In [1]:

```
import pandas as pd
import json
import csv
import tweepy
```

Loading in hatEval 2019 Data

Below is a description of the initial SemEval training data this project will use. The targets of abuse in this dataset are immigrants and women.

The columns can be understood as follows:

HS - Hate Speech

This column indicates whether the content of the tweet is hate speech or not. If the value is 1 then it is hate speech, if it is 0 then it is not hate speech.

TR - Target of Tweet

First, the tweet will have to have been confirmed as hate speech before this information is considered.

If the value is 1 then the target of this tweet is an individual, if it is 0 then the target is a group.

AG - Aggression

Again, firstly this tweet will have to be confirmed as hate speech before this column is used

If the value is 1 then the tweet is aggressive, if the value is 0 then it is not considered aggressive.

In [2]:

```
path = r'Raw_Data\semeval_2018_task5_hateval\public_development_en\dev_en.tsv'
dev = pd.read_csv(path, sep='\t');
dev.drop('id', inplace = True, axis = 1)
dev.reset_index(drop = True, inplace = True)
dev.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 4 columns):
text    1000 non-null object
HS      1000 non-null int64
TR      1000 non-null int64
AG      1000 non-null int64
dtypes: int64(3), object(1)
memory usage: 31.4+ KB
```

There's two sets of data here, one called dev_en and the other called train_en. I'm uncertain as to whether the dev_en dataset is a subset of the train_en dataset. To test this, I'll merge the two dataframes and then remove duplicates and see if there's any effect.

In [3]:

```
path = r'Raw_Data\semeval_2018_task5_hateval\public_development_en\train_en.tsv'
train = pd.read_csv(path, sep='\t');
train.drop('id', inplace = True, axis = 1)
train.reset_index(drop = True, inplace = True)
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9000 entries, 0 to 8999
Data columns (total 4 columns):
text    9000 non-null object
HS      9000 non-null int64
TR      9000 non-null int64
AG      9000 non-null int64
dtypes: int64(3), object(1)
memory usage: 281.4+ KB
```

Also manually inspecting the code using the search function on excel (I exported to csv file using this code - *new.to_csv('new.csv', sep=',', index = False)*

I found that the dev dataset was not a subset of the train dataset. therefore I will use both to train my model.

In [6]:

```
dfs = [train, dev]
alldf = pd.concat(dfs)
alldf.HS.value_counts()
```

Out[6]:

```
0    5790
1    4210
Name: HS, dtype: int64
```

In [14]:

```
alldf.head(10)
```

Out[14]:

	text	HS	TR	AG
0	Hurray, saving us $$$ in so many ways @potus @realDonaldTrump #LockThemUp #BuildTheWall #EndDACA #BoycottNFL #BoycottNike	1	0	0
1	Why would young fighting age men be the vast majority of the ones escaping a war & not those who cannot fight like women, children, and the elderly?It's because the majority of the refugees are not actually refugees they are economic migrants trying to get into Europe.... https://t.co/Ks0SHbtYqn	1	0	0
2	@KamalaHarris Illegals Dump their Kids at the border like Road Kill and Refuse to Unite! They Hope they get Amnesty, Free Education and Welfare Illegal #FamilesBelongTogether in their Country not on the Taxpayer Dime Its a SCAM #NoDACA #NoAmnesty #SendThe	1	0	0
3	NY Times: 'Nearly All White' States Pose 'an Array of Problems' for Immigrants https://t.co/ACZKLhdMV9 https://t.co/CJAISXCzR6	0	0	0
4	Orban in Brussels: European leaders are ignoring the will of the people, they do not want migrants https://t.co/NeYFyqvYIX	0	0	0
5	@KurtSchlichter LEGAL is. Not illegal. #BuildThatWall	1	0	0
6	@RitaPanahi @826Maureen @RealCandaceO Antifa are just a pack of druggie misfits that no one loves, being the violent thugs they are is their cry for attention and their hit of self importance.#JuvenileDelinquents	0	0	0
7	Ex-Teacher Pleads Not guilty To Rape Charges https://t.co/D2mGu3VT5G	0	0	0
8	still places on our Bengali (Sylheti) class! it's London's 2nd language! know anyone interested @SBSisters @refugeecouncil @DocsNotCops https://t.co/sOx6shjvMx	0	0	0
9	DFID Africa Regional Profile: July 2018 https://t.co/npfZCriW0w	0	0	0

Loading in OffensEval Data

This data is a bit more tricky but it'll be useful to try out and inspect if the data can be reliably used in this project. We must ask ourselves if offensive language necessarily equates to hate speech.

If this data is used then we must include a reference, it is mentioned in the README section of the data.

The column names in the file are the following:

id tweet subtask_a subtask_b subtask_c

The labels used in the annotation are listed below.

(A) Level A: Offensive language identification

- (NOT) Not Offensive - This post does not contain offense or profanity.
- (OFF) Offensive - This post contains offensive language or a targeted (veiled or direct) offense

In our annotation, we label a post as offensive (OFF) if it contains any form of non-acceptable language (profanity) or a targeted offense, which can be veiled or direct.

(B) Level B: Automatic categorization of offense types

- (TIN) Targeted Insult and Threats - A post containing an insult or threat to an individual, a group, or others (see categories in sub-task C).
- (UNT) Untargeted - A post containing non-targeted profanity and swearing.

Posts containing general profanity are not targeted, but they contain non-acceptable language.

(C) Level C: Offense target identification

- (IND) Individual - The target of the offensive post is an individual: a famous person, a named individual or an unnamed person interacting in the conversation.
- (GRP) Group - The target of the offensive post is a group of people considered as a unity due to the same ethnicity, gender or sexual orientation, political affiliation, religious belief, or something else.
- (OTH) Other – The target of the offensive post does not belong to any of the previous two categories (e.g., an organization, a situation, an event, or an issue)

So bearing all this in mind I believe that if we only get data from this set that has the OFF flag for level A, a TIN flag for level B and a GRP flag for level 3, we can extract data that is offensive and is targetting a specific group which may be the "same ethnicity, gender or sexual orientation, political affiliation, religious belief, or something else."

We may lose some tweets that could be useful for training our model in that they possibly target an individual in a hate speech way, but our overall dataset would be contaminated with too many tweets that are irrelevant to our investigation in that they are offensive but not in a hate speech manner

A problem lies here though in that not all offensive targetting of groups may be considered hate speech. Someone could tweet something like *"All Democrats are repulsive idiots"* and this may not be considered hate speech by the masses.

Likewise can we say that someone tweeting, *"These evangelical christians are uneducated and fucking barbaric!" #MyBodyMyChoice* may not be considered hate speech by the of majority people.

(Although some might, but I'm mainly interested in hate speech agreed upon by a majority of people)

In [15]:

```
path = r'Raw_Data\OLIDv1.0\olid-training-v1.0.tsv'
offens = pd.read_csv(path, sep='\t');
offens.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13240 entries, 0 to 13239
Data columns (total 5 columns):
id           13240 non-null int64
tweet        13240 non-null object
subtask_a    13240 non-null object
subtask_b    4400 non-null object
subtask_c    3876 non-null object
dtypes: int64(1), object(4)
memory usage: 517.3+ KB
```

In [17]:

```
offens = offens.dropna()
offens.reset_index(drop = True, inplace = True)
```

We may later want to use id for an index, but until we can be sure we've removed all redundant data lets not for now

In [5]:

```
offens = offens[offens["subtask_a"] == 'OFF']
offens = offens[offens["subtask_b"] == 'TIN']
offens = offens[offens["subtask_c"] == 'GRP']
offens.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1074 entries, 2 to 3873
Data columns (total 5 columns):
id           1074 non-null int64
tweet        1074 non-null object
subtask_a    1074 non-null object
subtask_b    1074 non-null object
subtask_c    1074 non-null object
dtypes: int64(1), object(4)
memory usage: 50.3+ KB
```

In [6]:

```
offens.head()
```

Out[6]:

	id	tweet	subtask_a	subtask_b	subtask_c
2	52415	@USER was literally just talking about this lo...	OFF	TIN	GRP
7	12681	@USER @USER Kind of like when conservatives wa...	OFF	TIN	GRP
8	82904	@USER @USER Da fuck is going on people? Ther...	OFF	TIN	GRP
10	12609	The only thing the Democrats have is lying and...	OFF	TIN	GRP
12	14726	@USER @USER @USER That's expected if you placa...	OFF	TIN	GRP

In [7]:

```
tweets = offens["tweet"]
pd.set_option('display.max_colwidth', -1)
tweets.head()
```

Out[7]:

```
2       @USER was literally just talking about this lol all mass shootings
like that have been set ups. it’s propaganda used to divide us on major is
sues like gun control and terrorism
7       @USER @USER Kind of like when conservatives wanna associate everyo
ne to their left as communist antifa members?
8       @USER @USER Da fuck is going on people?   There's the men's room a
nd the women's room  Pick one and stick w it 🤔
10      The only thing the Democrats have is lying and stalling to stop Tr
ump from being #President.  What have they done for you lately. #Trump #Ka
vanaugh #MAGA #DEMSUCK
12      @USER @USER @USER That's expected if you placate the violent lefti
sts/terrorists. #KavanaughConfirmation #woke
13      4 out of 10 British people are basically full-on racists. 4 out of
10 voters vote for the Conservatives. Coincidence!???!???
23      @USER I mean it worked for gun control right? URL
28      @USER Fuck em
35      @USER This all I get 😢 damn these haters deleting you IG for no r
eason... URL
```

As expected, these tweets - however tasteless some might be, would not be considered by most people to be hate speech. The other tweets above are clear and, in my opinion, inarguable instances of hate speech. Besides, it is only 1000 tweets or so to lose so it's not a massive loss of potential data.

