CSE427 – Final Project #2: Sentiment Analysis and Text Processing

M. Neumann

Due 05/05/2016 1pm (**no extension!**)

Project Goal

Sentiment Analysis tries to assess the *emotion* or *mood* in a text document. Essentially it can be computed by comparing the set of words in each document to an existing dictionary of positive words, negative words, and neutral words and by analyzing the most frequent positive or negative expressions. Throughout the course we collected your **emotional** descriptions for each homework assignment. Now, you will be able to use this data for sentiment analysis!

Goal: Explore and use the text (pre-)processing features in HIVE to assess whether the homework reviews homework review data gathered throughout the semester were positive or negative. You will also investigate how the homeworks on the different parts of the course were perceived. Finally, you will predict the emotions of your own reviews and compare them to what your true emotion (when writing the review) was. The average scores on this task will be graded as a **competition** among all groups doing this project (10% of the grade will depend on this).

Getting Started

Update your SVN repository, you will find additional materials for the *final project* in the folder `final_project/text`.

Download the homework review data from HERE:

```
https://classes.cec.wustl.edu/cse427/hw_reviews.zip
```

Indicating Group Work

Use the file `partners.txt` to indicate group work. Follow these instructions exactly, to ensure to get credit!

- `partners.txt` needs to include up to 3 wustlkeys in the first three lines (one line per wustlkey)
- **first line/wustlkey is the repository, where the solution is located**. We will **only** consider the submission in this repository!

- Every student in a group needs to have **the same** `partners.txt` in the `final_project/spark` folder in their repository (indicating that the partnerships are **mutually accepted**)!
- If you do not have a partner, try to find one! If you want to submit on your own, indicate your wusltkey in the first line of `partners.txt` and leave the second line blank.

Usage Agreement

By using the dataset of homework reviews gathered throughout the semester:

- I agree to **delete** this dataset once the project has been completed.
- I will not redistribute this data in any form.

Problem 1: Explore HIVE and Customer Review Data

Step 1: Data Management with HIVE

This problem defines **milestone 1** and prepares the data for Step 2.

Complete Lab 7. Find the instructions on the course webpage.

Step 2: Gaining Insight with Sentiment Analysis

In this problem you will explore the text processing features in HIVE to analyze customers' comments and product ratings. This warm-up exercise will introduce useful HIVE commands needed for the sentiment analysis tasks in the second part of the final project. This problem is part of **milestone 2**. You can find more information on HIVE's text processing features in the `TextProcessing_Hive.pdf` file provided in your SVN repository.

Customer ratings and feedback are great sources of information for both customers and retailers like *Dualcore*. However, customer comments are typically free-form text and must be handled differently. Fortunately, HIVE provides extensive support for text processing. Follow the instructions in `Problem1_Step2.pdf` provided in the project folder in your SVM repository.

Add a folder called `milestone2` to the `final_project/text` folder in your SVN repository. Submit your HiveQL statements in this folder. Use the file names given in the step by step instructions (you should have 4 `hql` files in total).
Add the new files/folders to your SVN repo before committing:

```
$ svn add milestone2
$ svn add milestone2/your_scripts
$ svn commit -m 'milestone 2 submission' .
```

Problem 2: Data Preparation

This problem prepares the homework review dataset and is part of **milestone 2**. Submit your programs for data pre-processing as submission of **milestone 2**. Detailed submission instructions

follow below.

Step 1: Filter out invalid reviews

The homework review data stored in the `hw_reviews` folder is organized as follows. Each review is stored in a txt file `hwx_review_id.txt` and all reviews for one homework are stored in a subfolder `hwx`. x is a placeholder for the homework number (1,...,8) and id is a placeholder for the review ID (1, ..., 96). These IDs are unique within homeworks but not across homeworks. Further, these are randomly shuffled number across homework, that is, `hw1_review_22.txt` and `hw2_review_22.txt` were not written by the same person.

Filter out all empty reviews and all reviews shorter than 50 words. You can delete these files as we will not need them anymore. Further, remove all non-word characters (especially newline and tab characters) from all files and put the homework number at the beginning of each file and separate it from the actual text using the tab character. Put a newline at the very end of each review. This step does not require HIVE, you can do that part in MAPREDUCE or any programming language of your choice.

