In [1]:
```
import os
import numpy as np
import pandas as pd
from tqdm import tqdm
```

Work with data

Read data

Emodji and labels look like:

Read train data

In [2]:
```
with open('./data/train/english_train.text', 'r') as f:
    texts = [l.strip() for l in f]
```

In [3]:
```
with open('./data/train/english_train.labels', 'r') as f:
    labels = [int(l.strip()) for l in f]
```

In [4]:
```
# with open('./data/data_us/tweet_by_ID_20_11_2017__03_39_34.txt.ids',
 'r') as f:
#     ids = [l.strip() for l in f]
```

Read trial data (only texts)

In [5]:
```
with open('./data/test/english_test.text', 'r') as f:
    texts_trial = [l.strip() for l in f]
```

Read mapping for emodji and labels

In [6]:
```
with open('./data/mapping/english_mapping.txt', 'r') as f:
    maps = [l.strip().split() for l in f]
```

In [7]:
```
emodji = [maps[l][1] for l in labels]
```

Look at the data

In [8]:
```
texts[:5]
```

Out[8]:
```
['A little throwback with my favourite person @ Water Wall',
 'glam on @user yesterday for #kcon makeup using @user in #featherette,
…',
 'Democracy Plaza in the wake of a stunning outcome #Decision2016 @ NBC
News',
 'Then &amp;amp; Now. VILO @ Walt Disney Magic Kingdom',
 'Who never... @ A Galaxy Far Far Away']
```

In [9]:
```
labels[:5]
```

Out[9]:
```
[0, 7, 11, 0, 2]
```

In [10]:
```
emodji[:5]
```

Out[10]:
```
['❤', '✨ ', '🇺🇸 ', '❤', '😂 ']
```

Preprocessing data

We clean our data from:

- URLS
- Punctiation
- Symbols '#' and '@'
- Stop-words

We also transform it into lowercase and use stemming.

In [11]:
```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize

from string import punctuation
from nltk.stem.snowball import EnglishStemmer

import re

import preprocessor as p
from preprocessor.api import clean, tokenize, parse
```

In [12]:
```
# import nltk
# nltk.download('stopwords')
```

In [13]:
```
# p.set_options(p.OPT.URL)
```

In [14]:
```
translator = str.maketrans("", "", punctuation)
stemmer = EnglishStemmer()
```

In [17]:
```
def preproc_eng(texts):
    clear_texts = []
    count = 0
    for text in texts:
        # TODO: hack
        text = re.sub('\s[\@]\s', '', text)

        text = ' '.join([word for word in text.split() if word not in (s
topwords.words('english'))])
        # delete punctuation
        text = word_tokenize(text.translate(translator))

        # stemming
        text = [stemmer.stem(w) for w in text]
        # preprocessing as tweet
        text = clean(' '.join(text))
        clear_texts.append(text)

        # Increment
        count += 1
        if count % 5000 == 0:
            print(str(count) + "/" + str(len(texts)))
    return clear_texts
```

