**Course Code**: COMPSCI 5078

**Course Title**: Web Science

**Assessment Title**: Msc individual coursework

**Report Topic**: Social Media Emotion Data Set

**Student Name**: Tong Shi

**Student ID**: 2431206s

**GitHub information for code**: https://github.com/stbmilu/wscw.

**GitHub information for data:** https://github.com/stbmilu/wscw/tree/master/Final_results

## Content

# 1.0 Introduction

The perspective of this research is to collect 900 reasonably clean tweets for six classes, i.e. happy, excitement, anger, surprise, fear, pleasant. The data is fetched by twitter API based on Tweepy python library. The coding language is python and IDE chosen in this research is Vim. Data is stored using Mongodb, and handle by Pymongo python library to perform data process. The time for data collection is from March 7th 04:00 to March 8th 06:00.

| Library | Version | Aim |
|---|---|---|
| tweepy | 3. 8. 0 | Data fetch |
| nltk | 3. 4. 5 | Text process |
| emoji | 0. 5. 4 | Text process |
| numpy | 1. 17. 2 | Extract results |
| pymongo | 3. 10. 1 | Data store |
| pandas | 1. 0. 1 | Data to csv file |
| contractions | 0. 0. 24 | Text process |

Table1: The important python libraries used in this research.

This report will demonstrate the data fetch methods, data processing methods, and crowdsourcing scheme, and discuss the final results to verify whether the tweet data classification is good or not.

# 2.0 Data crawl and Rules

## 2.1 Streaming API and rest API

To collect tweets, this research is planned to use streaming API. However, it is found, for "anger" and "fear" class, fetch duration is too long, because streaming API is for collecting current published tweet. Hence, this research changed to rest API to collect tweets published at past. Tweets fetched by streaming API is used as test data and included in the time duration as mentioned in the introduction section. The actual tweets used in this research is from rest API collected data. The more specific fetch duration/time period will be provided in the following section.

### 2.1.1 Data crawl code

The data crawler uses open authentication (Roger and Josh, 2019) framework to access twitter APIs. The access keys and consumer keys are obtained from twitter developer account.

For rest API:
```
auth = OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET.
api = API(auth, wait_on_rate_limit=True).
Cursor(api.search, q=query, lang="en", tweet_mode='extended').items(amount)
db_collection.insert_one(tweet._json)
```

The python program uses open authentication handler to indentfiy the KEYs generated from twitter developer account. Then, the program creates an authenticated API and use this API in "Cursor" querying method to fetch data by specfiying the amount, language (in this research is english), and query keywords. Finally tweets are inserted into Mongodb.

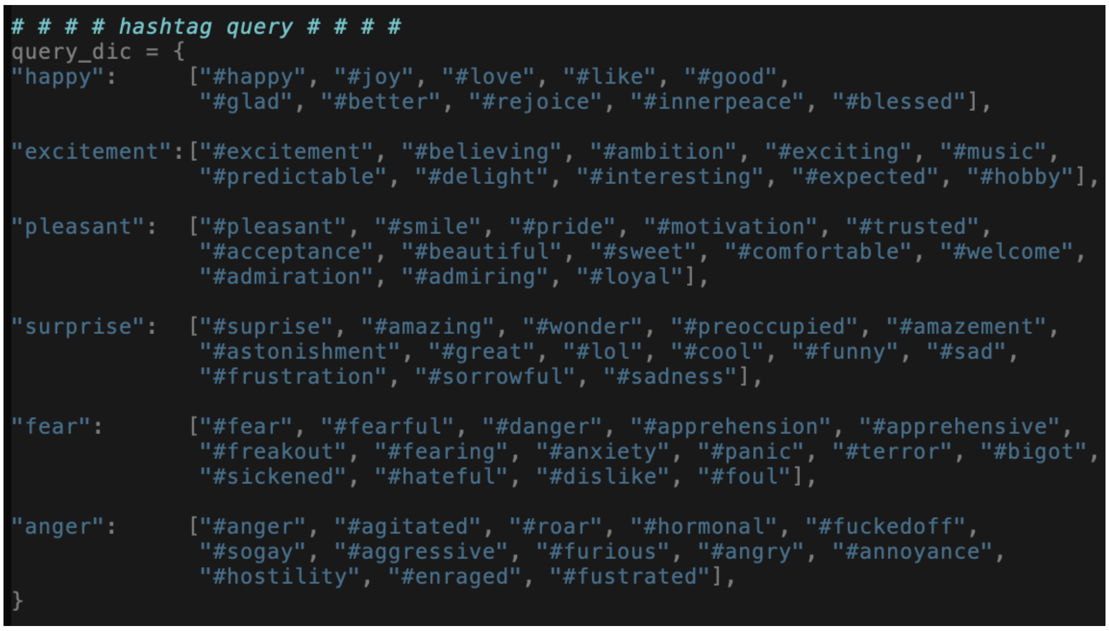
For streaming API:

```
auth = OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
auth.set_access_token(ACCESS_TOKEN, ACCESS_TOKEN_SECRET)
listener = TwitterListener(tweets_filename, collection, max)
stream = Stream(auth, listener)
stream.filter(track=keyword_list, languages=["en"])
```

