CSI5386

Natural Language Processing

Assignment 1

Social media corpus analysis

Submitted By:

Deepshi Mediratta 300086013

Nikhil Oswal 300074118

Part 1: Corpus processing: tokenization, word counting, and multi-word expressions

PROBLEM STATEMENT

Implement a word tokenizer for Twitter messages that splits the text of the messages into tokens and separates punctuation marks and other symbols from the words.

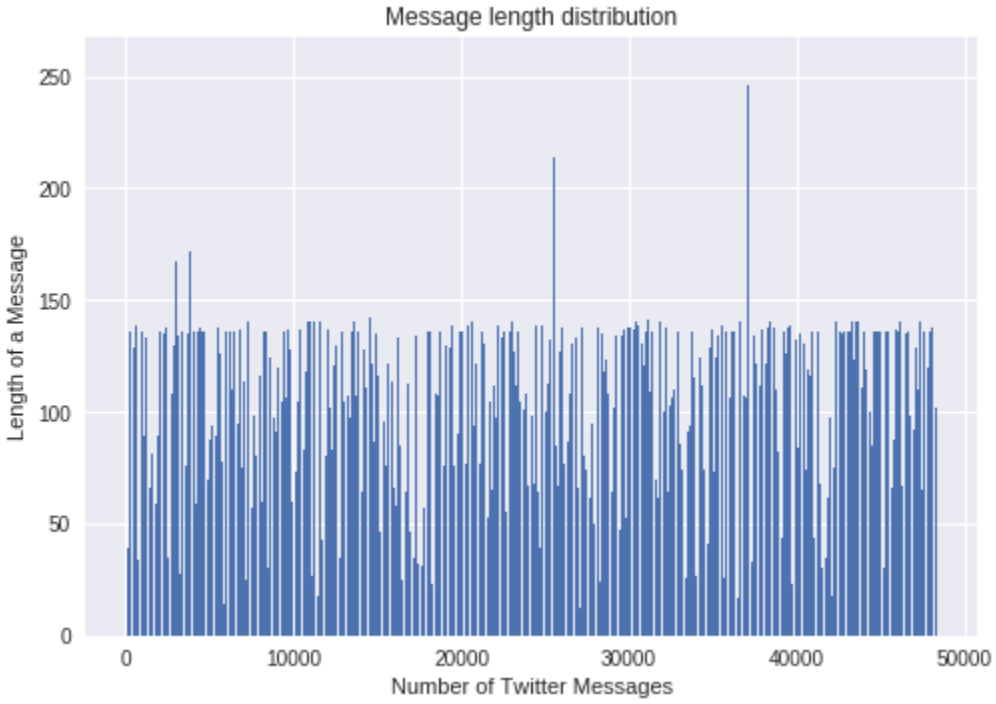
SOLUTION

1. **DATASET ANALYSIS**

   Initially, we performed some exploratory data analysis on the corpus to observe the structure of the data. The Twitter corpus was composed of 48,401 tweets (including retweets). The messages are saved in a text file where each line represents a single Twitter message.

   The average length of messages in the corpus is 16 words or 117 characters.

   Twitter has a limit of 140 characters per message. We observe from the distribution of length of messages that a few of them clearly exceed this limit. We decide to remove this from our dataset as these are unlikely to be a twitter message.



We observe the quality of the microblog text and following are the challenges:

- @user: represents the user who posted the tweet or the user name who was tagged in the tweet.
  Example: "@HannahDevJobs Groubal gives power to frustrated consumers. Tell us about your customer service complaint today. http://www.groubal.com"
  In this message, @HannahDevJobs is a user name which does not reveal any meaningful information to us.
- URLs: Profile links/image URL/Link to a tweet.
  Example: In the example above, http://www.groubal.com, represents a link to a website.
- Users have made spelling mistakes.
  Example: "C'mon, get into the Android world.."
- Users have used emoticons to express their views and emotions." Example: ♪♪, ♥, *-*
- Language used by the users is very informal, sometimes the messages are a mix of two languages.
  Example: "BBC-news更新 Devine denies expenses 'deceit' http://bbc.in/hQAywP #bbc". This message is a mix of English and Chinese Words.
- The text consists of misspellings, slangs, abbreviations.
- Usage of repetitive punctuation marks to emphasize emotions.
  Example: The usage of many exclamation marks in this message "The game is over the world big game so over too!!!!!!!!!!"
- A mix of lowercase and uppercase words are used to emphasize emotions.
  Example: "good thing this game RULES"

This proves that the corpus that we have has unstructured text, it contains a significant amount of noise which may not contain any useful information. If we want to analyze the text to solve some business problem, we need to improve the quality of the text that we are working with. We remove any irrelevant information such as HTML characters, emoticons, slangs, punctuations etc.

2. **TOOLS USED**

We used BeautifulSoup, Contractions, Numpy, Matploptlib, NLTK, Textblob and CMU's POS Tagger for this exercise.

3. **PREPROCESSING**

Pre-processing techniques used by us to clean the data for analysis:

a) Remove HTML tags

The corpus contains certain messages which are wrapped within HTML tags. These HTML tags are not relevant for text analysis. So, we remove the HTML tags and fetch the text that is wrapped in between them.

Original Message: """#1 China Wholesale Store Clothing/Shoes/Handbags/MP4/DVDs/ Cell Phones/Watches/Electronics."""

Cleaned Message: #1 China Wholesale Store Clothing/Shoes/Handbags/MP4/DVDs/ Cell Phones/Watches/Electronics.

b) **Remove Numbers**

Very few messages contain numbers, it is safe to decide that we can remove this from the text.

Original Message: http://bit.ly/8FNoOE United Continental to cut up to 500 jobs in Houston http://tinyurl.com/4ple5t6 http://bit.ly/8FNoOE

Cleaned Message: http://bit.ly/FNoOE United Continental to cut up to jobs in Houston http://tinyurl.com/plet http://bit.ly/FNoOE

c) **Remove Escape characters**

We used 'utf-8-sig' encoding while reading the text from .txt files to ensure that we do not capture the escape characters such as '&' but the actual character that it represents that is, an ampersand.

Original Message: NYR Organic- Organic Beauty Inside & Out! http://bit.ly/ha2ZJi

Cleaned Message: NYR Organic- Organic Beauty Inside & Out! http://bit.ly/ha2ZJi

d) **Contraction**

Words like 'don't' and 'wouldn't' are replaced with 'do not' and 'would not'

Original Message: Don't you wish State Farm was really there with a new boyfriend or girlfriend? haa :)

Cleaned Message: do not you wish State Farm was really there with a new boyfriend or girlfriend? haa :)

e) **Lowercase**

We decide to change the complete corpus to lowercase. Having different cases for the same word would lead to redundant words in the token's list which we want to avoid. If we do not do this, it could tamper with the statistics (type/token ratio etc)

Original Message: Good morning world!

