESULT



[redacted]t · 13m

@anushbaskaran17 ! Check out our new shoe collection! For more details call +91-123456789! Grab the offer soon!

CONCLUSION

- Bots can be utilised for a positive outlook in terms of marketing. The bot can be tailored to work towards attracting a customer base who is genuinely interested in utilising the product after analysing their sentiment towards it.
- This has been found to reduce spam, owing to the selective nature of delivering its advertisements. Also, having sentiments of the users known, the user has ample scope to review, revise and upgrade their products.
- The efficiency in terms of time and energy has also been optimised by automating the process using bots. This in turn helps in reducing delays, and cuts monetary costs too.

REFERENCES

- Weblink's:
    - https://marcobonzanini.com/2015/03/02/mining-twitter-data-with-python-part-1/
    - http://hanj.cs.illinois.edu/bk3/bk3_slidesindex.htm
    - http://dmml.asu.edu/smm/slides/
    - http://ecs.syr.edu/faculty/reza/tutorials/ICDM13/TutorialICDM13SMM.pdf
    - http://blogs.ischool.berkeley.edu/i290-abdt-s12/
- Journal / Conference Papers:
    - “Sarcasm Detection on Twitter: A Behaviour Modelling Approach”, Proceedings of the Eighth ACM Conference on Web Search and Data Mining. Pp 97-106.

REFERENCES

- Journal / Conference Papers:
  - “Behaviour Analysis on Social Media”, IEEE Intelligent Systems, Vol.29, Issue 4, 2014, Pp 9-11.
  - “The Rise of Social Bots”, Available: https://arxiv.org/abs/1407.5225.
  - “Detecting and Analysing automated activity on Twitter”, International Conference on Passive and Active Network Measurement, 2011, Pp 102-111.