Although these tweets can be still useful as examples of inane tweets that are not hate speech. Possibly though we shall exclude all tweets that satisfy all three levels (like the tweets above) just to err on the side of caution when training our model

ICVSM 2017 hate speech dataset

A dataset that usefully has benign tweets, offensive language and hate speech. Having offensive language tweets that is not hate speech will be very useful to my model because I can imagine confusion with some models can arise when trying to distinguish between offensive language, laced with profanity and hate speech.

What the columns mean:

Count

Number of CrowdFlower users who coded each tweet (min is 3, sometimes more users coded a tweet when judgments were determined to be unreliable by CF).

Hate_speech:

Number of CF users who judged the tweet to be hate speech.

Offensive_language

Number of CF users who judged the tweet to be offensive.

Neither

Number of CF users who judged the tweet to be neither offensive nor non-offensive.

Class

Class label for majority of CF users. 0 - hate speech, 1 - offensive language, 2 - neither

In [22]:

```
sometweets = 'Raw_Data\ICVSM_2017.csv'
sometweets = pd.read_csv(sometweets, sep=',');
sometweets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24783 entries, 0 to 24782
Data columns (total 7 columns):
Unnamed: 0            24783 non-null int64
count                 24783 non-null int64
hate_speech           24783 non-null int64
offensive_language    24783 non-null int64
neither               24783 non-null int64
class                 24783 non-null int64
tweet                 24783 non-null object
dtypes: int64(6), object(1)
memory usage: 1.3+ MB
```

Further refining tweet database into hate speech

In [10]:

```
hatetweets = sometweets[sometweets["class"] == 0]
pd.set_option('display.max_colwidth', -1)
hatetweets.head(20)
```

Out[10]:

	Unnamed: 0	count	hate_speech	offensive_language	neither	class	
85	85	3	2	1	0	0	"@Blackman38Tide @HowdyDov
89	90	3	3	0	0	0	"@CB_Baby alsarabsss" hes a be
110	111	3	3	0	0	0	"@DevilGrimz: @Vig gay, blacklistec # ht
184	186	3	3	0	0	0	"@MarkRoundtreeJr: https://t.co/RNvD there's blac
202	204	3	2	1	0	0	"@NoChillPaz: "At I http://t.co/R
204	206	3	2	1	0	0	"@NotoriousBM95: @ is a snake and a cc isn't a cripple like yo
219	221	6	3	2	1	0	"@RTNBA: Drakes r released by Nike/. http://t.co/QCtPLxH
260	263	3	2	1	0	0	"@TheoMaxximus: #G http://t.co/3ycrSr ye
312	317	3	2	1	0	0	"@ashlingwilde: @ supposed to
315	320	3	2	1	0	0	"@bigbootybishc lassen cc , the coa youll get r
349	354	3	2	1	0	0	"@jayswaggk #blondeproblems"
352	357	3	2	1	0	0	"@jgabsss: Stacey I http://t.co/PDLG4
437	443	3	2	1	0	0	"Don't worry about the about the nigga you I r
459	466	3	3	0	0	0	"Hey go look at that found the kidnapped g a nigge
519	528	3	2	1	0	0	"Let's kill cracker ba I

	Unnamed: 0	count	hate_speech	offensive_language	neither	class	
526	535	3	2	1	0	0	"My grandma us monkey all the time she did refer to a brol
531	540	3	2	1	0	0	"Nah its You 😂😂& yo i thought some1 pl on that fagg ht
540	549	3	3	0	0	0	"Our people". Now is t race 2 stand up and s the mongerIs turn tl
565	574	3	2	1	0	0	"These sour apple

In [11]:

```
hatetweets.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1430 entries, 85 to 24777
Data columns (total 7 columns):
Unnamed: 0            1430 non-null int64
count                 1430 non-null int64
hate_speech           1430 non-null int64
offensive_language    1430 non-null int64
neither               1430 non-null int64
class                 1430 non-null int64
tweet                 1430 non-null object
dtypes: int64(6), object(1)
memory usage: 89.4+ KB
```

For the most part, this database seems to reliably have hate speech tweets annotated correctly so this database will be used as data for the hate speech segment of the classification

A New Database - ICVSM_2018 Abusive Behaviour Database

In all, about 100,000 tweets. Labelled one of four categories: Normal, abusive, hateful and spam. A detailed explanation as to how the tweets in this datbase were categorised can be found here:
https://arxiv.org/pdf/1802.00393.pdf (https://arxiv.org/pdf/1802.00393.pdf)

In [7]:

```
icvsm = r'Raw_Data\ICVSM_2018_dataset\hatespeech_text_label_vote.csv'
icvsm = pd.read_csv(icvsm, sep='\t', names = \
                    ["tweets", "majority label", "votes on majority label"  ]);
icvsm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 99996 entries, 0 to 99995
Data columns (total 3 columns):
tweets                    99996 non-null object
majority label            99996 non-null object
votes on majority label   99996 non-null int64
dtypes: int64(1), object(2)
memory usage: 2.3+ MB
```

In [8]:

```
pd.set_option('display.max_colwidth', -1)
icvsm.head(10)
```

Out[8]:

	tweets	majority label	votes on majority label
0	Beats by Dr. Dre urBeats Wired In-Ear Headphones - White https://t.co/9tREpqfyW4 https://t.co/FCaWyWRbpE	spam	4
1	RT @Papapishu: Man it would fucking rule if we had a party that was against perpetual warfare.	abusive	4
2	It is time to draw close to Him 🙏🏻 Father, I draw near to you now and always ♡ https://t.co/MVRBBX2aqJ	normal	4
3	if you notice me start to act different or distant.. it's bc i peeped something you did or i notice a difference in how you act & ian fw it.	normal	5
4	Forget unfollowers, I believe in growing. 7 new followers in the last day! Stats via https://t.co/bunPHQNXhj	normal	3
5	RT @Vitiligoprince: Hate Being sexually Frustrated Like I wanna Fuck But ion wanna Just fuck anybody	abusive	4
6	Topped the group in TGP Disc Jam Season 2! Onto the Semi-Finals! @HighHorseGames https://t.co/N7LE8IX7Rm	normal	3
7	That daily baby aspirin for your #heart just might be preventing colon #cancer too. https://t.co/2TLawmHhEe	normal	5
8	I liked a @YouTube video from @mattshea https://t.co/niSeJrLKHq THE BLUE ARMY IS COMING! - Ancient Warfare 2	normal	2
9	RT @LestuhGang_: If your fucking up & your homies dont tell you that your fucking up, those ain't your homies	abusive	4

In [9]:

```
hatefultweets = icvsm[icvsm["majority label"] == "hateful" ]
hatefultweets.head(20)
```

Out[9]:

	tweets	majority label	votes on majority label
21	Bad day for #Kyrgyzstan. Suspect in St. Petersburg metro bombing identified as #Kyrgyz, opposition politician Japarov reportedly kills self.	hateful	2
34	I'm over the fucking moon we've cleared up the definition of an act of war. Now, about that slap on the wrist we just gave Syria.	hateful	3
37	RT @ynaoivw: nah bitch i hate u https://t.co/fHX8y7esMH	hateful	3
71	RT @WaysThingsWork: I fucking hate people 😂😂 https://t.co/Qz5gihmcQF	hateful	3
106	@VanJones68 You looked like a complete pathetic idiot tonight. How could you be so stupid Mr. Ghetto. My God you… https://t.co/eEpAih1GAj	hateful	3
147	RT @___DestinyJadai: Never argue with a bitch you can kill by just stepping out on your WORST day 😂 we both know why that bitch mad sis 🤷🏾…	hateful	4
152	RT @Woodparkweirdo: hell yeah I VAPE V: hate A: women P E	hateful	3
194	I talked well to you all along. I do not feel like I'm answering you bad. Damn it!!	hateful	2
221	RT @Jedi_Pite_Bre: Which state allowed 800K illegal aliens to get a license? You guessed it right-the retarded state of California https://…	hateful	3
234	@domy1337 @MainGame6 this is polish hooker look man, come here and you'll meet them next to some forests on the road lmao	hateful	2
245	ADULT SIBLING ABUSE SURVIVORS are often gossiped about for being the family idiot, misfit, or crazy relative. https://t.co/F6SvXWrTFm	hateful	3
287	@TLDRJones @sahluwal @IvankaTrump Learn to spell before you comment...Ivanka is a major nusinesz woman with a world… https://t.co/BKWgAgFzsr	hateful	4
313	Just bought a bunch of Tupperware and grown up colt thinks it's great but little colt hates how much he's turning into his parents	hateful	2
318	@JayFoee_ just another dumbass bronco fan swear I hate this fanbase dawg 😭	hateful	4
337	RT @iamwilliewill: This what happens when you separate yo self from niggas who don't eat they food cold. You FLOURISH... https://t.co/FzTIA…	hateful	3
350	@IronofMercy “Unlike you, I'm not a weakling and have a high Insanity resistance. No I'm lying, but I don't fight it.”	hateful	2
352	I don't give a fuck about NONE of y'all UGLY bitches at Riverdale lmfao get mad hoe	hateful	3
364	RT @ItsMeGrizz: Bad bitches don't take days off https://t.co/eazGi8KnNh	hateful	3
394	A kiosk woman tried to scam me to buy $200 skin cream. Lady, I worked at Lush...I know all the tricks.	hateful	2
400	@JuneGrayy @_BriaPlease @SchoolEm_Jonesy Stupid as fuck and mad cuz she couldn't read 😂😂😂	hateful	5

In the paper the authors explicitly state that the term hateful is supposed to describe hate speech. In fact, they use a near identical description as I used in my report to define hate speech *(Language used to express hatred towards a targeted individual or group, or is intended to be derogatory, to humiliate, or to insult the members of the group, on the basis of attributes such as race, religion, ethnic origin, sexual orientation, disability, or gender.)* ***and state that they told the annotators to define hateful as such.***

Most of the above tweets in my opinon, and among the opinions of an informal sampling I took (friends and family), do not qualify as hate speech. Possibly a few of the annotators got confused and thought that the content had hateful intent rather than hate speech and thought they had to annotate to that definition. The only possible hate speech tweets agreed by a majority of the simple sampling I took was tweet 106 and 221.