Cleaned Message: good morning world!

f) **URLs**

A lot of twitter message in the corpus contains URLs to pages which either do not exist anymore or are irrelevant. These can be removed from consideration.

Original Message: Londoners step up cuts protests http://bit.ly/eRwYD9

Cleaned Message: Londoners step up cuts protests

g) **Remove twitter handle**

In the corpus, we observe that almost all message contains a username, their presence does not reveal any information so we can remove them from the messages to reduce the number of tokens.

Original Message: How exciting! RT @BunchesUK: Hello! What's happening in your world? We're all gearing up for #Valentines with bouquets flying out the door.

Cleaned Message: How exciting! RT : Hello! What's happening in your world? We're all gearing up for #Valentines with bouquets flying out the door.

h) **Remove RT**

A few messages contain the word 'RT' which indicate that the message is a retweet. We keep the message as it could indicate how many times the tweet was passed along. Removing them could de-emphasize the importance of the message. But the word RT does not have any other meaning for us, so it is safe to remove it from the text.

Original Message: How exciting! RT : Hello! What's happening in your world? We're all gearing up for #Valentines with bouquets flying out the door.

Cleaned Message: How exciting! : Hello! What's happening in your world? We're all gearing up for #Valentines with bouquets flying out the door.

4. **TOKENIZATION**

After cleaning the corpus, we deeply analyzed the cleaned data set, that we name "**PreprocessedData.txt".** We pass this cleaned corpus to POS tagger to word tokenize it.

```
Nikhils-MacBook-Pro:ark-tweet-nlp-0.3.2 nikhiloswal$ ./runTagger.sh --model
 model.ritter.txt --output-format conll PreprocessedCorpus.txt >> microblog
2011_tokenized.txt
Detected text input format
Tokenized and tagged 48403 tweets (813031 tokens) in 66.7 seconds: 726.0 tw
eets/sec, 12195.2 tokens/sec
```

The attached file **"microblog2011_tokenized.txt"** is the output for POS Tagger.

MESSAGE 1		
Token	**Tag**	**Confidence**
save	VB	0.9739
bbc	NN	0.504
world	NN	0.9921
service	NN	0.9519
from	IN	0.9966
savage	NNP	0.5006
cuts	VBZ	0.5115

MESSAGE 11		
Token	**Tag**	**Confidence**
if	IN	0.9709
you	PRP	0.9995
are	VBP	0.9940
interested	VBN	0.7357
in	IN	0.9747
professional	JJ	0.8531
global	JJ	0.9055
translation	NN	0.9929
services	NNS	0.8965

MESSAGE 2		
Token	**Tag**	**Confidence**
a	DT	0.9867
lot	NN	0.9462
of	IN	0.9877
people	NNS	0.9180
always	RB	0.9842
make	VB	0.9253
fun	NN	0.5489
about	IN	0.9875
the	DT	0.9969
end	NN	0.9808
of	IN	0.9978
the	DT	0.9963
world	NN	0.9809
but	CC	0.9577
the	DT	0.9956
question	NN	0.9975
is	VBZ	0.9930
..	:	0.9534
"	''	0.9818
are	VBP	0.9796
you	PRP	0.9974
ready	JJ	0.6266
for	IN	0.9979
it	PRP	0.9878
?..	.	0.9338

MESSAGE 3		
Token	**Tag**	**Confidence**
rethink	VB	0.8814
group	NN	0.9405
positive	JJ	0.5425
in	IN	0.9747
outlook	NN	0.9604
:	:	0.9711
technology	NN	0.9666
staffing	NN	0.9582

MESSAGE 12		
Token	**Tag**	**Confidence**
fitness	NN	0.4978
first	JJ	0.6196
to	TO	0.9942
float	VB	0.9494
but	CC	0.9740
is	VBZ	0.9837
not	RB	0.9942
the	DT	0.9858
full	JJ	0.9279
service	NN	0.9077
model	NN	0.9810
dead	JJ	0.5631
?	.	0.9958

MESSAGE 13		
Token	**Tag**	**Confidence**
david	NNP	0.9900
cook	VB	0.6814
!	.	0.9848
has	VBZ	0.9367
the	DT	0.9937
mostest	JJS	0.9608
beautiful	JJ	0.9897
smile	NN	0.9704
in	IN	0.9966
the	DT	0.9978
world	NN	0.9864
!	.	0.9974

MESSAGE 14		
Token	**Tag**	**Confidence**
piss	VB	0.5167
off	RP	0.9379
.	.	0.9949
cnt	MD	0.5828
stand	VB	0.8793
lick	VB	0.7441
asses	NNS	0.6441

specialist	NN	0.9477
the	DT	0.9800
rethink	NN	0.7577
group	NN	0.9962
expects	VBZ	0.9550
revenues	NNS	0.9733
to	TO	0.9952
be	VB	0.9989
“	''	0.4924
marg	NN	0.8365
...	:	0.9934

MESSAGE 15		
Token	**Tag**	**Confidence**
beware	VB	0.2220
the	DT	0.9902
blue	JJ	0.7276
meanies	NNS	0.9777
:	:	0.9722
#cuts	HT	0.9438
#thebluemeanies	HT	0.9695

MESSAGE 4		
Token	**Tag**	**Confidence**
'	''	0.9102
zombie	NN	0.5602
'	''	0.9520
fund	NN	0.8794
manager	NN	0.9803
phoenix	NNP	0.4681
appoints	VBZ	0.8723
new	JJ	0.9895
ceo	NN	0.9547
:	:	0.9783
phoenix	NNP	0.4866
buys	VBZ	0.9604
up	RP	0.7912
funds	NNS	0.9446
that	IN	0.4451
have	VBP	0.5815
been	VBN	0.9889
closed	VBN	0.8279
to	TO	0.9939
new	JJ	0.8854
business	NN	0.9007
and	CC	0.9891
...	:	0.9868

MESSAGE 16		
Token	**Tag**	**Confidence**
como	NNP	0.4831
perde	NN	0.3918
os	NN	0.5374
dentes	VBZ	0.5520
no	DT	0.9176
world	NN	0.9963
of	IN	0.9923
warcraft	NN	0.6148
-	:	0.9639
via	IN	0.7925
alisson	NNP	0.8449

