Cheatography

Natural Language Processing with Python & nltk Cheat Sheet

by RJ Murray (murenei) via cheatography.com/58736/cs/15485/

Handling Text

`text='Some words'`	assign string
`list(text)`	Split text into character tokens
`set(text)`	Unique tokens
`len(text)`	Number of characters

Accessing corpora and lexical resources

`from nltk.corpus import brown`	import CorpusReader object
`brown.words(text_id)`	Returns pretokenised document as list of words
`brown.fileids()`	Lists docs in Brown corpus
`brown.categories()`	Lists categories in Brown corpus

Tokenization

`text.split(" ")`	Split by space
`nltk.word_tokenizer(text)`	nltk in-built word tokenizer
`nltk.sent_tokenize(doc)`	nltk in-built sentence tokenizer

Lemmatization & Stemming

`input="List listed lists listing listings"`	Different suffixes
`words=input.lower().split(' ')`	Normalize (lower-case) words
`porter=nltk.PorterStemmer`	Initialise Stemmer
`[porter.stem(t) for t in words]`	Create list of stems
`WNL=nltk.WordNetLemmatizer()`	Initialise WordNet lemmatizer
`[WNL.lemmatize(t) for t in words]`	Use the lemmatizer

Part of Speech (POS) Tagging

`nltk.help.upenn_tagset('MD')`	Lookup definition for a POS tag
`nltk.pos_tag(words)`	nltk in-built POS tagger
	<use an alternative tagger to illustrate ambiguity>

Sentence Parsing

`g=nltk.data.load('grammar.cfg')`	Load a grammar from a file
`g=nltk.CFG.fromstring("""...""")`	Manually define grammar
`parser=nltk.ChartParser(g)`	Create a parser out of the grammar
`trees=parser.parse_all(text)`	
`for tree in trees: ... print tree`	
`from nltk.corpus import treebank`	
`treebank.parsed_sents('wsj_0001.mrg')`	Treebank parsed sentences

Text Classification

`from sklearn.feature_extraction.text import Co ectorizer`	
`vect=CountVectorizer().fit(X_train)`	Fit bag
`vect.get_feature_names()`	Get fe
`vect.transform(X_train)`	Conve

Entity Recognition (Chunking/Chinking)

`g="NP: {<DT>?<JJ>*<NN>}"`	Regex chunk grammar
`cp=nltk.RegexpParser(g)`	Parse grammar
`ch=cp.parse(pos_sent)`	Parse tagged sent. using grammar
`print(ch)`	Show chunks
`ch.draw()`	Show chunks in IOB tree
`cp.evaluate(test_sents)`	Evaluate against test doc
`sents=nltk.corpus.treebank.tagged_sents()`	
`print(nltk.ne_chunk(sent))`	Print chunk tree

RegEx with Pandas & Named Groups

`df=pd.DataFrame(time_sents, columns=['text'])`
`df['text'].str.split().str.len()`
`df['text'].str.contains('word')`
`df['text'].str.count(r'\d')`
`df['text'].str.findall(r'\d')`
`df['text'].str.replace(r'\w+day\b', '???')`
`df['text'].str.replace(r'(\w)', lambda x: x.groups()[0][:3])`
`df['text'].str.extract(r'(\d?\d):(\d\d)')`
`df['text'].str.extractall(r'((\d?\d):(\d\d) ?([ap]m))')`
`df['text'].str.extractall(r'(?P<digits>\d)')`