To mitigate this, I'll further explore and see if the tweets, with high agreement among annotators that it contains hateful content, is a better suit to the hate speech definition.

In [10]:

```
morehatefultweets =\
hatefultweets[hatefultweets["votes on majority label"] > 7 ]
morehatefultweets.head(10)
```

Out[10]:

	tweets	majority label	votes on majority label
10333	Hanging at UB on Friday, Unsacred headlines the metal showcase at Macrock on Saturday, then heading to Damaged City for Marked Men on Sunday	hateful	16
13919	This was a proportionate response by the United States. It is not designed to overthrow the Assad regime..... https://t.co/PqdEWMQrhy	hateful	18
15481	I had two exams and had to choose 1 or the other to study for and the 1 I didn't study for I passed with an 81 🎉	hateful	8
22478	South Korean OF Kim makes Orioles opening day roster #BaltimoreOrioles #OriolesOpeningDay #Orioles #Orioles https://t.co/erAcYNSsUb	hateful	10
29915	Black people low budget cookouts have quarter legs for the old heads and bullshit hotdogs for the "kids" 😭😭😭😭😭	hateful	14
33403	📄 #WinterEvent2017 On the attraction of two perfectly conducting plat on-the-attraction-of-two-perfectly-conducting-plates.pdf	hateful	22
37228	#DickCavett asks Art Garfunkel who broke up Simon & Garfunkel. AG doesn't remember. I'm guessing suppression @decadesnetwork.	hateful	20
38192	Damn niggas was comparing this season to the farm which is really bad https://t.co/OVCBErCZDC	hateful	10
39529	RT @isabelaseraffim: insomnia ain't a joke bruh I'm really a fucking zombie at this point	hateful	20
43602	RT @iamwilliewill: This what happens when you separate yo self from niggas who don't eat they food cold. You FLOURISH... https://t.co/FzTIA…	hateful	15

This does not seem to be much better. Possibly this database should be discarded because of the unreliability of annotators. With the exception of the normal tweets and spam, which do not appear to be annotated correctly and are difficult to get wrong because the meaning of what is normal and what is spam is not nuanced.

In [11]:

```
icvsm['majority label'].value_counts()
```

Out[11]:

```
normal     53851
abusive    27150
spam       14030
hateful    4965
Name: majority label, dtype: int64
```

Using Twitter API to mine hate speech tweets (Waseem and Hovy 2016)

In [32]:

```
path = r'https://raw.githubusercontent.com/ZeerakW/hatespeech/master/NAACL_SRW_2016.csv'

ids = pd.read_csv(path, sep=',', names = ['id', 'label']);

ids.head()
```

Out[32]:

	id	label
0	572342978255048705	racism
1	572341498827522049	racism
2	572340476503724032	racism
3	572334712804384768	racism
4	572332655397629952	racism

In [9]:

```
sometweets.head(10)
```

Out[9]:

	Unnamed: 0	count	hate_speech	offensive_language	neither	class	t
0	0	3	0	0	3	2	!!! RT @mayasolovely: woman you shouldn't corr about cleaning up your h & as a man you s always take the trash
1	1	3	0	3	0	1	!!!!! RT @mleew17: boy cold...tyga dwn bad for cuff hoe in the 1st p
2	2	3	0	3	0	1	!!!!!!! RT @UrKindOfB Dawg!!!! RT @80sbaby4life ever fuck a bitch and she st cry? You be confused a
3	3	3	0	2	1	1	!!!!!!!! RT @C_G_Ande @viva_based she look t
4	4	6	0	6	0	1	!!!!!!!!!!!! RT @ShenikaRol The shit you hear about me be true or it might be faker the bitch who told it &#5
5	5	3	1	2	0	1	!!!!!!!!!!!!!!!!!"@T_Madison_x shit just blows me..claim y faithful and down for some but still fucking with 😂😂&#128
6	6	3	0	3	0	1	!!!!!!"@__BrighterDays: I ca just sit up and HATE on an bitch .. I got too much shit
7	7	3	0	3	0	1	!!!!“@selfiequee cause I'm tired of you big bi coming for us s girls!!&#
8	8	3	0	3	0	1	" & you might not g bitch back & thats
9	9	3	1	2	0	1	" @rhythmixx_ :hobbies inc fighting Mariam"\n\r

In [25]:

```
#conda install -c conda-forge tweepy
```

Below is a tweepy method to obtain tweets via ID. Twitter API only allows us to extract tweets 100 at a time, as there are rate limits - therfore, we must set the wait_on_rate_limit parameter to True.

In [34]:

```
def lookup_tweets(tweet_IDs, api):
    full_tweets = []
    tweet_count = len(tweet_IDs)

    try:
        for i in range((tweet_count // 100) + 1):
            # Catch the last group if it is less than 100 tweets
            end_loc = min((i + 1) * 100, tweet_count)
            full_tweets.extend(
                api.statuses_lookup(id_=tweet_IDs[i * 100:end_loc])
            )
        return full_tweets
    except tweepy.TweepError:
        print('Something went wrong, quitting...')
```

Open a previously created json file with twitter api credentials and also set wait_on_rate_limit so as not to exceed wait limits

In [35]:

```
# Load credentials from json file
with open("twitter_credentials.json", "r") as file:
    creds = json.load(file)


auth = tweepy.OAuthHandler(creds['CONSUMER_KEY'], creds['CONSUMER_SECRET'])
auth.set_access_token(creds['ACCESS_TOKEN'], creds['ACCESS_SECRET'])

api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True)
```

Run below kernel to load tweets via ID - May take a while

In [36]:

```
good_tweet_ids = list(ids['id']) #Create a list of tweet ids to look up
results = lookup_tweets(good_tweet_ids, api) #apply function
```

Then wrangle the data into one dataframe

In [37]:

```
temp = json.dumps([status._json for status in results]) #create JSON
newdf = pd.read_json(temp, orient='records')
full = ids.merge(newdf,  how='left', on = 'id')
tweetSet = full[['id', 'label', 'text']]
tweetSet = tweetSet.drop_duplicates(subset = 'id')
pd.set_option('display.max_colwidth', -1)
tweetSet.head(10)
```

Out[37]:

	id	label	text
0	572342978255048705	racism	So Drasko just said he was impressed the girls cooked half a chicken.. They cooked a whole one #MKR
2	572341498827522049	racism	Drasko they didn't cook half a bird you idiot #mkr
4	572340476503724032	racism	Hopefully someone cooks Drasko in the next ep of #MKR
6	572334712804384768	racism	of course you were born in serbia...you're as fucked as A Serbian Film #MKR
7	572332655397629952	racism	These girls are the equivalent of the irritating Asian girls a couple years ago. Well done, 7. #MKR
8	575949086055997440	racism	#MKR Lost the plot - where's the big Texan with the elephant sized steaks that they all have for brekkie ?
10	551659627872415744	racism	NaN
11	551763146877452288	racism	NaN
12	551768543277355009	racism	NaN
13	551769061055811584	racism	NaN

In [38]:

```
print(tweetSet.info())
dups = len(tweetSet) - tweetSet['text'].count()
print(" \nAmount of tweet IDs not returning tweets - ", dups)
#print(tweetSet['text'].unique().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 16849 entries, 0 to 16994
Data columns (total 3 columns):
id       16849 non-null int64
label    16849 non-null object
text     10866 non-null object
dtypes: int64(1), object(2)
memory usage: 526.5+ KB
None

Amount of tweet IDs not returning tweets -  5983
```

Suspended Accounts:

Although from our original dataset with tweet IDs we have 16,907 tweets total, we only retrieve 10920 tweets with our function. This is because in many cases the twitter accounts associated with these tweets have been suspended.

As we can see above, some tweet IDs have retrieved NaN for the tweet text because of this, demonstrated below is an attempt to retrieve one of these tweets.

