

## Project II: Sentimental Analysis

### Goal:

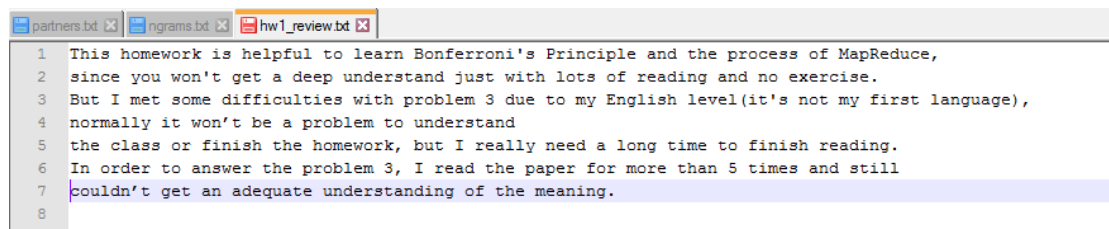
Sentimental analysis refers to the use of text analysis to identify and extract subjective information in source materials. It's widely applied to reviews and social media for a variety of applications.

In this project, we used Hive and java to process reviews for CSE427s homework to find out the sentimental information inside, e.g, which homework assignment got the most positive and which the most negative on average across all students, which positive word and negative word students used most frequently, which topic is the favorite?

### Target Description:

Students wrote reviews for CSE427s homework, theoretically there are 8 homework and 96 students, so we should have 768 reviews in total, but we only consider reviews with over 50 words valid for analysis, we have ??? Valid reviews.

A valid review example is as below:

A screenshot of a text editor window with three tabs: 'partners.txt', 'ngrams.txt', and 'hw1\_review.txt'. The 'hw1\_review.txt' tab is active, showing a review text with line numbers 1 through 8 on the left margin. The text of the review is: '1 This homework is helpful to learn Bonferroni's Principle and the process of MapReduce, 2 since you won't get a deep understand just with lots of reading and no exercise. 3 But I met some difficulties with problem 3 due to my English level(it's not my first language), 4 normally it won't be a problem to understand 5 the class or finish the homework, but I really need a long time to finish reading. 6 In order to answer the problem 3, I read the paper for more than 5 times and still 7 couldn't get an adequate understanding of the meaning. 8'.

### Approach Description:

#### Step 1: Filter out invalid reviews

We only consider reviews with over 50 words, so we need to filter out invalid reviews, we chose to use Hive in VM.

We created a table and load all reviews into it, we selected all the reviews with the condition: where the count(\*) is over 50, then we stored the selected reviews in a folder for the next step.

#### Step 2: Text pre-processing

We used hive to eliminate all the punctuations so that we can easily compare each word with the bags of words, beside normally a review has many words that are meaningless to analyze, e.g the word "a", "the" and "are". Removing those stop words from the beginning will speed up the whole project.

We locally wrote java code to realize the preprocessing part.

TextPre.java do works that add id number to a text file, and organize words in those text files, like: format all words to their lower case, etc.

RemoveStopWords.java removes all the stops words in the text file.

After finishing those scripts, we added them to hive as udf to use that function

```
hive> ADD JAR removestopwords.jar;
```

```
hive> CREATE TEMPORARY FUNCTION remove AS 'default1.Removestopwords';
```

Take hw1\_review\_5.txt as an example:

## **BEFORE PROGRESSING:**

Overall, this homework assignment was easy and straightforward for me. There was good variety in the questions and I enjoyed reading the required article. The second problem was frustrating for a while because I arrived at a correct answer but didn't know it due to rounding error. It didn't take too long and was overall an enjoyable process

## **AFTER PROGRESSING:**

1 homework assignment easy straightforward good variety questions enjoyed reading required article problem frustrating arrived correct answer didn't know it due to rounding error didn't take too long enjoyable process

### **Step 3: Basis analysis**

We used java with the script named "analysis1.java" to realize some requirements, which is provided in the svn repository.

Then, we added this java class to hive as udf.

By counting the positive and negative words in reviews we can get the information that which homework assignment got the most positive and which the most negative on average across all students.

Besides, we also counted the positive and negative words frequency and got the most frequently used words in each category.

### **Step 4: N-grams**

From the result of basis analysis, most words make sense, however, there are some words that do not make sense: pig and problem. Pig as an academic term for the software PIG and problem probably refer to the problem number in the homework. Besides, words occurred together can provide much more information than a single word, for example, we usually get nothing from "time", but sense negative sentiment from "long time".

To fix those problems, we need to combine the result of n-grams to do further analysis. Gladly, hive has a built-in NGRAM function.

We loaded all processed homework reviews into table reviews, and do 2-grams and 3-grams over the entire homework reviews, then we retrieved the 3 most positive and negative 2-grams and 3-grams by comparing with positive.txt and negative.txt. Since many n-grams are mealiness, e.g "la la", "problem 2", "ha ha ha" and "blood blood blood", we set to retrieve 25 top n-grams and determined the final results by ourselves.

In order to make the script more efficient, we passed the path and n as parameters in our command line.

From the result, we absolutely got more information using n-grams than basis analysis. For example, we got "lot time", "learned lot", "time consuming" and "assignment hard finish" from which we can easily sense the sentiment.

### Step 5: favorite topics

Since different homework are for different topics, by analyzing them we can find out information for each assignments, e.g the favorite topic and whether the emotion changed during the semester. Our approach for this step is quite like last one.

We created 2 tables to store the first half (hw1-hw6) reviews and the second half (hw7-hw8) reviews and created 2 other tables to store hw1-hw2 reviews and hw3-hw6 reviews, loaded data into the corresponding tables and did N-grams over records stored in these tables, then joined the n-gram results with positive.txt and negative.txt. Then counted the number of records.

The scripts are the same, the difference between this step with the last one is that we used several tables for different homework reviews in this step. And passing parameters from the command line makes the script convenient.

#### Code script

```
$hive> SET n=2;
$hive> SELECT EXPLODE(NGRAMS(SENTENCES(LOWER(review))),${hiveconf:n},25))
        AS bigrams
        FROM reviews;
$hive> SET n=3;
$hive> SELECT EXPLODE(NGRAMS(SENTENCES(LOWER(review))),${hiveconf:n},25))
        AS bigrams
        FROM reviews;
```

### Step 6: Does your system agree with your own emotions?

One way to verify the result is to test the result on some test examples. So we used our own reviews to test if the results from the previous steps are right.

- (a) In review\_label.txt
- (b) The result basically agrees.