In [18]:
```
texts_clear = preproc_eng(texts)
texts_trial_clear = preproc_eng(texts_trial)
```

```
100/90000
200/90000
300/90000
400/90000
500/90000
600/90000
700/90000
800/90000
900/90000
1000/90000
1100/90000
1200/90000
1300/90000
1400/90000
1500/90000
1600/90000
1700/90000
1800/90000
1900/90000
2000/90000
2100/90000
2200/90000
2300/90000
2400/90000
2500/90000
2600/90000
2700/90000
2800/90000
2900/90000
3000/90000
3100/90000
3200/90000
3300/90000
3400/90000
3500/90000
3600/90000
3700/90000
3800/90000
3900/90000
4000/90000
4100/90000
4200/90000
4300/90000
4400/90000
4500/90000
4600/90000
4700/90000
4800/90000
4900/90000
5000/90000
5100/90000
5200/90000
5300/90000
5400/90000
5500/90000
5600/90000
5700/90000
```

```
5800/90000
5900/90000
6000/90000
6100/90000
6200/90000
6300/90000
6400/90000
6500/90000
6600/90000
6700/90000
6800/90000
6900/90000
7000/90000
7100/90000
7200/90000
7300/90000
7400/90000
7500/90000
7600/90000
7700/90000
7800/90000
7900/90000
8000/90000
8100/90000
8200/90000
8300/90000
8400/90000
8500/90000
8600/90000
8700/90000
8800/90000
8900/90000
9000/90000
9100/90000
9200/90000
9300/90000
9400/90000
9500/90000
9600/90000
9700/90000
9800/90000
9900/90000
10000/90000
10100/90000
10200/90000
10300/90000
10400/90000
10500/90000
10600/90000
10700/90000
10800/90000
10900/90000
11000/90000
11100/90000
11200/90000
11300/90000
11400/90000
```

```
11500/90000
11600/90000
11700/90000
11800/90000
11900/90000
12000/90000
12100/90000
12200/90000
12300/90000
12400/90000
12500/90000
12600/90000
12700/90000
12800/90000
12900/90000
13000/90000
13100/90000
13200/90000
13300/90000
13400/90000
13500/90000
13600/90000
13700/90000
13800/90000
13900/90000
14000/90000
14100/90000
14200/90000
14300/90000
14400/90000
14500/90000
14600/90000
14700/90000
14800/90000
14900/90000
15000/90000
15100/90000
15200/90000
15300/90000
15400/90000
15500/90000
15600/90000
15700/90000
15800/90000
15900/90000
16000/90000
16100/90000
16200/90000
16300/90000
16400/90000
16500/90000
16600/90000
16700/90000
16800/90000
16900/90000
17000/90000
17100/90000
```

```
17200/90000
17300/90000
17400/90000
17500/90000
17600/90000
17700/90000
17800/90000
17900/90000
18000/90000
18100/90000
18200/90000
18300/90000
18400/90000
18500/90000
18600/90000
18700/90000
18800/90000
18900/90000
19000/90000
19100/90000
19200/90000
19300/90000
19400/90000
19500/90000
19600/90000
19700/90000
19800/90000
19900/90000
20000/90000
20100/90000
20200/90000
20300/90000
20400/90000
20500/90000
20600/90000
20700/90000
20800/90000
20900/90000
21000/90000
21100/90000
21200/90000
21300/90000
21400/90000
21500/90000
21600/90000
21700/90000
21800/90000
21900/90000
22000/90000
22100/90000
22200/90000
22300/90000
22400/90000
22500/90000
22600/90000
22700/90000
22800/90000
```

```
22900/90000
23000/90000
23100/90000
23200/90000
23300/90000
23400/90000
23500/90000
23600/90000
23700/90000
23800/90000
23900/90000
24000/90000
24100/90000
24200/90000
24300/90000
24400/90000
24500/90000
24600/90000
24700/90000
24800/90000
24900/90000
25000/90000
25100/90000
25200/90000
25300/90000
25400/90000
25500/90000
25600/90000
25700/90000
25800/90000
25900/90000
26000/90000
26100/90000
26200/90000
26300/90000
26400/90000
26500/90000
26600/90000
26700/90000
26800/90000
26900/90000
27000/90000
27100/90000
27200/90000
27300/90000
27400/90000
27500/90000
27600/90000
27700/90000
27800/90000
27900/90000
28000/90000
28100/90000
28200/90000
28300/90000
28400/90000
28500/90000
```

```
28600/90000
28700/90000
28800/90000
28900/90000
29000/90000
29100/90000
29200/90000
29300/90000
29400/90000
29500/90000
29600/90000
29700/90000
29800/90000
29900/90000
30000/90000
30100/90000
30200/90000
30300/90000
30400/90000
30500/90000
30600/90000
30700/90000
30800/90000
30900/90000
31000/90000
31100/90000
31200/90000
31300/90000
31400/90000
31500/90000
31600/90000
31700/90000
31800/90000
31900/90000
32000/90000
32100/90000
32200/90000
32300/90000
32400/90000
32500/90000
32600/90000
32700/90000
32800/90000
32900/90000
33000/90000
33100/90000
