ML & NLP Week 10

MSc Artificial Intelligence and Data Analytics, UOM

Sentiment Analysis with Naive Bayes Classifier

Import Data :

The movies reviews corpus is splitted in half into two subsets one with positive and one with the negative reviews, each of them contains 1000 records.

Read & Preprocess the Data :

For each category of review keep a list with all the words appeared --*pos_reviews*-- and --*neg_reviews*--. Then we clean the two lists by removing all stopwords and the punctuation and finally keep only the unique words from the initial lists.

Feature Extraction:

I used two type of features one is using uni-grams and the other is using bi-grams.
By applying the above processing in previous step we create a bag_of_words type of features referred also as uni-gram kind of features --*see **sentiment_analysis.py** at lines:**77-91***--.
Then I created a bi-gram type of features/input data by removing a different kind of stop-words because in bi-grams some words (adjectives,adverbs, etc) play important role --*see **sentiment_analysis.py** at lines:**92-96***--. Comment out one section at a time to test the difference of the results.

Train & Test set Split:

Firstly we shuffle the two category's vocabulary/list and get ***20%*** of total words as test set by getting the first *200* words from each type of review/category vocabulary. Then keep the rest of the vocabularies for training. This creates a Test_set size equals to 400 and a Train_set of 1600 words.

Classifier:

Classifying the train data using a Naive Bayes Classifier model and then calculate the accuracy on test data. For 10 runs using uni-grams bag_of_words, I got a mean accuracy of **71.9 %**. While for 10 runs using bi-grams bag_of_words I got a mean accuracy of **81.2 %**. Noticing that by using a set of words the Naive Bayes classifier achieves a 10% gain on accuracy. Also printed the *15 words* that contribute the most in the results by showing the likelihood matrix. This matrix says how many times each word appeared in negative context and how many on positive. So, the likelihood matrix helps to see how a word can helps to classify similar words into one of the two types.

Resources:

Movies Reviews Corpus info

Corpus LazyLoad info

NLP with NB example

BagOfWords

Author:

Tassos Karageorgiadis

| December 2020 |