Web Science Course Work: Social Media Emotion Data Set

Yuyang Zhang (2414208z)

Source Code:

`https://github.com/helloqiu/WebScienceCourseWork`

Sample Data Link:

`https://github.com/helloqiu/WebScienceCourseWork/releases/download/1.0/mongo_sample.tar.gz`

College of Science and Engineering
University of Glasgow

March 2020

Contents

1 Introduction **2**

1.1 External Packages 2

1.2 Data Collection 3

2 Data Craw and Rules **4**

2.1 Tweets Fetching 4

2.2 Pre-processing 4

2.3 Categorization 6

3 Crowdsourcing **8**

3.1 Data 8

3.2 Question 8

3.3 Quality 9

3.4 Accuracy Result 9

4 Conclusion **10**

Bibliography **11**

1 Introduction

In this course work, the task is to build an emotion annotated data set from Twitter. This report contains three sections. In the first section, packages used in the codes and the time of data collection will be listed. Then, the methods of crawling tweets, content pre-processing and categorizing will be discussed. Finally, there will be some analysis based on crowdsourcing results.

1.1 External Packages

```
backcall==0.1.0
beautifulsoup4==4.8.2
certifi==2019.11.28
chardet==3.0.4
cycler==0.10.0
decorator==4.4.1
emot==2.1
future==0.18.2
idna==2.8
inexactsearch==1.0.2
ipython==7.12.0
ipython-genutils==0.2.0
jedi==0.16.0
kiwisolver==1.1.0
matplotlib==3.2.0
nltk==3.4.5
numpy==1.18.1
oauthlib==3.1.0
pandas==1.0.1
parso==0.6.1
pexpect==4.8.0
pickleshare==0.7.5
progressbar2==3.47.0
prompt-toolkit==3.0.3
ptyprocess==0.6.0
Pygments==2.5.2
pymongo==3.10.1
pyparsing==2.4.6
pyspellchecker==0.5.4
```

```
python-dateutil==2.8.1
python-twitter==3.5
python-utils==2.3.0
pytz==2019.3
requests==2.22.0
requests-oauthlib==1.3.0
silpa-common==0.3
six==1.14.0
soundex==1.1.3
soupsieve==2.0
spellchecker==0.4
traitlets==4.3.3
urllib3==1.25.8
wcwidth==0.1.8
yapf==0.29.0
```

Code 1: requirements.txt

All external packages with their version numbers that are used in this project are shown in Code 1.

1.2 Data Collection

All tweets used in this project were collected during the time period from 20:35:14 06/03/2020 to 21:04:43 06/03/2020. There are 60000 collected tweets in total.

2 Data Craw and Rules

2.1 Tweets Fetching

In this coursework, data were collected by using the streaming API provided by Twitter [1]. In order to fetch the data more easily, the *twitter* package was used in the python codes.

```
api = twitter.Api(consumer_key=config['consumer_key'],
            consumer_secret=config['consumer_secret'],
            access_token_key=config['access_token_key'],
            access_token_secret=config['access_token_secret'],
            sleep_on_rate_limit=True)

stream = api.GetStreamFilter(languages=['en'],
           locations=UK_BOUNDS,
           track=keywords)
```

Code 2: Fetch Tweets

As shown in Code 2, there are three parameters assigning to the streaming API: *languages*, *track* and *locations*. As required, languages are set to English only, and the locations of the tweets are limited within the UK. To get more tweets related to the six emotion classes, keywords are set to the related words according to the NRC lexicon. After connecting to the streaming API, an HTTP connection will be established and Twitter's server will keep pushing matched tweets to the crawler client. The crawler then will store all collected tweets to the MongoDB as raw tweets.