MESSAGE 5		
Token	**Tag**	**Confidence**
latest	JJS	0.6381
::	:	0.5571
top	JJ	0.8602
world	NN	0.9841
releases	NNS	0.7567

MESSAGE 6		
Token	**Tag**	**Confidence**
cdt	NNP	0.5019
presents	VBZ	0.8419
alice	NNP	0.8762
in	IN	0.9597
wonderland	NN	0.9618
-	:	0.9687
catonsville	NNP	0.4957
dinner	NN	0.9760
has	VBZ	0.9807
posted	VBN	0.9449
'	''	0.9463
cdt	NN	0.6107
presents	VBZ	0.7664
alice	NNP	0.8615
in	IN	0.9740
wonderland	NN	0.9706
'	''	0.9594
to	TO	0.9896
the	DT	0.9936
...	:	0.9930

MESSAGE 17		
Token	**Tag**	**Confidence**
how	WRB	0.9814
exciting	JJ	0.9676
!	.	0.9844
:	UH	0.6434
hello	UH	0.9999
!	.	0.9847
what	WP	0.9803
is	VBZ	0.9965
happening	VBG	0.8828
in	IN	0.9796
your	PRP$	0.9739
world	NN	0.9932
?	.	0.9903
we	PRP	0.9891
are	VBP	0.9969
all	DT	0.9681
gearing	VBG	0.8853
up	RP	0.9203
for	IN	0.9987
#valentines	HT	0.7701
with	IN	0.9958
bouquets	NNS	0.9777
flying	VBG	0.9531
out	RP	0.6980
the	DT	0.9941
door	NN	0.9944
.	.	0.9968

MESSAGE 7		
Token	**Tag**	**Confidence**
territory	NN	0.9318
manager	NN	0.9930
:	:	0.9830
location	NN	0.9782
:	:	0.9780
calgary	NNP	0.8052
,	,	0.9889
alberta	NNP	0.9846
,	,	0.9889
canada	NNP	0.9884
job	NN	0.9568
category	NN	0.9790
:	:	0.9717
bu	NN	0.3815
...	:	0.9980
#jobs	HT	0.9762

MESSAGE 8		
Token	**Tag**	**Confidence**
i	PRP	0.9781
cud	MD	0.7355
murder	NN	0.5771
sum	NN	0.6417
today	NN	0.9962
n	CC	0.8607
not	RB	0.9948
even	RB	0.9942
flinch	VB	0.6605
i	PRP	0.9915
am	VBP	0.9555
tht	DT	0.2673
fukin	RB	0.6578
angry	JJ	0.9794
today	NN	0.9955

MESSAGE 18		
Token	**Tag**	**Confidence**
i	PRP	0.9866
would	MD	0.9663
very	RB	0.9744
much	RB	0.8503
appreciate	VBP	0.8385
it	PRP	0.9960
if	IN	0.9856
people	NNS	0.9669
would	MD	0.9839
stop	VB	0.9955
broadcasting	NN	0.9037
asking	VBG	0.9790
me	PRP	0.9893
to	TO	0.9966
add	VB	0.9911
people	NNS	0.9672
on	IN	0.9961
bbm	NNP	0.8763
.	.	0.9971

MESSAGE 19		
Token	**Tag**	**Confidence**
sam	NNP	0.9569
i	PRP	0.9701
knw	VBP	0.8937
you	PRP	0.9989
are	VBP	0.9946
a	DT	0.9957
cricket	NN	0.9904
fan	NN	0.9783
are	VBP	0.9617
you	PRP	0.9984
watching	VBG	0.9279
any	DT	0.9165
of	IN	0.9922
the	DT	0.9963
world	NN	0.9836
cup	NN	0.9873
matches	VBZ	0.7315

MESSAGE 9		
Token	**Tag**	**Confidence**
bbc	NNP	0.6699
news	NN	0.7415
-	:	0.9818
today	NN	0.9776
-	:	0.9657
free	JJ	0.9902
school	NN	0.9962
funding	NN	0.8790
plans	NNS	0.7315
'	''	0.9378
lack	NN	0.7382
transparency	NN	0.9705
'	''	0.9115
-	:	0.9644
…	UH	0.6744

MESSAGE 10		
Token	**Tag**	**Confidence**
manchester	NNP	0.9817
city	NN	0.8986
council	NN	0.9968
details	NNS	0.9668
saving	VBG	0.9776
cuts	VBZ	0.4558
plan	NN	0.9613
:	:	0.9806
...	:	0.9953
depressing	JJ	0.9187
.	.	0.9927
apparently	RB	0.9780
we	PRP	0.9939
are	VBP	0.9931
th	DT	0.6214
most	JJS	0.6385
deprived	VBN	0.4490
&	CC	0.9364
top	JJ	0.7569
hardest	JJS	0.7105
hit	NN	0.5791

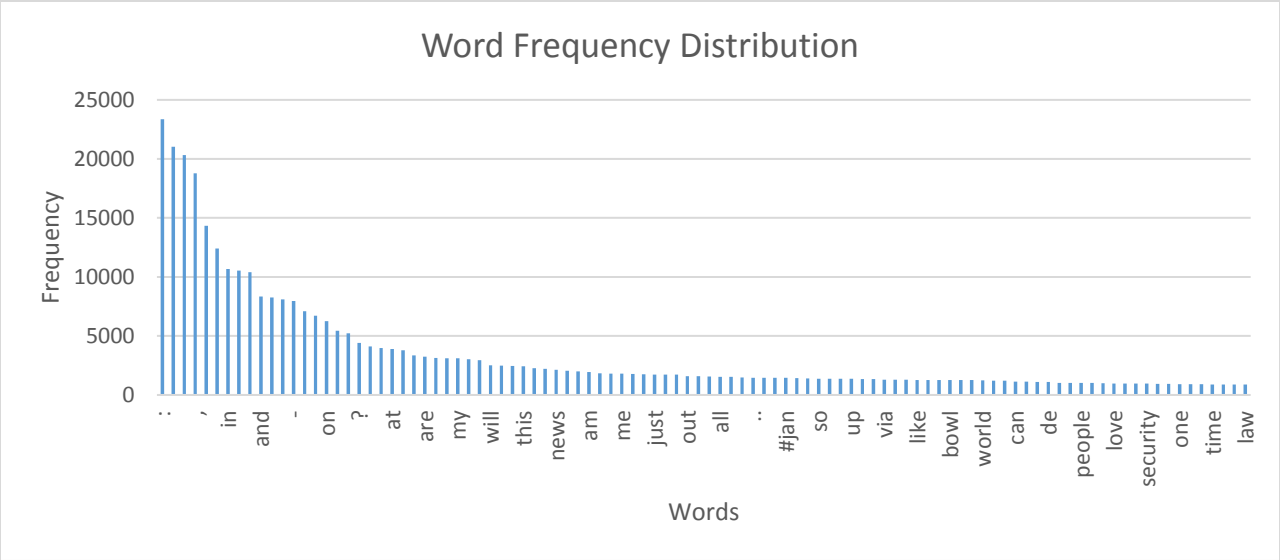
MESSAGE 20		
Token	**Tag**	**Confidence**
john	NNP	0.9816
baer	NNP	0.9599
:	:	0.9801
who	WP	0.9422
did	VBD	0.8878
not	RB	0.9880
see	VB	0.9888
this	DT	0.9485
coming	VBG	0.4781
?	.	0.9905
:	:	0.9146
to	TO	0.9870
those	DT	0.9780
who	WP	0.9501
know	VBP	0.7995
ed	VBN	0.2962
and	CC	0.9933
		midge
rendell	NNP	0.9694
-	:	0.9763
heck	UH	0.4761
,	,	0.9908
to	TO	0.9913
the	DT	0.9913
philly	NNP	0.9628
world	NN	0.9624
at	IN	0.9984
la	NNP	0.9819
...	:	0.9952