In [21]:

```
try:
    tweet = api.get_status(551763146877452288) # Index 11 in above tweetSet dataset
except tweepy.TweepError as err:
    if err.api_code == 63:
        print(err.reason)
    else:
        print(tweet.text)
```

```
[{'code': 63, 'message': 'User has been suspended.'}]
```

In [39]:

```
#Dropping rows where tweet I'd failed to obtain associated tweet text
tweetSet.dropna(inplace=True)

#Manually inspecting tweets
pd.set_option('display.max_colwidth', -1)
tweetSet.head(20)
```

Out[39]:

	id	label	text
0	572342978255048705	racism	So Drasko just said he was impressed the girls cooked half a chicken.. They cooked a whole one #MKR
2	572341498827522049	racism	Drasko they didn't cook half a bird you idiot #mkr
4	572340476503724032	racism	Hopefully someone cooks Drasko in the next ep of #MKR
6	572334712804384768	racism	of course you were born in serbia...you're as fucked as A Serbian Film #MKR
7	572332655397629952	racism	These girls are the equivalent of the irritating Asian girls a couple years ago. Well done, 7. #MKR
8	575949086055997440	racism	#MKR Lost the plot - where's the big Texan with the elephant sized steaks that they all have for brekkie ?
491	575174115667017728	racism	RT @PhxKen: SIR WINSTON CHURCHHILL: "ISLAM IS A DANGEROUS IN A MAN AS RABIES IN A DOG" http://t.co/kCXgKD70SK
1266	569294066984202240	racism	RT @TheRightWingM: Giuliani watched his city attacked & people jump to their deaths. He's entitled to say WTF he wants about the guy shield…
1942	446460991396917248	racism	RT @YesYoureRacist: At least you're only a tiny bit racist RT @AnMo95: I'm not racist, but my dick is!
1943	489938636956135424	racism	@MisfitInChains @oldgfatherclock @venereveritas13 SANTA JUST *IS* WHITE
1944	502315096052596736	racism	RT @Dreamdefenders: Eric Holder from #ferguson: "I understand that mistrust. I am the Attorney General, but I am also a Black Man" http://t…
1945	502508643204755456	racism	RT @AntonioFrench: I spent the morning at the Board of Elections getting maps/data to start registering every black person in #Ferguson. ht…
1946	537442614534864896	racism	RT @harikondabolu: In his #Ferguson testimony, Darren Wilson said he thought "it looks like a demon" before he shot Mike Brown. Ironic word…
1947	538130705382129664	racism	"He can't be a server at our restaurant, that beard makes him look like a terrorist." Everyone laughs. #fuckthanksgiving
1948	574589943982379008	racism	@GarrettaBrown85 @5Candrew Why do people even talk about white privilege when the majority of food stamp recipients are white people!!
1973	576395841440641024	racism	RT @ThePatriot143: Bet👉3 'Race Pimps' enjoyed watching #Ferguson protesters being interviewed on @seanhannity show as much as I did. http:/…
1974	572346080911736832	sexism	These two are revolting #MKR #MKR2015
1976	572348198062170112	sexism	#katieandnikki stop calling yourselves pretty and hot..you're not and saying it a million times doesn't make you either...STFU #MKR
1978	572319306387599360	sexism	The menus look like they were made by a 5 year old little girl...in this case just the mental age of a 5 year old girl I guess #MKR

	id	label	text
1979	572347842456522752	sexism	Wish these blondes were in that How To Get Away With Murder show....#MKR

Hard to say at this point whether this dataset can reliably be used for hate speech detection in my task. Undoubtedly there is quite a bit of hate speech tweets in this dataset, however a lot of this does depend on the context of the individual they are targeting at the time - the tweets talking about #mkr - (a show called my kitchen rules in australia) is deemed sexist or racist at times only because of the subject they are talking about. Likewise there is deep politcal complexities in some of the tweets above such as the ones about ferguson and 9/11. Will have to consult supervisor.

Data from annotater reliability study - uses much of the Waseem & Hovy dataset directly above but has some extra data

The below data is from a study where the partcipants aimed to examine how reliable amateur annotaters were compared to experts in annotating hate speech. They used a lot of data from the above Waseem and Hovy 2016 dataset, (there is an overlap of 2,876 tweets).

I only retrieved the opinions of the expert annotaters, as in the paper of the study itself they claim that the amateur annotaters are unreliable. Maybe this data retrieval gets some useful data, if anything at least it may give me some benign tweets which again is useful to the BERT classifier I'll be using later

In [43]:

```
path = r'https://raw.githubusercontent.com/ZeerakW/hatespeech/master/NLP%2BCSS_2016.csv'

ids = pd.read_csv(path, sep='\t', usecols = ["TweetID", "Expert"], index_col=False);

ids.head()
```

Out[43]:

	TweetID	Expert
0	597576902212063232	neither
1	565586175864610817	neither
2	563881580209246209	neither
3	595380689534656512	neither
4	563757610327748608	neither

In [44]:

```
ids.rename(columns = {'TweetID':'id'}, inplace = True)
ids.rename(columns = {'Expert': 'label'}, inplace = True)
ids.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6909 entries, 0 to 6908
Data columns (total 2 columns):
id        6909 non-null int64
Expert    6909 non-null object
dtypes: int64(1), object(1)
memory usage: 108.0+ KB
```

In [45]:

```
good_tweet_ids = list(ids['id']) #Create a list of tweet ids to look up
results = lookup_tweets(good_tweet_ids, api) #apply function
```

In [46]:

```
temp = json.dumps([status._json for status in results]) #create JSON
newdf = pd.read_json(temp, orient='records')
full = ids.merge(newdf,  how='left', on = 'id')
```

In [47]:

```
tweetSet1 = full[['id', 'Expert', 'text']]
tweetSet1 = tweetSet1.drop_duplicates(subset = 'id')
pd.set_option('display.max_colwidth', -1)
dups1 = len(tweetSet1) - tweetSet1['text'].count()
tweetSet1.dropna(inplace=True)
tweetSet1.head(10)
```

Out[47]:

	id	Expert	text
0	597576902212063232	neither	Cisco had to deal with a fat cash payout to the FSF *and* allow an external party to do constant reviews of their FOSS license compliancy.
1	565586175864610817	neither	@MadamPlumpette I'm decent at editing, no worries ^.^
2	563881580209246209	neither	@girlziplocked will read. gotta go afk for a bit - still bringing stuff in from car after week long road trip.
3	595380689534656512	neither	guys. show me the data. show me your github. tell me your story. show me something that makes me think you're not a bag of useless opinions.
4	563757610327748608	neither	@tpw_rules nothings broken. I was just driving througg a lot of water.
5	563082741370339330	neither	ur face is classified as a utility by the FCC.
6	596962098845851648	neither	@lysandraws yay! Absolutely. I'm not gone until November :)
7	563874350038675457	neither	RT @kashiichan: "It really feels like the @twitter DM can be the hand-on-the-knee of social communication." http://t.co/7mFseL5zfE #stopwad…
8	597240424873394176	neither	@SirenSailor rtfm. http://t.co/jaMXHikl3u
10	595306172833353728	neither	@Popehat who wouldn't?

Data content of both datasets after dropping null rows:

In [48]:

```
print("There are", len(tweetSet.index), "tweets in the first dataframe with",\
      dups, "duplicates")

print("And", len(tweetSet1.index), "tweets in the second dataframe with",\
     dups1, "duplicates")
```

```
There are 10866 tweets in the first dataframe with 5983 duplicates
And 6240 tweets in the second dataframe with 669 duplicates
```

Combining the two datasets and saving as csv

Hopefully with the merge function I'll be able to reliably combine the two datasets and have no duplicates in the final result. As stated above there is supposed to be an overlap of 2,786 tweets so this final set of data should be 16,907 + 6,909 - 2,786 - 5983 - 669 = 14,378 in the overall combined set.

In [132]:

```
fullSet = pd.merge(tweetSet, tweetSet1,  how='outer', on = ['id', 'text', 'label'])
fullSet.drop_duplicates(subset= ['id'], keep = 'last', inplace = True)
fullSet.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 13172 entries, 3 to 17046
Data columns (total 3 columns):
id       13172 non-null int64
label    13172 non-null object
text     13172 non-null object
dtypes: int64(1), object(2)
memory usage: 411.6+ KB
```

Not the exact number of tweets we were seeking, but perhaps this is because when that datset was first collected - all of those tweets were available to be retrieved; wheras now they can't be because the accounts they belong to are suspended.

In [26]:

```
#Below we combined both sets of data to get a local copy of the dataframe:

#fullSet.to_csv('Waseem_Hovy_2016.csv', sep = ',', encoding='utf-8', index = False, header

path = r'Raw_Data\Waseem_Hovy_2016.csv'

ids = pd.read_csv(path, sep=',');

print(ids.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13172 entries, 0 to 13171
Data columns (total 3 columns):
id       13172 non-null int64
label    13172 non-null object
text     13172 non-null object
dtypes: int64(1), object(2)
memory usage: 308.8+ KB
None
```

Cleaning and merging the Datasets to create an overall database of tweets

We will be implementing a transfer learning approach to NLP in this project and so this database will be designed with that in mind. There will be three steps in this process

1. **Train a BERT sentence encoder on the whole dataset to learn tweet language**
2. **Transfer the knowledge learned from the BERT model onto a neural network model classifying tweets that are offensive**
3. **Transfer the knowledge learned from the prvious model and attempt to classify either offensive or hate speech**

Because of this the overall database will be simple, there will be three columns. One column for the text content for a tweet, one column denoting if it's an offensive tweet or not (offensive could mean that it contains profanity but not necessairly harmful in intent) and one column denoting if the tweet is judged to be hate speech or not. All tweets that are hate speech will be considered offesnive.

In [3]:

```
path = r'Raw_Data\OLIDv1.0\olid-training-v1.0.tsv'
offens = pd.read_csv(path, sep='\t');
offens.head(6)
```

Out[3]:

	id	tweet	subtask_a	subtask_b	subtask_c
0	86426	@USER She should ask a few native Americans wh...	OFF	UNT	NaN
1	90194	@USER @USER Go home you're drunk!!! @USER #MAG...	OFF	TIN	IND
2	16820	Amazon is investigating Chinese employees who ...	NOT	NaN	NaN
3	62688	@USER Someone should'veTaken" this piece of sh...	OFF	UNT	NaN
4	43605	@USER @USER Obama wanted liberals & illega...	NOT	NaN	NaN
5	97670	@USER Liberals are all Kookoo !!!	OFF	TIN	OTH

As mentioned before, we cannot say with confidence that any of these tweets can be strictly characterised as hate speech - although it can help with the second stage of our transfer learning model, in that it has a wealth of offensive tweets.

We will exclude the tweets that satisfy all three subtasks; (that the tweet is offensive, is targeted and is directed at a group), because although we saw instances above that may not be hate speech in this category, nevertheless we'll err on the side of caution so as to not contaminate the database by annotating tweets as hate speech when they aren't.

