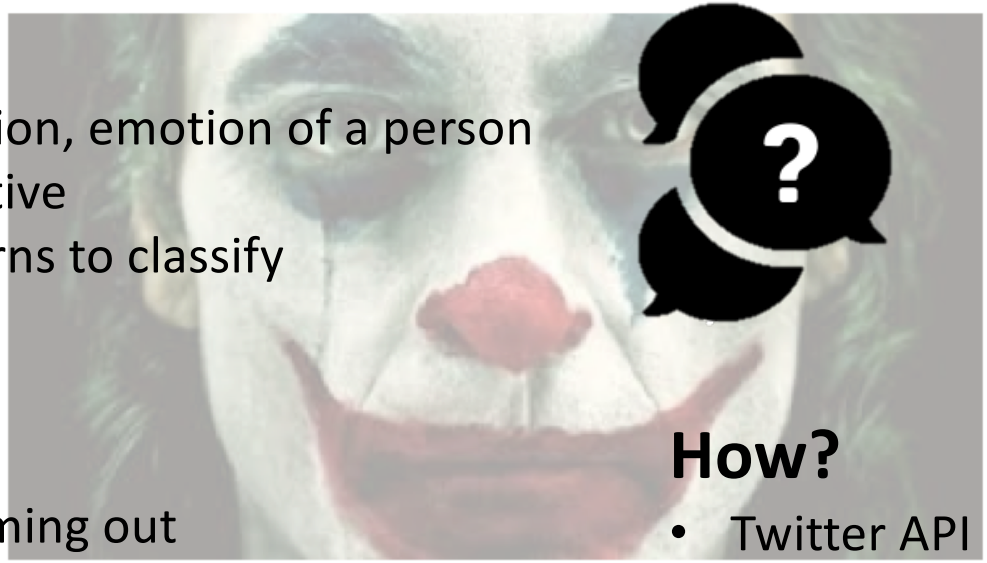
Sentiment Analysis of Tweets about the movie *Joker*

Capstone Project 2

Coline Zeballos

Proposal

« Sentiment analysis of Tweets about *Joker* »



What?

- Attitude, opinion, emotion of a person
- Positive/negative
- Computer learns to classify

Who?

- Fans
- Movie specialist
- Producers
- Psychologists
- Psychiatrists

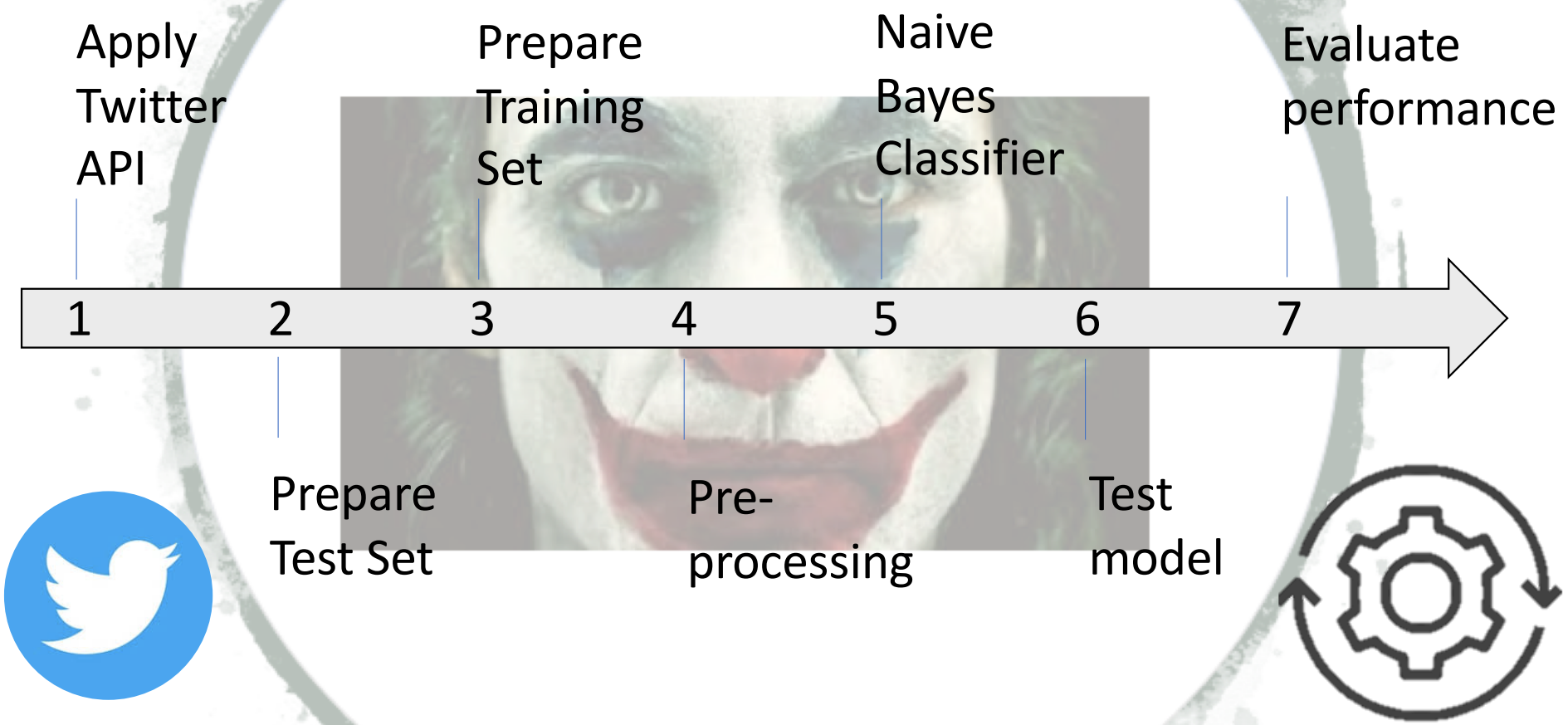
Why?

