Getting started

Should have python, git and pip installed

- Install numpy and matplotlib
- Download NLTK
  - www.nltk.org/install.html
  - sudo pip install -U nltk
- Test NLTK
  - python
  - import nltk
- Download data
  - python
  - import nltk
  - nltk.download()
- In NLTK downloader:
  - Under corpora, download:
    - gutenberg, brown, state_union, stopwords, words
  - Under models, download:
    - averaged_perceptron_tagger, maxent_ne_chunker, punkt
- GitHub code and readme download:
  - https://github.com/ab6/QConSF-2016.git

Introduction to NLP using NLTK and Python

Amber McKenzie

QCon San Francisco
November 11, 2016

Who am I?

dialogtech

C B M

University of South Carolina

UNIVERSITY OF SOUTH CAROLINA

COMPUTATIONAL DATA ANALYTICS GROUP

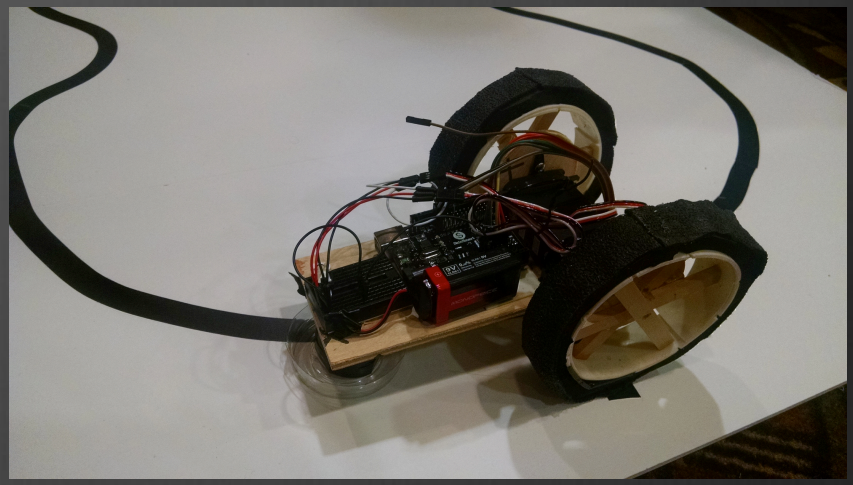
AT THE OAK RIDGE NATIONAL LABORATORY

Contact information

Amber McKenzie, Ph.D.
Data Scientist

DialogTech
www.dialogtech.com

mckenzie.amber@gmail.com
amber.mckenzie@dialogtech.com
https://nlprunner.wordpress.com



Resources

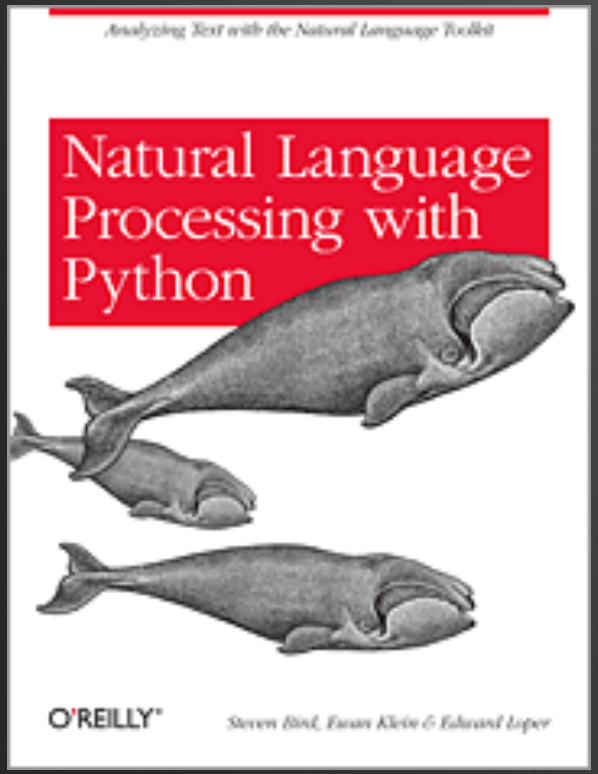
- NLTK book
  - http://www.nltk.org/book/
- Matplotlib tutorial
  - http://matplotlib.org/users/pyplot_tutorial.html
- Other resource links on the module write-ups

Agenda

- Introduction to Natural Language ToolKit (NLTK)
- Module 1: document classification
- Module 2: historical text analysis

- GitHub code and readme download:
  - https://github.com/ab6/QConSF-2016.git

Natural Language ToolKit (NLTK)



- Suite of text processing libraries
- Number of available pre-built models and corpora
- Easy to get started with basic NLP
  - Building blocks for more robust applications
  - Both linguistic and statistical analysis libraries
- www.nltk.org

NLTK data

- Over 50 different data sources available:
  - News: Brown, Reuters
  - Historical: State of the Union and inaugural addresses
  - Literary: Project Gutenberg, Shakespeare
  - Product reviews
  - Web text and twitter
- http://www.nltk.org/nltk_data/

Module Introduction

- Designed as an overview to NLP and to be used as a springboard to start NLP applications and analyses
- Module 1 – Document classification
  - Classify documents into literary categories
  - Identify features, process data, create model
- Module 2 – Historical data analysis
  - Conduct statistical analyses on dataset
  - Examine various text characteristics and conduct NLP-based data analysis

Modules

- Module explanation and solutions provided in GitHub repo
- Explanation
  - High level overview
  - Fine-grained steps
  - Hints and resources
- Solutions
  - Code for core requirements
- Helper functions for Module 2 provided also
- GitHub code and readme download:
  - https://github.com/ab6/QConSF-2016.git

NLP, data analysis, and ML

- State of the art in NLP is trending towards statistical and supervised learning methods
- NLP techniques used to extract features of text in data analysis and machine learning applications
- Deep learning used to model underlying processes behind language

Text features

- Words
  - Existence and frequency
  - Types: word features and functions
  - Specific: concepts and entities
  - Stopwords
- Phrases and collocations
  - What words occur together
  - Google's ngram corpus

NLTK Intro – data access

```
>>> import nltk
>>> from nltk.corpus import gutenberg
>>> print (gutenberg.fileids())
[u'austen-emma.txt', u'austen-persuasion.txt', u'austen-sense.txt', u'bible-kjv.txt', u'blake-poems.txt', u'bryant-stori
es.txt', u'burgess-busterbrown.txt', u'carroll-alice.txt', u'chesterton-ball.txt', u'chesterton-brown.txt', u'chesterton
-thursday.txt', u'edgeworth-parents.txt', u'melville-moby_dick.txt', u'milton-paradise.txt', u'shakespeare-caesar.txt',
u'shakespeare-hamlet.txt', u'shakespeare-macbeth.txt', u'whitman-leaves.txt']
```

```
>>> ids = gutenberg.fileids()
>>> emmaWords = gutenberg.words("austen-emma.txt")
>>> emmaWords = gutenberg.words(ids[0])
>>> print (emmaWords[:20])
[u'[', u'Emma', u'by', u'Jane', u'Austen', u'1816', u']', u'VOLUME', u'I', u'CHAPTER', u'I', u'Emma', u'Woodhouse', u','
, u'handsome', u',', u'clever', u',', u'and', u'rich']
```

```
>>> emmaText = gutenberg.raw(ids[0])
>>> print (emmaText[:47])
[Emma by Jane Austen 1816]

VOLUME I

CHAPTER I
```

Text representation

- Bag-of-words vs. frequency counts
- Frequency distributions
  - Stopword lists
  - Term frequency-inverse document frequency (TF-IDF)
- Features are dictated by the data and the target application
  - For some applications, frequencies are useful. For others, just a binary representation of words is good
  - Other linguistic features include morphology, capitalization, part-of-speech
- Word embeddings

FreqDist

```
>>> lowerWords = [word.lower() for word in emmaWords]
>>> fdist = nltk.FreqDist(lowerWords)
>>> mostCommon = fdist.most_common(10)
>>> print (mostCommon)
[(',', 11454), ('.', 6928), ('to', 5239), ('the', 5201), ('and', 4896), ('of', 4291), ('i', 3178),
('a', 3129), ('it', 2528), ('her', 2469)]
>>> words, freqs = map(list, zip(*mostCommon))
>>> print (words)
[',', '.', 'to', 'the', 'and', 'of', 'i', 'a', 'it', 'her']
```

Module 1: doc classification

- Brown corpus: documents and category tags
- Identify set of feature words
- Create bag-of-words representation
- Breaking into training and test set
- Create and test model

Basic NLP

- Tokenization
  - Considerations: white space, hyphens, apostrophes, etc.
- Sentence segmentation
- Part-of-speech tagging (POS)
- Syntax
  - Used to identify relationships between concepts
- Named entity recognition (NER)
  - With or without categories

Tokenization and sentence segmentation

```
>>> tokens = nltk.word_tokenize(emmaText)
>>> print (tokens[:30])
[u'[', u'Emma', u'by', u'Jane', u'Austen', u'1816', u']', u'VOLUME', u'I', u'CHAPTER', u'I', u'Emma', u'Woodhouse', u','
, u'handsome', u',', u'clever', u',', u'and', u'rich', u',', u'with', u'a', u'comfortable', u'home', u'and', u'happy', u
'disposition', u',', u'seemed']
```

```
>>> sents = nltk.sent_tokenize(emmaText)
>>> print (sents[4:6])
[u'Between _them_ it was more the intimacy\nof sisters.', u"Even before Miss Taylor had ceased to hold the nominal\noffi
ce of governess, the mildness of her temper had hardly allowed\nher to impose any restraint; and the shadow of authority
 being\nnow long passed away, they had been living together as friend and\nfriend very mutually attached, and Emma doing
 just what she liked;\nhighly esteeming Miss Taylor's judgment, but directed chiefly by\nher own."]
```

*Note the new-line characters.

Bigrams and Collocations

```
>>> nltk.FreqDist(nltk.bigrams(lowerWords)).most_common(10)
[((',', 'and'), 1881), (('mr', '.'), 1153), (("'", 's'), 932), ((';', 'and'), 866), (('."', '"'), 7
57), (('mrs', '.'), 699), (('to', 'be'), 607), (('.', 'i'), 570), ((',', 'i'), 568), (('of', 'the')
, 559)]
```

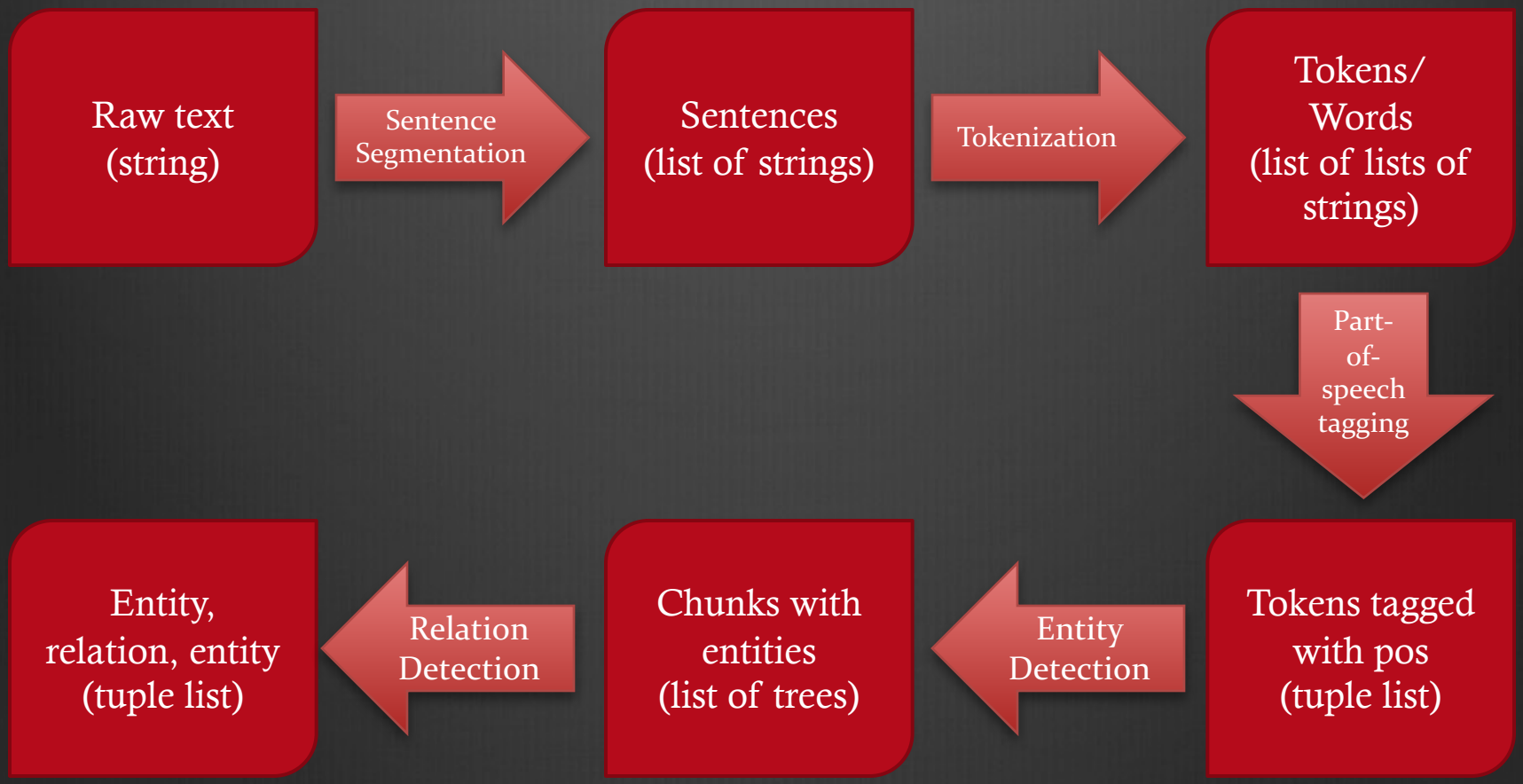
```
>>> bigram_measures = nltk.collocations.BigramAssocMeasures()
>>> finder = BigramCollocationFinder.from_words(lowerWords)
>>> finder.apply_freq_filter(3)
>>> finder.nbest(bigram_measures.pmi, 10)
[('caro', 'sposo'), ('&', 'c'), ('frozen', 'maid'), ('brunswick', 'square'), ('extensive', 'grounds
'), ('nicely', 'dressed'), ('sore', 'throat'), ('mill', 'farm'), ('thousand', 'pounds'), ('william'
, 'larkins')]
```

POS tagging and NER

```
>>> sent
[u'She', u'dearly', u'loved', u'her', u'father', u',', u'but', u'he', u'was', u'no', u'companion', u'for', u'her', u'.']
>>> nltk.pos_tag(sent)
[(u'She', 'PRP'), (u'dearly', 'RB'), (u'loved', 'VBD'), (u'her', 'PRP'), (u'father', 'NN'), (u',', ','), (u'but', 'CC'),
 (u'he', 'PRP'), (u'was', 'VBD'), (u'no', 'DT'), (u'companion', 'NN'), (u'for', 'IN'), (u'her', 'PRP$'), (u'.', '.')]
```

```
>>> nltk.ne_chunk(pos)
Tree('S', [Tree('GPE', [(u'Highbury', 'NNP')]), (u',', ','), (u'the', 'DT'), (u'large', 'JJ'), (u'and', 'CC'), (u'populo
us', 'JJ'), (u'village', 'NN'), (u',', ','), (u'almost', 'RB'), (u'amounting', 'VBG'), (u'to', 'TO'), (u'a', 'DT'), (u't
own', 'NN'), (u',', ','), (u'to', 'TO'), (u'which', 'WDT'), Tree('PERSON', [(u'Hartfield', 'NNP')]), (u',', ','), (u'in'
, 'IN'), (u'spite', 'NN'), (u'of', 'IN'), (u'its', 'PRP$'), (u'separate', 'JJ'), (u'lawn', 'NN'), (u',', ','), (u'and',
'CC'), (u'shrubberies', 'NNS'), (u',', ','), (u'and', 'CC'), (u'name', 'NN'), (u',', ','), (u'did', 'VBD'), (u'really',
'RB'), (u'belong', 'JJ'), (u',', ','), (u'afforded', 'VBD'), (u'her', 'PRP'), (u'no', 'DT'), (u'equals', 'NNS'), (u'.',
'.')])
>>> nltk.ne_chunk(pos, binary = True)
Tree('S', [Tree('NE', [(u'Highbury', 'NNP')]), (u',', ','), (u'the', 'DT'), (u'large', 'JJ'), (u'and', 'CC'), (u'populou
s', 'JJ'), (u'village', 'NN'), (u',', ','), (u'almost', 'RB'), (u'amounting', 'VBG'), (u'to', 'TO'), (u'a', 'DT'), (u'to
wn', 'NN'), (u',', ','), (u'to', 'TO'), (u'which', 'WDT'), Tree('NE', [(u'Hartfield', 'NNP')]), (u',', ','), (u'in', 'IN
'), (u'spite', 'NN'), (u'of', 'IN'), (u'its', 'PRP$'), (u'separate', 'JJ'), (u'lawn', 'NN'), (u',', ','), (u'and', 'CC')
, (u'shrubberies', 'NNS'), (u',', ','), (u'and', 'CC'), (u'name', 'NN'), (u',', ','), (u'did', 'VBD'), (u'really', 'RB')
, (u'belong', 'JJ'), (u',', ','), (u'afforded', 'VBD'), (u'her', 'PRP'), (u'no', 'DT'), (u'equals', 'NNS'), (u'.', '.')]
)
```

Information extraction process



Module 2: historical data analysis

- Data: State of the Union addresses
- Goal: Conduct text-based data analysis using a variety of features
- Methods
  - Length of words, unique words
  - Bigrams and collocations
  - Named entities
- Additional analyses
  - Part of speech frequencies
  - Sentiment
  - Categorical grouping

Interesting Finding

- Arthur Vandenberg in 1997 collocation
  - Why?
- 50 years before the address, Vandenberg became chairman of the Senate Foreign Relations Committee.
- In his address, Clinton was advocating for foreign diplomacy and citing the historical landmarks of the Truman Doctrine and NATO, in which Vandenberg played a large part.

Contact information

Amber McKenzie, Ph.D.
Data Scientist

DialogTech
www.dialogtech.com

mckenzie.amber@gmail.com
amber.mckenzie@dialogtech.com
https://nlprunner.wordpress.com