In [4]:

```
offens.drop(offens[(offens["subtask_a"] == 'OFF') & \
                   (offens["subtask_b"] == 'TIN') & \
                   (offens["subtask_c"] == 'GRP')].index, inplace = True)
offens.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 12166 entries, 0 to 13239
Data columns (total 5 columns):
id           12166 non-null int64
tweet        12166 non-null object
subtask_a    12166 non-null object
subtask_b    3326 non-null object
subtask_c    2802 non-null object
dtypes: int64(1), object(4)
memory usage: 570.3+ KB
```

We don't need the columns annotating who the tweets are targeted against and if they are indeed targeted as this information is not important to our investigation. Thus, we'll drop these columns

In [5]:

```
cols = ['subtask_b', 'subtask_c']
offens.drop(cols, inplace = True, axis = 1)
offens.reset_index(drop = True, inplace = True)
offens.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12166 entries, 0 to 12165
Data columns (total 3 columns):
id           12166 non-null int64
tweet        12166 non-null object
subtask_a    12166 non-null object
dtypes: int64(1), object(2)
memory usage: 285.3+ KB
```

In [6]:

```
offens.rename(columns = {'subtask_a':'Offensive', 'tweet': 'Tweet'}, inplace = True)
offens['Hate_Speech'] = 'N'
offens.loc[offens['Offensive'] == 'OFF', 'Offensive'] = "Y"
offens.loc[offens['Offensive'] == 'NOT', 'Offensive'] = "N"
offens.head()
```

Out[6]:

	id	Tweet	Offensive	Hate_Speech
0	86426	@USER She should ask a few native Americans wh...	Y	N
1	90194	@USER @USER Go home you’re drunk!!! @USER #MAG...	Y	N
2	16820	Amazon is investigating Chinese employees who ...	N	N
3	62688	@USER Someone should'veTaken" this piece of sh...	Y	N
4	43605	@USER @USER Obama wanted liberals & illega...	N	N

ICVSM 2017 dataset

In [7]:

```
sometweets = pd.read_csv(r'Raw_Data\ICVSM_2017.csv', sep=',');

cols = ['Unnamed: 0', 'count', 'hate_speech','offensive_language','neither']
sometweets.drop(cols, inplace = True, axis = 1)
sometweets.reset_index(drop = True, inplace = True)
sometweets['Hate_Speech'] = 'N'
sometweets['Offensive'] = 'N'
sometweets.rename(columns = {'tweet': 'Tweet'}, inplace = True)
sometweets.head(1)
```

Out[7]:

	class	Tweet	Hate_Speech	Offensive
0	2	!!! RT @mayasolovely: As a woman you shouldn't...	N	N

Remember the annotation mentioned above, the column we're interested in here is the column "class".

Denoted 0 - for hate speech, 1 - for offensive speech and 2 for neither. This fits perfectly into our existing dataframe structure.

We can drop all columns that aren't either class or tweet and create new columns to match the previous dataframe.

Based on what the class label is on this dataframe, we can populate the Hate_Speech and Offensive columns

In [8]:

```
sometweets.loc[sometweets['class'] == 0, 'Hate_Speech'] = "Y"
sometweets.loc[sometweets['class'] == 0, 'Offensive'] = "Y"
sometweets.loc[sometweets['class'] == 1, 'Offensive'] = "Y"
sometweets.head(5)
```

Out[8]:

	class	Tweet	Hate_Speech	Offensive
0	2	!!! RT @mayasolovely: As a woman you shouldn't...	N	N
1	1	!!!!! RT @mleew17: boy dats cold...tyga dwn ba...	N	Y
2	1	!!!!!!! RT @UrKindOfBrand Dawg!!!! RT @80sbaby...	N	Y
3	1	!!!!!!!!! RT @C_G_Anderson: @viva_based she lo...	N	Y
4	1	!!!!!!!!!!!! RT @ShenikaRoberts: The shit you...	N	Y

In [9]:

```
hatetweets = sometweets[sometweets["Hate_Speech"] == 'Y']
print("There are", len(hatetweets.index), "tweets denoted as hate speech")
#If it went right it should be 1430 entries
offenstweets = sometweets[sometweets["Offensive"] == 'Y']
print("There are", len(offenstweets.index), "tweets denoted as offensive")
```

```
There are 1430 tweets denoted as hate speech
There are 20620 tweets denoted as offensive
```

HatEval 2019 dataset

In [10]:

```
path = r'Raw_Data\semeval_2018_task5_hateval\public_development_en\dev_en.tsv'
dev = pd.read_csv(path, sep='\t');

path1 = r'Raw_Data\semeval_2018_task5_hateval\public_development_en\train_en.tsv'
train = pd.read_csv(path1, sep='\t');
#The IDs below aren't actually tweet IDs, just unrelated numbers

dev.drop('id', inplace = True, axis = 1)
dev.reset_index(drop = True, inplace = True)

train.drop('id', inplace = True, axis = 1)
train.reset_index(drop = True, inplace = True)

semeval = pd.concat([train,dev], axis=0)
semeval.drop_duplicates(subset = 'text',inplace=True)
semeval.reset_index(drop = True, inplace = True)
semeval.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9998 entries, 0 to 9997
Data columns (total 4 columns):
text    9998 non-null object
HS      9998 non-null int64
TR      9998 non-null int64
AG      9998 non-null int64
dtypes: int64(3), object(1)
memory usage: 312.6+ KB
```

In [11]:

```
semeval.head(5)
```

Out[11]:

	text	HS	TR	AG
0	Hurray, saving us $$$ in so many ways @potus @...	1	0	0
1	Why would young fighting age men be the vast m...	1	0	0
2	@KamalaHarris Illegals Dump their Kids at the ...	1	0	0
3	NY Times: 'Nearly All White' States Pose 'an A...	0	0	0
4	Orban in Brussels: European leaders are ignori...	0	0	0

A slight problem with this dataset is that it doesn't have a column denoting whether tweets are offensive or not which makes it not useful to our second stage of classification.

However, it does classify tweets as hate speech or not, and reliably too. Getting reliably annotated hate speech tweets online is rare so we'll use these tweets for the first and third stages of our model.

For tweets in this dataframe that aren't hate speech, we will annotate the 'Offensive' column of our dataset as 'Unknown'

In [12]:

```
semeval.rename(columns = {'text': 'Tweet', 'HS' : 'Hate_Speech'}, inplace = True)
semeval['Offensive'] = 'Unknown'
semeval.loc[semeval['Hate_Speech'] == 1, 'Hate_Speech'] = "Y"
semeval.loc[semeval['Hate_Speech'] == 0, 'Hate_Speech'] = "N"
semeval.loc[semeval['Hate_Speech'] == "Y", 'Offensive'] = "Y"
cols = ['TR', 'AG']
semeval.drop(cols, inplace = True, axis = 1)
semeval.head(10)
```

Out[12]:

	Tweet	Hate_Speech	Offensive
0	Hurray, saving us $$$ in so many ways @potus @...	Y	Y
1	Why would young fighting age men be the vast m...	Y	Y
2	@KamalaHarris Illegals Dump their Kids at the ...	Y	Y
3	NY Times: 'Nearly All White' States Pose 'an A...	N	Unknown
4	Orban in Brussels: European leaders are ignori...	N	Unknown
5	@KurtSchlichter LEGAL is. Not illegal. #BuildT...	Y	Y
6	@RitaPanahi @826Maureen @RealCandaceO Antifa a...	N	Unknown
7	Ex-Teacher Pleads Not guilty To Rape Charges h...	N	Unknown
8	still places on our Bengali (Sylheti) class! i...	N	Unknown
9	DFID Africa Regional Profile: July 2018 https:...	N	Unknown

ICVSM 2018 Abusive dataset

The data labelled as hateful is not very reliable in this dataset, although some may be hate speech.

To avert the risk of contaminating the tweet data we've already labelled as hate speech inn other datasers, *we'll drop tweets labelled as hateful in this set.* This will result in losing around 5000 tweets.

However, the rest of the tweets can be used to train the BERT model - (approx 95,000 tweets) so it's not all doom and gloom

In [13]:

```
icvsm = r'Raw_Data\ICVSM_2018_dataset\hatespeech_text_label_vote.csv'
icvsm = pd.read_csv(icvsm, sep='\t', names = \
                    ["tweets", "majority label" ], index_col = False);

#We're dropping the tweets labelled as 'hateful' as they're unreliably annotated.
icvsm.drop(icvsm.loc[icvsm['majority label']=='hateful'].index, inplace=True)

print("\nThere are", len(icvsm.index), "tweets in this dataset")
```

```
There are 95031 tweets in this dataset
```

In [14]:

```
icvsm['Hate_Speech'] = 'N'
icvsm['Offensive'] = 'N'
icvsm.rename(columns = {'tweets': 'Tweet'}, inplace = True)
icvsm.loc[icvsm['majority label'] == 'abusive', 'Offensive'] = "Y"
icvsm.drop('majority label', inplace = True, axis = 1)
icvsm.head(12)
```

Out[14]:

	Tweet	Hate_Speech	Offensive
0	Beats by Dr. Dre urBeats Wired In-Ear Headphon...	N	N
1	RT @Papapishu: Man it would fucking rule if we...	N	Y
2	It is time to draw close to Him 🙏&#127...	N	N
3	if you notice me start to act different or dis...	N	N
4	Forget unfollowers, I believe in growing. 7 ne...	N	N
5	RT @Vitiligoprince: Hate Being sexually Frustr...	N	Y
6	Topped the group in TGP Disc Jam Season 2! Ont...	N	N
7	That daily baby aspirin for your #heart just m...	N	N
8	I liked a @YouTube video from @mattshea https:...	N	N
9	RT @LestuhGang_: If your fucking up & your...	N	Y
10	Uber finds one allegedly stolen Waymo file – o...	N	N
11	@Move_Fwd give up. You've lost. You will not c...	N	N

