INLS 613 - Text Mining
Assignment 2
Due: Wednesday, September 24, 2020 by 11:55pm via Sakai

1 Objective

The goal of this homework is to give you exposure to the practice of training and testing a machine-learning model for predictive analysis of text. You will use a toolkit called LightSIDE. As in HW1, the task is to classify movie reviews into *positive* and *negative* sentiment. You will be training a Naive Bayes classifier to predict whether a movie review expresses a positive or negative opinion. The *primary* goal of the assignment is for you to learn to experiment with different feature representations and do error analysis, and to review some concepts we've covered in class.

2 Software Details and Data

LightSIDE is a text-mining toolkit built by Elijah Mayfield at Carnegie Mellon University. You are expected to learn how to use it as part of this exercise. The user's manual is a great place to start![1] You can download and install LightSIDE on your own computers.

The training and test sets for this assignment can also be found inside the `/LightSIDE/data/hw2` directory. These are smaller versions than the datasets used for Assignment #1.

3 LightSIDE Exercises

Complete the following exercises. Please provide your answers using complete paragraphs. Make sure you use a Naive Bayes classifier (the default classifier) and, with the exception of Exercise #5 below, make sure you use unigram features and select the "Skip stopwords in N-Grams" option.[2]

1. Generate a feature table using unigram features. LightSIDE provides several metrics that measure the degree of co-occurrence between a feature and a target class value (in our case, *positive* and *negative*). This and the next question focus on precision. Suppose you have the following contingency table:

	positive	negative
word w occurs	a	b
word w does not occur	c	d

   The precision of term w with respect to the *positive* class is given by $\frac{a}{(a+b)}$, and the precision of term w with respect to the *negative* class is given by $\frac{b}{(a+b)}$.

   Sort the feature table by descending order of precision with respect to the positive class. You can do this by selecting *positive* in the "Evaluations to Display" tab and then double-clicking on the header labeled "precision". This functionality is illustrated in Figure 1.a.

---

[1]http://ankara.lti.cs.cmu.edu/side/LightSide_Researchers_Manual.pdf
[2]This option tells LightSIDE to remove stopwords from your unigram feature representation.

At the top of the list, you will find some terms you might expect. For example, "breathtaking" has perfect precision with respect to the *positive* class. There are, however, a few surprises. For example, "ultimately" has perfect precision with respect to the *positive* class. Choose a feature from the top of the list that you did not expect to be perfectly correlated with a particular class value and look at the instances in train.csv where the term appears. List the term, its precision value, and provide an explanation for why the term has a high co-occurrence with the target class value. Is the term associated with some language phenomenon that you did not anticipate, or is it a statistical anomaly. **(20%)**

2. Now, look at the middle of the list. Here, you will find terms that have a precision of 0.5 for a given class value. In other words, these are terms that occur an equal number of times in *positive* and *negative* reviews. Again, here you will find some terms you might expect. For example, "zombies" occurs an equal number of times in *positive* and *negative* reviews. There are, however, a few surprises. For example, "depressing" (an arguably negative term) also appears an equal number of times in *positive* and *negative* reviews. Choose a feature from the middle of the list (with a precision value of about 0.50) that you did not expect to be *uncorrelated* with a particular class value and look at the instances in train.csv where the term appears. List the term, its precision value, and try to provide an explanation for why the term has a low co-occurrance with the target class value that you expected it to have a high co-occurence with. **(20%)**

3. Train a model on train.csv and then re-apply this model to train.csv (yes, the same file). How does the accuracy of the model on train.csv (trained on train.csv and tested on train.csv) compare to its performance on test.csv (trained on train.csv and tested on test.csv)? This functionality is illustrated in Figure 1.b.

   You'll notice that the model's accuracy on train.csv is not perfect. In other words, the model fails to correctly predict some instances it used to estimate its own model parameters! Use the "predict labels" functionality in LightSIDE (Figure 1.c) to output the model's predictions on train.csv. Select a false positive or false negative prediction, copy/paste it into your report, and provide an explanation for why you think the instance is particularly difficult. **(20%)**

4. LightSIDE allows you to set a threshold on the minimum number of training set instances that must contain a particular feature in order for that feature to make it into the feature representation (Figure 1.a).

   Let's call this threshold t. If we set $t = 1$, then every term in the training set makes it into the feature representation. If we set $t = 2$, then only terms that appear in *at least* 2 training set instances make into the feature representation. If we set $t = 5$, then only terms that appear in *at least* 5 training set instances make into the feature representation. You should convince yourself that greater values of t lead to fewer features.

   Fill out the table below with the accuracy numbers from training models with $t = 1, 2, 3, 4, 5$ and evaluating these models on the training set (second column) and on the test set (third column). Using LightSIDE, this will require you to construct five different *feature tables* in the 'Extract Features' tab and perform 10 train/test operations in the 'Build Models' tab.

   How does the accuracy of the classifier change for these different values of this threshold on the training set and on the test set? Write a short paragraph explaining what is happening and why. **(20%)**

threshold	training set accuracy	test set accuracy
1	?	?
2	?	?
3	?	?
4	?	?
5	?	?

5. Explore different feature representations in order to improve the accuracy of your Naive Bayes classifier on test.csv. You can generate different feature representations by selecting different types of features from the "configuration plugin" pane (Figure 1.a) and by adjusting the threshold parameter mentioned in the previous question (Figure 1.a) . Note that LightSIDE can be slow to learn a model and apply a model when the feature representation is large. Provide a couple of paragraphs description of what you tried, what worked, and what did not work.

The student that achieves the best accuracy (on test.csv) will be given ONE bonus point towards his/her final grade and, of course, fame and glory. **(20%)**

4 Submission

Submit all your answers in the form of a report. Please submit Word and PDF formats only. As in HW1, include your best accuracy for Exercise #5 in your report.

Find "train.csv" under data/hw directory

This "rare threshold" tells LightSIDE to only keep features that appear in at least n instances

Select "pos" or "neg" to evaluate the co-occurrence of features with respect to the pos/neg class according to these metrics

For all questions except for #5, select this configuration: unigrams and stopwords removed

Click here to reverse the order of features based on the metric selected in "Evaluations to Display" (in this case, precision).

(a) Feature Representation Interface

Here, you can select any of the feature representations constructed in the "Extract Features" tab.

Clicking "train" will automatically train a model using the data imported in the "Extract Features" tab and will test the model on the "Supplied Test Set"

Evaluation outcomes from train/test experiment

In this assignment, you will always supply a test set. In some cases, train.csv and in other cases test.csv.

(b) Model Building and Evaluation Interface

Choose the model you want to apply. You will be able to select models trained in the "Build Models" tab.

Import the file you want to make predictions for.

You can use this button to export this prediction data to a csv file (that you can open in excel)

Use this display to compare the prediction and true values.

(c) Prediction/Annotation Interface

4

Figure 1: LightSIDE Screen Shots