Step 2: Text Pre-processing

Now, put the data into HIVE using two fields: `hw_number` as INT and `review` as STRING. One important part of text processing is **removing the stop words** so that they do not skew the results of your sentiment analysis. A feature of stop words is that they occur frequently in natural language text, but without adding sentiment, or meaning, to the text. Using HIVE pre-process your text data into all **lower-case letters** and remove the stop-words listed in the file `english.stop` provided in your SVN repository from the data set. You will need to store the stop words as HIVE table as well. This blog post might also be informative: `http://bigdatabloggin.blogspot.com/2012/08/trending-topics-in-hive-ngrams.html`.

Add a your HIVE scripts and one pre-processed example review named `prep_review.txt` to the `milestone2` folder. **Do NOT add any other data!**
Add the new files/folders to your SVN repo before committing:

```
$ svn add milestone2/your_scripts
$ svn add milestone2/prep_review.txt
$ svn commit -m 'milestone 2 submission' .
```

Problem 3: Sentiment Analysis on the Homework Review Data

Step 1: Basic Analysis

(a) Find out which homework assignment was perceived the most positive and which the most negative on average across all students. Use the `pos-words.txt` and `neg-words.txt` lists of words provided in your SVN repository.

(b) What positive and negative words are used most frequently? Give the 5 most frequently used words in each category. Do your results make sense? What is the frequency of those words?

Step 2: N-grams

(a) N-grams are a more informative way of analyzing text as they leverage the context or words. Write a HIVE script compute the n-grams of a given corpus of text documents. Both n and the path to the text documents should be provided as a parameter. Run your script on the homework review corpus using n=2 and n=3.

(b) Now, retrieve the 3 most positive 2- and 3-grams. Again use the lists of `positive.txt` and `negative.txt` words to determine if an n-gram is positive or negative. Do your results make sense? What is the frequency of those n-grams in the corpus?

Step 3: What are the Favorite Topics?

(a) Is there a difference between the emotions about the homework assignments in the first half of the semester (hw1-hw6) and the ones in the second half (hw7-hw9)?

(b) Is there a difference between the emotions about the homework assignments on theory (hw1 and hw2), MAPREDUCE (hw3-hw6, hw9) and the ones on PIG (hw7-hw8)? If so, which topic was liked the most?

Step 4: Does your system agree with your own emotions?

The accuracy of a sentiment analysis system is, in principle, how well it agrees with human judgments. However, according to research human raters typically agree 79% of the time. Thus, a 70% accurate program is doing nearly as well as humans, even though such accuracy may not sound impressive. If a program were "right" 100% of the time, humans would still disagree with it about 20% of the time, since they disagree that much about any answer.

(a) Get your own past reviews (all reviews of all group members) and label those as being positive or negative. (This is a manual process. Thus, you do not need to write a program! Just read the statements and assign them to either being positive or negative.)

As I would love to use this data in my research and also to analyze and improve the course, I ask you to store and submit this data in a txt file (one per group) called `review_labels.txt` with the following comma separated fields (again the grading of the project or any other grade in the course does **NOT** depend on your labels!!):

- wustlkey,
- hw_number,
- label

Use one line per label. Do not include any other information! Note that we will not use your personal information. The wustlkey will only be used to map your label to your review. **Please, let me know, if you do not want me to use your data in my research outside the course (so that I can delete your reviews).**

(b) Run your final program through **this set of your own reviews** and compare the resulting predictions with your labels by computing the average accuracy.

(c) Do the result agree with your original moods? If your program is right 70% of the time (i.e., your accuracy is above 70%), then congratulations, you have done a brilliant job! If your

accuracy is lower than that, discuss at least two improvements to your sentiment analysis approach (you may of course implement those and improve your result)! **We will grade this part as a competition!** 10% of the grade for the results will be assigned according to a ranking of the quality of all teams' predictions (i.e., your reported accuracy)!

Step 5: Documentation of Approach and Results (Report)

Write the project report documenting your sentiment analysis approach, your implementations, and the obtained results. This report should be readable for an informed outsider and it should not require the reader to look at or run any code.

Final Submission Instructions

Submit your report including **documentation**, as well as, **results** as `project_report.pdf` by adding it to the `final_project/text` folder in your SVN repository. Submit your implementations by adding your HIVE scripts to the `final_project/text/src` folder in your SVN repository. **Do NOT add any data!**

Add the new files/folders to your SVN repo before committing:

```
$ svn add src/*
$ svn add project_report.pdf
$ svn add review_labels.txt
$ svn commit -m 'final project submission' .
```