The python program creates a data listener and authenticate the stream API with this listener. Then the program streaming tweets by specfiying language (english in this research) and keywords wanted for tweets.

## 2.2 Crawling strategies

The actual data fetch method is to use rest API. This research creats a query dictionary for six emotion classes filling with associated query hashtags. Each hashtag is used as data fetch keyword. The hashtags choosen in this dictionary are unduplicated and most likely ones to represent relevant emotion class. For instance, if a user use "#joy" in a tweet, the emotion most prossibly is happy. For each class, tweets are fetched using resprective hastages. This sheme aims for better data clean process later on. The dictionary is shown as below.



```
# # # # hashtag query # # # #
query_dic = {
"happy":     ["#happy", "#joy", "#love", "#like", "#good",
              "#glad", "#better", "#rejoice", "#innerpeace", "#blessed"],

"excitement":["#excitement", "#believing", "#ambition", "#exciting", "#music",
              "#predictable", "#delight", "#interesting", "#expected", "#hobby"],

"pleasant":  ["#pleasant", "#smile", "#pride", "#motivation", "#trusted",
              "#acceptance", "#beautiful", "#sweet", "#comfortable", "#welcome",
              "#admiration", "#admiring", "#loyal"],

"surprise":  ["#suprise", "#amazing", "#wonder", "#preoccupied", "#amazement",
              "#astonishment", "#great", "#lol", "#cool", "#funny", "#sad",
              "#frustration", "#sorrowful", "#sadness"],

"fear":      ["#fear", "#fearful", "#danger", "#apprehension", "#apprehensive",
              "#freakout", "#fearing", "#anxiety", "#panic", "#terror", "#bigot",
              "#sickened", "#hateful", "#dislike", "#foul"],

"anger":     ["#anger", "#agitated", "#roar", "#hormonal", "#fuckedoff",
              "#sogay", "#aggressive", "#furious", "#angry", "#annoyance",
              "#hostility", "#enraged", "#fustrated"],
}
```

Figure1: Hashtag query dictionary for twitter data fetch.

At first, this research chose all hashtags for one emotion using streaming API in order to speed up data fetch. Unfortunately, it is still very slow. For "anger" class, in 10 hours, only 70 tweets were generated. Hence, this research changed to rest API to fetch data using randomly selected hashtags for each emotion class. The code is shown as below:

```
def rest_data_fetch(class_name, db_collection):
        starttime = datetime.datetime.now()
        Rest_api = TwitterClient()
        for i in range(0, 4):
                query = random.sample(query_dic[class_name], 1)
                Rest_api.cursor_data(query, db_collection, 250)
# main crawl code is shown in the above 2.1.1 Data crawl code section
        endtime = datetime.datetime.now()
        duration = (endtime - starttime).seconds
        print("-------------------------------")
        print("End of the fetching, time duration: " + str(duration))
```

As shown in the code, using the query dictionnary, this research randomly selects hashtag keywords in different progress chunks of a single data fetch process, expecting to contain as much hashtag keywords as possible. For large sample amount, more hashtag keywords will be included. For each fetch process, the data amount is record by Mongodb compass and time is record as well. The actual data amount and associated time period for each class is record as below.

| Emotion | Anger | Happy | Excitement | Pleasant | Fear | Surprise | Total |
|---|---|---|---|---|---|---|---|
| Amount | 1009 | 1695 | 1000 | 2340 | 2006 | 1000 | 9050 |
| Time(s) | 57 | 57 | 45 | 62 | 62 | 25 | 308 |

Table2:Total and individual class distribution along with time period in which data collected.

