Lab 3: Sentiment Classification

This assignment has several different parts and it is due in two weeks. The necessary data files are available in Canvas.

For this lab you will work with a set of restaurant review data from Yelp.[1] The task is focused on building and evaluating binary classifiers using the training data and to then use a trained model to make predictions about whether reviews express primarily positive or negative sentiment. You have flexibility in choosing which methods you use for the task, but you will be asked to train a model, make predictions, and try to improve the accuracy of predictions. Our objective is for you to gain experience using supervised learning tools for classification tasks.

Data

The files you need to use are available as tab separated value (TSV) files available in a zip file (lab4.zip) in Canvas. Each line of one of the data files consists of 3 tab-delimited fields described below. In addition to the review data, there is a sentiment lexicon which you are permitted to use if you wish. The reviews are encoded as UTF-8 and the data should be nearly if not entirely English. The data are split into three portions: ***train, dev,*** and ***test***. The train portion should be used to build your models. The dev partition is used for your experiments and evaluation. Finally, we ask you to submit results with the assignment for what you think is your best model on the test set. The splits are evenly balanced between positive and negative examples. There are 2,000 examples in each file; 1,000 of both positive and negative reviews.

Col.	Field	Description
1	Stars	Use 4 stars as the positive class and 2 stars as the negative class. Zero is used in the test set.
2	Docid	A unique hashed ID
3	Text	Review text

Examples

4 eI8cFUAXogFAeT7hpkJWEg We're always looking for non-chain restaurants by the Waterfront (they do exist...), and this has become a favorite. I've had the Hot Roast Beef & Fries and the Corned Beef on Rye, and both were delicious. Every beer I've had here has been ice cold, and the service has been strong too. It's a great place to watch a Pirates game too.

4 DhN7VzbcYL41wimoOEryrA Great service, great food at a reasonable price! The cheeseburger was as big as my head and delicious and the onion rings were just as good. Our waitress was friendly, attentive and quick to refill drinks. Can't wait to go back!

2 pwnZqW3BlhvtAP66n5CBqg Blah. Always excited to try new south side fair I headed in on Friday, the place was dead and the TV's were blaring. I ordered the pulled pork sandwich and waited about 30 minutes for my order and it was horrible. The pork was fatty and dry and they tried to disguise it by adding a gallon of BBQ sauce on top. The sauce itself was too sweet and the side of beans tasted like chili. Overall it was barely edible. The tables were sticky and needed a serious wipe down. Hopefully the service and food improves with age.

2 BIeDBg4MrEd1NwWRlFHLQQ Decent but terribly inconsistent food. I've had some great dishes and some terrible ones, I love chaat and 3 out of 4 times it was great, but once it was just a fried greasy mess (in a bad way, not in the good way it usually is.) Once the matar paneer was great, once it was oversalted and the peas were just plain bad. I don't know how they do it, but it's a coinflip between good food and an oversalted overcooked bowl. Either way, portions are generous.

After you download the files you will need to write a program that can process the TSV format and create feature vectors. You will then build at least two classifiers from the training set. You can use any classification method (*e.g.,* Naïve Bayes, decision trees, SVMs, regular expressions, language models *etc*...), but at least one method you try should be bag-of-word based (*i.e.,* words, stems, or character n-grams should be the principal features). You are strongly encouraged to work with open source machine learning toolkits such as NLTK, sklearn, spaCy, SVMlight, HuggingFace, etc...

[1] This is real user-generated content. Though we attempted to remove it, there may be vulgar language or objectionable content.

Tasks

(a) **Study the training data.** Examine train.tsv. Do you see indications of positive or negative sentiment (e.g., words like 'good' or 'terrible'). Are reviews balanced or polarized?

- 4 points. Identify some useful features for positive or negative sentiment. Give 10 example features and the relative frequency you observe in both classes (e.g., 'good', 33.4% (pos), 7.6% (neg).)
- 2 points. Report any other observations about the data that are helpful for this task.

(b) **Train a classifier**. Using the training partition build a supervised model using a learning model of your choice. We suggest a simple bag of words model as a baseline feature representation. Then make predictions for the dev and test partitions and write those to a file.

- 4 points. Describe your approach. Include details such as the algorithm used, important parameters, the type of features, and the total number of features.
- 2 points. Print out a feature representation for the first document in the dev set. (Skip 'zeros'.)
- 4 points. Print ***docid [tab] prediction*** for the first 10 documents in the dev file. Prediction should be 2 or 4

(c) **Evaluate your predictions**. Using your predictions from part (b), compute precision, recall and F_1 scores for just the positive class over the full dev set. Show the work in your computation (*i.e.,* the numerators and denominators for precision and recall). **R**ecall is the percentage of positive predictions (*i.e.,* 4 stars) in the test file that are correctly predicted to belong to the positive class. **P**recision is the percentage of positive predictions in the predictions file which are indeed correct according to the test file labels. For reference: $F_1 = 2*P*R/(P+R)$.

- 4 points. Calculate and report Precision, Recall, and F_1 scores.
- 4 points. Reasonable results (e.g., F_1 score > 65). Around 80% is probably not very hard to attain.
- 4 points. Find a few mistakes that the classifier makes. Present a couple of incorrect predictions that you find interesting along with a short comment.

(d) **Build a second classifier**. Repeat steps (b) and (c) using a *different* machine learning algorithm[2].

- 2 points. Provide the same information requested for step (b)
- 2 points. Evaluate and provide the same information from step (c)
- 2 points. Briefly compare your results between the two classifiers.

(e) **Feature engineering.** For one of your classifiers explore using additional features beyond bags of words to improve performance. For example, you could use an English sentiment dictionary such as SentiWordNet which is included with the lab data. That file requires some pre-processing to extract sentiment terms; I've heard that SentiWordNet might be available in NLTK. Some other ideas would be to look at word bigrams (*e.g.,* "delicious food", "horrible service"), specifically handling negation (see J&M chapter 4), use of punctuation (!!!!), or emoticons :-).

- 3 points. Describe any features you added and the effect of using them on the dev set.
- 3 points. In Canvas submit predictions for the test.tsv file using what you think is your best model. The file should be named YOURJHED.txt. The format should be ***docid [tab] prediction***. Prediction should be either 2 or 4

There are 40 points for this lab. Please include your name prominently on the first page. You should submit a single PDF that contains the requested output for the tasks above and your source code. You should also submit in Canvas a single YOURJHED.txt file with your predictions for the test partition.

[2] You can try more than one algorithm, but the requirement is to use at least one additional algorithm. And if you skip this part, it's only 15% of the assignment.