5. **How many tokens did you find in the corpus? How many types (unique tokens) did you have? What is the type/token ratio for the corpus? The type/token ratio is defined as the number of types divided by the number of tokens.**
   The number of tokens found in the corpus is **812740**.
   The number of unique tokens found in the corpus is **63169**.
   Type token ratio for the corpus **812740:63169** or **12.8661**

6. **For each token, print the token and its frequency in a file called Tokens.txt (from the most frequent to the least frequent) and include the first 100 lines in your report.**
   The following table lists the top 100 most frequent tokens.

Token	Frequency	Token	Frequency	Token	Frequency	Token	Frequency
:	23366	with	3140	#egypt	1556	world	1243
the	21031	‘	3133	all	1549	if	1224
.	20319	my	3109	egypt	1547	they	1203
,	18767	that	3021	what	1469	can	1136
to	14344	new	2934	..	1460	how	1122
…	12399	will	2514	no	1456	more	1112
in	10669	&	2486	now	1452	de	1091
of	10547	from	2461	#jan	1449	union	1032
a	10407	this	2435	|	1421	he	1024
and	8349	be	2259	was	1397	people	1019
i	8265	have	2228	so	1383	who	1017
is	8092	news	2134	super	1379	or	985
-	7965	by	2043	an	1371	love	978
for	7092	do	1998	up	1363	airport	974
!	6722	am	1948	…	1358	day	966
on	6253	egyptian	1825	obama	1351	security	966
“	5437	your	1812	via	1289	release	946
you	5216	me	1802	media	1288	$	929
?	4419	us	1769	social	1281	one	919
(	4109	state	1750	like	1277	his	917
not	3979	just	1733	about	1276	president	913
at	3904	as	1732	get	1276	time	893
it	3790	we	1727	bowl	1275	today	890
)	3358	out	1601	but	1272	good	888
are	3252	has	1590	white	1270	law	879

The following bar chart represents the word frequency distribution of the first 100 tokens.



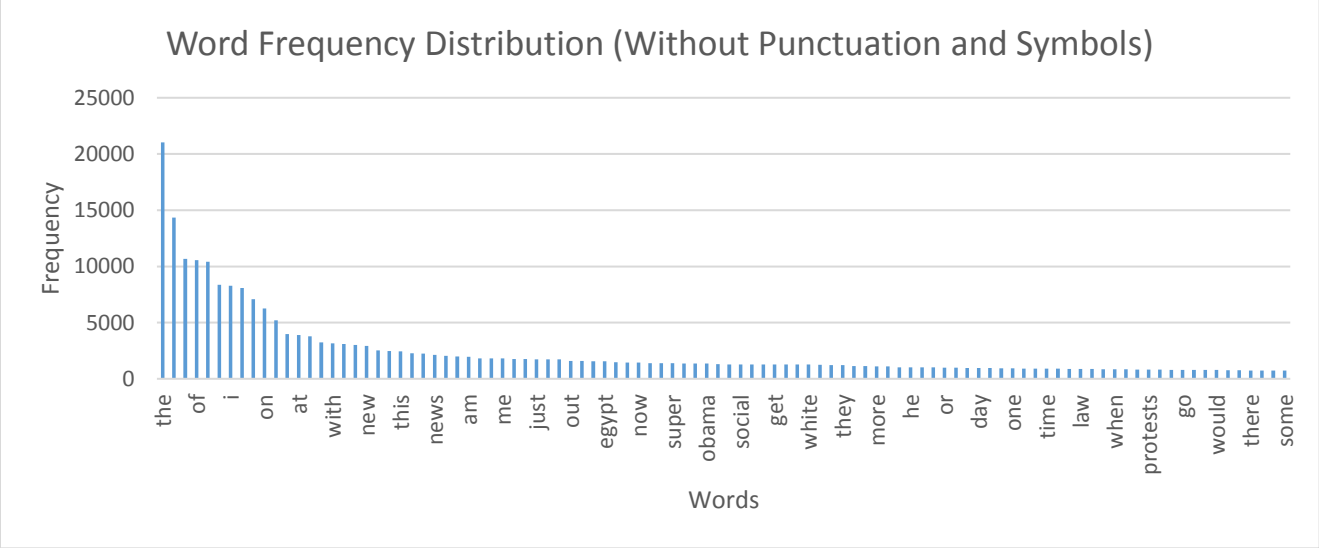
7. **How many tokens appeared only once in the corpus?**
   The number of tokens that had frequency one is **37333**.

8. **From the list of tokens, extract only words, by excluding punctuation and other symbols. How many words did you find? List the top 100 most frequent words in your report, with their frequencies. What is the type/token ratio when you use only word tokens (called lexical diversity)?**
   Handling Punctuation and other symbols
   At this point, we have word tokens that we generated with the POS tagger, we work on this to remove the punctuations and other symbols from the list of tokens.
   Subset of Tokens: ['save', 'bbc', 'world', 'service', 'from', 'savage', 'cuts', '', 'a', 'lot', 'of', 'people', 'always', 'make', 'fun', 'about', 'the', 'end', 'of', 'the', 'world', 'but', 'the', 'question', 'is', '..', '"', 'are', 'you', 'ready', 'for', 'it', '♪♪', '♥', '*-*']
   Subset of Tokens without punctuations: ['save', 'bbc', 'world', 'service', 'from', 'savage', 'cuts', 'a', 'lot', 'of', 'people', 'always', 'make', 'fun', 'about', 'the', 'end', 'of', 'the', 'world', 'but', 'the', 'question', 'is', 'are', 'you', 'ready', 'for', 'it']
   Before removing punctuations and other symbols we had 812740 tokens. We were left with **641911 words** when we excluded punctuations and other symbols from out list of tokens, which means we were able to remove 170829 characters from our list of tokens.
   The number of **unique tokens** is **47738**.
   Type/ Token ratio or the **lexical diversity** is **641911: 47738** or **13.4465**

The following table lists the top 100 most frequent tokens, without punctuation and other symbols.

