MINOR PROJECT

Documentation

ON

PREDICTING THE SEMANTIC ORIENTATION OF COMMUNICATION MADE OVER SOCIAL NETWORKING

Submitted by: -

B Tech CSE – OSS (V Semester)

Harshal Mittal	500046288	R100215030
Arpit Ahlawat	500046661	R100215015
Mansi Sahu	500046553	R100215040
Shubhangi Srivastava	500044400	R100215071

Under the guidance of

Mr. P. Srikanth
(Assistant Professor, SoES,UPES)

UPES

School of Computer Science

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

Dehradun-248007

2017

Table of Contents

School of Computer Science and Engineering

University of Petroleum & Energy Studies, Dehradun- 248007

Project Proposal Approval Form (2017-18)

Minor I

Project Title

Predicting the Semantic Orientation of Communication Made over Social Networking

Abstract:

Developing a state-of-the-art sentiment analysis system that detects the sentiment of short informal textual messages such as tweets and SMS (message-level task), the sentiment of a word or a phrase within a message (term-level task). The system is based on a supervised statistical text classification approach leveraging a variety of surface form, semantic and sentiment features. The sentiment features are primarily derived from novel high-coverage tweet-specie sentiment lexicons. These lexicons are automatically generated from tweets with sentiment-word hashtags and emoticons. To adequately capture the sentiment of words in negated contexts, a separate sentiment lexicon is generated for negated words.

I. Introduction

Sentiment Analysis involves determining the evaluative nature of a piece of text. For example, a message or post review on social network can express a positive, negative, or neutral sentiment (or polarity).

Automatically identifying sentiment expressed in text has a number of applications, including tracking sentiment towards products, movies, politicians etc., improving customer relation models, detecting happiness and well-being, and improving automatic dialogue systems.

In our sentiment analysis system, we are utilizing three freely available, manually created, general-purpose sentiment lexicons. In addition, we are generating two high-coverage sentiment lexicons from about 2.5 million corpus using sentiment markers within them. These lexicons capture many peculiarities of the social media language. such as common intentional and unintentional miss-spellings (e.g., gr8, lovin, coul, holys**t),elongations (e.g., yesssss, mmmmmmm, uugghh), and abbreviations (e.g., lmao, wtf). They also include words that are not usually considered to be expressing sentiment, but that are often associated with positive/negative feelings (e.g., party, birthday, vulgar).

This Project describes a state-of-the-art sentiment analysis system addressing two tasks:

(a) Detecting the sentiment of short informal textual messages (message-level task).

(b) Detecting the sentiment of a word or a phrase within a message (term-level task).The system is based on a **supervised statistical text classification approach[1]** leveraging a variety of surface-form, semantic, and sentiment features. Given only limited amounts of training data, statistical sentiment analysis systems often benefit from the use of manually or automatically built sentiment lexicons. Sentiment lexicons are lists of words (and phrases) with prior associations to positive and negative sentiments. Some lexicons can additionally provide a sentiment score for a term to indicate its strength of evaluative intensity. Higher scores indicate greater intensity. For example, an entry great (positive, 1.2) states that the word great has positive polarity with the sentiment score of 1.2. An entry acceptable (positive, 0.1) specifies that the word acceptable has positive polarity and its intensity is lower than that of the word great.

II. Problem Statement

Due to undue advantage of mass outreach of social network, many offensive/vulgar data gets broadcasted.

III. Literature Review-

<u>Learning Word Vectors for Sentiment Analysis (2015) Andrew L. Maas[6]</u>

Read recently research paper to get guidance of work and algorithm used.

1. Naive Bayes' probability algorithm.
2. Word Vector space
3. High Balanced Tree (AVL).

<u>Thumbs up? Sentiment Classification using Machine Learning Techniques(2002) Bo Pang and Lillian Lee[6]</u>

Another research paper to learn about the method for making scoring for the sentiments in the large lexicon on social media.

Bag of words model

<u>Naive Bayes</u>:

One approach to text classification is to assign to a given document d the class

$c^* = \arg\max c\ P(c \mid d)$.

We derive the *Naive Baye's* (NB) classifier by first
observing that by Baye's' rule,

$$P(c \mid d) = P(c)P(d \mid c)/ P(d) ,$$

where $P(d)$ plays no role in selecting c^*.

