Homework 10:
Web Crawling. POS Tagging

Benjamin Roth, Marina Sedinkina
Symbolische Programmiersprache

Due: Thursday January 25, 2017, 16:00

In this exercise you will:

- process any text from the given URL
- test different stemming and lemmatizing approaches
- find homographs within a text

Exercise 1: Raw Text Processing [7 points]

This homework will be graded using unit tests by running: `python3 -m unittest -v hw10_crawling/test_raw.py`

Implement following methods that can process any text from the given URL:

- `get_text(url)` - returns a list of clean paragraphs (no HTML markup) from the given URL. Find all paragraphs in the article, extract the text of each paragraph and save it in a list. `Hint:` Use `urllib` and `BeautifulSoup`.
- `make_text(raw)` - should tokenize the text and return an NLTK text object (`nltk.Text`).
- `lower(text)` - takes as input an NLTK text object "text" and returns list of lower case tokens
- `content(tokens)` - filters list of tokens by removing stopwords
- `stem_porter(tokens)` - takes as input list of tokens and returns list of stemmed tokens (use PorterStemmer)
- `stem_lancasters(tokens)` - takes as input list of tokens and returns list of stemmed tokens (use Lancaster Stemmer)
- `lemmatize(tokens)` - takes as input list of tokens and returns list of lemmas (use WordNetLemmatizer)

Exercise 2: Homographs [4 points]

This homework will be graded using unit tests by running: `python3 -m unittest -v hw10_crawling/test_homographs.py`

Use NLTK to find all homographs within a given text. We use the following definition: Distinct words that have the same written form are called homographs. In other words, homographs are words with the same spelling and different POS.

- Implement a function `fill_dict.py` that stores mapping between words and all the possible POS of this word. `Hint:` use `defaultdict`, a subclass of the built-in `dict` class. Setting `defaultdict` to set makes the `defaultdict` useful for building a dictionary of sets:

```
>>> s = [('red', 1), ('blue', 2), ('red', 3), ('blue', 4),
         ('red', 1), ('blue', 4)]
>>> d = defaultdict(set)
>>> for k, v in s:
...     d[k].add(v)
...
>>> d.items()
[('blue', set([2, 4])), ('red', set([1, 3]))]
```

- Complete a function `filter_dict.py` that deletes an entry from the dictionary, if this entry is not a homograph.

- Complete a function `filter_dict_nominalization.py` that deletes an entry from the dictionary, if this entry is not a homograph of nominalization or denominalization, restricted to the cases where a verb is used as a noun or the other way round (e.g. "to conflict" vs. "conflict")

- Implement a function `find_homographs.py` that returns a dictionary to a given tagged text, which holds homographs. Use already implemented methods.