## 2.3 Processing of Tweets

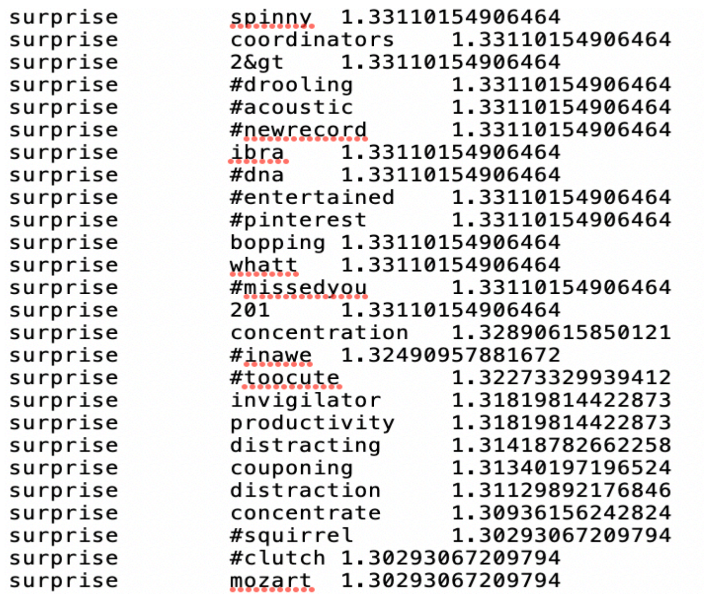
Inevitably lot of tweets will be ambiguous, overlap between classes. To deal with ambiguous, this research performs Duplicate process, Hashtag process, Emo-process, text and duplicate process in proper order.

### 2.3.1 Duplicate process

This research found, after data fetch, some tweet full texts are exactly the same. Hence in the data process produce, this research first filtered out the tweets with same full text using a text buffer. Each tweet text had been checked whether in the buffer or not before being appended to the buffer. Then final reasonable unduplicated tweets in this buffer had been inserted to a new database. Moreover, another duplicate process will be performed after text process.

### 2.3.2 Hashtag process

For hashtag process, this research use NRC lexicon (Saif and Peter, 2013), to loop through all hashtags in a tweet text. "Joy" class in NRC is used for identifying "happy" tweets, "anticipation" in NRC is used for identifying "excitement" tweets. The "sadness" and "surprise" classes are combined to identify "surprise" tweets in this research. In this process program, this research offers each tweet a blank score form for six emotion. For each tweet, this research extracts all hashtags from it. If the hashtag exists in NRC lexicon, the associated score will be added to relevant emotion. If the final maximum score is in the current emotion class of a tweet, the tweet can be kept in that class, otherwise it will be filter out.



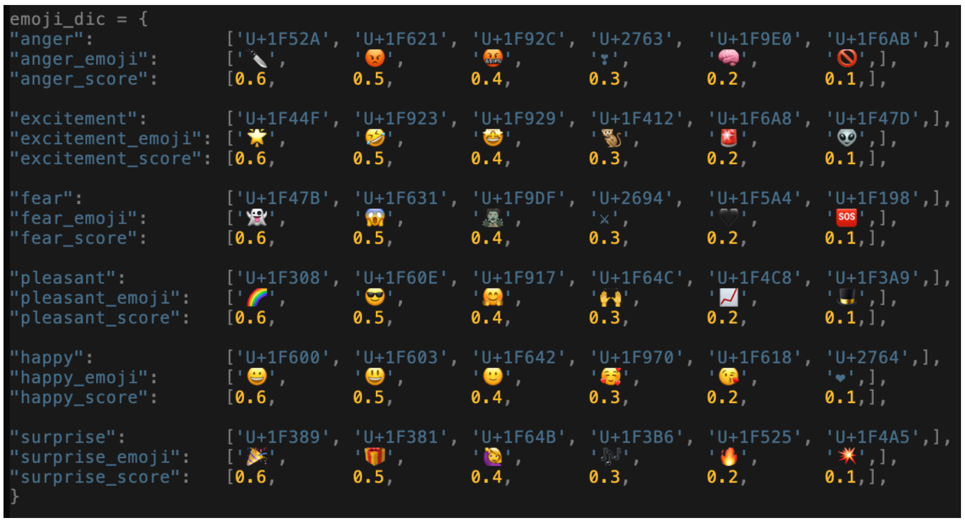
```
surprise        spinny  1.33110154906464
surprise        coordinators    1.33110154906464
surprise        2&gt    1.33110154906464
surprise        #drooling       1.33110154906464
surprise        #acoustic       1.33110154906464
surprise        #newrecord      1.33110154906464
surprise        ibra    1.33110154906464
surprise        #dna    1.33110154906464
surprise        #entertained    1.33110154906464
surprise        #pinterest      1.33110154906464
surprise        bopping 1.33110154906464
surprise        whatt   1.33110154906464
surprise        #missedyou      1.33110154906464
surprise        201     1.33110154906464
surprise        concentration   1.32890615850121
surprise        #inawe  1.32490957881672
surprise        #toocute        1.32273329939412
surprise        invigilator     1.31819814422873
surprise        productivity    1.31819814422873
surprise        distracting     1.31418782662258
surprise        couponing       1.31340197196524
surprise        distraction     1.31129892176846
surprise        concentrate     1.30936156242824
surprise        #squirrel       1.30293067209794
surprise        #clutch 1.30293067209794
surprise        mozart  1.30293067209794
```

Figure2: Data chunk in NRC lexicon with emotion, hashtag, keyword and associated score.