<u>Sentiment lexicon Scoring</u>:

Sentiment Score (w) = PMI (w,positive) – PMI (w,negative)
PMI stands for pointwise mutual information

PMI (w,positive) = $\log_2$ (freq (w,postive)*N)/freq (w)* freq (positive)). ... (1)

Where freq (w,postive) is the number of times a term w occurs in positive lexicons, freq (w) is the total frequency of the term in the corpus, freq (postive) is the total number of tokens in positive lexicon and N is the total number of tokens in the corpus. PMI (w,negative) is calculated in a similar way thus equation 1 is simplified to

Sentiment score(w)= $\log_2$(freq(w,positive) * N/freq(w) * freq(positive))

Where freq(w,positive) is the number of times a term w occurs in positive lexicons, freq(w) is the total frequency of term w in the corpus, freq(positive) is the total number of tokens in positive lexicons and N is the total number iof tokens in the corpus .PMI(w,negative)is calculated in a similar way.Thus , equation 1 is simplified to:
Sentiment score (w) = $\log_2$(freq(w,positive)*freq(negative)/freq(w,negative)*freq(positive))

Since PMI is known to be a poor estimator of association for low-frequency events, we ignore terms that occurred less than five times in each (positive and negative) group of lexicons.

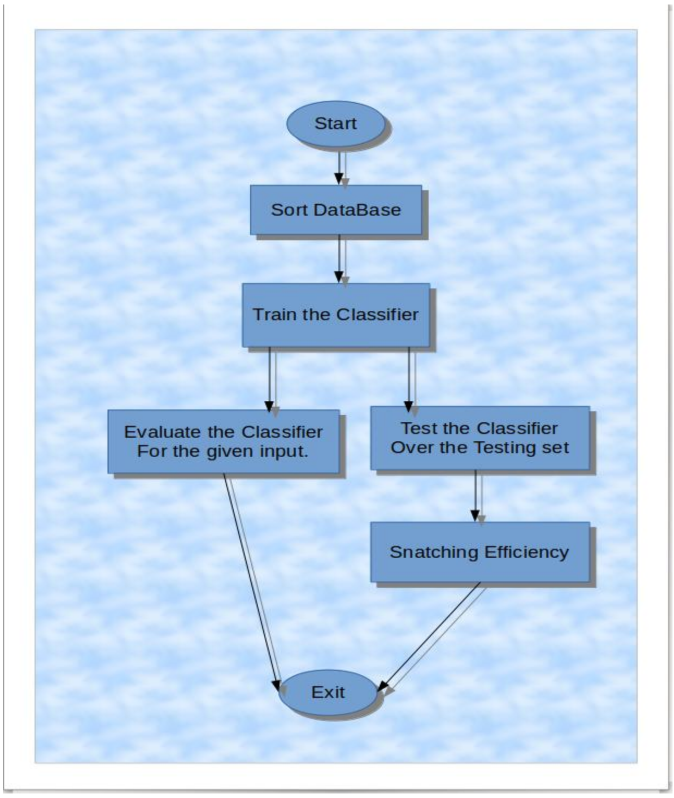
Ref .http://www.etenberg.org/ebooks/10681

IV. Objective

To represent and evaluate a miniature framework that retrieve semantic orientation information using data collected from large corpus in social networking.

V. Methodology

Main Idea is to build a Model so that we can analyse the semantic orientation of the social networks.
So achieve this, we:

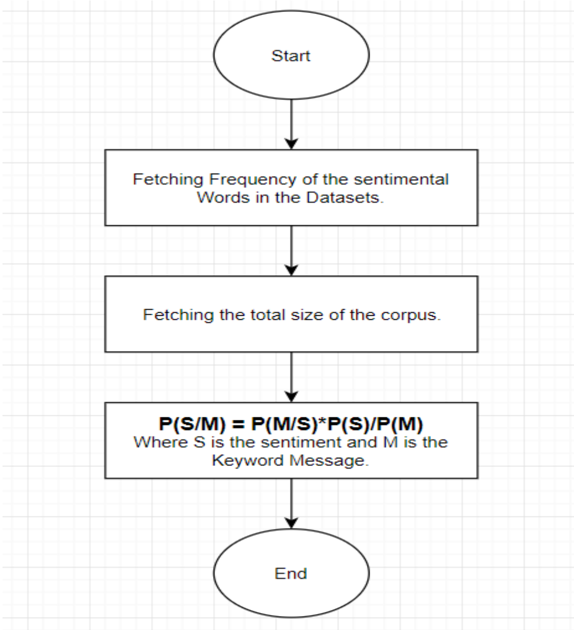


Overall Flow Chart

1. Collect the data from the large corpus in the social media.
2. Analyse the collection of data.
3. Then we are going to refine the data i.e. arranging and structuring the data collected with the help of some mathematical tools like Sorting and Arranging in AVL Tree by specialized method that is rotation.
4. Create classes of sentiment as follows
    a. Happy
    b. Sad
    c. Angry
    d. **Non-Decent (major focus).**
5. Algorithm for analysing the sentiment. There we'll use:
    a. Customised Naive Baye's algorithm.

    Let S be a Sentiment among stated above and M be the message then, probability for the M to be happy S is:

$$P(S_{happy}/M) = P(M/S_{happy})*P(S_{happy})/P(M)$$



And Hence for other sentiment also.

b. Binary Tree Searching Algorithm.

Step1. Take a string (message) as a input.
Step2. Using user defined function we'll split the sentences into words.
Step3. Analyse these words by checking their frequency.
Step4. These words gets searched in binary tree of our collected data set.
Step5. If word found at the root node then return 1. Else if the word is less than the root word (Alphabetical Order) Search the left sub-Binary Tree. Otherwise search in the right-sub-Binary Tree.
Step6. Repeat the entire process for the rest of the sentiment in that message .

6. Checking the efficiency of the algorithm.
   a. Making a Customised Confusion Matrix for detecting the efficiency of the customised classifier.
7. Testing the Algorithm by different data sets.

Classification of the Classifier:

Classification is probably the most frequently studied problem in machine learning and it has led to a large number of important algorithmic and theoretic developments over the past century. In its simplest form it reduces to the question: given a pattern x drawn from a domain X, estimate which value an associated binary random variable y 2 f_1g will assume. For instance, given pictures of apples and oranges, we might want to state whether the object in question is an apple or an orange. Equally well, we might want to predict whether a home owner might default on his loan, given income data, his credit history, or whether a given e-mail is spam or ham. The ability to solve this basic problem already allows us to address a large variety of practical settings. There are many variants exist with regard to the protocol in which we are required to make our estimation: 10 1 Introduction Fig. 1.6. Left: binary classi_cation. Right: 3-class classi_cation. Note that in the latter case we have much more degree for ambiguity. For instance, being able to distinguish stars from diamonds may not su_ce to identify either of them correctly, since we also need to distinguish both of them from triangles.

Naive Baye's Algorithm:

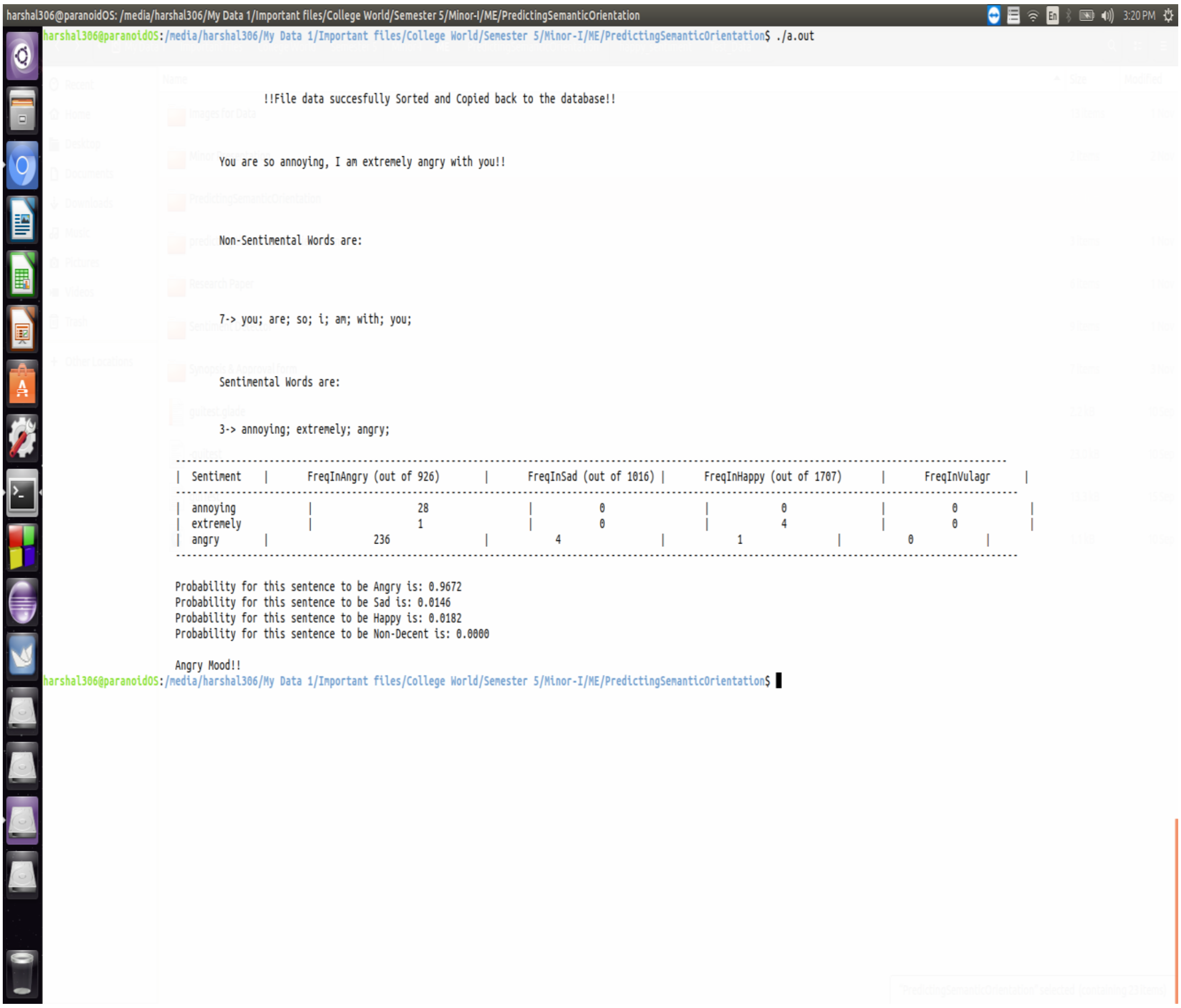
```
Train(X;Y) freads documents X and labels Yg
Compute dictionary D of X with n words.
Compute m;mham and mspam.
Initialize b := log c+logmham/logmspam to set the rejection threshold
Initialize p 2 R2_n with pij = 1, wspam = n, wham = n.
for Count occurrence of each wordg
i denotes the number of times word j occurs in document xig
for i = 1 to m do
        if yi = spam then
        for j = 1 to n do
                p0;j   p0;j + xj
                wspam   wspam + xj
        end for
        else
        for j = 1 to n do
        p1;j   p1;j + xj
        wham   wham + xj
        end for
        end if
end for
fNormalize counts to yield word probabilitiesg
for j = 1 to n do
p0;j   p0;j=wspam
p1;j   p1;j=wham
end for
```

VI. Work Samples:

The basic working sample is conducted by the Test cases for our 4 classification of the respective objective.

1. Angry Sentiments:

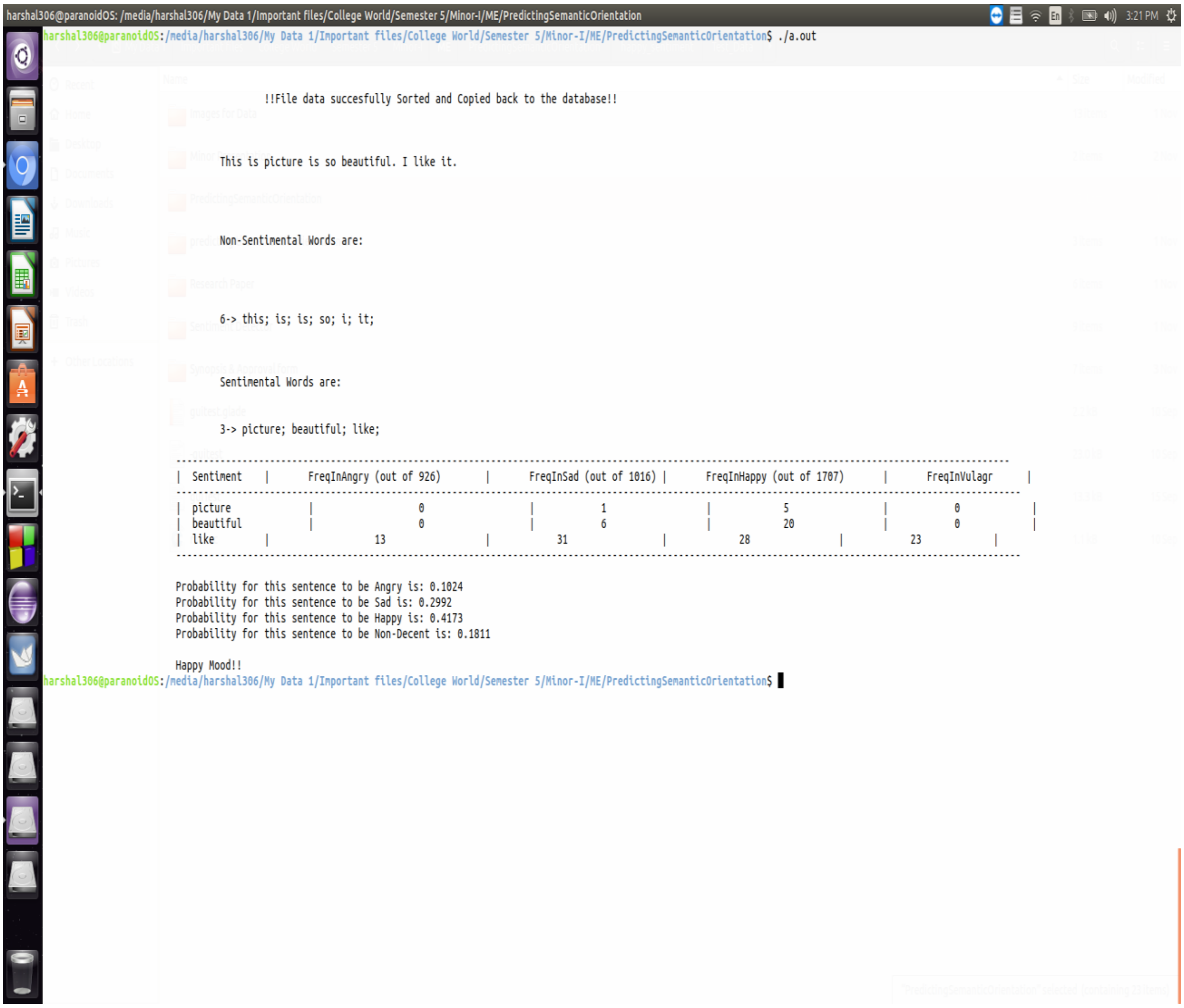
Angry sentiments are basically, searched by reference of saying and by the basic keywords used are: angry; annoy; bitter; frustrated; fired; burning etc.
For this our classifier predicts the angry sentiment with accuracy of 92%



Input: You are so annoying, I am extremely angry with you!!
Output: Angry mood (96%)

2. Happy Sentiment:

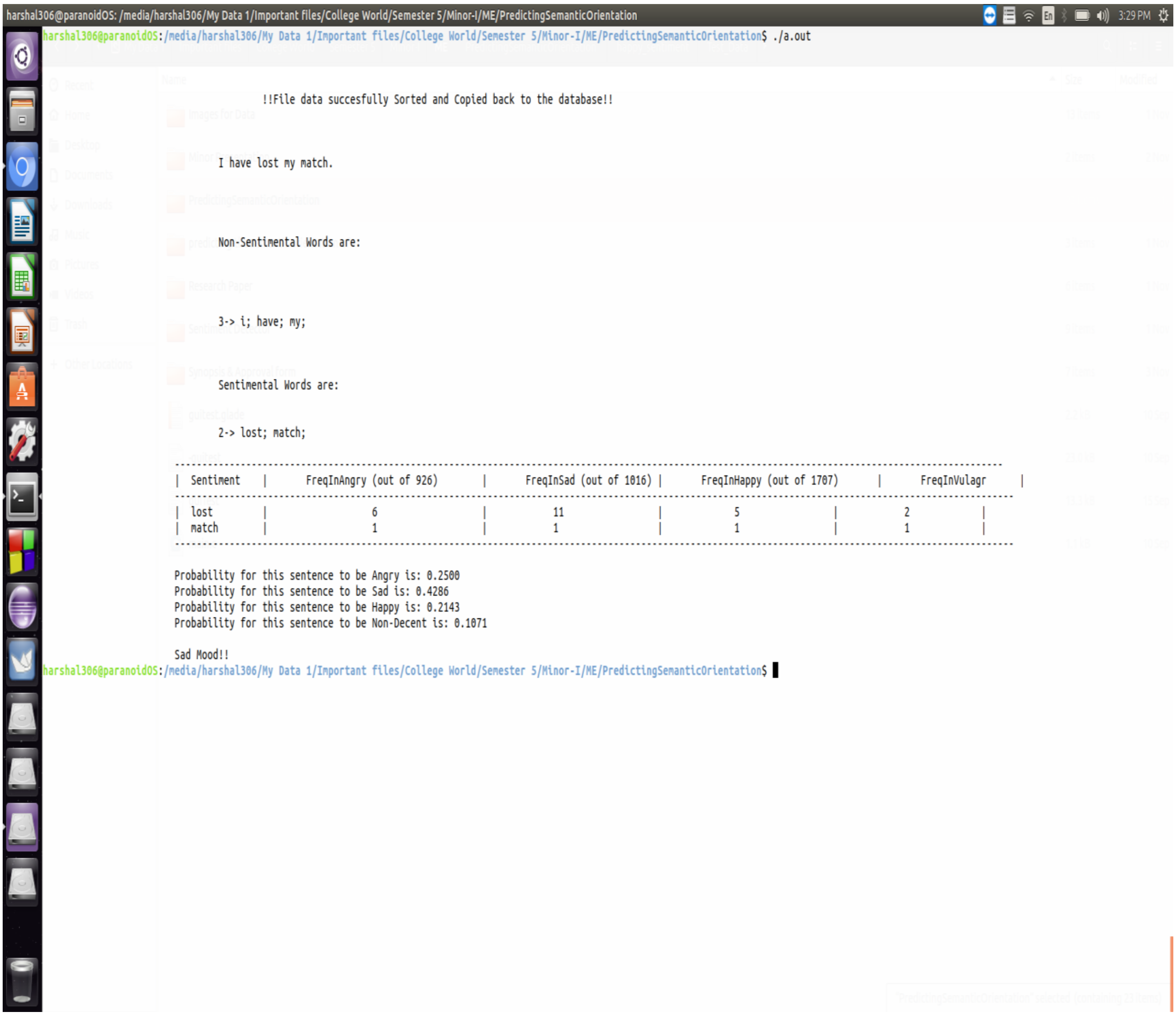
Happy sentiments are basically, searched by reference of saying and by the basic keywords used are: happy; glad; cheerful; like; awesome; etc.
For this our classifier predicts the angry sentiment with accuracy of 90%



