Social Media & Text Analysis

part 2 - Intro to NLP

Ohio State University
Instructor: Wei Xu
Website: socialmedia-class.org
Basic Text Processing

• Tokenization:

```python
import nltk
nltk.download('punkt')
sentence = "At eight o'clock in the morning, Arthur didn't feel well."
tokens = nltk.word_tokenize(sentence)
print(tokens)
```

['At', 'eight', "o'clock", 'in', 'the', 'morning', ',', 'Arthur', 'did', 'n't', 'feel', 'well', '.']

breaking text up into words, phrases, symbols, or other meaningful elements called tokens.
Basic Text Processing

• Tokenization:

```python
import nltk
	nltk.download('punkt')
	sentence = "At eight o'clock in the morning, Arthur didn't feel well."

tokens = nltk.word_tokenize(sentence)
print (tokens)
```

To start using Python NLTK (Natural Language Toolkit) library.
Basic Text Processing

• Tokenization:

```python
import nltk
nltk.download('punkt')

sentence = "At eight o'clock in the morning, Arthur didn't feel well."

tokens = nltk.word_tokenize(sentence)
print (tokens)
```

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
['At', 'eight', 'o'clock', 'in', 'the', 'morning', ',', ',', 'Arthur', ',']

Load in a pre-trained tokenizer for English named “Punkt”.
Basic Text Processing

• Tokenization:

```python
import nltk

nltk.download('punkt')

sentence = "At eight o'clock in the morning, Arthur didn't feel well."

tokens = nltk.word_tokenize(sentence)

print(tokens)
```

Calling the `word_tokenize()` function in the `nltk` module.
Try it Out!

• We will use Google's Colab programming environment:

What if we try to tokenize some tweets?
Try it Out!

- We will use **Google's Colab** programming environment:

What if we try to tokenize some tweets?

```python
[20] tweet1 = "@someone did you check out this #superawesome!! <3"
    print (nltk.word_tokenize(tweet1))

[ ] ['@', 'someone', 'did', 'you', 'check', 'out', 'this', '#', 'superawesome', '!', '!', '<', '3']
Twitter-specific Tokenizer

`Twokenize` is another tokenizer specifically designed for processing Twitter data. Google Colab doesn't have it built-in, so we will first use `pip` installer to install the `Twokenize` package.

```
[16] !pip install -q twokenize
```

For the same example tweet as well, Twokenize appears to work even better:

```
[25] import twokenize

tweet = "my heart.. broken T____T</3"

print (twokenize.tokenizeRawTweetText(tweet))
```

```python
[my', 'heart', '..', 'broken', 'T____T', '</3']
```
NLP Pipeline

- Language Identification
- Tokenization
- Part-of-Speech (POS) Tagging
- Shallow Parsing (Chunking)
- Named Entity Recognition (NER)
- Stemming
- Normalization
More Resources

• NLTK (Natural Language Toolkit):
  - NLTK Book: http://www.nltk.org/book — free online!