In [28]:

```
#Combined Waseem and Hovy datasets below
path = r'Raw_Data\Waseem_Hovy_2016.csv'

ids = pd.read_csv(path, sep=',');

print(ids.info())
print("\nThere are 5 different labels for how the tweets have been annotated:")
print(ids['label'].unique())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13172 entries, 0 to 13171
Data columns (total 3 columns):
id       13172 non-null int64
label    13172 non-null object
text     13172 non-null object
dtypes: int64(1), object(2)
memory usage: 308.8+ KB
None

There are 5 different labels for how the tweets have been annotated:
['racism' 'sexism' 'none' 'neither' 'both']
```

If a tweet has been labelled as racism, sexism or both - we annotate as hate speech and offensive. Otherwise we label as not hate speech; but we can't be sure if some of the tweets that have been labelled as none or neither aren't offensive - so we'll label that as Unknown

In [29]:

```
ids['Hate_Speech'] = 'Y'
ids['Offensive'] = 'Y'

ids.rename(columns = {'text': 'Tweet'}, inplace = True)

ids.loc[ids['label'] == 'none', 'Hate_Speech'] = "N"
ids.loc[ids['label'] == 'none', 'Offensive'] = "Unknown"

ids.loc[ids['label'] == 'neither', 'Hate_Speech'] = "N"
ids.loc[ids['label'] == 'neither', 'Offensive'] = "Unknown"

ids.drop('label', inplace = True, axis = 1)
ids.reset_index(drop = True, inplace = True)

print("Hate Speech Classification Summary:\n" \
     "\nThere are", ids['Hate_Speech'].value_counts()['Y'], \
      "tweets labelled as hate speech and", \
      ids['Hate_Speech'].value_counts()['N'], \
      "tweets labelled as not hate speech" )

print("\nThere are", ids['Offensive'].value_counts()['Unknown'],
     "tweets that in the previous data were not labelled as hate speech," \
      " however we can't be sure they don't contain content that's offensive,",\
     "hence they're labelled as Unknown for that category")


ids.head(5)
```

```
Hate Speech Classification Summary:

There are 3534 tweets labelled as hate speech and 9638 tweets labelled as no
t hate speech

There are 9638 tweets that in the previous data were not labelled as hate sp
eech, however we can't be sure they don't contain content that's offensive,
hence they're labelled as unknown
```

Out[29]:

	id	Tweet	Hate_Speech	Offensive
0	572334712804384768	of course you were born in serbia...you're as ...	Y	Y
1	572332655397629952	These girls are the equivalent of the irritati...	Y	Y
2	575949086055997440	#MKR Lost the plot - where's the big Texan wi...	Y	Y
3	575174115667017728	RT @PhxKen: SIR WINSTON CHURCHHILL: "ISLAM IS ...	Y	Y
4	569294066984202240	RT @TheRightWingM: Giuliani watched his city a...	Y	Y

Testing for possible duplicate tweets

Where tweet ID is available, we want to use it at the beginning to root out possible duplicate tweets which may have overlapped over the datasets.

Removing duplicates via tweet ID is much more reliable, but after this we'll attempt to remove duplicate tweets via the text content.

There are only two datasets that have an accompanying ID, the ICVSM dataset has IDs along with their label as another csv file in the directory. I'll test now to see if there's any overlap between the ICVSM_2018 tweets and the Waseem_And Hovy tweets

In [31]:

```
icvsm1 = r'Raw_Data\ICVSM_2018_dataset\hatespeech_id_label.csv'
icvsm1 = pd.read_csv(icvsm1, sep=',', names = \
                     ["id", "majority label" ], index_col = False);

#Again dropping 'hateful' tweets as we won't be using them
icvsm1.drop(icvsm1.loc[icvsm1['majority label']=='hateful'].index, \
            inplace=True)


icvsm1.head(2)
print("\nThere are", len(icvsm1.index), "tweets in this dataset")
```

```
There are 95031 tweets in this dataset
```

In [12]:

```
path = r'Raw_Data\Waseem_Hovy_2016.csv'

ids1 = pd.read_csv(path, sep=',');

ids1.head(2)
```

Out[12]:

	id	label	text
0	572334712804384768	racism	of course you were born in serbia...you're as fucked as A Serbian Film #MKR
1	572332655397629952	racism	These girls are the equivalent of the irritating Asian girls a couple years ago. Well done, 7. #MKR

In [13]:

```
ids1.label.value_counts()
```

Out[13]:

```
neither    5497
none       4141
sexism     3415
racism     94
both       25
Name: label, dtype: int64
```

In [33]:

```
print("\nThere are", len(icvsm1.index), "tweets in the ICVSM dataset and", \
      len(ids1.index), "tweets in the Waseem_Hovy dataset")

print("\nThe end merge of datasets should have", len(icvsm1.index) + len(ids1.index),\
      "tweets in the dataset assuming no duplicates")
```

```
There are 95031 tweets in the ICVSM dataset and 13172 tweets in the Waseem_H
ovy dataset

The end merge of datasets should have 108203 tweets in the dataset assuming
no duplicates
```

In [34]:

```
dt = pd.merge(icvsm1, ids1,  how='outer', on = ['id'])
duplicateRows = dt[dt.duplicated(subset = ['id'], keep = False)]
print("When using the duplicated method there are", \
      len(duplicateRows), "duplicate ids identified, we'll obviously retain",\
     "half of these because we'll keep the first instance of the duplicate tweet",\
     "this will result in", int(len(duplicateRows)/2), "tweets being dropped")
duplicateRows.head(10)
```

```
When using the duplicated method there are 330 duplicate ids identified, w
e'll obviously retain half of these because we'll keep the first instance of
the duplicate tweet this will result in 165 tweets being dropped
```

Out[34]:

	id	majority label	label	text
9	849087242987593728	abusive	NaN	NaN
10	849087242987593728	abusive	NaN	NaN
13	849282894682050564	abusive	NaN	NaN
14	849282894682050564	abusive	NaN	NaN
36	849881409284182016	normal	NaN	NaN
37	849881409284182016	normal	NaN	NaN
46	848975292794318848	normal	NaN	NaN
47	848975292794318848	normal	NaN	NaN
52	850346419164553218	normal	NaN	NaN
53	850346419164553218	normal	NaN	NaN

As shown above, identifying duplicate tweets via id is quite reliable as the ids above are identical. They're often entered in consecutive indexes in the database which leads me to believe it must have been human error in the large ICVSM 2017 database of 100,000 tweets. This isn't a huge amount of error when considerig the overall size of the dataframe, however we do need to weed out these duplicates because they may give us inaccurate scores when we do cross validation evaluation down the line

In [35]:

```
dt.drop_duplicates(subset= ['id'], keep = 'first', inplace = True)
print("When using the drop_duplicates method there are",\
(len(icvsm1.index) + len(ids1.index) - len(dt.index)), "duplicate tweets\n")

print(dt.info())
dt.head()
```

```
When using the drop_duplicates method there are 165 duplicate tweets

<class 'pandas.core.frame.DataFrame'>
Int64Index: 108038 entries, 0 to 108202
Data columns (total 4 columns):
id                108038 non-null int64
majority label    94866 non-null object
label             13172 non-null object
text              13172 non-null object
dtypes: int64(1), object(3)
memory usage: 4.1+ MB
None
```

Out[35]:

	id	majority label	label	text
0	849667487180259329	abusive	NaN	NaN
1	850490912954351616	abusive	NaN	NaN
2	848791766853668864	abusive	NaN	NaN
3	848306464892604416	abusive	NaN	NaN
4	850010509969465344	normal	NaN	NaN

Let's see if dropping duplicate entries via text content is a reliable method

In [36]:

```
icvsm1 = r'Raw_Data\ICVSM_2018_dataset\hatespeech_text_label_vote.csv'
icvsm1 = pd.read_csv(icvsm1, sep='\t', names = \
                     ["text", "label" ], index_col = False);

icvsm1.drop(icvsm1.loc[icvsm1['label']=='hateful'].index, inplace=True)

dt = pd.merge(icvsm1, ids1,  how='outer', on = ['text', 'label'])

duplicateRows = dt[dt.duplicated(subset = ['text'], keep = False)]

print("When using the duplicated method there are", \
      len(duplicateRows), "duplicate ids identified, we'll obviously retain",\
     "half of these because we'll keep the first instance of the duplicate tweet",\
     "this will result in", int(len(duplicateRows)/2), "tweets being dropped")
pd.set_option('display.max_colwidth', -1)
duplicateRows.head(30)
```

```
When using the duplicated method there are 8809 duplicate ids identified, w
e'll obviously retain half of these because we'll keep the first instance of
the duplicate tweet this will result in 4404 tweets being dropped
```

Out[36]:

	text	label	id
1	RT @Papapishu: Man it would fucking rule if we had a party that was against perpetual warfare.	abusive	NaN
2	RT @Papapishu: Man it would fucking rule if we had a party that was against perpetual warfare.	abusive	NaN
3	RT @Papapishu: Man it would fucking rule if we had a party that was against perpetual warfare.	abusive	NaN
7	RT @Vitiligoprince: Hate Being sexually Frustrated Like I wanna Fuck But ion wanna Just fuck anybody	abusive	NaN
8	RT @Vitiligoprince: Hate Being sexually Frustrated Like I wanna Fuck But ion wanna Just fuck anybody	abusive	NaN
12	RT @LestuhGang_: If your fucking up & your homies dont tell you that your fucking up, those ain't your homies	abusive	NaN
13	RT @LestuhGang_: If your fucking up & your homies dont tell you that your fucking up, those ain't your homies	abusive	NaN
16	RT @ennoia3: That's one way he pulls you in RT@amysreedusxx norman fucking reedus just threw candy at me when will your fav ever https://t.…	abusive	NaN
17	RT @ennoia3: That's one way he pulls you in RT@amysreedusxx norman fucking reedus just threw candy at me when will your fav ever https://t.…	abusive	NaN
20	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
21	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
22	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
23	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
24	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
25	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
26	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
27	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN

	text	label	id
28	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
29	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
30	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
31	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
32	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
33	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
34	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
35	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
36	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
37	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
38	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN
39	RT @EiramAydni: Im a nasty ass freak when I like you..	abusive	NaN

In [37]:

```
dt.drop_duplicates(subset= ['text'], keep = 'first', inplace = True)
print("\nOverall, there are", len(dt.index), "tweets in this dataset with",\
     (len(icvsm1.index) + len(ids1.index) - len(dt.index)), "duplicate tweets\n")
```

```
Overall, there are 101139 tweets in this dataset with 7064 duplicate tweets
```

There weren't 5005 duplicates as predicted but even more with 7064, this is likely because as we can see above there are some tweets duplicated more than once. These tweets likely weren't picked up by checking via id because they are likely from different individual sources, but they are retweets so they have the same text content.

