NLTK classification

Joanna

April 2021

1 Ex 1

Work with the tutorial: Learning to Classify Text http://www.nltk.org/book/ch06.html part 1,2 and 3. Do NOT just blindly copy and paste the code from the book

All examples should be tested and written to single .py file. Submit the file with your comments.

2 Ex 2

Work with the movie review corpus which is appropriate for sentiment analysis.

1. Check the corpus, size, how many reviews there are, how many of them are positive/negative. (If you think of any other information you can obtain, please do, use examples from previous classes.)
2. Use a short review "Mr. Matt Damon was outstanding, fantastic, excellent, wonderfully subtle, superb, terrific, and memorable in his portrayal of Mulan." See how the classifier classifies this short and fake movie review.
   (a) Define a variable storing the review.
   (b) lowercase the text
   (c) tokenize the text
   (d) generate word feature dictionary using document_features function
   (e) Classify (classifier.classify) prepared features - save results
   (f) Calculate and save the classification probability to class 'pos' (classifier.prob_classify).
   (g) Calculate and save the classification probability to class 'neg' (classifier.prob_classify).
3. Change "Matt Damon" to "Steven Seagal". See how the classifier classifies this short and fake movie review. Compare with previous results. Save and comment on them.

4. Try short review "Mr. Matt Damon was outstanding, fantastic.". Save and compare results - comment on them. Hint: the classifier learn from 2,000 presence/absence word features.

5. Using the movie review document classifier generate a list of the 30 features that the classifier finds to be most informative. Please explain why these particular features are informative? Are they surprising?

Submit the code file including also your comments.