Exercises

1. In the W3C Date Time Format, dates are represented like this: 2009-12-31. Replace the ? in the following Python code with a regular expression, in order to convert the string '2009-12-31' to a list of integers [2009, 12, 31]:

[int(n) for n in re.findall(?, '2009-12-31')]

1. Are you able to write a regular expression to tokenize text in such a way that the word don't is tokenized into do and n't? Explain why this regular expression won't work: «n't|\w+».
2. Try to write code to convert text into hAck3r, using regular expressions and substitution, where e → 3, i → 1, o → 0, l → |, s → 5, . → 5w33t!, ate → 8. Normalize the text to lowercase before converting it. Add more substitutions of your own. Now try to map s to two different values: $ for word-initial s, and 5 for word-internal s.

Exercises

1. Write code to access a favourite webpage and extract some text from it. For example, access a weather site and extract the forecast top temperature for your town or city today.
2. Write a function unknown() that takes a URL as its argument, and returns a list of unknown words that occur on that webpage. In order to do this, extract all substrings consisting of lowercase letters (using re.findall()) and remove any items from this set that occur in the Words Corpus (nltk.corpus.words). Try to categorize these words manually and discuss your findings.

Exercises

1. For the following cases, use the Porter Stemmer to normalize some tokenized text, calling the stemmer on each word. Do the same thing with the Lancaster Stemmer and see if you observe any differences.
   1. Capture the text that a user inputs.
   2. Read the text from a local file.
   3. Read the book “King Edward III by Shakespeare” from free online books in Project Gutenberg
   4. Read the html format of “CHAPTER VII. THE HOUSE IN SOHO” of the book “THE SECRET ADVERSARY by Agatha Christie” from free online books in Project Gutenberg

Exercises

1. Word features can be very useful for performing document classification, since the words that appear in a document give a strong indication about what its semantic content is. However, many words occur very infrequently, and some of the most informative words in a document may never have occurred in our training data. One solution is to make use of a lexicon, which describes how different words relate to one another. Using WordNet lexicon, augment the movie review document classifier presented in this tutorial to use features that generalize the words that appear in a document, making it more likely that they will match words found in the training data.
2. Using the movie review document classifier discussed in this tutorial, generate a list of the 30 features that the classifier finds to be most informative. Can you explain why these particular features are informative? Do you find any of them surprising?

Exercises

1. Use any features you can think to build the best name gender classifier you can. Begin by splitting the Names Corpus into three subsets: 500 words for the test set, 500 words for the dev-test set, and the remaining 6900 words for the training set. Then, starting with the example name gender classifier, make incremental improvements. Use the dev-test set to check your progress. Once you are satisfied with your classifier, check its final performance on the test set. How does the performance on the test set compare to the performance on the dev-test set? Is this what you'd expect?
2. The synonyms strong and powerful pattern differently (try combining them with chip and sales). What features are relevant in this distinction? Build a classifier that predicts when each word should be used.

Exercises

1. Search the web for "spoof newspaper headlines", to find such gems as: British Left Waffles on Falkland Islands, and Juvenile Court to Try Shooting Defendant. Manually tag these headlines to see if knowledge of the part-of-speech tags removes the ambiguity.
2. Tokenize and tag the following sentence: They wind back the clock, while we chase after the wind. What different pronunciations and parts of speech are involved?
3. Write programs to process the Brown Corpus and find answers to the following questions:
   1. Which nouns are more common in their plural form, rather than their singular form? (Only consider regular plurals, formed with the -s suffix.)
   2. Which word has the greatest number of distinct tags? What are they, and what do they represent?
   3. List tags in order of decreasing frequency. What do the 20 most frequent tags represent?
   4. Which tags are nouns most commonly found after? What do these tags represent?