no neural nets

PYTHON-3 TO BE USED

Natural Language Process: COMS4705

Kathy McKeown, Fall 2017

Due: Sept. 26th, 2017, midnight

Just use count based vectors

Homework 1: Republican or Democrat? (100 points)

Please post any questions concerning this assignment to Piazza via Courseworks (*http://courseworks.columbia.edu*), under the Homework 1 topic.

1 General Instructions

The main goal of Homework 1 is to build a classifier that will predict whether a person is a Democrat or a Republican depending on their Twitter posts. Your job is to see how accurate a classifier you can build, depending on the machine learning approach and on the various features you will be asked to implement.

As data, you are given a set of tweets labeled as democrat or republican. These tweets were posted during the 2014 midterm election and were self-labeled using a website called WeFollow where people post under either democrat or republican (this website is no longer live). The data files provided to you are formatted with one tweet per line, with one line containing the tweet text, a tab, and the label text (i.e. "democrat" or "republican").

We have divided the data into training (40k tweets), development (5k) and test (5k) sets, available on the course website. The training and development sets are available to you, but we will evaluate you on our reserved test set. Therefore, we recommend testing your trained model on the development set to ensure that it generalizes well.

Programming portion

You will use sci-kit learn to do the homework. You are to experiment with SVMs, Naïve Bayes, and logistic regression. You should build three models containing 1. unigrams, 2. bigrams and 3. trigrams. In addition, you should try a fourth model that is the best of the three n-gram models (or any combination of the three n-gram models) but also tests emojis as well as at least three types of style features that are typically seen on Twitter (e.g., multiple exclamation points, word lengthening ("soooooooooo cool"), all caps ("AWESOME!"). You could use feature selection on the fourth model to get the highest accuracy.

For each model, you may select the model (SVM, Naïve Bayes, logistic regression) that produces the best accuracy. If this model has tunable parameters, you should tune them to perform well on the development set (you may tune parameters you find in the scikit-learn documentation; some examples are kernel and C for SVM; and penalty, C, and fit_intercept for logistic regression).

You must write the code yourself. Don't use publicly available code (please refer to the Academic Integrity policy if you have questions in this regard). You can look at and modify the text classification example in sci-kit learn called document_classification_20newsgroups.py. Your system should show the accuracy, top twenty features, and a contingency table for each model.

Your submission must include a README file as specified in Section 2.2 below. Also include code for your program. See section 2.1 for details on the deliverables.

Written portion

1. Provide a written description of your best model and a justification of the features that you used or did not use. Why do you think they are helpful or not?
2. You are to do an error analysis. For your best performing model, select five interesting examples that did not work. Indicate why you think they were classified incorrectly. If you could add additional features, what features do you think would help improve performance?

2 Grading

You will be graded on the following elements:

2.1 Classifier (60 points total)

Your deliverables are the following (50 points):

- A **classify.py** file that takes the train and test files as command line arguments (e.g., can be run as 'python classify.py train.txt test.txt'), trains and saves your best-scoring model to a file, and tests that model on the test set and prints its accuracy (10 points)
- An **analyze.py** file that takes a trained model and the test file as command line arguments (e.g., can be run as 'python analyze.py model.pkl test.txt'), and prints the top 20 features and a contingency table to the console (15 points)
- A wrapper, **hw1.py**, that runs both files to print the following to the console: the accuracy of your best-trained model, and the accuracy, top 20 features, and contingency table for each required model (20 points)
- **model.pkl**, the best-scoring model saved with pickle (5 points)
- Your **README**, described below (2.3)

Accuracy (10 points)

Your system will receive up to 10 points depending on whether the best model produced accuracy above or below a threshold we will specify. Expect this threshold to be released Monday (September 18th). You will receive a full 10 points if above this threshold, and will lose 2 points for each 2 points loss in accuracy.

2.2 Written Answers (30 points)

Justification of best performing model: the description is complete and provides information about why this model (and the features it uses or doesn't use) was the best. (20 points)

Error analysis: Selected tweets should be different and the explanation should clearly explain why the features failed to predict the correct class. (10 points)

Your deliverables are the following:

- A **written.pdf** file containing your name, email address, the homework number, and your answers for the written portion.

2.3 Software Engineering (includes documentation) (10 points)

Your README file must include the following:

- Your name and email address.
- Homework number.
- Information on how to train and test your classifier.
- A description of special features (or limitations) of your classifier.

Within Code Documentation:

- Code should be documented in a meaningful way. This can mean expressive function/variable names as well as commenting.
- Informative method/function/variable names.
- Efficient implementation.

3 Submission instructions

Push the software (including README) to your class Bitbucket (which you created in HW0) under a hw1/ directory by the deadline. Put the written assignment on CourseWorks.

4 Academic Integrity

Copying or paraphrasing someone's work (code included), or permitting your own work to be copied or paraphrased, even if only in part, is not allowed, and will result in an automatic grade of 0 for the entire assignment or exam in which the copying or paraphrasing was done. Your grade should reflect your own work. If you believe you are going to have trouble completing an assignment, please talk to the instructor or TA in advance of the due date.