CIS192 Python Programming

NLP

Harry Smith

University of Pennsylvania

NOvember 15, 2017

Outline

Natural Language Processing

Natural language processing

From Wikipedia, the free encyclopedia

Natural language processing is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction. Many challenges in NLP involve: natural language understanding, enabling computers to derive meaning from human or natural language input; and others involve natural language generation.

Natural Language Processing

source: researchperspectives.org

Language is Hard

How can a computer:

- Recognize parts of speech of sentences?
- Tell whether a sentence is positive or negative?
- Figure out what words are most commonly used?
- Summarize text?

Some Applications of NLP

- Sentiment analysis
- Spam filtering
- Plagiarism detection
- Document categorization
- Summarization
- Text search
- Much more...

Natural Language Tool Kit (NLTK)

- NLP toolkit for English in Python
- Developed at Penn in 2001!
- nltk.org

Terminology

- Corpus: a body of text
- Token: Each meaningful "entity" in a string
  - Depending on context, tokens can be words, sentences, paragraphs
- Part of Speech: categories that words are assigned to
  - noun, verb, adjective, ...
- Stopwords: most common words in a language, filtered out before NLP tasks
  - the, is, at, which, on, ...

Outline

Word Tokenization

```
>>> nltk.word_tokenize('The mitochondria is the
powerhouse of the cell.')
['The', 'mitochondria', 'is', 'the', 'powerhouse', '
of', 'the', 'cell', '.']
```

Sentence Tokenization

```
>>> sentences = "Prof. Sanjeev Khanna taught CIS 320
 last spring. It was a great class...and I wasn't
able to get off the waitlist for CIS 677."
>>> nltk.sent_tokenize(sentences)
['Prof. Sanjeev Khanna taught CIS 320 last spring.',
 "It was a great class...and I wasn't able to get
 off the waitlist for CIS 677."]
```

Outline

Counting Words in a Corpus

Before today:

```
>>> counts = defaultdict(int)
>>> for word in words:
      counts[word] += 1
```

Better:

```
>>> counts = FreqDist(words)
>>> counts.most_common(10) #=> [('the', 49), ...]
```

Neat!

Outline

Creating "random" sentences from a corpus

- In probability theory, Markov Chains are "memoryless"
- "Future state depends on current state only"
- To create a "random" sentence:
  - Take your current word
  - Add a new word that typically appears after your current word
  - Repeat!

Outline

Part of Speech Tagging

- Use `nltk.pos_tag(list_of_tokens)` to identify part of speech tags
- `nltk.help.upenn_tagset` shows what each tag code means

Outline

Free Word Association

- After hearing a word, what's the word that immediately comes to mind?
- "dog" $\rightarrow$ "cat" $\rightarrow$ "meow" $\rightarrow$ ...
- How would we do this?

Free Word Association

- After hearing a word, what's the word that immediately comes to mind?
- "dog" $\rightarrow$ "cat" $\rightarrow$ "meow" $\rightarrow$ ...
- Simple way:
  - For each token in our corpus, count the occurrences of surrounding tokens

Outline

Sentiment Analysis

- Is some particular text is positive or negative (and to what degree?)
- How might we do this?

Sentiment Analysis

- Is some particular text is positive or negative (and to what degree?)
- How might we do this?
  - Machine learning (last two lectures)
    - ★ Try to "learn" the sentiment-relevant features of text
    - ★ Need lots of training data
    - ★ Data driven approach
  - Rule-based methods
    - ★ "Rule of thumb": uses heuristics to determine sentiments
    - ★ Needs little training data
    - ★ Good for production: fast, but harder to initially create
    - ★ VADER: popular rule based model aimed for social media

Outline

Next Steps

- CIS 526: Machine Translation
- CIS 530: Computational Linguistics
- Kaggle for large datasets, competitions
- awesome-nlp: curated list of NLP resources on GitHub