BrAna – Brand Analysis Platform

Group No. 11

Project Guide:
Dr. A. J. Agrawal

Group Members:
Megha Gupta (**ROLL NO. 12**)
Toshal Patel (**ROLL NO. 31**)

Current Scenario

PROBLEMS FACED

- Each brand/company needs basic strategies for excelling among their competition.
- Social Media in today's world is an excellent and prime means to reach maximum people.
- Many startups/ small scale companies cannot understand the actual reactions of people.
- Many online, offline surveys conducted are not very honest, does not reflect true customer emotions

NEED

- Branding
- Competition
- Understand the audience.
- Identify and improve growth opportunities (Sentiment analysis and data driven strategies).
- Connect to the customers and Expand the market

BrAna

What is Brana?

It is a brand analysis platform which helps the user to understand the audience reaction towards the company's products.

Methodology

How will BrAna be built?

BrAna consits of three phases –

PHASE 1 – Building the Data Set

Extracting the twitter data related to the brand/ company/ startup.

- **PYTHON**
- **REST- API**
- **NLP tools developed by University of Washington**
- **Linux**

PHASE 2 – Named Entity Recognition

Performing Named Entity Recognition, identifying the entities and chunking out the important data.

- **PYTHON**
- **NLP tools developed by University of Washington**
- **Linux**

PHASE 3 – Sentiment Analysis

Performing Sentiment Analysis on chunks of data for extracted, and report the results.

- **PYTHON**
- **Linux**

Example for Brand Analysis

“Most products launched by Ramdevâ™ Patanjali are below standard! Many fail quality test, RTI inquiry finds... https://t.co/4PLtDf2ias.”

Helps in brand analysis
Key terms –

- Patanjali
    - Fails
    - Below standard

Targets

Present state:

- Twitter Data extraction.
- Background work required for Named-Entity Recognition – Research and algorithms.

By the end of this semester:

Entire execution of Named-Entity Recognition for Twitter Data.

By next semester:

- Sentiment Analysis for Brands.
- Compilation of all phases.

Building the Data Set

Python and REST API

Using REST-API to extract data from Twitter

```python
    import simplejson as json

# Import the necessary methods from "twitter" library
from twitter import Twitter, OAuth, TwitterHTTPError, TwitterStream

# Variables that contains the user credentials to access Twitter API
ACCESS_TOKEN = '906776625932410881-eObfTfRaV6rRO1CK78ipjUZt5W2TFRV'
ACCESS_SECRET = 'A4BM1OFhYCkJgi6LubbD4I95cjltPbpG1zyYAdaEX5QAx'
CONSUMER_KEY = 'tC3spu2BsCON7qTe7jXL0adYO'
CONSUMER_SECRET = 'CnmhWML4XfQ7Qk78NYMhu9j0fEs6bwsbXjczgPPUuKEh7kkl0k'

oauth = OAuth(ACCESS_TOKEN, ACCESS_SECRET, CONSUMER_KEY, CONSUMER_SECRET)

# Initiate the connection to Twitter Streaming API
twitter_stream = TwitterStream(auth=oauth)

# Get a sample of the public data following through Twitter
iterator = twitter_stream.statuses.filter(track="Patanjali", language="en")

# Create a file to write all the tweets
fhandle = open("tweets_patanjali.txt", "w")

# Print each tweet in the stream to the screen
# Here we set it to stop after getting 1000 tweets.
# You don't have to set it to stop, but can continue running
# the Twitter API to collect data for days or even longer.
tweet_count = 1000
for tweet in iterator:
    tweet_count -= 1
    # Twitter Python Tool wraps the data returned by Twitter
    # as a TwitterDictResponse object.
    # We convert it back to the JSON format to print/score
    fhandle.write(json.dumps(tweet))

    # The command below will do pretty printing for JSON data, try it out
    # print json.dumps(tweet, indent=4)

    if tweet_count <= 0:
        break
```

Twitter data extracted

Using ‘jq’ for extracting ‘text’ of tweets json
jq is like sed for JSON data - you can use it to slice and filter and map and transform structured data

jq query –

type tweetspatanjali.txt | jq -r ".text".

