Web and Text Analytics

Course Project 3: Helpfulness Score Prediction

Background

You will be provided with data collected from Amazon reviews. These data will be in JSON format and contain several fields, including the review text and the helpfulness score.

You aim will be to develop to train a machine learning algorithm on these reviews, and to use the trained model to predict the helpfulness scores of new reviews.

Data

The data will be made available to you on request. An overview is available at https://snap.stanford.edu/data/web-Amazon.html.

Requirements

Cleaning & Preprocessing (5 marks)

You are required to clean the data as follows:

- Remove stopwords
- Remove punctuation symbols (these are not useful as we will adopt a bag-of-words models)
- Convert to lower-case

You will also need to:

- Discard reviews without helpfulness scores
- Convert the scores into a suitable numeric representation. For e.g., if the helpfulness score in the data is given as [2, 3], then it should be converted to 0.67 (i.e. 2/3)

You may need to perform other cleaning and pre-processing activities.

Feature Selection (10 marks)

- Compute the relevancy of words using tf-idf
- Select only those words whose relevancy score > a user-defined threshold. To select a threshold:
    - View the words and their respective tf-idf scores (sorted in ascending/descending order of the score)
    - Find out the threshold via inspection.

Training & Predictions (10 marks)

- Split your dataset into a training and a test set (and a validation set if you want)
- Train a linear regression (LR) and a support vector regression (SVR) on the reviews in the training set
- Estimate the errors (e.g. MSE).
- Use both methods (LR, SVR) to estimate the scores of reviews in the test set.
- Which methods give the best performance?

Extra (10 marks)

- Incorporate the number of words as an additional feature in your regressions.
- Retrain your models with this new feature set and estimate its performance on your testing reviews, as described above.
- Consider the problem as one of classification, such that reviews with a score of >0.6 belong to the class 0 (or positive) and all other reviews belong to the class 1. (It's recommended that you create a new training set, with reviews and their corresponding classes).
- Train a RandomForest classifier (or any other classifier of your choice) on the training data again.
- Generate the confusion matrix using cross-validation.
- Apply your classifier on the reviews in the test set and comment on the results

Deliverables & Deadline

- Short report (2-3 pages), describing the entire framework:
    - Tools used for the various steps above
    - Results:
        - Intermediate results, e.g. tf-idf scores of 10 words, chi-square scores of 10 words
        - Final results: confusion matrices and performance scores
- Source code
- Submissions to be made via lol@. Zip the report and the source codes together (or give the github link in your report).

Deadline: Thurs. 29th Nov. , midnight.

Teams will present their work in my office during the lecture hours of the 30th Nov. Details will follow.