33200/90000
33300/90000
33400/90000
33500/90000
33600/90000
33700/90000
33800/90000
33900/90000
34000/90000
34100/90000
34200/90000
```

```
34300/90000
34400/90000
34500/90000
34600/90000
34700/90000
34800/90000
34900/90000
35000/90000
35100/90000
35200/90000
35300/90000
35400/90000
35500/90000
35600/90000
35700/90000
35800/90000
35900/90000
36000/90000
36100/90000
36200/90000
36300/90000
36400/90000
36500/90000
36600/90000
36700/90000
36800/90000
36900/90000
37000/90000
37100/90000
37200/90000
37300/90000
37400/90000
37500/90000
37600/90000
37700/90000
37800/90000
37900/90000
38000/90000
38100/90000
38200/90000
38300/90000
38400/90000
38500/90000
38600/90000
38700/90000
38800/90000
38900/90000
39000/90000
39100/90000
39200/90000
39300/90000
39400/90000
39500/90000
39600/90000
39700/90000
39800/90000
39900/90000
```

```
40000/90000
40100/90000
40200/90000
40300/90000
40400/90000
40500/90000
40600/90000
40700/90000
40800/90000
40900/90000
41000/90000
41100/90000
41200/90000
41300/90000
41400/90000
41500/90000
41600/90000
41700/90000
41800/90000
41900/90000
42000/90000
42100/90000
42200/90000
42300/90000
42400/90000
42500/90000
42600/90000
42700/90000
42800/90000
42900/90000
43000/90000
43100/90000
43200/90000
43300/90000
43400/90000
43500/90000
43600/90000
43700/90000
43800/90000
43900/90000
44000/90000
44100/90000
44200/90000
44300/90000
44400/90000
44500/90000
44600/90000
44700/90000
44800/90000
44900/90000
45000/90000
45100/90000
45200/90000
45300/90000
45400/90000
45500/90000
45600/90000
```

```
45700/90000
45800/90000
45900/90000
46000/90000
46100/90000
46200/90000
46300/90000
46400/90000
46500/90000
46600/90000
46700/90000
46800/90000
46900/90000
47000/90000
47100/90000
47200/90000
47300/90000
47400/90000
47500/90000
47600/90000
47700/90000
47800/90000
47900/90000
48000/90000
48100/90000
48200/90000
48300/90000
48400/90000
48500/90000
48600/90000
48700/90000
48800/90000
48900/90000
49000/90000
49100/90000
49200/90000
49300/90000
49400/90000
49500/90000
49600/90000
49700/90000
49800/90000
49900/90000
50000/90000
50100/90000
50200/90000
50300/90000
50400/90000
50500/90000
50600/90000
50700/90000
50800/90000
50900/90000
51000/90000
51100/90000
51200/90000
51300/90000
```

```
51400/90000
51500/90000
51600/90000
51700/90000
51800/90000
51900/90000
52000/90000
52100/90000
52200/90000
52300/90000
52400/90000
52500/90000
52600/90000
52700/90000
52800/90000
52900/90000
53000/90000
53100/90000
53200/90000
53300/90000
53400/90000
53500/90000
53600/90000
53700/90000
53800/90000
53900/90000
54000/90000
54100/90000
54200/90000
54300/90000
54400/90000
54500/90000
54600/90000
54700/90000
54800/90000
54900/90000
55000/90000
55100/90000
55200/90000
55300/90000
55400/90000
55500/90000
55600/90000
55700/90000
55800/90000
55900/90000
56000/90000
56100/90000
56200/90000
56300/90000
56400/90000
56500/90000
56600/90000
56700/90000
56800/90000
56900/90000
57000/90000
```

```
57100/90000
57200/90000
57300/90000
57400/90000
57500/90000
57600/90000
57700/90000
57800/90000
57900/90000
58000/90000
58100/90000
58200/90000
58300/90000
58400/90000
58500/90000
58600/90000
58700/90000
58800/90000
58900/90000
59000/90000
59100/90000
59200/90000
59300/90000
59400/90000
59500/90000
59600/90000
59700/90000
59800/90000
59900/90000
60000/90000
60100/90000
60200/90000
60300/90000
60400/90000
60500/90000
60600/90000
60700/90000
60800/90000
60900/90000
61000/90000
61100/90000
61200/90000
61300/90000
61400/90000
61500/90000
61600/90000
61700/90000
61800/90000
61900/90000
62000/90000
62100/90000
62200/90000
62300/90000
62400/90000
62500/90000
62600/90000
62700/90000
```

```
62800/90000
62900/90000
63000/90000
63100/90000
63200/90000
63300/90000
63400/90000
63500/90000
63600/90000
63700/90000
63800/90000
63900/90000
64000/90000
64100/90000
64200/90000
64300/90000
64400/90000
64500/90000
64600/90000
64700/90000
64800/90000
64900/90000
65000/90000
65100/90000
65200/90000
65300/90000
65400/90000
65500/90000
65600/90000
65700/90000
65800/90000
65900/90000
66000/90000
66100/90000