### 2.3.3 Emo-process

For emo-process, this research found the raw data seldom has emoticons but emojis, which reflects, instead of hand typing punctuations, people usually use emojis as an easier way to describe their feeling. Hence, to preform emo-process, this research generates a dictionary containing 36 most commonly used emojis representing six emotion classes, and weights them from score 0.6 to 0.1 according to degree level. For instance, a smile emoji might express happy ang pleasant, however, a laugh emoji is more probably express happy than pleasant. So, to treat this degree level, this research weight laugh emoji as 0.5 and smile as 0.3 in the happy class. To avoid ambiguous, this research selects emojis which are as much mutually exclusive as possible, no duplicates among six classes.

To perform emo-process, for each tweet, this research scores six emotion classes by looping through all emojis in a tweet text. If the maximum score is above zero and is not the current class of a tweet, the tweet will be filter out, otherwise it will be kept in its own class. The emo-dictionary for emo-process is shown as below:



```
emoji_dic = {
"anger":            ['U+1F52A', 'U+1F621', 'U+1F92C', 'U+2763',  'U+1F9E0', 'U+1F6AB',],
"anger_emoji":      ['🔪',       '😡',       '🤬',       '❣',       '🧠',       '🚫',],
"anger_score":      [0.6,       0.5,       0.4,       0.3,       0.2,       0.1,],

"excitement":       ['U+1F44F', 'U+1F923', 'U+1F929', 'U+1F412', 'U+1F6A8', 'U+1F47D',],
"excitement_emoji": ['🌟',       '🤣',       '🤩',       '🐒',       '🚨',       '👽',],
"excitement_score": [0.6,       0.5,       0.4,       0.3,       0.2,       0.1,],

"fear":             ['U+1F47B', 'U+1F631', 'U+1F9DF', 'U+2694',  'U+1F5A4', 'U+1F198',],
"fear_emoji":       ['👻',       '😱',       '🧟',       '⚔',       '🖤',       '🆘',],
"fear_score":       [0.6,       0.5,       0.4,       0.3,       0.2,       0.1,],

"pleasant":         ['U+1F308', 'U+1F60E', 'U+1F917', 'U+1F64C', 'U+1F4C8', 'U+1F3A9',],
"pleasant_emoji":   ['🌈',       '😎',       '🤗',       '🙌',       '📈',       '🎩',],
"pleasant_score":   [0.6,       0.5,       0.4,       0.3,       0.2,       0.1,],

"happy":            ['U+1F600', 'U+1F603', 'U+1F642', 'U+1F970', 'U+1F618', 'U+2764',],
"happy_emoji":      ['😀',       '😃',       '🙂',       '🥰',       '😘',       '❤',],
"happy_score":      [0.6,       0.5,       0.4,       0.3,       0.2,       0.1,],

"surprise":         ['U+1F389', 'U+1F381', 'U+1F64B', 'U+1F3B6', 'U+1F525', 'U+1F4A5',],
"surprise_emoji":   ['🎉',       '🎁',       '🙋',       '🎶',       '🔥',       '💥',],
"surprise_score":   [0.6,       0.5,       0.4,       0.3,       0.2,       0.1,],
}
```

Figure3: Emo-dictionary used in emo-process with emotion and associated degree level.

### 2.3.4 Text process and final duplicate process

This research finds some noise patterns among tweet texts, such as punctuations, mentions, retweet sign, URLs, last few hashtags, incorrect spelling like "looooove", and contractions. All noise pattern has been removed in text process mainly use re and nltk python library. Particularly, this research loop through each word in a text to find the char which appears more than three times continuously and shorten them to just 1 char as a method to correct spelling, like "looooove" to "love". This research also uses re and contractions python library to identify contractions and replace them to correct one like "don't" to "do not". Most punctuations has been removed while some punctuations like "!,.:?;", which might express thoughts, are kept in the text.