{"created_at": "Fri Oct 06 18:19:11 +0000 2017", "id": 916367245902618624, "id_str":
"916367245902618624", "text": "@AmitShah And help patanjali to grow ....",
"display_text_range": [10, 46], "source": "<a href=\"http://twitter.com/download/android\"
rel=\"nofollow\">Twitter for Android", "truncated": false, "in_reply_to_status_id":
916342460003086337, "in_reply_to_status_id_str": "916342460003086337", "in_reply_to_user_id":
1447949844, "in_reply_to_user_id_str": "1447949844", "in_reply_to_screen_name": "AmitShah",
"user": {"id": 820109877045592064, "id_str": "820109877045592064", "name": "SATYAMEVA JAYATE"
, "screen_name": "GROUPTOBEMADE", "location": null, "url": null, "description": null,
"translator_type": "none", "protected": false, "verified": false, "followers_count": 5,
"friends_count": 102, "listed_count": 0, "favourites_count": 2, "statuses_count": 2,
"created_at": "Sat Jan 14 03:26:46 +0000 2017", "utc_offset": null, "time_zone": null,
"geo_enabled": false, "lang": "en", "contributors_enabled": false, "is_translator": false,
"profile_background_color": "F5F8FA", "profile_background_image_url": "",
"profile_background_image_url_https": "", "profile_background_tile": false,
"profile_link_color": "1DA1F2", "profile_sidebar_border_color": "C0DEED",
"profile_sidebar_fill_color": "DDEEF6", "profile_text_color": "333333",
"profile_use_background_image": true, "profile_image_url": "http://pbs.twimg.com
/profile_images/897056843582758912/fbX8xb-q_normal.jpg", "profile_image_url_https": "https
://pbs.twimg.com/profile_images/897056843582758912/fbX8xb-q_normal.jpg", "default_profile":
true, "default_profile_image": false, "following": null, "follow_request_sent": null,
"notifications": null}, "geo": null, "coordinates": null, "place": null, "contributors": null
, "is_quote_status": false, "quote_count": 0, "reply_count": 0, "retweet_count": 0,
"favorite_count": 0, "entities": {"hashtags": [], "urls": [], "user_mentions":
[{"screen_name": "AmitShah", "name": "Amit Shah", "id": 1447949844, "id_str": "1447949844",
"indices": [0, 9]}]}, "symbols": []}, "favorited": false, "retweeted": false, "filter_level":
"low", "lang": "en", "timestamp_ms": "1507313951360"}

1. Part Of Speech (POS) tagging

- Each word assigned to its most frequent tag and assign each Out of Vocabulary (OOV) to the most common POS tag.

My dog also likes eating sausage.

Tagging -

My/PRP$ dog/NN also/RB likes/VBZ eating/VBG sausage/NN ./.

- Tweets contain greater OOVs
  - Eg. the use of the word “n” for “in”

- Drawbacks of state-of-the-art Stanford Tagger:
  - Misclassification due to unreliable capitalization.
  - common nouns misclassified as proper noun, vice versa.
  - Different grammar of tweets – “watching angels and demons”.

```
CC  coordinating conjunction
CD  cardinal digit
DT  determiner
EX  existential there (like: "there is" ... think of it like "there exists")
FW  foreign word
IN  preposition/subordinating conjunction
JJ  adjective   'big'
JJR adjective, comparative  'bigger'
JJS adjective, superlative  'biggest'
LS  list marker 1)
MD  modal   could, will
NN  noun, singular 'desk'
NNS noun plural 'desks'
NNP proper noun, singular   'Harrison'
NNPS    proper noun, plural 'Americans'
PDT predeterminer   'all the kids'
POS possessive ending   parent's
PRP personal pronoun    I, he, she
PRP$    possessive pronoun  my, his, hers
RB  adverb  very, silently,
RBR adverb, comparative better
RBS adverb, superlative best
RP  particle    give up
TO  to  go 'to' the store.
UH  interjection    errrrrrrrm
VB  verb, base form take
VBD verb, past tense    took
VBG verb, gerund/present participle taking
VBN verb, past participle   taken
VBP verb, sing. present, non-3d take
VBZ verb, 3rd person sing. present  takes
WDT wh-determiner   which
WP  wh-pronoun  who, what
WP$ possessive wh-pronoun   whose
WRB wh-abverb   where, when
```