Token	Frequency	Token	Frequency	Token	Frequency	Token	Frequency
the	21031	by	2043	media	1288	one	919
to	14344	do	1998	social	1281	his	917
in	10669	am	1948	like	1277	president	913
of	10547	egyptian	1825	about	1276	time	893
a	10407	your	1812	get	1276	today	890
and	8349	me	1802	bowl	1275	good	888
i	8265	us	1769	but	1272	law	879
is	8092	state	1750	white	1270	house	868
for	7092	just	1733	world	1243	over	847
on	6253	as	1732	if	1224	when	847
you	5216	we	1727	they	1203	show	839
not	3979	out	1601	can	1136	our	818
at	3904	has	1590	how	1122	protests	815
it	3790	all	1549	more	1112	got	808
are	3252	egypt	1547	de	1091	service	789
with	3140	what	1469	union	1032	go	788
my	3109	no	1456	he	1024	going	787
that	3021	now	1452	people	1019	video	779
new	2934	was	1397	who	1017	would	776
will	2514	so	1383	or	985	lol	764
from	2461	super	1379	love	978	after	763
this	2435	an	1371	airport	974	there	744
be	2259	up	1363	day	966	says	743
have	2228	obama	1351	security	966	its	740
news	2134	via	1289	release	946	some	739

The following bar chart represents the word frequency distribution, without punctuation and symbols, of the first 100 most frequent tokens.



9. **From the list of words, exclude stopwords. List the top 100 most frequent words and their frequencies.**

Removing stopwords

We use the list of stopwords provided in stopwords.txt

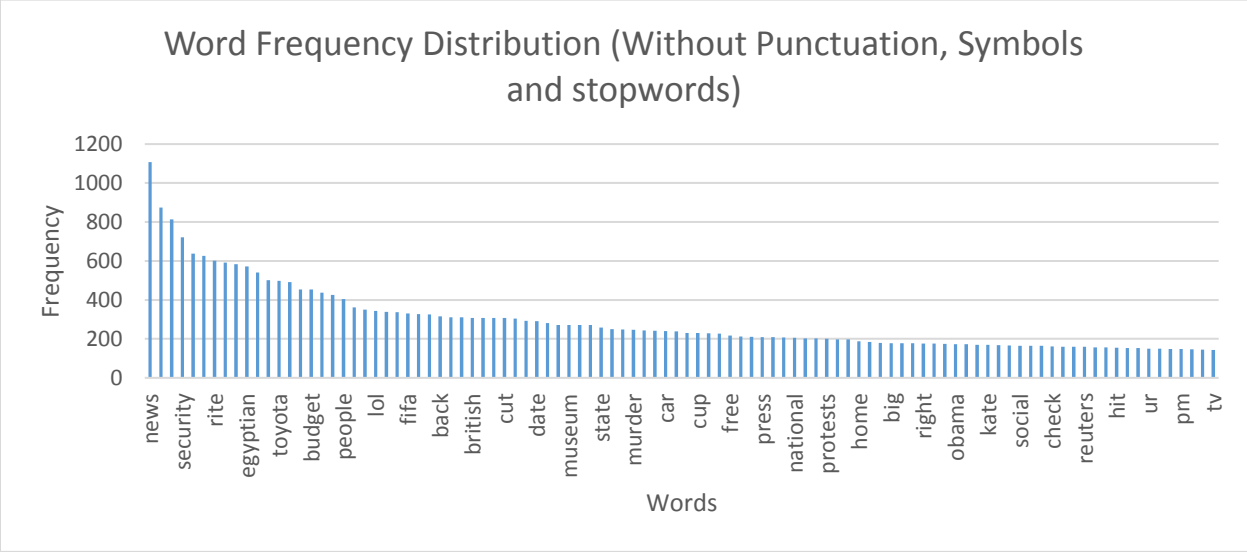
Total words = 151425

Unique words = 28064

The following table lists the top 100 most frequent tokens, without punctuation, other symbols and stopwords.

Token	Frequency	Token	Frequency	Token	Frequency	Token	Frequency
news	1107	mexico	327	assange	231	obama	172
release	875	protesters	325	cup	230	lawsuit	172
world	813	back	315	top	228	game	170
security	722	says	311	live	227	kate	170
white	638	pakistan	310	free	217	help	168
phone	626	british	308	th	212	life	166
rite	602	computer	308	president	211	social	165
taco	592	soccer	308	press	209	house	165
return	583	cut	307	cairo	209	media	164
egyptian	572	haiti	304	released	207	check	161
bell	541	video	292	national	205	mubarak	160
crash	502	date	291	online	202	recall	160
toyota	498	court	281	ap	202	reuters	159
bbc	492	war	272	protests	201	movie	156
cuts	454	museum	272	business	197	government	156
budget	454	man	271	staff	197	hit	155
love	437	drug	271	home	188	black	153
egypt	425	state	259	work	184	meat	153
people	405	yo	250	watch	180	ur	150
police	362	know	249	big	178	hacking	149
peace	351	murder	246	twitter	178	uk	148
lol	343	beef	244	clinton	178	pm	148
service	339	think	242	right	176	blog	146
today	337	car	240	wikileaks	176	shit	145
fifa	331	stripes	238	post	174	tv	143

The following bar chart represents the word frequency distribution, without punctuation, symbols and stopwords, of the first 100 most frequent tokens.