- Recent coming out
- Divided opinions
- Many tweets
- Interesting to get an opinion with real data

How?

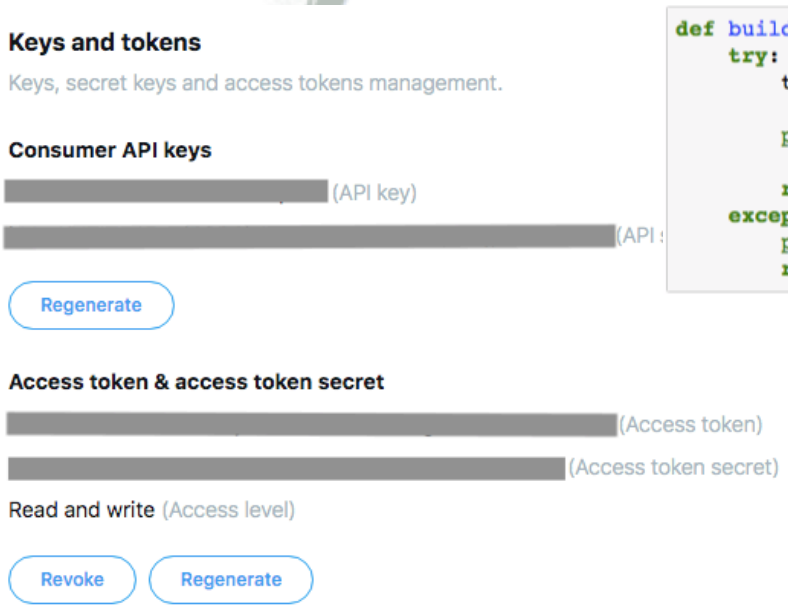
- Twitter API
- Keywords: « joker », « movie »
- Labels: « positive », « negative »

Methodology



Preparing the Test set

Twitter authentication credentials



Function to fetch 100 Tweets on *Joker*

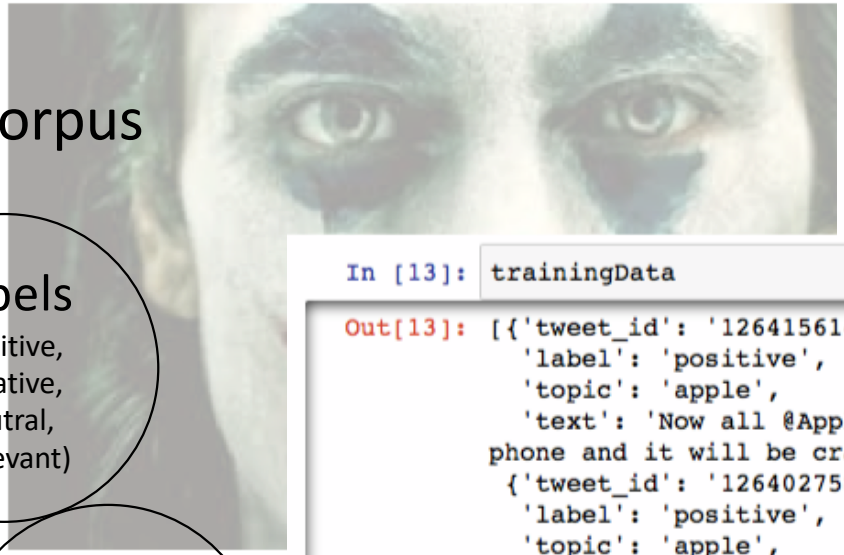
```
def buildTestSet(search_keyword):
    try:
        tweets_fetched = api.GetSearch(search_keyword, count = 100)

        print("Fetched " + str(len(tweets_fetched)) + " tweets for the term " + search_keyword)

        return [{"text":status.text, "label":None} for status in tweets_fetched]
    except:
        print("Unfortunately, something went wrong..")
        return None
```

```
print(testDataSet[0:4])
```

```
[{'text': 'Finally saw the joker.\n\nWhile the technical
aspects of the movie are stunning and the acting incredib
le.\n\nThat migh… https://t.co/QFIi10T0oH', 'label': None
}, {'text': '"Joker" became the first R-rated movie to hi
t $1 billion at the box-office, and there are\u200b rumor
s of a sequel https://t.co/k6U8yuxsNC', 'label': None}, {
'text': "Would you like to see a sequel to Joaquin Phoeni
x's #Joker? https://t.co/InEGd1EvME", 'label': None}, {'t
ext': 'RT @1woo17: They had done this before for other mo
vies such as:\nBTS Bring The Soul: The Movie\nWeathering
With You\nJoker\nSpiderman: Far From…', 'label': None}]
```