For final duplicate process, as mentioned before, after data fetch, data has been cleaned by duplicate process immediately. However, these only filter out tweet with exact same text. This research aims at filtering out tweets with the same core text, as some tweets has same core text but different URLs which cannot be filter out on the first process. Luckily, after text process, this research has gained the purely core tweet text. In the program, this research creates a text buffer once more. Final tweet texts had been appended to the buffer before checking whether the text is already in the buffer or not. This process is regarded as final duplicated process and successfully filter out tweets which have the same pure core text.

### 2.3.5 Over all flow:

The raw data amount and data amount with each data process produce is shown as below:

| Class | Raw amount | Duplicate process amount | Hashtag process amount | emo_process amount | text and duplicate process amount | Final extract amount |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| Anger | 1009 | 624 | 263 | 261 | 237 | 150 |
| Excitement | 1000 | 563 | 256 | 241 | 219 | 150 |
| Fear | 2006 | 1068 | 506 | 490 | 461 | 150 |
| Happy | 1695 | 798 | 288 | 279 | 259 | 150 |
| Pleasant | 2340 | 1365 | 311 | 269 | 237 | 150 |
| Surprise | 1000 | 786 | 277 | 257 | 229 | 150 |
| Total | 9050 | 5204 | 1901 | 1797 | 1642 | 900 |

Table3: The effect of whole data process.

The overall flow and combination of these data processing methods are shown as below:

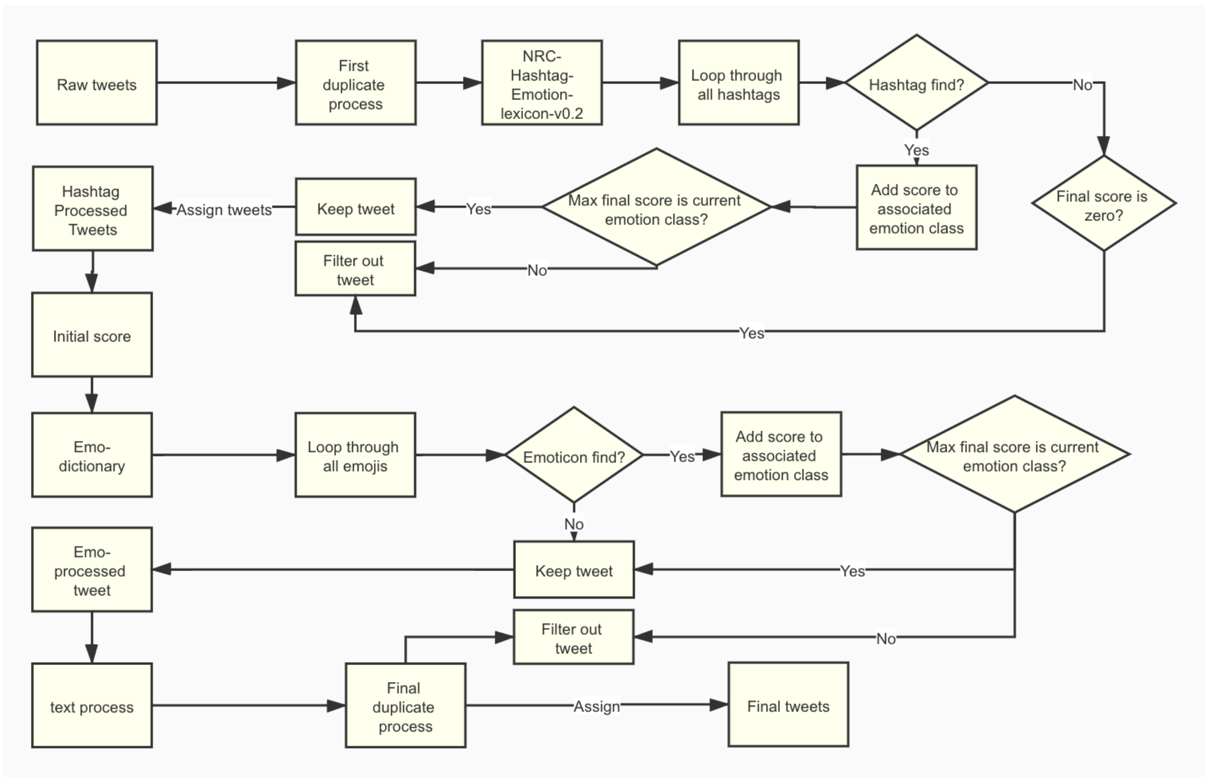


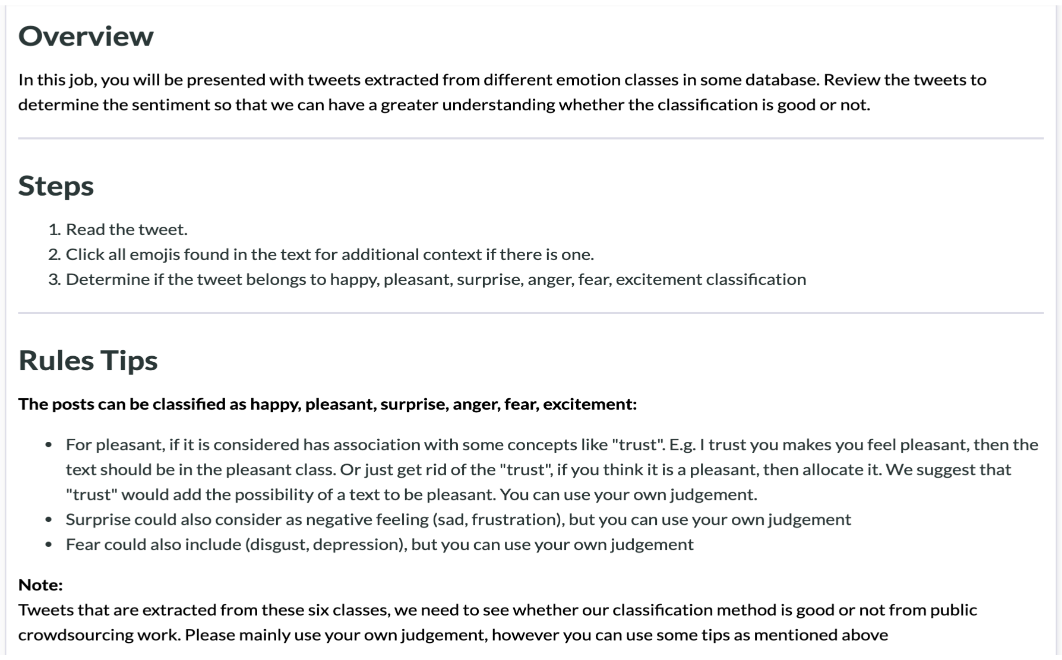
Figure4: Diagram for the combination of data processing strategies.

This research preforms Duplicate process, Hashtag process, Emo-process, text and duplicate process in proper order. The results demonstrate that with 9050 raw data, after data process, only 1642 data had been remained. Final 900 reasonably clean data have assigned to corresponding emotion class.

# 3.0 Crowdsourcing

## 3.1 Crowdsourcing details

This research randomly selected 20 data from each final emotion class, as a total 120 data samples for crowdsourcing. Data is uploaded to Figure eight interface platform. Five judgements are required for each data row, and Contributors can get 2¢ for each judgement. The job design is shown as below.



### Overview

In this job, you will be presented with tweets extracted from different emotion classes in some database. Review the tweets to determine the sentiment so that we can have a greater understanding whether the classification is good or not.

### Steps

1. Read the tweet.
2. Click all emojis found in the text for additional context if there is one.
3. Determine if the tweet belongs to happy, pleasant, surprise, anger, fear, excitement classification