2.2 Pre-processing

In order to analyse tweets, a stage of pre-processing has been added to the whole process. The flow chart of this procedure is shown in Figure 1.

```
URL = re.compile(
    r'(https?:\/\/(?:www\.|(?!www))[a-zA-Z0-9][a-zA-Z0-9-]+
    [a-zA-Z0-9]\.[^\s]{2,}|www\.[a-zA-Z0-9][a-zA-Z0-9-]+
    [a-zA-Z0-9]\.[^\s]{2,}|https?:\/\/(?:www\.|(?!www)[a-zA-Z0-9]
    +\.[^\s]{2,}|www\.[a-zA-Z0-9]+\.[^\s]{2,})'
)
USERNAME = re.compile(r'@[a-zA-Z0-9_]{0,15}')
```

Code 3: Regular Expression

In this process, first, the tweet is filtered by checking if it has "text" property and if there is a tweet with the same ID or text stored in the database. If the the tweet is a retweet, the text or extended text of the original tweet will be used, and if it is a extended tweet, the extended text will be used. Then, urls and usernames such as "@someone" are removed by using regular expressions which are shown in Code 3, and emoticons are translated to emotions which are added to the tweet object as an extra property. After that, the text content is tokenized by using the *nltk* package, and then the slangs such as "looove" are converted to "love". Finally, the tweet is stored into a new collection for further processing.

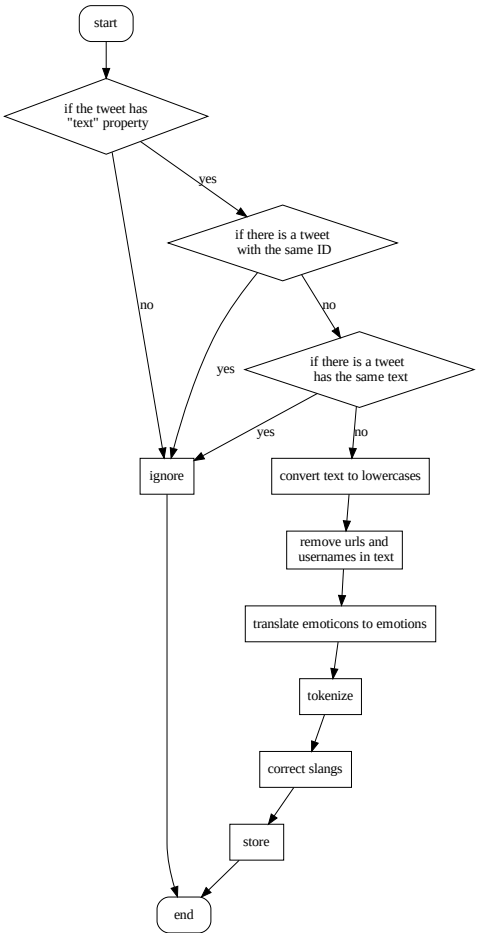

Figure 1: Flow Chart of Pre-processing

2.3 Categorization

After the pre-processing, the tweet can be categorized into classes according to the text, emoticons and hashtags.

	text	emoticon	hashtag
mark	0.5	3.0	3.0

Table 1: Marks

This process is done by calculating the weight for each emotion. The marks of these three properties are shown in Table 1.

The weight from text is calculated according to NRC lexicon. Every word of the text will be searched for in it. There are 10 emotions in the lexicon. So, these emotions are mapped to the six classes.

```
TAGS = {
    'anger': 'angry',
    'disgust': 'fear',
    'fear': 'fear',
    'joy': 'happy',
    'sadness': 'surprise',
    'surprise': 'surprise',
    'negative': 'surprise',
    'positive': 'pleasant',
    'trust': 'pleasant',
    'anticipation': 'excitement',
}
```

Code 4: NRC Lexicon Map

The weight from emoticon is calculated by iterating through the "emoticon" property added by pre-processing. It can be directly added to the six emotions.

The weight from hashtag is calculated in the same way of text. The emotions of the hashtags are from the NRC lexicon.

```
    ...
    {
       "emotions_weight":{
         "excitement":0.5,
         "happy":1.0,
         "pleasant":4.5,
```

```
      "surprise":0,
      "fear":0,
      "angry":0
    },
    "emotion":"pleasant",
    "max_emotion_weight":4.5
}
...
```

Code 5: Tweet Example

After the categorization process, an example of the output is shown in Code 5.