10. **Compute all the pairs of two consecutive words (excluding stopwords and punctuation). List the most frequent 100 pairs and their frequencies in your report. Also compute the type/token ratio when you use only word tokens without stopwords (called lexical density)?**

Biagrams

Type/Token Ratio is 357724: 252461

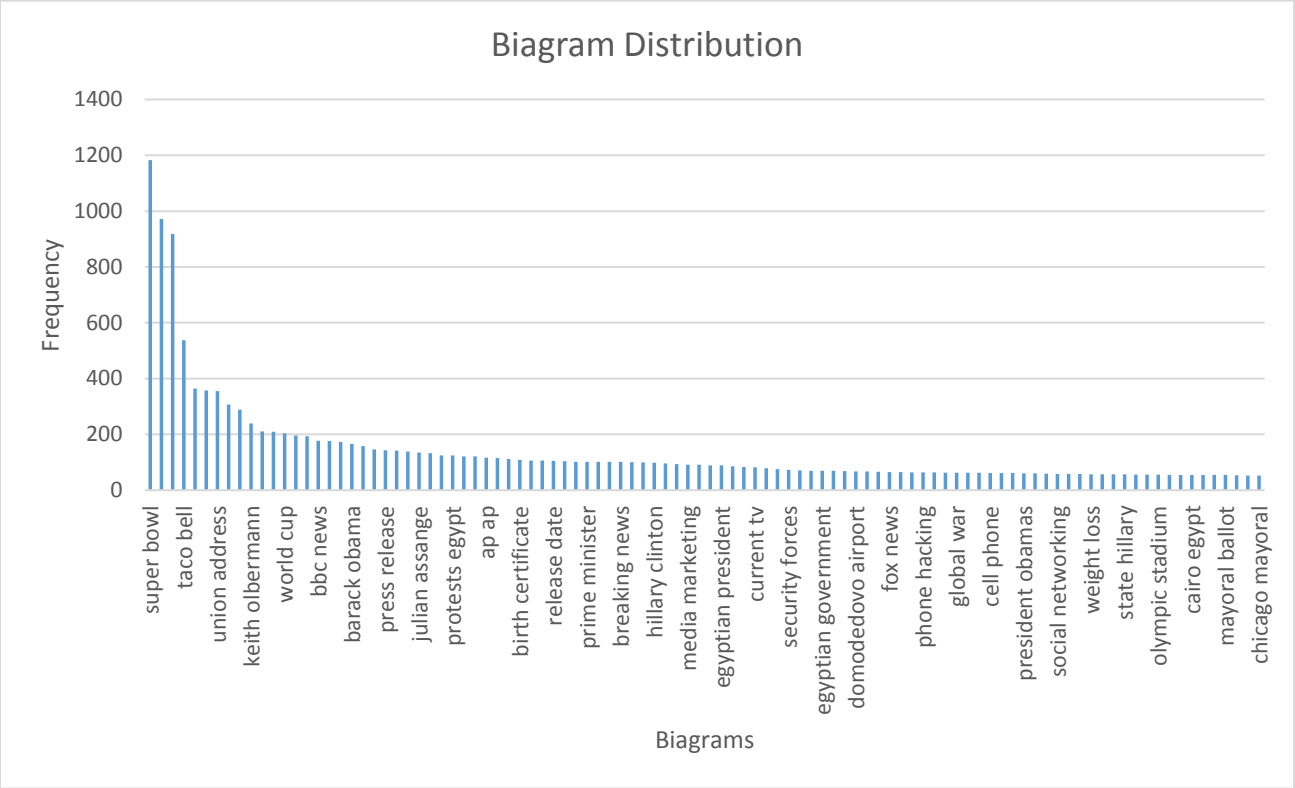
The following table lists the top 100 most frequent pairs of two consecutive words.

Biagram	Frequency	Biagram	Frequency
super bowl	1183	president barack	133
social media	972	hosni mubarak	125
state union	918	protests egypt	124
taco bell	537	tahrir square	121
egypt jan	364	supreme court	121
white house	357	ap ap	116
union address	355	youtube video	115
global warming	307	world news	112
jan egypt	288	birth certificate	108
keith olbermann	239	egyptian protesters	106
bowl xlv	210	obamas state	106
president obama	209	release date	105
world cup	204	egyptian museum	104
white stripes	196	world service	102
moscow airport	193	prime minister	102
bbc news	177	blog post	102

rahm emanuel	176	glenn beck	102
united states	173	breaking news	101
barack obama	166	middle east	100
health care	158	mph kt	99
egypt protests	146	hillary clinton	98
press release	143	tcot tlot	96
budget cuts	142	president hosni	93
customer service	138	media marketing	91
julian assange	135	egyptian protests	91

Biagram	Frequency	Biagram	Frequency
bbc world	89	cell phone	62
egyptian president	89	cowboys stadium	62
federal judge	86	anthony hopkins	61
egyptian police	83	president obamas	60
current tv	82	state tv	60
union speech	79	care law	59
bid date	75	social networking	58
security forces	73	climate change	58
egyptian people	71	shorty award	58
secretary state	69	weight loss	57
egyptian government	69	fifa soccer	57
airport security	69	judge rules	57
share friends	68	state hillary	57
domodedovo airport	67	jan jan	56
special olympics	67	nominate shorty	56
international airport	66	olympic stadium	56
fox news	65	green bay	55
egyptian embassy	65	fifa world	55
tear gas	64	cairo egypt	55
phone hacking	64	cairo jan	55
unemployment rate	64	toyota recalls	55
egyptian army	63	mayoral ballot	54
global war	63	iranelection iran	53
kate middleton	63	obama state	52
gabrielle giffords	63	chicago mayoral	52

The following bar chart represents the biagram distribution, without punctuation, symbols and stopwords, of the first 100 most frequent biagrams.



11. **Extract multi-word expressions (composed of two or more words, so that the meaning of the expression is more than the composition of the meanings of its words). You can use an existing tool or your own method (explain what tool or method you used). List the most frequent 100 expressions extracted. Make sure they are multi-word expressions and not just n-grams or collocations.**

    After removal of stop words, punctuations and symbols, we use the cleaned corpus and **Textblob** to extract phrases from Twitter messages.