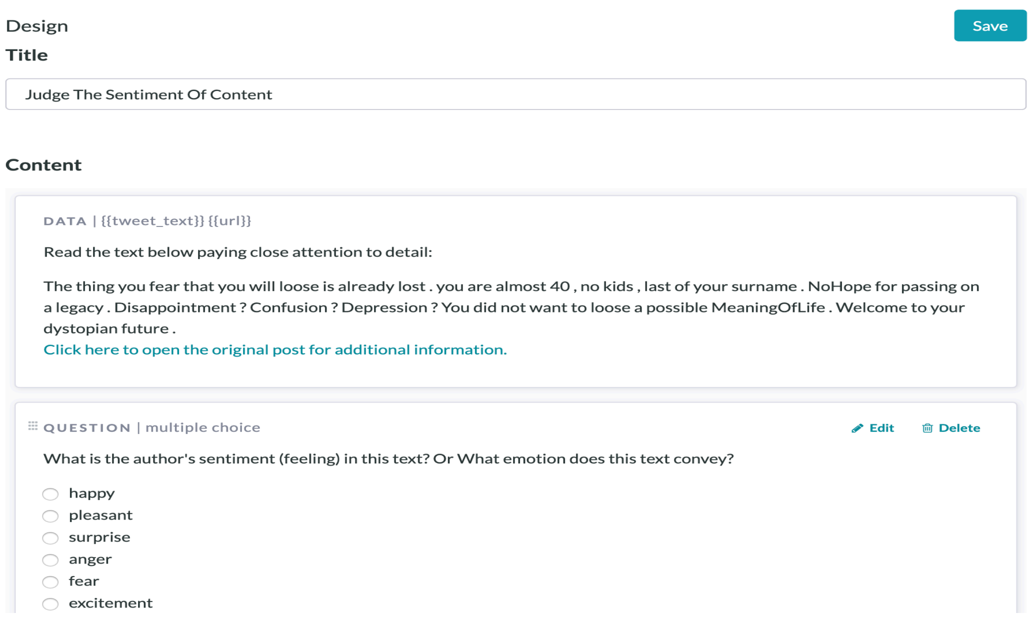
### Rules Tips

**The posts can be classified as happy, pleasant, surprise, anger, fear, excitement:**

- For pleasant, if it is considered has association with some concepts like "trust". E.g. I trust you makes you feel pleasant, then the text should be in the pleasant class. Or just get rid of the "trust", if you think it is a pleasant, then allocate it. We suggest that "trust" would add the possibility of a text to be pleasant. You can use your own judgement.
- Surprise could also consider as negative feeling (sad, frustration), but you can use your own judgement
- Fear could also include (disgust, depression), but you can use your own judgement

**Note:**
Tweets that are extracted from these six classes, we need to see whether our classification method is good or not from public crowdsourcing work. Please mainly use your own judgement, however you can use some tips as mentioned above

Figure5: Figure eight crowdsourcing job instruction.



Design
Title
Judge The Sentiment Of Content

Content

DATA | {{tweet_text}} {{url}}

Read the text below paying close attention to detail:

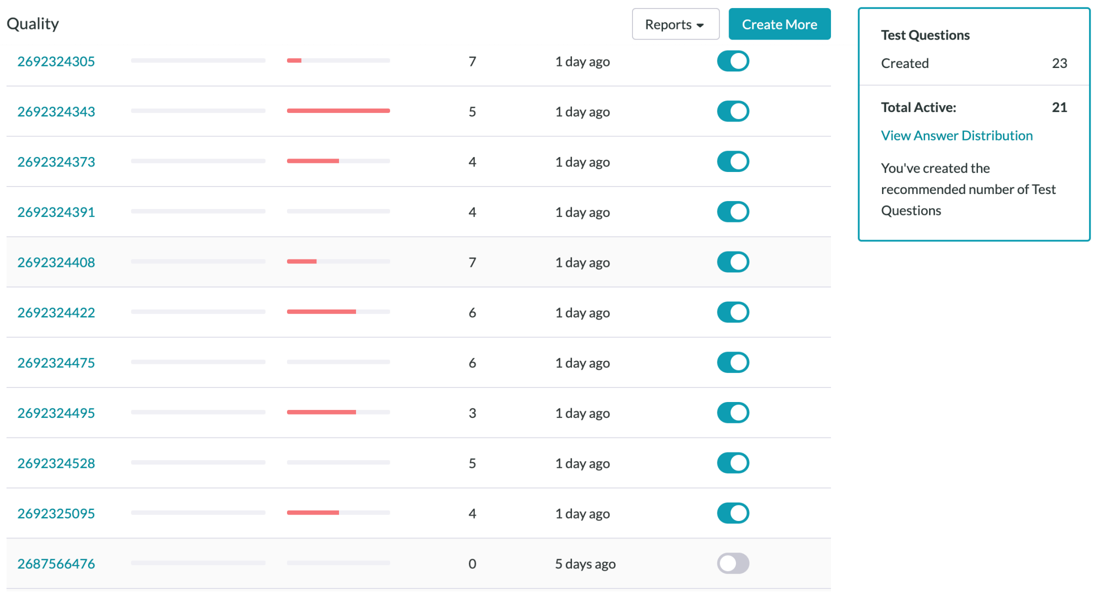
The thing you fear that you will loose is already lost . you are almost 40 , no kids , last of your surname . NoHope for passing on a legacy . Disappointment ? Confusion ? Depression ? You did not want to loose a possible MeaningOfLife . Welcome to your dystopian future .
Click here to open the original post for additional information.

QUESTION | multiple choice

What is the author's sentiment (feeling) in this text? Or What emotion does this text convey?

- happy
- pleasant
- surprise
- anger
- fear
- excitement

Figure6: Figure eight crowdsourcing job content and question.