3 Crowdsourcing

To analyse the accuracy of the categorization, the crowdsourcing method has been used.

3.1 Data

For each emotion class, 20 tweets are randomly extracted as a sample. The contributors are asked to judge the emotion of the provided content.

The crowdsourcing process is performed on *Figure Eight* platform. A sample of the input data is shown in Figure 2.

@rinkara I used to RP with people in their 20's when I was a teen and I never ran into anything weird. The people who turned out to be creepy and abusive were younger then me kids should use caution sure, but not every relationship is inherently predatory	surprise	1236027716870303746
#UASpringFest is coming and we are SUPER excited to tell you about it! Stay tuned to our social media for more news coming soon. https://t.co/FKMvkBHrLQ	excitement	1236027716920651777

Figure 2: Data Sample

The text extracted from the tweets are the only contents that are provided to the contributors.

3.2 Question

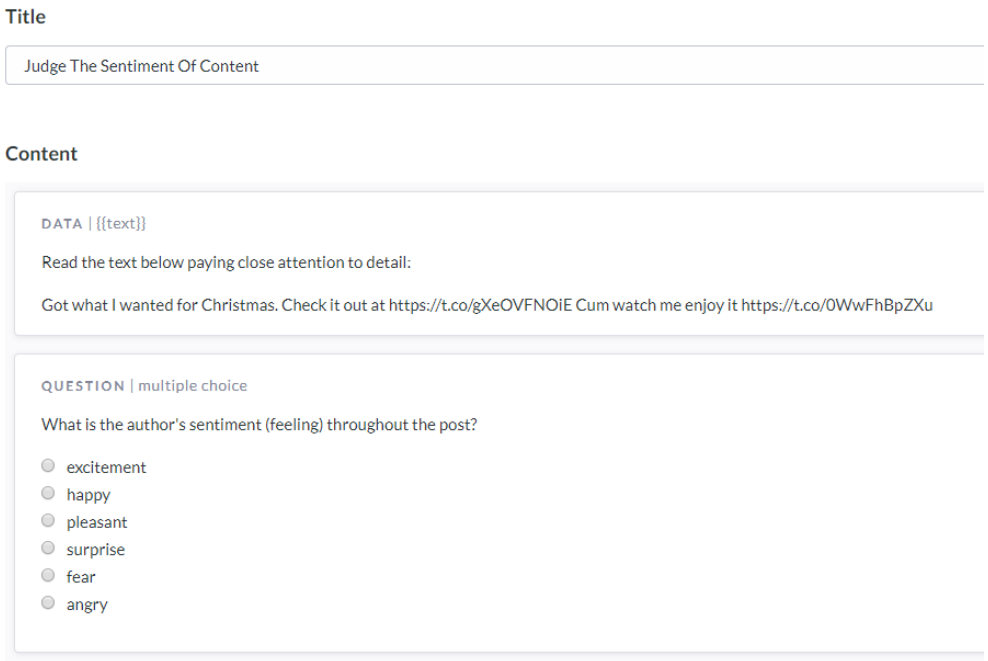

Figure 3: Crowdsourcing Question Design

The question design for the crowdsourcing is shown in figure 3. We provide the text to the contributors and ask them about what the author's sentiment of this post is.