    `social media resources missed sm`

    `social media resources missed internet online strategies product lau`

    The multiword expression extracted considering the above tweets is "`social media resources`"

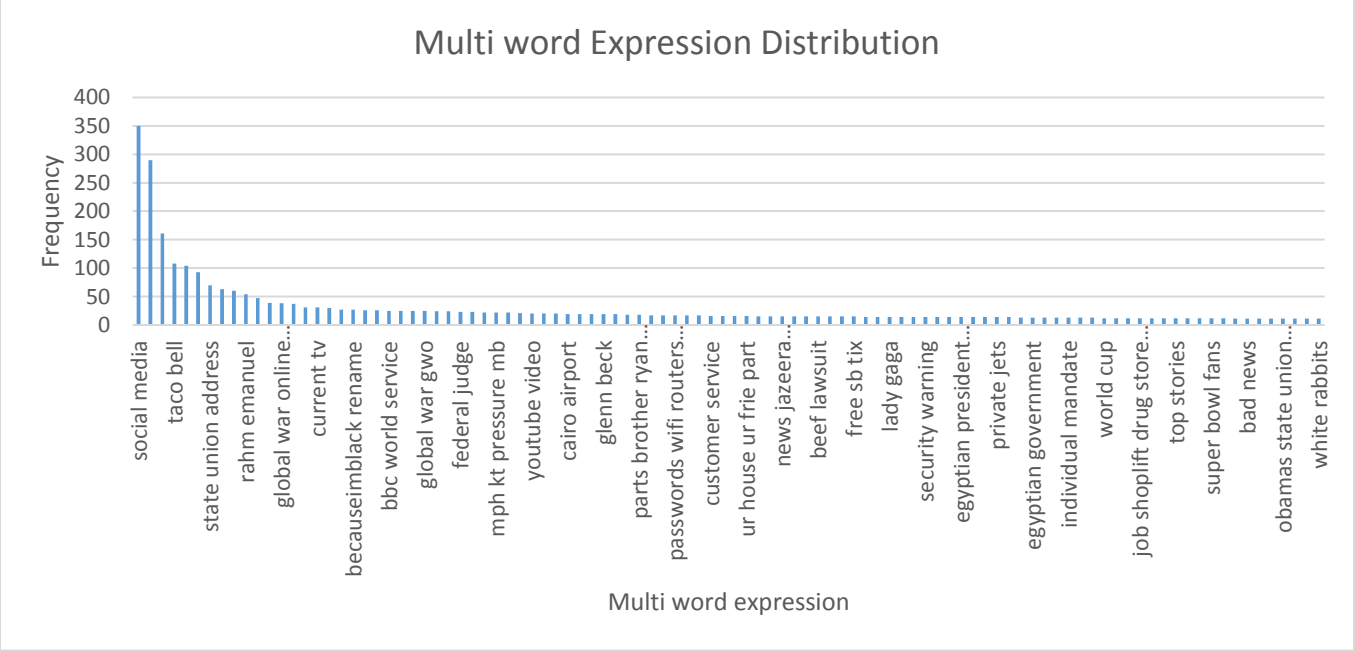
The following table lists the top 100 most frequent multi word expressions.

Multi word expressions	Frequency	Multi word expressions	Frequency
social media	350	free shipping	24
super bowl	290	russian media	24
white house	161	federal judge	23
taco bell	108	protests egypt	23
white stripes	104	egyptian army	22
state union	93	mph kt pressure mb	22
state union address	70	mph kt pressure	22
super bowl xlv	63	prime minister	21
keith olbermann	60	youtube video	20
rahm emanuel	54	tahrir square	20
supreme court	47	super bowl xlv badge	20
president obama	39	cairo airport	19
global war online iphone click link	38	egyptian protests	19
egypt protests	37	egyptian police	19
egypt jan	31	glenn beck	19
current tv	31	special olympics	19
blog post	30	white iphone	18
warm weather	27	parts brother ryan butler	18
becauseimblack rename	27	fifa soccer	17
julian assange	26	pandas chill dude racism stupid	17
mph kt humidity	26	passwords wifi routers protesters communicate world jan	17
bbc world service	25	federal judge florida	17
jan egypt	25	super bowl sunday	17
squad iphone ipod ipad	25	customer service	16
global war gwo	25	taco bell fights	16

Multi word expressions	Frequency	Multi word expressions	Frequency
fans wout seats nite field postgame	16	egyptian government	13
ur house ur frie part	16	renewable energy	13
mexico city wyou	15	noire release date	13
airport security	15	individual mandate	13
news jazeera egyptians form humanchain egypt museum	15	seats superbowl	13
suicide bomber	15	creative coalition sundance donate moms theatre piven theatre	13
free press	15	world cup	12
beef lawsuit	15	super bowl ads	12
cairo residents	15	egypt protesters	12

free merchandise food beverage	15	job shoplift drug store mobsterworld	12
free sb tix	15	oprah winfrey family	12
bbc news	14	national museum	12
social network	14	top stories	12
lady gaga	14	oprah winfrey	12
squad iphone ipod ipad gl	14	illinois court	12
th pm liverpool perisur	14	super bowl fans	12
security warning	14	olbermann msnbc	12
email address spammers	14	cell phone	11
cairo museum	14	bad news	11
egyptian president hosni mubarak	14	haiti aid secretary state hillary clinton	11
super bowl commercials	14	hillary clinton	11
white girl	14	obamas state union address	11
private jets	14	pharaonic mummies	11
social media resources	14	egyptian museum	11
release purr mx deets wwwkatyperrycom	13	white rabbits	11

The following bar chart represents the multi word expression distribution of the first 100 most frequent expressions.



Part 2: Part-of-Speech Tagging

PROBLEM STATEMENT

Use this corpus of 10,000 Twitter messages, already tokenized. It is in the format of one sentence per line. Run one or more part-of-speech (POS) taggers on the corpus, and compute the tagging accuracy. Use the PennTreebank tags plus 4 extra tags for Twitter messages: USR, HT, URL, RT for @usernames, #hastags, urls, and re-tweet symbols.

SOLUTION

1. **TOOLS USED**
   a) CMU Twitter NLP and POS tagger: Fast and robust Java-based tokenizer and part-of-speech tagger for tweets.
   b) model.ritter_ptb_alldata_fixed.20130723: A model that gives a Penn Treebank-style tagset for Twitter. Trained from a fixed version of Ritter et al. EMNLP 2011's annotated data.

2. **POS TAGGING**
   a) **Generating POS Tags**

```
./runTagger.sh --model model.ritter.txt --output-format conll
POS_tagger_input.txt >> POS_tagger_output.txt
```

```
Nikhils-MacBook-Pro:ark-tweet-nlp-0.3.2 nikhiloswal$ ./runTagger.sh --model
 model.ritter.txt --output-format conll POS_tagger_input.txt >> POS_tagger_
output.txt
Detected text input format
Tokenized and tagged 10000 tweets (98152 tokens) in 9.4 seconds: 1066.1 twe
ets/sec, 10464.0 tokens/sec
```

```
Number of Tweets - 10000 | Number of Tokens - 98152
```

   b) **Sample Output**

Format - conll (<Token> <POS Tag> <Confidence>)

```
DREAM    NN    0.8291

Too      RB    0.9617
much     JJ    0.5336
hw NN    0.8034
```

Format - pretsv (<Tweet> <POS Tags> <Confidence > <Tweet>)

```
DREAM    NN    0.8291      DREAM
Too much hw    RB JJ NN    0.9617 0.5336 0.8034   Too much hw
```

c) **POS Tagger's output for the first 20 sentences**

The following table lists the first 20 sentences from the POS tagger's output.