66200/90000
66300/90000
66400/90000
66500/90000
66600/90000
66700/90000
66800/90000
66900/90000
67000/90000
67100/90000
67200/90000
67300/90000
67400/90000
67500/90000
67600/90000
67700/90000
67800/90000
67900/90000
68000/90000
68100/90000
68200/90000
68300/90000
68400/90000
```

```
68500/90000
68600/90000
68700/90000
68800/90000
68900/90000
69000/90000
69100/90000
69200/90000
69300/90000
69400/90000
69500/90000
69600/90000
69700/90000
69800/90000
69900/90000
70000/90000
70100/90000
70200/90000
70300/90000
70400/90000
70500/90000
70600/90000
70700/90000
70800/90000
70900/90000
71000/90000
71100/90000
71200/90000
71300/90000
71400/90000
71500/90000
71600/90000
71700/90000
71800/90000
71900/90000
72000/90000
72100/90000
72200/90000
72300/90000
72400/90000
72500/90000
72600/90000
72700/90000
72800/90000
72900/90000
73000/90000
73100/90000
73200/90000
73300/90000
73400/90000
73500/90000
73600/90000
73700/90000
73800/90000
73900/90000
74000/90000
74100/90000
```

```
74200/90000
74300/90000
74400/90000
74500/90000
74600/90000
74700/90000
74800/90000
74900/90000
75000/90000
75100/90000
75200/90000
75300/90000
75400/90000
75500/90000
75600/90000
75700/90000
75800/90000
75900/90000
76000/90000
76100/90000
76200/90000
76300/90000
76400/90000
76500/90000
76600/90000
76700/90000
76800/90000
76900/90000
77000/90000
77100/90000
77200/90000
77300/90000
77400/90000
77500/90000
77600/90000
77700/90000
77800/90000
77900/90000
78000/90000
78100/90000
78200/90000
78300/90000
78400/90000
78500/90000
78600/90000
78700/90000
78800/90000
78900/90000
79000/90000
79100/90000
79200/90000
79300/90000
79400/90000
79500/90000
79600/90000
79700/90000
79800/90000
```

```
79900/90000
80000/90000
80100/90000
80200/90000
80300/90000
80400/90000
80500/90000
80600/90000
80700/90000
80800/90000
80900/90000
81000/90000
81100/90000
81200/90000
81300/90000
81400/90000
81500/90000
81600/90000
81700/90000
81800/90000
81900/90000
82000/90000
82100/90000
82200/90000
82300/90000
82400/90000
82500/90000
82600/90000
82700/90000
82800/90000
82900/90000
83000/90000
83100/90000
83200/90000
83300/90000
83400/90000
83500/90000
83600/90000
83700/90000
83800/90000
83900/90000
84000/90000
84100/90000
84200/90000
84300/90000
84400/90000
84500/90000
84600/90000
84700/90000
84800/90000
84900/90000
85000/90000
85100/90000
85200/90000
85300/90000
85400/90000
85500/90000
```

```
85600/90000
85700/90000
85800/90000
85900/90000
86000/90000
86100/90000
86200/90000
86300/90000
86400/90000
86500/90000
86600/90000
86700/90000
86800/90000
86900/90000
87000/90000
87100/90000
87200/90000
87300/90000
87400/90000
87500/90000
87600/90000
87700/90000
87800/90000
87900/90000
88000/90000
88100/90000
88200/90000
88300/90000
88400/90000
88500/90000
88600/90000
88700/90000
88800/90000
88900/90000
89000/90000
89100/90000
89200/90000
89300/90000
89400/90000
89500/90000
89600/90000
89700/90000
89800/90000
89900/90000
90000/90000
100/10000
200/10000
300/10000
400/10000
500/10000
600/10000
700/10000
800/10000
900/10000
1000/10000
1100/10000
1200/10000
```

```
1300/10000
1400/10000
1500/10000
1600/10000
1700/10000
1800/10000
1900/10000
2000/10000
2100/10000
2200/10000
2300/10000
2400/10000
2500/10000
2600/10000
2700/10000
2800/10000
2900/10000
3000/10000
3100/10000
3200/10000
3300/10000
3400/10000
3500/10000
3600/10000
3700/10000
3800/10000
3900/10000
4000/10000
4100/10000
4200/10000
4300/10000
4400/10000
4500/10000
4600/10000
4700/10000
4800/10000
4900/10000
5000/10000
5100/10000
5200/10000
5300/10000
5400/10000
5500/10000
5600/10000
5700/10000
5800/10000
5900/10000
6000/10000
6100/10000
6200/10000
6300/10000
6400/10000
6500/10000
6600/10000
6700/10000
6800/10000
6900/10000
```

```
7000/10000
7100/10000
7200/10000
7300/10000
7400/10000
7500/10000
7600/10000
7700/10000
7800/10000
7900/10000
8000/10000
8100/10000
8200/10000
8300/10000
8400/10000
8500/10000
8600/10000
8700/10000
8800/10000
8900/10000
9000/10000
9100/10000
9200/10000
9300/10000
9400/10000
9500/10000
9600/10000
9700/10000
9800/10000
9900/10000
10000/10000
```

