Project Report
“Classify the sentiment of sentences from the Rotten Tomatoes dataset ”

Akash Bulbule – 800836626
Naveen Mysore – 800812071

Project Summary

The Sentiment analysis for rotten tomatoes dataset corpus aims at not only predicting the sentiment for the movie review but also learning the sentiment of new words,encountered in the new review not present in the training set. The advantage of learning the sentiments of unencountered words is better sentiment prediction.

Dataset used

We used the train.tsv dataset having rotten tomato reviews parsed through the stanford parser to break the reviews into set of partial sentences with sentiment attached to it as shown below.

train.tsv

PhraseId	SentenceId	Phrase	Sentiment
11	1	demonstrating the adage	2
12	1	demonstrating	2
13	1	the adage	2
14	1	the	2
15	1	adage	2

We used a user interface to enter the review and the output was the sentiment for the review and also the updated sentiment value for words not encountered in the trainset.

Computation Steps

The computation involved four distinct steps:

Preprocessing: The python script takes care of preprocessing the input entered by the user. We are using ntlk python library which is a natural language processing library for generating uni words and bi words for the user`s input. We will be using this words as inputs for our sentiment analysis.

1) Bucket Processing: We separated the training set data into 5 distinct buckets by running 5 iterations of MapReduce for each of the sentiment below:
   0-very negative
   1-negative
   2-neutral
   3-positive
   4-very positive
   Thus this helped segregate the training data into the 5 sentiment buckets.

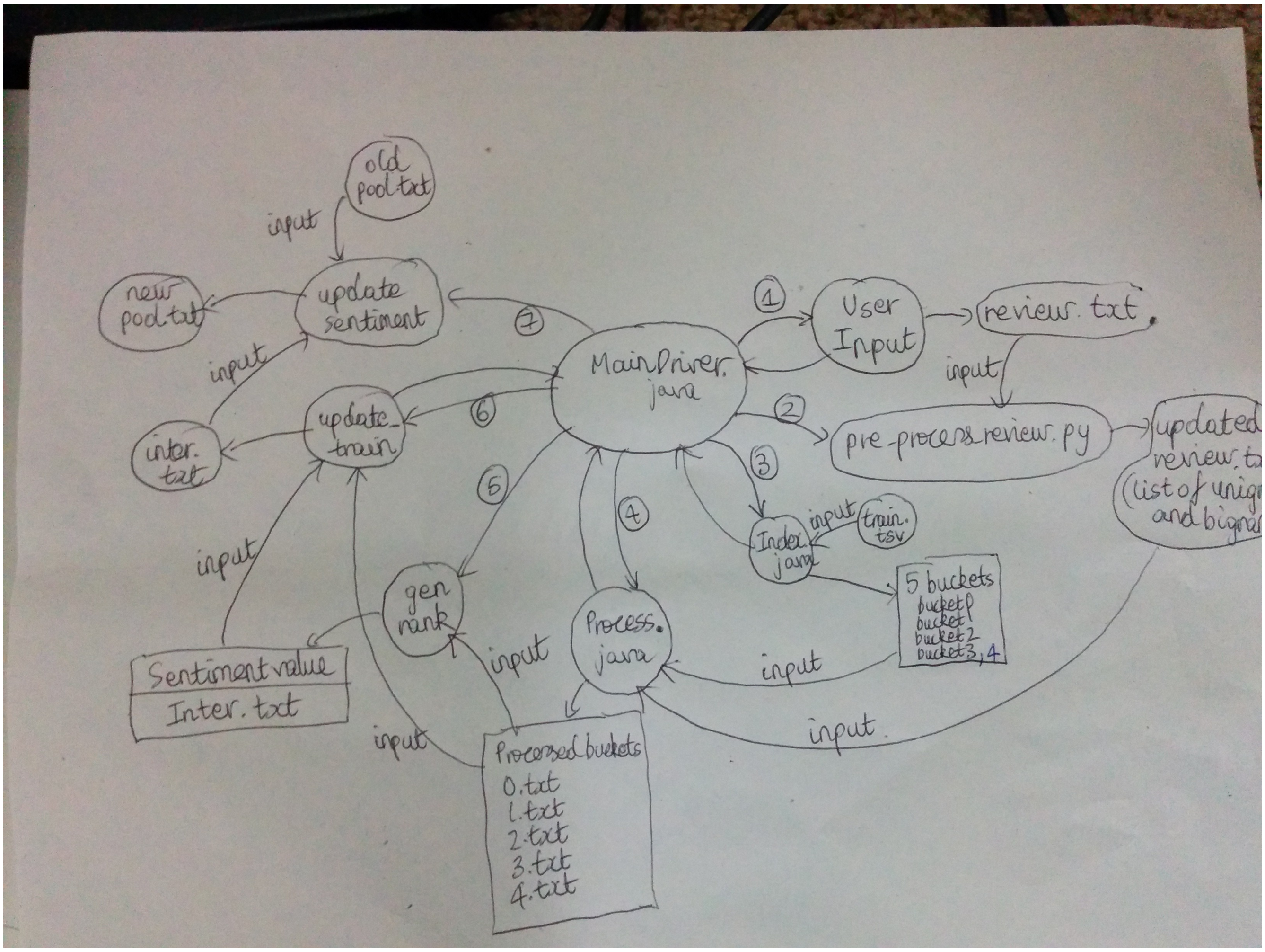
2) We took the processed review as input from the user and compared with each of the bucket using MapReduce and attached the sentiment values to those query unigrams and bigrams. We removed some of the unigrams which were already a part of bigrams
   for eg: unigrams “this” , “is” were replaced by bigram “this is” since the sentiment value is already attached to “this is”. We ran this compare step for 5 iterations comparing the unigrams,bigrams with each of the bucket. The result will be 0.txt,1.txt,2.txt,3.txt,4.txt in the prerank folder. unigrams, bigrams and corresponding sentiment value (0,1,2,3 or 4) attached to them in their respective files.
3) Sentiment Calculation: We calculate the sentiment value by running mapreduce over the files 0.txt,1.txt,2.txt,3.txt,4.txt,5.txt and calculating the sentiment value by adding up the sentiment values for the unigrams,bigrams and taking an average.We focused on unigrams,bigrams which added sentiment to the review (positive/negative) and ignored the ones which had a neutral sentiment. Thus this helped us focus our attention to words which added sentiment to the movie review and hence predict the sentiment of the review. Thus this completes the step of sentiment analysis for the movie review. (We have considered double values for sentiment like 3.67 and 2.7 which can be rounded to 4 or 3 to give the actually sentiment class)
4) We then retrieved a list of unigrams,bigrams which do not already have an assigned sentiment value in train.tsv. For eg: words like “Harry Potter” used by were not present in the train.tsv. And for reviews like “The movie was as great as a Harry Potter movie” it is difficult to determine the sentiment as only the word great adds value to it. It would be great if we could somehow learn the sentiment of “Harry Potter” which in this case should be 3 or 4 and add it to the dataset.
   Thus we used the last mapreduce to handle this “learning” process. We created two files inter.txt and pool.txt
   inter.txt – It is the new list of unigrams,bigrams unencountered in the train.tsv with the attached sentiment value to it which is retrieved from the sentiment value of the review in step 3.(All unigrams,bigrams are attached with the same sentiment value)
   pool.txt – it is the set of already updated unigrams,bigrams through previous iterations of step 3.