<Token_POS Tag>
DREAM_NN
Too_RB much_JJ hw_NN
high_JJ school_NN is_VBZ weird_JJ
I_PRP feel_VBP .._: Blah_UH ._.
I_PRP Love_VBP One_CD Direction_NNP
Can_MD I_PRP make_VBP a_DT pie_NN with_IN potatoes_NNS ?_.
After_IN so_RB many_JJ days_NNS of_IN just_RB trying_VBG ,_, finally_RB made_VBD it_PRP of_IN bed_NN for_IN a_DT run_NN at_IN 6_CD ._. Hah_UH
I_PRP ca_MD n't_RB express_VB how_WRB I_PRP feel_VBP in_IN a_DT text_NN !_.
Finally_RB
@smosh_USR awesome_JJ about_IN food_NN battle_NN 2012_CD
I_PRP should_MD probably_RB finish_VB my_PRP$ homework_NN
I_PRP _' m_VBP so_RB sleepy_JJ right_RB now_RB !_. !_. #earlybedtime_HT
Life_NN _' s_VBZ most_RBS important_JJ promises_NNS might_MD never_RB be_VB spoken_VBN ._.
@JCSweetGirl_USR Hi_UH !_.
@nessamaders_USR aaaawn_UH *-*_UH
@djherrold_USR just_RB ask_VB if_IN you_PRP can_MD get_VB a_DT picture_NN with_IN him_PRP ._. I_PRP _' m_VBP sure_JJ it_PRP _' ll_MD make_VB his_PRP$ day_NN ._.
Me_PRP beating_VBG this_DT trend_NN bad_JJ tonight_NN #ThugLife_HT
@ALAXASS_USR #idontevenknowyournamebro_HT
The_DT fact_NN that_IN @Brittney_9_USR and_CC @brynnmariecee_USR gain_VB up_RP on_IN me_PRP in_IN child_NN development_NN <<<_UH #realjerks_HT
Dreaming_VBG about_IN you_PRP ._.

3. ACCURACY

```
java -Xmx2g -cp ark-tweet-nlp-0.3.2.jar
cmu.arktweetnlp.RunTagger --input-format conll --model
model.ritter.txt POS_tagger_output.txt
```

```
Nikhils-MacBook-Pro:ark-tweet-nlp-0.3.2 nikhiloswal$ java -Xmx2g -cp ark-tw
eet-nlp-0.3.2.jar cmu.arktweetnlp.RunTagger --input-format conll --model mo
del.ritter.txt POS_tagger_output.txt > POSTaggerAccuracy.txt
98152 / 98152 correct = 1.0000 acc, 0.0000 err
10000 tweets in 8.4 seconds, 1187.1 tweets/sec
```

Number of Tokens - 98152 | Accuracy - 100% | Error - 0%

The following table compares a few tokens of the expected output to the actual output.

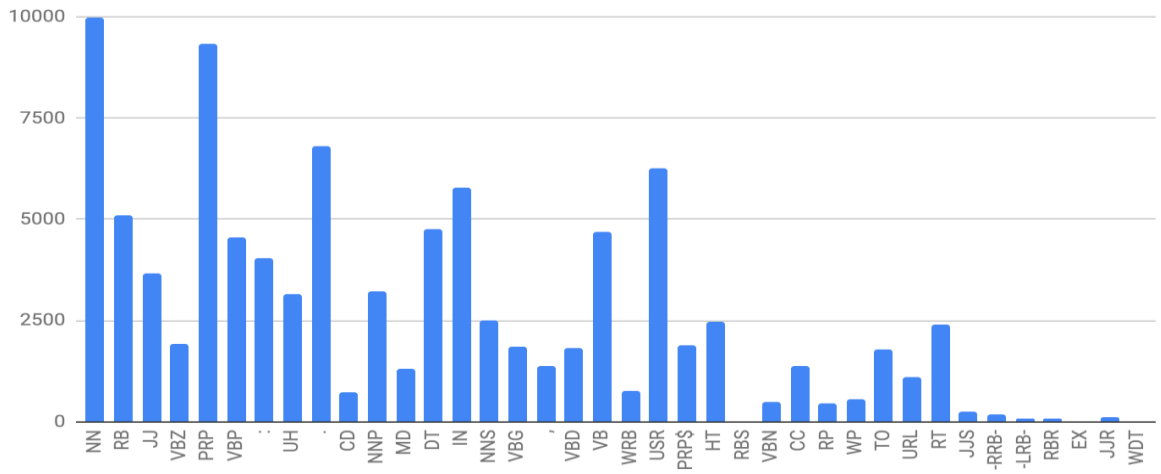
Actual Output	Expected Output
DREAM_NN	DREAM_NN
Too_RB much_JJ hw_NN	Too_RB much_JJ hw_NN
high_JJ school_NN is_VBZ weird_JJ	high_JJ school_NN is_VBZ weird_JJ
I_PRP feel_VBP .._: Blah_UH ._.	I_PRP feel_VBP .._: Blah_UH ._.

The following table represents all possible POS tags and their respective frequency.

POS Tags	Count	POS Tags	Count
NN	9984	WRB	753
RB	5101	USR	6271
JJ	3668	PRP$	1879
VBZ	1916	HT	2481
PRP	9340	RBS	6
VBP	4545	VBN	491
:	4049	CC	1361
UH	3146	RP	436
.	6817	WP	551
CD	716	TO	1786
NNP	3209	URL	1094
MD	1303	RT	2415
DT	4758	JJS	233
IN	5768	-RRB-	161

NNS	2485	-LRB-	66
VBG	1853	RBR	62
,	1381	EX	18
VBD	1809	JJR	114
VB	4701	WDT	6

The following bar chart represents the distribution of POS tags and their frequency.



Distribution of Tasks

	Task	Owner
Part 1	Initial Corpus Analysis	Deepshi
	Preprocessing - HTML Elements, Numbers, Escape Character, Twitter Handles	Deepshi
	Preprocessing - Contraction, Lowercase, URL, RT	Nikhil
	Tokenization	Nikhil
	Removal of Punctuations and Symbols	Deepshi
	Removal of Stop words	Nikhil
	Consecutive words	Deepshi, Nikhil
	Multiword Expression	Deepshi
Part 2	Tokenization	Nikhil, Deepshi
	Accuracy	Nikhil
	Report	Nikhil, Deepshi