In [19]:
```
texts_clear[:5]
```

Out[19]:
```
['a littl throwback favourit personwat wall',
 'glam user yesterday kcon makeup use user featherette…',
 'democraci plaza wake stun outcom decision2016nbc news',
 'then amp now vilowalt disney magic kingdom',
 'who nevera galaxi far far away']
```

Build model

Baseline 1

Firstly, build the simplest model with TF_IDF as feautures and LogitRegression Classifier

Best score: 45.256

In [21]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
```

In [22]:
```
tf = TfidfVectorizer()
```

Split our data to train and to validation, get scores

In [27]:
```
def get_scores_valid(X, y, C=1.0, ratio=0.9, seed=14):
    '''
    X, y – выборка
    ratio – в каком отношении поделить выборку
    C, seed – коэф-т регуляризации и random_state
              логистической регрессии
    '''

    idx_split = int(ratio * len(X))
    X_train = X[:idx_split]
    X_valid = X[idx_split:]
    y_train = y[:idx_split]
    y_valid = y[idx_split:]

    X_train_tf = tf.fit_transform(X_train)
    X_valid_tf = tf.transform(X_valid)

    logit = LogisticRegression(C=C, n_jobs=-1, random_state=seed) # remo
ved dual=True

    logit.fit(X_train_tf, y_train)

    valid_pred = logit.predict(X_valid_tf)

    valid_pred.dtype = np.int
    np.savetxt('res.txt', valid_pred, fmt='%d')
    np.savetxt('goldres.txt', np.array(y_valid), fmt='%d')
```

Select parameters

In [32]:
```
Cs = np.logspace(-3, 1, 10)
scores =[]
count = 0
for C in Cs:
    print("C value: ", C)
    get_scores_valid(texts_clear, labels, C=C)
    %run ./tools/evaluationscript/scorer_semeval18.py goldres.txt res.tx
t
    count += 1
    print("{}/{}\n".format(count, len(Cs)))
```

```
C value:  0.001
Macro F-Score (official): 1.783
-----
Micro F-Score: 21.7
Precision: 21.7
Recall: 21.7
1/10

C value:  0.0027825594022071257
Macro F-Score (official): 1.783
-----
Micro F-Score: 21.7
Precision: 21.7
Recall: 21.7
2/10

C value:  0.007742636826811269
Macro F-Score (official): 1.783
-----
Micro F-Score: 21.7
Precision: 21.7
Recall: 21.7
3/10

C value:  0.021544346900318832
Macro F-Score (official): 3.877
-----
Micro F-Score: 22.856
Precision: 22.856
Recall: 22.856
4/10

C value:  0.05994842503189409
Macro F-Score (official): 8.647
-----
Micro F-Score: 26.322
Precision: 26.322
Recall: 26.322
5/10

C value:  0.1668100537200059
Macro F-Score (official): 13.617
-----
Micro F-Score: 29.933
Precision: 29.933
Recall: 29.933
6/10

C value:  0.46415888336127775
Macro F-Score (official): 17.001
-----
Micro F-Score: 31.689
Precision: 31.689
Recall: 31.689
7/10

C value:  1.2915496650148828
```

```
Macro F-Score (official): 20.07
-----
Micro F-Score: 32.0
Precision: 32.0
Recall: 32.0
8/10

C value:  3.593813663804626
Macro F-Score (official): 19.289
-----
Micro F-Score: 29.333
Precision: 29.333
Recall: 29.333
9/10

C value:  10.0
Macro F-Score (official): 18.992
-----
Micro F-Score: 27.4
Precision: 27.4
Recall: 27.4
10/10
```

In [25]:
```
C_best = 10.0
```

Check best model on trial data

In [46]:
```
logit = LogisticRegression(n_jobs=-1, random_state=14, C=C_best)
```

In [47]:
```
X_train_tf = tf.fit_transform(texts_clear)
X_test_tf = tf.transform(texts_trial_clear)
```

In [48]:
```
logit.fit(X_train_tf, labels)
```

Out[48]:
```
LogisticRegression(C=10.0, class_weight=None, dual=False, fit_intercept
=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=-1, penalty='l2', random_
state=28,
                   solver='lbfgs', tol=0.0001, verbose=0, warm_start=Fa
lse)
```

In [49]:
```
res = logit.predict(X_test_tf)
```

In [50]:
```
res.dtype = np.int
```

In [51]:
```
np.savetxt('res.txt', res, fmt='%d')
```

In [52]:
```
%run ./tools/evaluationscript/scorer_semeval18.py ./data/test/english_test.labels res.txt
```

```
Macro F-Score (official): 20.121
-----
Micro F-Score: 27.17
Precision: 27.17
Recall: 27.17
```

In [ ]: