Text Wrangling in NLP

MAS.S60

Rob Speer

Catherine Havasi

Text isn't always nice to you

- You may have noticed this when doing the assignment

Problem Solvers

- Crazy HTML/XML? – `BeautifulSoup`
- RSS/Atom feeds? – `feedparser`
- CSV? – `csv` in the standard library
- Excel? – **just save it as CSV** (and use the "excel" dialect)
- Word or PDF? There are packages…

Tokenization

- How do you split words?
- How do you split sentences?

Exceptions to tokenization

- NLTK doesn't necessarily do the right thing in your domain
- There isn't one right answer, just ad-hoc solutions

Regular expressions for the win!

- Suppose social media users are commenting on events
- If it's a negative event, they say #TopicFail
- If it's a positive event, they say #TopicFTW
- The topic might have CamelCaseWords
- Your task: recognize these strings and separate the words

Stemming

```
>>> porter = nltk.PorterStemmer()
>>> lancaster = nltk.LancasterStemmer()
>>> [porter.stem(t) for t in tokens]
['DENNI', ':', 'Listen', ',', 'strang', 'women', 'lie', 'in', 'pond',
'distribut', 'sword', 'is', 'no', 'basi', 'for', 'a', 'system', 'of', 'govern',
'.', 'Suprem', 'execut', 'power', 'deriv', 'from', 'a', 'mandat', 'from',
'the', 'mass', ',', 'not', 'from', 'some', 'farcic', 'aquat', 'ceremoni', '.']
>>> [lancaster.stem(t) for t in tokens]
['den', ':', 'list', ',', 'strange', 'wom', 'lying', 'in', 'pond', 'distribut',
'sword', 'is', 'no', 'bas', 'for', 'a', 'system', 'of', 'govern', '.', 'suprem',
'execut', 'pow', 'der', 'from', 'a', 'mand', 'from', 'the', 'mass', ',', 'not',
'from', 'som', 'farc', 'aqu', 'ceremony', '.']
```

- NLTK has multiple options for stemming
  - Porter – very aggressive stemmer, but often used in indexing text
  - Lancaster – similar, sometimes outputs more recognizable word forms

Snowball (Porter) stemming

- Does the word end with a known ending?
- Is there enough before the ending?
- Then replace the ending with something simpler.

relation**al** (al ->)
rela**tion** (tion -> t)
relat

rel**ate** (ate -> at)
relat

relat**ing** (ing ->)
relat

correla**tion** (tion -> t)
correl**at** (at ->)
correl

Lemmatizers – Morphy

```
>>> wnl = nltk.WordNetLemmatizer()
>>> [wnl.lemmatize(t) for t in tokens]
['DENNIS', ':', 'Listen', ',', 'strange', 'woman', 'lying', 'in', 'pond',
'distributing', 'sword', 'is', 'no', 'basis', 'for', 'a', 'system', 'of',
'government', '.', 'Supreme', 'executive', 'power', 'derives', 'from', 'a',
'mandate', 'from', 'the', 'mass', ',', 'not', 'from', 'some', 'farcical',
'aquatic', 'ceremony', '.']
```

- Reduces words in WordNet to their roots
- women to woman
- Misses “lying” because it doesn’t know it’s looking for verbs right now

MBLEM

- “Memory-Based Lemmatizer”
- Based on TIMBL, a framework for trainable classifiers
- Decision tree based on the end of the word
- We ported it to Python in `simplenlp`

Oh shit, here comes the Internet

- Let's get some real world text from Twitter
- Have or set up a Twitter account
- At your command prompt:

```
easy_install tweetstream
```

Unicode

- It’s 2012. You can’t pretend one byte = one character anymore.
- Python can deal with Unicode, if you know what to ask it to do...

Assignment

- Find a trending topic on Twitter that's in a language you don't know
  - Note: this probably means you need to sample text on different days!
- What does it mean?
- What are the interesting words people are using along with that topic?
  - Probability distributions will help you here