Quality

| | | |
|---|---|---|
| 2692324305 | 7 | 1 day ago |
| 2692324343 | 5 | 1 day ago |
| 2692324373 | 4 | 1 day ago |
| 2692324391 | 4 | 1 day ago |
| 2692324408 | 7 | 1 day ago |
| 2692324422 | 6 | 1 day ago |
| 2692324475 | 6 | 1 day ago |
| 2692324495 | 3 | 1 day ago |
| 2692324528 | 5 | 1 day ago |
| 2692325095 | 4 | 1 day ago |
| 2687566476 | 0 | 5 days ago |

Test Questions

Created 23

Total Active: 21

View Answer Distribution

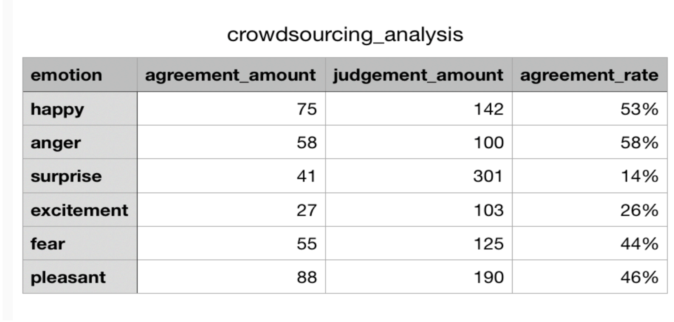
You've created the recommended number of Test Questions

Figure7: Figure eight crowdsourcing job test questions

In order to verify whether the final emotion classification is clean enough or not, this job asks contributors to read the tweet content and answer the question "what emotion does the text convey?" as shown in figure3. Contributors need to select an emotion class to answer the question. If they don't quite understand this job, they can read the instruction in figure4 first. To grantee the job quality, this job creates 21 test questions as a way to eliminating contributors who repeatedly make mistakes during the task, contributors need to pass them to work on later. The final results are collected as a dataset to compare with our own emotion classification

## 3.2 Results and discussion

After crowdsourcing, this research calculates the amount of total judgements and the amount of agreements for each emotion class as shown in the csv tabular file below. To verify our own justification, agreement rate is calculated as well, which reflect the percentage of agreements. It is found the agreement rate of "surprise" class is lowest with 14% and "anger" class is highest with 58%. Four classes have agreement rate below 50%, only "happy" and "anger", which represents two polarity emotions and antithesis, have agreement rate above 50%.


crowdsourcing_analysis

| emotion | agreement_amount | judgement_amount | agreement_rate |
|---|---|---|---|
| happy | 75 | 142 | 53% |
| anger | 58 | 100 | 58% |
| surprise | 41 | 301 | 14% |
| excitement | 27 | 103 | 26% |
| fear | 55 | 125 | 44% |
| pleasant | 88 | 190 | 46% |

Figure8: Crowdsourcing analysis with agreements, total judgements and agreement rate.

The general result shows low agreement rate after crowdsourcing. These results reflect several limitations which cannot be ignored. For the crowdsourcing side, contributors might be tried when facing 120 rows data, this might result in judgement deviation. For the data process side, first, in the hashtag process, the NRC lexicon is only used to identify hashtags but not the other words. One tweet might contain the core emotion in the pure full text. All words in a tweet need to be identified using NRC lexicon. Besides, the NRC lexicon has many duplicate hashtags through different emotion classes, which overlaps when doing hashtag scoring for each emotion classes. Second, the emo-dictionary generated in this research is not large enough, more emojis and emoticons needs to be taken into account. It is also found in the final results, very few core tweet texts are advertisements, which need to be identified by more enhanced algorithm and filtered out in text process.

## 4.0 Conclusion

To conclude, this research fetched 9050 tweets to perform emotion labelling. After the data process as Duplicate process, Hashtag process, Emo-process, Text process and Final duplicate process, a total amount of 1642 tweets remain, and 900 tweets were extracted as reasonably clean dataset with 150 tweets per emotion class. However, the crowdsourcing shows unsatisfied agreement rate results, which reflects several processing limitations such as identifying all words using NRC, overlap in NRC, inadequate emoji/emoticon dictionary. Overall, this research provides a data processing method for emotion labeling, the tweets are "reasonably" clean, more enhanced data processing could be discussed in the future.

## Reference

1. Roger A. Grimes and Josh Fruhlinger. (2019). What is OAuth? How the open authorization framework works. https://www.csoonline.com/article/3216404/what-is-oauth-how-the-open-authorization-framework-works.html

2.NRC lexicon. https://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm

3.Svitlana Galeshchuk. (2019, 09). Working with Twitter Data in Python. https://medium.com/analytics-vidhya/working-with-twitter-data-b0aa5419532