We do NOT want duplicate text entries in our final set, as it will contaminate our validation set when we eventually do use cross-validation to evaluate the performance of our model.

Thus, we will drop duplicate tweets via dropping by text, it's also seemingly identified duplicates reliably which is a plus.... Below I'll inspect a little further just to verify it's accurate

In [38]:

```
pd.set_option('display.max_colwidth', -1)
duplicateRows.head(200)
```

Out[38]:

	text	label	id
1	RT @Papapishu: Man it would fucking rule if we had a party that was against perpetual warfare.	abusive	NaN
2	RT @Papapishu: Man it would fucking rule if we had a party that was against perpetual warfare.	abusive	NaN
3	RT @Papapishu: Man it would fucking rule if we had a party that was against perpetual warfare.	abusive	NaN
7	RT @Vitiligoprince: Hate Being sexually Frustrated Like I wanna Fuck But ion wanna Just fuck anybody	abusive	NaN
8	RT @Vitiligoprince: Hate Being sexually Frustrated Like I wanna Fuck But ion wanna Just fuck anybody	abusive	NaN
...	...	...	...
313	RT @mdlbird: 22 fucking years ago https://t.co/GhTeY9qoOl	abusive	NaN
314	RT @mdlbird: 22 fucking years ago https://t.co/GhTeY9qoOl	abusive	NaN
315	RT @mdlbird: 22 fucking years ago https://t.co/GhTeY9qoOl	abusive	NaN
316	RT @mdlbird: 22 fucking years ago https://t.co/GhTeY9qoOl	abusive	NaN
317	RT @mdlbird: 22 fucking years ago https://t.co/GhTeY9qoOl	abusive	NaN

200 rows × 3 columns

I have a suspicion that most of the duplicate entries come from within the ICVSM database.

I'll test this quickly below by calculating the amount of duplicates within the ICVSM database, and subtracting it from the overlap I *believed* was between the two datasets, when I was assuming there were no duplicates within the ICVSM dataset

In [39]:

```
duplicateRows1 = icvsm1[icvsm1.duplicated(subset = ['text'], keep = False)]

difference = len(duplicateRows) - len(duplicateRows1)

print("The actual overlap between ICVSM 2018 and the Waseem & Hovy Database is",\
     difference)
```

```
The actual overlap between ICVSM 2018 and the Waseem & Hovy Database is 0
```

Okay so all the duplicates were within the ICVSM dataset

Finally combining all of the data

All of the previous kernels in the section before the last one where you were testing for duplicates must be ran in order to get a cleaned, consistent version of each set

In [40]:

```
#Drop duplicates within the icvsm_2018 dataset first to get a fair approximation
#of how many tweets are actually overlapped between all the datasets
icvsm.drop_duplicates(subset= ['Tweet'], keep = 'first', inplace = True)

pdlist = [offens, sometweets, semeval, icvsm, ids]
final_df = pd.concat(pdlist, sort = True, axis = 0)
cols = ['class', 'id']
final_df.drop(cols, inplace = True, axis = 1)
final_df.reset_index(drop = True, inplace = True)
#final_df.drop_duplicates(subset= ['Tweet'], keep = 'first', inplace = True)
print(final_df.info())
final_df.head(5)
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148086 entries, 0 to 148085
Data columns (total 3 columns):
Hate_Speech    148086 non-null object
Offensive      148086 non-null object
Tweet          148086 non-null object
dtypes: object(3)
memory usage: 3.4+ MB
None
```

Out[40]:

	Hate_Speech	Offensive	Tweet
0	N	Y	@USER She should ask a few native Americans what their take on this is.
1	N	Y	@USER @USER Go home you’re drunk!!! @USER #MAGA #Trump2020 👊🇺🇸👊 URL
2	N	N	Amazon is investigating Chinese employees who are selling internal data to third-party sellers looking for an edge in the competitive marketplace. URL #Amazon #MAGA #KAG #CHINA #TCOT
3	N	Y	@USER Someone should'veTaken" this piece of shit to a volcano. 😂"
4	N	N	@USER @USER Obama wanted liberals & illegals to move into red states

In [41]:

```
final_df1 = final_df.drop_duplicates(subset= ['Tweet'], keep = 'first')
final_df1.reset_index(drop = True, inplace = True)
diff = len(final_df.index) - len(final_df1.index)

print("The amount of tweets lost in the final dataframe by using the drop duplicates by tex
      diff, "which is {0:.2}%".format(diff/len(final_df.index) * 100), \
      "of the original dataset")
```

```
The amount of tweets lost in the final dataframe by using the drop duplicate
s by text entry is 29 which is 0.02% of the original dataset
```

In [42]:

```
print( "There are", len(final_df1.index), "tweets in the final dataset\n")
final_df1.info()
```

```
There are 148057 tweets in the final dataset

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 148057 entries, 0 to 148056
Data columns (total 3 columns):
Hate_Speech    148057 non-null object
Offensive      148057 non-null object
Tweet          148057 non-null object
dtypes: object(3)
memory usage: 3.4+ MB
```

In [24]:

```
final_df1.to_csv('final.csv', sep = ',', encoding='utf-8', \
                 index = False, header = True)

print("Hate Speech Column Labelling:\n", final_df1.Hate_Speech.value_counts())

print("\nOffensive Column Labelling:\n", final_df1.Offensive.value_counts())
```

```
Hate Speech Column Labelling:
 N    138883
Y     9174
Name: Hate_Speech, dtype: int64

Offensive Column Labelling:
 N          80127
Y          52504
Unknown    15426
Name: Offensive, dtype: int64
```

Pre-processing Tweets

In this section, some methods will be created to preprocess tweet data

In [19]:

```
import html
import nltk
from nltk.stem.porter import *
stopwords=stopwords = nltk.corpus.stopwords.words("english")

other_exclusions = ["#ff", "ff", "rt"]
stopwords.extend(other_exclusions)

stemmer = PorterStemmer()

final_df1 = pd.read_csv('final.csv', sep = ',')
```

Initial pre-process method

In [46]:

```
def preprocess(text_string):
    """
    Accepts a text string and replaces:
    1) urls with url
    2) lots of whitespace with one instance
    3) mentions with @user
    4) Also uses the html.unescape() method to convert unicode to text counterpart
    5) Replace & with and
    6) Remove the fact the tweet is a retweet if it is - knowing the tweet is
       a retweet does not help towards our classification task.
    This allows us to get standardized counts of urls and mentions
    Without caring about specific people mentioned
    """
    space_pattern = '\s+'
    giant_url_regex = ('http[s]?://(?:[a-zA-Z]|[0-9]|[#$-_@.&+]|'
        '[!*\(\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+')
    mention_regex = '@[\w\-]+'
    RT_regex = '(RT|rt)[ ]*@[ ]*[\S]+'

    # Replaces urls with URL
    parsed_text = re.sub(giant_url_regex, 'url', text_string)

    # Remove the fact the tweet is a retweet.
    # (we're only interested in the language of the tweet here)
    parsed_text = re.sub(RT_regex, ' ', parsed_text)

    # Replaces mentions with the common @USER
    parsed_text = re.sub(mention_regex, '@user',  parsed_text)

    # Converting unicode within tweet to text
    parsed_text = html.unescape(parsed_text)

    #Replace '&' with 'and' - might as well
    parsed_text = parsed_text.replace(r'&', r'and')

    #Remove excess whitespace at the end
    parsed_text = re.sub(space_pattern, ' ', parsed_text)

    #Set text to lowercase and strip
    parsed_text = parsed_text.lower()
    parsed_text = parsed_text.strip()

    return parsed_text
```

Results of text preprocessing

On Mentions:

Replaces @CeleyNichole, @white_thunduh with @USER

In [47]:

```
testtweet = final_df1['Tweet'][12234]

print("Original:", testtweet)
print("\nPreprocessed:", preprocess(testtweet))
```

```
Original: "@CeleyNichole: @white_thunduh how come you never bring me food" i
dont have a car retard

Preprocessed: "@user: @user how come you never bring me food" i dont have a
car retard
```

On URLs and tweets with unicode:

Replaces link at the end with URL and unicode string &#8220 with it's text counterpart "

In [48]:

```
testtweet1 = final_df1['Tweet'][14000]
print("Original:", testtweet1)
print("\nPreprocessed:", preprocess(testtweet1))
```

```
Original: &#8220;@nhalegood: When hoes feel like their photo didnt get enoug
h favorites http://t.co/ZDf98BpF94&#8221; (http://t.co/ZDf98BpF94&#8221;)

Preprocessed: “@user: when hoes feel like their photo didnt get enough favor
ites url
```

On Retweets (RT)

In [49]:

```
testtweet1 = final_df1['Tweet'][32156]
print("Original:", testtweet1)
print("\nPreprocessed:", preprocess(testtweet1))
```

```
Original: RT @simplyalize: "@xonayyy: &#8220;@ugglyyy: well ain't this bout
a bitch ... http://t.co/CVJadMYpg6&#8221; (http://t.co/CVJadMYpg6&#8221;) &#
128514;&#128514;&#128514;" BRUH&#128553;&#128514;&#128514;

Preprocessed: "@user: “@user: well ain't this bout a bitch ... url 😂😂😂"
 bruh😭😂😂
```

Replacing emojis with text

The tweet above shows that we have emojis in the tweet data and these could be perhaps crucial for identifying sarcasm in tweet speech.