   We ran a MapReduce task using inter.txt and pool.txt as input.
   a) For unigrams,bigrams not in pool.txt we simply emit the unigram,bigram with the sentiment value.
   b) For unigrams,bigrams already in pool.txt we take the average of the sentiment value from inter.txt and pool.txt and emit it .
   c) We delete the old pool.txt and replace it with the output of this mapreduce and name this as pool.txt

With every run of the above 4 steps of the project new unigrams,bigrams with unassigned sentiments are added to pool.txt with the sentiment value as the movie review present sentiment while the sentiment value of the already present unigrams,bigrams gets updated and thus with severals iterations based on movie reviews we are able to get accurate sentiments for the unigrams,bigrams and further a better sentiment value for the subsequent movie reviews.

The Project thus helps learn sentiment values of words with better prediction of sentiment subsequently.

Architecture implement for analysis



We first promt the user to enter the review and we store the review in review.txt. We then execute the python script to clean up the input and extract bi words and uni words and save it back in review.txt in this format (this movie –> movie-->good-->is good). We now move on to bucketing process where index.java will take input from train.tsv and creates five buckets. We next execute process.java which takes input from the buckets generated before and arranges them in an order for sentiment calculation. process.java takes review.txt as an input along with buckets and creates a set of new bucket files (0.txt, 1.txt, 2.txt...) which becomes input for next stage. gen_rank.java will take these new bucket files and calculated the sentiment value. update_train.java creates a new file called inter.txt which will hold all the words which were new i.e., the words which the user entered but not present in train.tsv.

We now move on to training part. We take inputs from new bucket files and the list of words which are not in the train.tsv but was involved in the user`s review. We take sentiment value calculated and list of words in inter.txt as an input here. We now move on to update_sentiment.java which will update sentiment values for the words which are already present in pool.txt and for words which are not present in pool.txt we just enter the new words followed up current rank.

Input – Output sample

```
------------------------------
----- Sentiment Analysis -----
Scale: 0-very negative, 1-negative, 2-neutral, 3-positive, 4-very positive
Review: This movie is awesome. The comedy involved is very subtle and keeps the audience engaging.
Sentiment Value: 3.3333333333333335
------------------------------
Words and their contribution: {subtle and=3.0, engaging=4.0, is=2.0, audience=2.0, comedy=3.0, the
audience=2.0, movie=2.0, The comedy=3.0, the=2.0, .=2.0, This=2.0, subtle=3.0, and=2.0,
involved=2.0, This movie=2.0, engaging .=4.0, very=2.0, keeps=2.0}
Bi grams involved: [The comedy, This movie, engaging ., subtle and, the audience]
Uni grams involved: [., This, and, audience, comedy, engaging, involved, is, keeps, movie, subtle, the,
very]
words contributed: [subtle and, The comedy, engaging .]
------------------------------
```

Challenges Faced

The sentiment analysis itself or the 3rd step and 4th step were the challenging part of the project because we had compute the sentiments using the mapreduce framework and also determine unigrams,bigrams which weren't assigned any sentiment, determine the new sentiment values and update the pool of unigrams,bigrams in pool.txt

Goals Completed

As per our proposed plan we were able to predict the sentiment value for the review and also determine the sentiment value of the key phrases not present in the training set and dynamically update its sentiment value.

Contributions

Naveen and Akash both contributed to the project equally.
Akash was focused on the Step 1 and Step 2
Naveen was focused on Step 3 and Step 4. We both assisted and helped each other in all the modules of the project.

Conclusion

The Project helps not only determine the sentiment for the movie reviews but also improve its prediction in subsequent iterations.

References

Kaggle competition- https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews
The github links:https://github.com/akashbulbule/sentiment https://github.com/NAVEENMN/sentiment