1. Part Of Speech (POS) tagging (contd.)

- Lexical variations -

‘2m’, ‘2ma’, ‘2mar’, ‘2mara’, ‘2maro’,‘tmmrw’, ‘tmo’, ‘tmoro’, ‘tmorrow’, ‘tmoz’, ‘tmr’, ‘tmro’, ‘tmrow’, ‘tmrrow’, ‘tmrrw, ‘tomorro’, ‘tomorrw’, ‘tomoz’, ‘tomrw’, ‘tomz’

▷ T-POS : POS tagging system with new tags for retweets, @usernames, #hashtags, urls.

- Uses Contitional Random Fields

Conditional random fields (CRFs) are a probabilistic framework for labelling and segmenting structured data, such as sequences, trees and lattices. Often applied in pattern recognition and machine learning and used for structured prediction.

- Brown clusters

(A method used to create **clusters** of words that are similar. It is an instance of a **Clustering** algorithm which generates a hierarchical **cluster** of words.)

1. Part Of Speech (POS) tagging (contd.)

Brown clusters example

Friday Monday Thursday Wednesday Tuesday Saturday Sunday weekends Sundays Saturdays
June March July April January December October November September August
people guys folks fellows CEOs chaps doubters commies unfortunates blokes
down backwards ashore sideways southward northward overboard aloft downwards adrift
water gas coal liquid acid sand carbon steam shale iron
great big vast sudden mere sheer gigantic lifelong scant colossal
man woman boy girl lawyer doctor guy farmer teacher citizen
American Indian European Japanese German African Catholic Israeli Italian Arab
pressure temperature permeability density porosity stress velocity viscosity gravity tension
mother wife father son husband brother daughter sister boss uncle
machine device controller processor CPU printer spindle subsystem compiler plotter
John George James Bob Robert Paul William Jim David Mike
anyone someone anybody somebody
feet miles pounds degrees inches barrels tons acres meters bytes
director chief professor commissioner commander treasurer founder superintendent dean custodian
liberal conservative parliamentary royal progressive Tory provisional separatist federalist PQ
had hadn't hath would've could've should've must've might've
asking telling wondering instructing informing kidding reminding bothering thanking deposing
that tha theat
head body hands eyes voice arm seat eye hair mouth

1. Part Of Speech (POS) tagging (contd.)

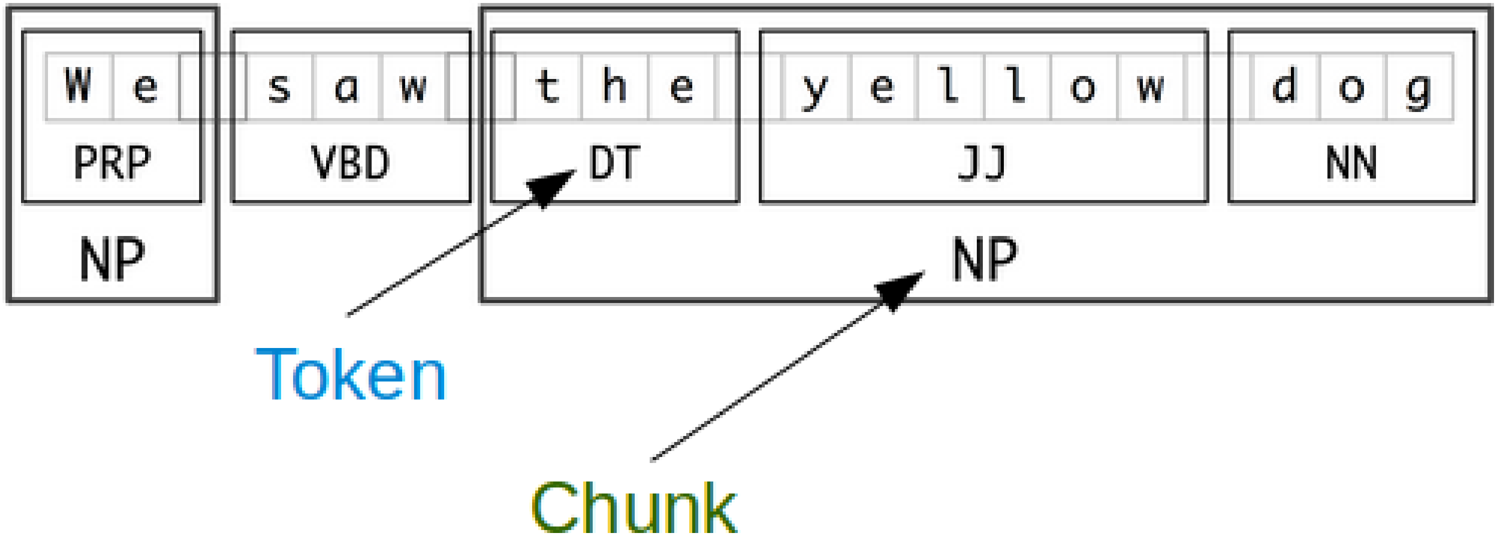
- POS dictionaries
- Spelling and contextual features

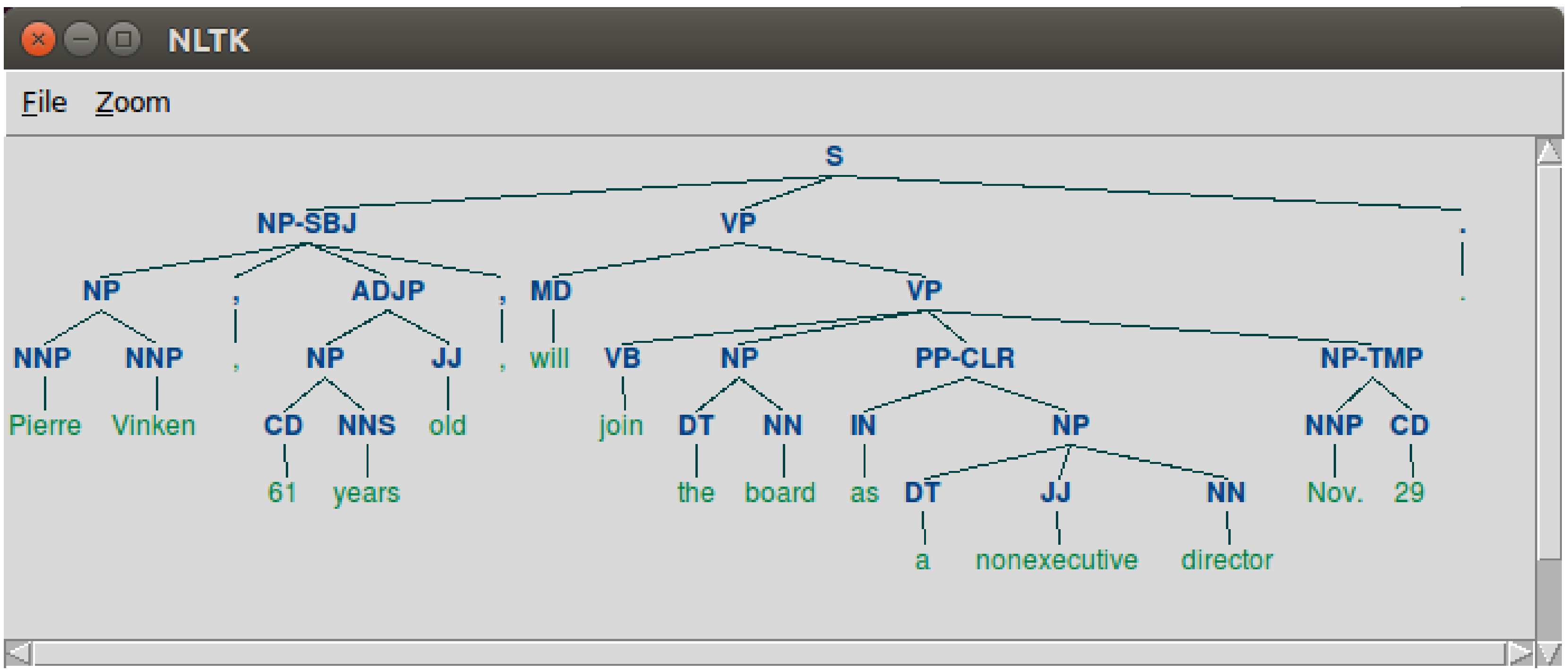
▷ Better than Stanford tagger, obtaining 26% reduction in error.

2. Shallow Parsing (or Chunking)

- Identifying non-recursive phrases – noun, verb, prepositional phrases in text
- Parser forms tree, POS tagger returns the lowest level of the tree, but, Shallow Parsing returns a specific part of the tree (eg. noun phrase) and eliminated the need to form an entire parsing tree.
- For example an adjective and a noun might combine to be a 'Noun Phrase', which might combine with another adjective to form another Noun Phrase (e.g. quick brown fox) (the exact way the pieces combine depends on the parser in question).

Source: Research paper, Named Entity Recognition in Tweets: An experimental study.


W e
s a w
t h e
y e l l o w
d o g
PRP
VBD
DT
JJ
NN
NP
NP
Token
Chunk


NLTK
File
Zoom
S
NP-SBJ
VP
.
NP
,
ADJP
,
MD
VP
.
NNP
NNP
,
NP
JJ
,
will
VB
NP
PP-CLR
NP-TMP
Pierre
Vinken
CD
NNS
old
join
DT
NN
IN
NP
NNP
CD
61
years
the
board
as
DT
JJ
NN
Nov.
29
a
nonexecutive
director

3. Capitalization

- Key feature for recognizing named entities is capitalization, but tweets have random capitalization.
- **T-CAP** – capitalization classifier

- Either *informative* or *uninformative* labels
  - Uninformative – non-entity capitalized words, or entity words which are not capitalized.

- Use Support Vector Machines for learning
  - Features used –
    - the fraction of words in the tweet which are capitalized,
    - the fraction which appear in a dictionary of frequently lowercase/capitalized words but are not lowercase/capitalized in the tweet,
    - the number of times the word 'I' appears lowercase
    - whether or not the first word in the tweet is capitalized.

Named Entity Recognition

- The state-of-the-art Stanford NER performs poorly on Twitter data – misclassifying data.
- Here, treating classification and segmentation as separate tasks.
- Using large number of randomly sampled tweets, because most words found in tweets are not part of an entity.

1. Segmenting Named Entities

- **T-SEG** – a sequence labelling task using IOB encoding for representing segments
- Each word either begins, is inside, or outside of a named entity.
- Conditional Random Fields for learning and inference.
- Brown clusters, and output of **T-POS, T-CHUNK,** and **T-CAP** are used in generating features.

2. Classifying Named Entities

- Twitter data contains many distinctive infrequent entity types, in any random sample of tweets, many types will occur a few times.
- Individual tweets often do not contain enough context to determine the type of entity. (Problem of many infrequent types)
- “KKTNY in 45 mins……….” – without any prior knowledge it is not enough context to determine what kind of entity “KKTNY” refers to.
- But we can determine it to be a TV show since it often occurs with verbs *watching* or *premieres.*

- For this problem – leverage large lists of entities and their types gathered from any open-domain ontology as a source of distant supervision, allowing large amount of unlabelled data in learning.

- Apply LabeledLDA – makes each entity string as a mixture of types rather than single hidden variable to represent the type of each mention.

References:

- Source: Research paper, Named Entity Recognition in Tweets: An experimental study.

- http://blog.echen.me/2012/01/03/introduction-to-conditional-random-fields/

- https://github.com/aritter/twitter<underscore>nlp