Preparing the Training Set

Critical to have a good training set!



Nick Sanders' Corpus

5512 Tweets

Labels (positive, negative, neutral, irrelevant)

Topics (apple, google, microsoft, twotter)

Tweet ID

```
tr_set.head()
```

	apple	positive	126415614616154112
0	apple	positive	126404574230740992
1	apple	positive	126402758403305474
2	apple	positive	126397179614068736
3	apple	positive	126395626979196928

```
In [13]: trainingData

Out[13]: [{'tweet_id': '126415614616154112',
  'label': 'positive',
  'topic': 'apple',
  'text': 'Now all @Apple has to do is get swype on the i
phone and it will be crack. Iphone that is'},
 {'tweet_id': '126402758403305474',
  'label': 'positive',
  'topic': 'apple',
  'text': "Hilarious @youtube video - guy does a duet wit
h @apple 's Siri. Pretty much sums up the love affair! ht
tp://t.co/8ExbnQjY"},
 {'tweet_id': '126397179614068736',
  'label': 'positive',
  'topic': 'apple',
  'text': '@RIM you made it too easy for me to switch to
@Apple iPhone. See ya!'},
```

Pre-processing Tweets

Words are most important
Remove items which don't bring value to sentiment analysis (punctuation, duplicates, URLs…)

Steps taken on both training and test set:

- Remove stop words
- Lowercase
- Remove URLs and usernames
- Remove hashtag sign
- Remove duplicate characters
- Break tweets' text into words (tokenize)

"@person1 retweeted @person2: Corn has got to be the most dellllicious crop in the world!!!! #corn #thoughts..."

⇩

"AT_USER rt AT_USER corn has got to be the most delicious crop in the world corn thoughts"

⇩

{"corn", "most", "delicious", "crop", "world", "corn", "thoughts"}

Naïve Bayes Classifier

Classification algorithm that relies on Baye's theorem

Widely used in NLP problems to predict the tag of text

Build vocabulary of words in training dataset
2700 words

```
word_features
```

```
dict_keys(['get', 'swype', 'iphone', 'crack', 'h
ums', 'love', 'affair', 'made', 'easy', 'switch'
lad', 'gave', 'sense', 'humor', 'via', 'great',
mpanies', 'experience', 'best', 'customer', 'ser
ng', 'haha', 'suggested', 'find', 'whores', 'hic
, 'actually', 'help', 'greatly', 'appreciated',
, 'offers', 'refurbished', '1st-gen', 'ipads',
```

Match tweet content against our vocabulary
→ Is word in vocabulary present in each tweet?
(True/False)

$$P(C|X) = \frac{P(X|C)\,P(C)}{P(X)}$$

Train classifier

```
NBayesClassifier = nltk.NaiveBayesClassifier.train(trainingFeatures)
```

```
Out[40]: [({'contains(get)': True, 'contains(swype)': True, 'conta
ins(iphone)': True, 'contains(crack)': True, 'contains(hi
larious)': False, 'contains(video)': False, 'contains(guy
)': False, 'contains(duet)': False, "contains('s)": False
, 'contains(siri)': False, 'contains(pretty)': False, 'co
ntains(much)': False, 'contains(sums)': False, 'contains(
love)': False, 'contains(affair)': False, 'contains(made)
': False, 'contains(easy)': False, 'contains(switch)': Fa
lse, 'contains(see)': False, 'contains(ya)': False, 'cont
ains(16)': False, 'contains(strangest)': False, 'contains
(things)': False, 'contains(said)': False, 'contains(far)
': False, 'contains(sooo)': False, 'contains(glad)': Fals
e, 'contains(gave)': False, 'contains(sense)': False, 'co
ntains(humor)': False, 'contains(via)': False, 'contains(
great)': False, 'contains(close)': False, 'contains(perso
nal)': False, 'contains(event)': False, 'contains(tonight
)': False, 'contains(regent)': False, 'contains(st)': Fal
se, 'contains(store)': False, 'contains(companies)': Fals
e, 'contains(experience)': False, 'contains(best)': False
```