Often NLP classifiers may miss things in translation when emojis are involved as they can often convey things not immediately obvious to machines in text like sarcasm.

We'll see if we can convert them to a text representation and also we'll see what our performance is like if we just leave them alone.

In [50]:

```
#demoji link -
#https://pypi.org/project/demoji/#files
#Find the install for the wheel file

#Put the below in anaconda prompt
#pip install demoji-0.1.5-py3-none-any.whl

import demoji
demoji.download_codes()
def emojiReplace(text_string):

    emoji_dict = demoji.findall(text_string)
    for emoji in emoji_dict.keys():
        text_string = text_string.replace(emoji, '{' + emoji_dict[emoji] + '} ' )

    return text_string

#demoji.replace(preprocess(testtweet1), repl = )

testtweet1 = final_df1['Tweet'][32156]
emoji_dict = demoji.findall(preprocess(testtweet1))

print("Tweet with emojis:", (preprocess(testtweet1)))
print("\nProcessed tweet:", emojiReplace(preprocess(testtweet1)))
```

```
Downloading emoji data ...
... OK (Got response in 0.68 seconds)
Writing emoji data to C:\Users\fionn\.demoji/codes.json ...
... OK
Tweet with emojis: "@user: “@user: well ain't this bout a bitch ... url 😂
😂😂" bruh😩😂😂

Processed tweet: "@user: “@user: well ain't this bout a bitch ... url {face
 with tears of joy} {face with tears of joy} {face with tears of joy} " bruh
{weary face} {face with tears of joy} {face with tears of joy}
```

In [51]:

```
testtweet1 = final_df1['Tweet'][18290]
print("Original:", preprocess(testtweet1))
print("\nPreprocessed:", emojiReplace(preprocess(testtweet1)))
```

```
Original: @user 😂😂💀💀💀💀bitch you outta line

Preprocessed: @user {face with tears of joy} {face with tears of joy} {skul
l} {skull} {skull} {skull} bitch you outta line
```

In [52]:

```
testtweet1 = final_df1['Tweet'][14340]
print("\nOriginal:", testtweet1)
print("Preprocessed:", preprocess(testtweet1))
```

```
Original: 1) Vader is a Sith. Not a Jedi. Duh. 2) Vader is also a whiny bitc
h. 3) YODA IS MOTHERFUCKING YODA. #micdrop
Preprocessed: 1) vader is a sith. not a jedi. duh. 2) vader is also a whiny
bitch. 3) yoda is motherfucking yoda. #micdrop
```

Contraction

This is simply the practice of replacing *contracted* words like "would've" with it' decontracted counterpart "would have".

This sounds quite simple but it does have the issue of being sometimes ambiguous, for example the word "how's" can be interpreted as "how is/how has/ how was"

Anyways I'll attempt to develop a function that addresses this. Although it can be argued that most cases aren't ambiguous and that the vast majoirty of the time the word how's would be how is and also not much is lost in translation when referring to hate speech, but I digress

Could enter code to do this here but for the moment hold back on it because perhaps BERT has already been pre-trained for this. We'll look at creating and implementing this code later and see if it improves results

Putting the cleaned tweet text in my dataframe

In [53]:

```
# Preprocessing the tweet text using custom function above
final_df1['cleaned_text'] = final_df1['Tweet'].apply(preprocess)

#Converting emojis in the tweet data to text
final_df1['cleaned_text'] = final_df1['cleaned_text'].apply(emojiReplace)

final_df1.head(20)
```

Out[53]:

	Hate_Speech	Offensive	Tweet	cleaned_text
0	N	Y	@USER She should ask a few native Americans what their take on this is.	@user she should ask a few native americans what their take on this is.
1	N	Y	@USER @USER Go home you’re drunk!!! @USER #MAGA #Trump2020 👊us👊 URL	@user @user go home you’re drunk!!! @user #maga #trump2020 {oncoming fist} {flag: United States} {oncoming fist} url
2	N	N	Amazon is investigating Chinese employees who are selling internal data to third-party sellers looking for an edge in the competitive marketplace. URL #Amazon #MAGA #KAG #CHINA #TCOT	amazon is investigating chinese employees who are selling internal data to third-party sellers looking for an edge in the competitive marketplace. url #amazon #maga #kag #china #tcot
3	N	Y	@USER Someone should'veTaken" this piece of shit to a volcano. 😂"	@user someone should'vetaken" this piece of shit to a volcano. {face with tears of joy} "
4	N	N	@USER @USER Obama wanted liberals & illegals to move into red states	@user @user obama wanted liberals and illegals to move into red states
5	N	Y	@USER Liberals are all Kookoo !!!	@user liberals are all kookoo !!!
6	N	Y	@USER @USER Oh noes! Tough shit.	@user @user oh noes! tough shit.
7	N	N	@USER Buy more icecream!!!	@user buy more icecream!!!
8	N	Y	@USER Canada doesn’t need another CUCK! We already have enough #LooneyLeft #Liberals f**king up our great country! #Qproofs #TrudeauMustGo	@user canada doesn’t need another cuck! we already have enough #looneyleft #liberals f**king up our great country! #qproofs #trudeaumustgo
9	N	N	@USER @USER @USER It’s not my fault you support gun control	@user @user @user it’s not my fault you support gun control
10	N	N	@USER What’s the difference between #Kavanaugh and @USER One of these men admitted to groping a 15 year old girl years ago. The other is going to be #confirmed to the SCJ #DemsareFrauds #DemsAreDone #WalkAwayDemocrats2018 #redwave #VoteRedSaveAmerica #trumptrain #MAGA URL	@user what’s the difference between #kavanaugh and @user one of these men admitted to groping a 15 year old girl years ago. the other is going to be #confirmed to the scj #demsarefrauds #demsaredone #walkawaydemocrats2018 #redwave #voteredsaveamerica #trumptrain #maga url

	Hate_Speech	Offensive	Tweet	cleaned_text
11	N	Y	@USER you are a lying corrupt traitor!!! Nobody wants to hear anymore of your lies!!! #DeepStateCorruption URL	@user you are a lying corrupt traitor!!! nobody wants to hear anymore of your lies!!! #deepstatecorruption url
12	N	N	@USER @USER @USER It should scare every American! She is playing Hockey with a warped puck!	@user @user @user it should scare every american! she is playing hockey with a warped puck!
13	N	N	@USER I like my soda like I like my boarders with a lot of ICE.	@user i like my soda like i like my boarders with a lot of ice.
14	N	N	@USER you are also the king of taste	@user you are also the king of taste
15	N	N	#MAGA @USER 🎶 Sing like no one is listening ♥ Love like you've never been hurt ✔ Vote GOP when no one is watching 🙉 And don't listen to Liberals' dirt URL	#maga @user {musical notes} sing like no one is listening {heart suit} love like you've never been hurt {check mark} vote gop when no one is watching {hear-no-evil monkey} and don't listen to liberals' dirt url
16	N	N	5/5: @USER The time is right for this House to respond to the concerns of all Canadians. Four out of five Canadians support stronger gun control and with good reason." #guncontrol #cdnpoli #cdnhist"	5/5: @user the time is right for this house to respond to the concerns of all canadians. four out of five canadians support stronger gun control and with good reason." #guncontrol #cdnpoli #cdnhist"
17	N	N	@USER Besides Jax's mom and maybe Ope he is hands down my favorite he's like the only good person on the show 😂	@user besides jax's mom and maybe ope he is hands down my favorite he's like the only good person on the show {face with tears of joy}
18	N	Y	@USER @USER @USER gun control! That is all these kids are asking for!	@user @user @user gun control! that is all these kids are asking for!
19	N	Y	@USER @USER @USER @USER LOL!!! Throwing the BULLSHIT Flag on such nonsense!! #PutUpOrShutUp #Kavanaugh #MAGA #CallTheVoteAlready URL	@user @user @user @user lol!!! throwing the bullshit flag on such nonsense!! #putuporshutup #kavanaugh #maga #callthevotealready url

In [54]:

```
print("Breakdown of tweets labelled as offensive:")
print(final_df1.Offensive.value_counts())

print("\nBreakdown of tweets labelled as Hate_Speech:")
print(final_df1.Hate_Speech.value_counts())

print("\nThere are", len(final_df1.index), "tweets in total")
```

```
Breakdown of tweets labelled as offensive:
N          80127
Y          52504
Unknown    15426
Name: Offensive, dtype: int64

Breakdown of tweets labelled as Hate_Speech:
N    138883
Y    9174
Name: Hate_Speech, dtype: int64

There are 148057 tweets in total
```

In [63]:

```
#Be careful storing this back as a csv file though because perhaps your
# pre-processing gets lost in the translation
final_df1.to_csv('final_clean.csv')
```

In [ ]:

```

```