

# Working with Text in Python(saved)

## Introduction to Text Mining



## Introduction to Text Mining

### APPLIED TEXT MINING IN PYTHON

V.G. Vinod Vydiswaran

Assistant Professor of Learning Health Sciences,  
Medical School and Assistant Professor of Information,  
School of Information



## Text data is growing fast!

- **Data continues to grow exponentially**
  - Estimated to be 2.5 Exabytes (2.5 million TB) a day
  - Grow to 40 Zettabytes (40 billion TB) by 2020 (50-times that of 2010)
- **Approximately 80% of all data is estimated to be unstructured, text-rich data**
  - >40 million articles (5 million in English) in Wikipedia
  - >4.5 billion Web pages
  - >500 million tweets a day, 200 billion a year
  - >1.5 trillion queries / searches on Google a year

# Data hidden in plain sight

The image shows a Twitter profile for 'UN Spokesperson' (@UN\_Spokesperson). The profile picture is the United Nations logo. The header shows statistics: 14.6K tweets, 994 following, 391K followers, 49 likes, and 3 lists. The bio states: 'Official Twitter account of the Office of the Spokesperson for United Nations Secretary-General Ban Ki-moon.' The location is 'New York, USA' and it was joined in May 2010. There are 3,008 photos and videos. The tweets section shows three tweets. The second tweet, from 17h ago, reads: '"Ethics are built right into the ideals and objectives of the United Nations" #UNSG @ NY Society for Ethical Culture bit.ly/2guVelr'. This tweet has 8 replies, 13 retweets, and 27 likes. Annotations with green boxes and arrows point to various elements: 'Social network' points to the profile picture; 'Author' points to the name 'UN Spokesperson'; 'Description' points to the bio; 'Location' points to 'New York, USA'; 'Tweet' points to the tweet text, with sub-points 'Topic' and 'Sentiment'; 'Time' points to the timestamp '17h'; 'Popularity' points to the engagement metrics (replies, retweets, likes).

**Social network**

**Author**

**Description**

**Location**

**Tweet**

- Topic
- Sentiment

**Time**

**Popularity**

## So, what can be done with text?

- Parse text
- Find / Identify / Extract relevant information from text
- Classify text documents
- Search for relevant text documents
- Sentiment analysis
- Topic modeling

## Finding specific words

- **Long words:** Words that are most than 3 letters long

```
>>> [w for w in text2 if len(w) > 3]
['Ethics', 'built', 'right', 'into', 'ideals', 'objectives', 'United', 'Nations']
```

- **Capitalized words**

```
>>> [w for w in text2 if w.istitle()]
['Ethics', 'United', 'Nations']
```

- **Words that end with s**

```
>>> [w for w in text2 if w.endswith('s')]
```

```
1 [w for w in text2 if w.endswith('s')]
```

## Finding unique words: using set()

```
>>> text3 = 'To be or not to be'
>>> text4 = text3.split(' ')
>>> len(text4)
6
>>> len(set(text4))
5
>>> set(text4)
set(['not', 'To', 'or', 'to', 'be'])
>>> len(set([w.lower() for w in text4]))
4
>>> set([w.lower() for w in text4])
set(['not', 'to', 'or', 'be'])
```

## Some word comparison functions ...

- `s.startswith(t)`
- `s.endswith(t)`
- `t in s`
- `s.isupper(); s.islower(); s.istitle()`
- `s.isalpha(); s.isdigit(); s.isalnum()`

## String Operations

- `s.lower(); s.upper(); s.titlecase()`
- `s.split(t)`
- `s.splitlines()`
- `s.join(t)`
- `s.strip(); s.rstrip()`
- `s.find(t); s.rfind(t)`
- `s.replace(u, v)`



## From words to characters

```
>>> text5 = 'ouagadougou'
>>> text6 = text5.split('ou')
>>> text6
['', 'agad', 'g', '']
>>> 'ou'.join(text6)
'ouagadougou'
```

```
>>> text5.split('')
Traceback (most recent call last):
  File "<stdin>", line 1, in
<module>
ValueError: empty separator
>>> list(text5)
['o', 'u', 'a', 'g', 'a', 'd',
'o', 'u', 'g', 'o', 'u']
>>> [c for c in text5]
['o', 'u', 'a', 'g', 'a', 'd',
'o', 'u', 'g', 'o', 'u']
```

## Cleaning text

```
>>> text8 = '    A quick brown fox jumped over the lazy dog. '
>>> text8.split(' ')
['', '', '\t', 'A', 'quick', 'brown', 'fox', 'jumped', 'over',
'the', 'lazy', 'dog.', '']
>>> text9 = text8.strip()
>>> text9.split(' ')
['A', 'quick', 'brown', 'fox', 'jumped', 'over', 'the',
'lazy', 'dog.']
```

## Changing text

- Find and replace

```
>>> text9
'A quick brown fox jumped over the lazy dog.'
>>> text9.find('o')
10
>>> text9.rfind('o')
40
>>> text9.replace('o', 'O')
'A quick brOwn fOx jumped Over the lazy dOg.'
```

# Handling larger texts

- **Reading files line by line**

```
>>> f = open('UNDHR.txt', 'r')
>>> f.readline()
'Universal Declaration of Human Rights\n'
```

- **Reading the full file**

```
>>> f.seek(0)
>>> text12 = f.read()
>>> len(text12)
10891
>>> text13 = text12.splitlines()
>>> len(text13)
158
>>> text13[0]
'Universal Declaration of Human Rights'
```

## File operations

- **`f = open(filename, mode)`**
- **`f.readline(); f.read(); f.read(n)`**
- **`for line in f: doSomething(line)`**
- **`f.seek(n)`**
- **`f.write(message)`**
- **`f.close()`**
- **`f.closed`**

## Issues with reading text files

```
>>> f = open('UNDHR.txt', 'r')
>>> text14 = f.readline()
'Universal Declaration of Human Rights\n'
```

- **How do you remove the last newline character?**

```
>>> text14.rstrip()
'Universal Declaration of Human Rights'
```

– Works also for DOS newlines (^M) that shows up as `'\r'` or `'\r\n'`

# Take home concepts

- Handling text sentences
- Splitting sentences into words, words into characters
- Finding unique words
- Handling text from documents