Testing the model

01 Run classifier on Test set (100 Tweets), get label (pos/neg) and calculate overall sentiment percentage % of Tweets

02 Results

```
Overall Negative Sentiment
Negative Sentiment Percentage = 52.0%
```

03 Interpretation

- Overall sentiment is negative at 52%
- Coincides with critics one can read in the newspaper and on the internet
- Balanced between positive and negative

Let's have a look at a couple of Tweets and the assigned label to check if the classification is intuitive or not.

Analyzing the performance of the model

```
# Check first 10 Tweets words and their attributed labels
test_labels[0]
```

```
array([list(['finally', 'saw', 'joker', 'technical', 'asp
ects', 'movie', 'stunning', 'acting', 'incredible', 'migh
```

```
test_labels[1]
```

```
array([list(['``', 'joker', "'`'", 'became', 'first', 'r-r
ated', 'movie', 'hit', '1', 'billion', 'box-office', 'are
\u200b', 'rumors', 'sequel']),
```

```
test_labels[2]
```

```
array([list(['would', 'like', 'see', 'sequel', 'joaquin',
'phoenix', "'s", 'joker']),
```

```
test_labels[3]
```

```
array([list(['rt', 'done', 'movies', 'bts', 'bring', 'sou
l', 'movie', 'weathering', 'joker', 'spiderman', 'far', '
```

```
test_labels[4]
```

```
array([list(['ashishchanchalani', 'ashishchanchalanimemes
', 'oldvideos', 'joker', 'think', 'downloaded', 'wrong',
```

```
test_labels[5]
```

```
array([list(['think', 'joker', 'new', 'favorite', 'charac
ter/movie', 'love', 'every', 'actor', '’', 'played', 'get
```

```
test_labels[6]
```

```
array([list(['rt', 'still', 'thinking', 'sana', 'harley',
'quinn', 'wanted', 'chaeyoung', 'joker', 'coz', 'chaeyoun
```

```
test_labels[7]
```

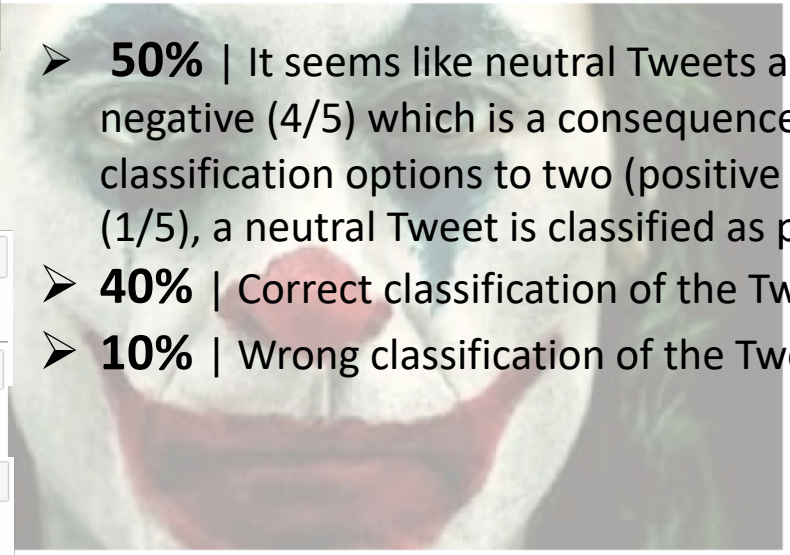
```
array([list(['joker', 'action', 'movie']), None, 'positiv
```

```
test_labels[8]
```

```
array([list(['saw', 'ford', 'v', 'ferrari', 'last', 'week
end', 'great', 'movie', 'would', 'top', 'list', '2019', "
```

```
test_labels[9]
```

```
array([list(['’', 'dope', 'joker', 'movie']), None, 'posi
tive'],
      dtype=object)
```



- **50%** | It seems like neutral Tweets are more often classified as negative (4/5) which is a consequence of simplifying the classification options to two (positive and negative). More rarely (1/5), a neutral Tweet is classified as positive
- **40%** | Correct classification of the Tweet
- **10%** | Wrong classification of the Tweet

This (subjective) performance evaluation task is telling us that the classification model is not optimal yet.

Conclusion

Possible improvements

- Trying different combinations of pre-processing steps
- Adding/removing some pre-processing steps
- Checking the impact of each pre-processing step on model performance
- Building our own Training Dataset, to control quality of the classification in the
- Training Dataset, as it is crucial to a good model performance.