3.3 Quality

In order to guarantee the quality of the judgement, some test questions has been created manully in the process of crowdsourcing. The missed percentage and number of judgment are shown in Table 2.

missed percentage	number of judgments	answers	text
0.01	100	excitement happy pleasant	@whitewinery St. Basil's is beautiful. Orthodox have some beautiful cathedrals - like this: https://t.co/HAa2emZhCc
0.22	94	angry	@eruditechalamet Hot damn
0.22	87	happy	@OuamroucheH @chancetotravel @MuchMorocco @Visit_Morocco_ Thankyou, we love ? Morocco ?? too. I'm glad to hear it ?
0.23	79	angry	love clingy but if im angry or its hot out? you need to back the fuck up
0.42	12	angry	Fuck what y'all Talkin bout uzi album hot as shit ????
0.29	17	happy pleasant	@KaraboTebedi Your skin looks like those beautiful stones that you can find that have these shapes and colors which make them beautiful
0.71	17	surprise	#BREAKING:Why won't @USArmy CONFIRM OR DENY whether @senatemajldr,one of the most powerful politicians who's currently up for reelection,was EVER the subject of a court martial? #TheResistance #MitchMcConnell #CNN #MSNBC #Yahoo #mondaythoughts #FBRParty https://t.co/dkpzf4x0Pz
0.5	12	pleasant angry	@jendoesthething Minori, you are NOT a delinquent. Takeru is just, you know, a type. Kinda tsun, hot cosplayer, athletic.
0.13	15	surprise fear angry	Should we hold our breath waiting for you to condemn Senator Schumer's threats against 2 Supreme Court justices? https://t.co/y55ztSFjSb
0.31	13	fear angry	WHY is she threatening a god?
0.29	14	fear	'What's the worst that could happen' oh no, I already know something bad is gonna happen

Table 2: Test Questions

3.4 Accuracy Result

From the result of the crowdsourcing, we can roughly calculate the accuracy of the classfication.

As shown in Figure 4, the class with the highest accuracy is happy, and the class with the lowest accuracy is pleasant. The overall accuracy is around 0.3. This may be caused by the fact that many words from the NRC lexicon have emotional overlaps, and the difference between happy, pleasant, surprise is vague.

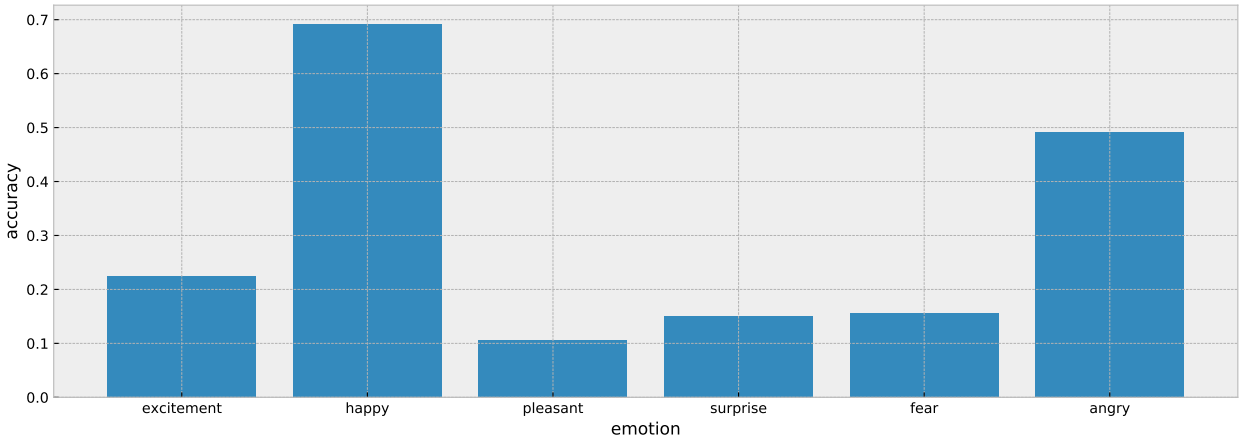

Figure 4: Accuracy

4 Conclusion

In this assignment, an emotion annotated data set from Twitter has been built. The strategies of classfication has been shown, and the crowdsourcing has been used to check the accuracy. In the result, we find that different classes have different accuracy, and this is probabily caused by emotional overlaps of the keywords.

References

[1] Twitter. Filter realtime Tweets. https://developer.twitter.com/en/docs/tweets/filter-realtime/api-reference/post-statuses-filter.