Extracting Tweet Entities

What is ENTITY EXTRACTION?

- It is a process where an algorithm takes a string of text as input and identifies relevant nouns (people, places and organizations) that are mentioned in that string.

- Below example extracts the text, screen names, and hashtags from the tweets that are collected and introduces a Python idiom called a ***double (***or ***nested) list comprehension.***

Example 1-6. Extracting text, screen names, and hashtags from tweets

```
status_texts = [ status['text']
                 for status in statuses ]

screen_names = [ user_mention['screen_name']
                 for status in statuses
                     for user_mention in status['entities']['user_mentions'] ]

hashtags = [ hashtag['text']
             for status in statuses
                 for hashtag in status['entities']['hashtags'] ]
```

- If you understand a (single) list comprehension, the code formatting should illustrate the double list comprehension as simply a collection of values that are derived from a nested loop as opposed to the results of a single loop.
- List comprehensions are particularly powerful because they usually yield substantial performance gains over nested lists and provide an intuitive (once you're familiar with them) yet terse syntax.

```
# Compute a collection of all words from all tweets
words = [ w
          for t in status_texts
              for w in t.split() ]

# Explore the first 5 items for each...

print json.dumps(status_texts[0:5], indent=1)
print json.dumps(screen_names[0:5], indent=1)
print json.dumps(hashtags[0:5], indent=1)
print json.dumps(words[0:5], indent=1)
```

- In Python, syntax in which square brackets appear after a list or string value, such as status texts[0:5], is indicative of slicing, whereby you can easily extract items from lists or substrings from strings.
- In this particular case, [0:5] indicates that you'd like the first five items in the list status texts (corresponding to items at indices 0 through 4).

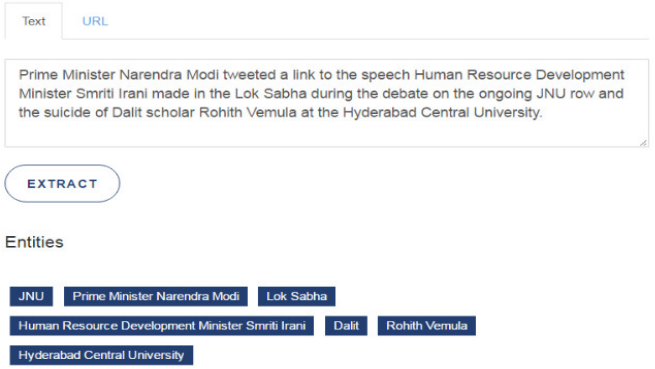

```
[
 "\u201c@KathleenMariee_: #MentionSomeOneImportantForYou @AhhlicksCruise...,
 "#MentionSomeoneImportantForYou My bf @Linkin_Sunrise.",
 "RT @hassanmusician: #MentionSomeoneImportantForYou God.",
 "#MentionSomeoneImportantForYou @Louis_Tomlinson",
 "#MentionSomeoneImportantForYou @Delta_Universe"
]
```

```
[
 "KathleenMariee_",
 "AhhlicksCruise",
 "itsravennn_cx",
 "kandykisses_13",
 "BMOLOGY"
]
[
 "MentionSomeOneImportantForYou",
 "MentionSomeoneImportantForYou",
 "MentionSomeoneImportantForYou",
 "MentionSomeoneImportantForYou",
 "MentionSomeoneImportantForYou"
]
```

```
[
 "\u201c@KathleenMariee_:",
 "#MentionSomeOneImportantForYou",
 "@AhhlicksCruise",
 ",",
 "@itsravennn_cx"
]
```

Use Cases of Entity Extraction

- Classifying content for news providers.



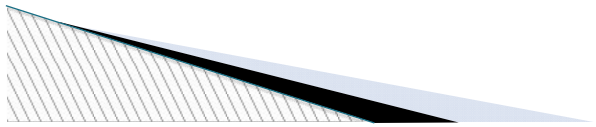
Analyzing Tweets and Tweet Entities with Frequency Analysis:-

- Manipulating data so that it can be counted and further manipulated in meaningful ways.

- Let's now take a closer look at what's in the data by computing a frequency distribution and looking at the top 10 items in each list.
- Among the more compelling reasons for mining Twitter data is to try to answer the question of what people are talking about right now.
- One of the simplest techniques you could apply to answer this question is basic frequency analysis, just as we are performing here.

- Example 1-7.
  Creating a basic frequency distribution from the words in tweets

```
from collections import Counter
for item in [words, screen_names, hashtags]:
c = Counter(item)
print c.most_common()[:10]   #top 10
print
```



- The result of the frequency distribution is a map of key/value pairs corresponding to terms and their frequencies.

```
[(u'#MentionSomeoneImportantForYou', 92), (u'RT', 34), (u'my', 10),
 (u',', 6), (u'@justinbieber', 6), (u'<3', 6), (u'My', 5), (u'and', 4),
 (u'I', 4), (u'te', 3)]

[(u'justinbieber', 6), (u'Kid_Charliej', 2), (u'Cavillafuerte', 2),
 (u'touchmestyles_', 1), (u'aliceorr96', 1), (u'gymleeam', 1), (u'fienas', 1),
 (u'nayely_1D', 1), (u'angelchute', 1)]

[(u'MentionSomeoneImportantForYou', 94), (u'mentionsomeoneimportantforyou', 3),
 (u'NoHomo', 1), (u'Love', 1), (u'MentionSomeOneImportantForYou', 1),
 (u'MyHeart', 1),  (u'bebesito', 1)]
```