Input: This Picture is so Beautiful. I like it.
Output: Happy Mood

3. Sad Sentiment:

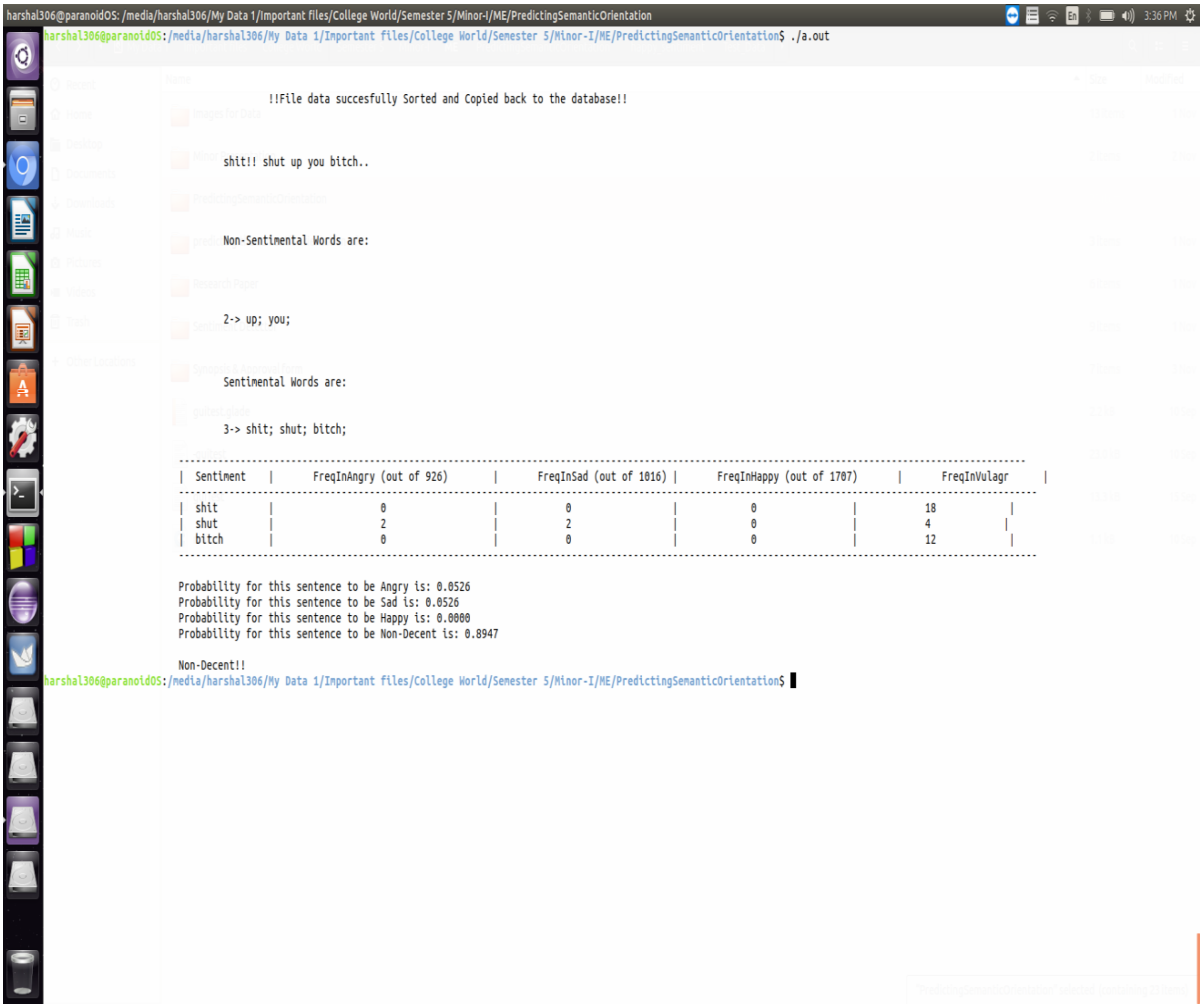
Sad sentiments are basically, searched by reference of saying and by the basic keywords used are: sad; down; cry; distressed; stressed; depression etc.
For this our classifier predicts the angry sentiment with accuracy of 91.7%



Input: I have lost my match.
Output: Sad Mood

4. Vulgar Sentiment:

Vulgar sentiments are basically, searched by reference of saying and by the basic keywords used are: motherfucker; bitch; fuck; and many more etc.
For this our classifier predicts the angry sentiment with accuracy of 91.7%



Input: Shutup!! you bitch!!
Output: Non-decent

5. Overall Efficiency (**Outcome**):

Overall efficiency of the classifier is calculated on the basis of the datasets of testing purpose and we have found that the efficiency of the classifier is 92.82% (**~93%**).

VII. System Requirements

Hardware Requirements

Processor	:	Intel(R) Core(TM)2 Quad CPU Q8400 @ 2.66GHz
RAM	:	1GB
HD	:	1GB

Software Requirements

Operating System	:	Linux
Compiler	:	GNU Compiler Collection (GCC Compiler)

VIII. Conclusion

Sentiment classification has seen a great deal of attention. Its application to many different domains of discourse makes it an ideal candidate for domain adaptation. This work addressed two important questions of domain adaptation. First, we showed that for a given source and target domain, we can significantly improve for sentiment classification the structural correspondence learning model. We chose pivot features using not only common frequency among domains but also mutual information with the source labels. We also showed how to correct structural correspondence misalignmentsby using a small amount of labeled target domain data. Second, we provided a method for selecting those source domains most likely to adapt well to given target domains. The unsupervised A-distance measure of divergence between domains correlates well with loss due to adaptation. Thus we can use the Adistance to select source domains to label which will give low target domain error. In the future, we wish to include some of the more recent advances in sentiment classification, as well as addressing the more realistic problem of ranking. We are also actively searching for a larger and more varied set of domains on which to test our techniques.

Detecting the sentiment of a word or a phrase within a message (term-level task).The system is based on a **supervised statistical text classification approach**[1] leveraging a variety of surface-form, semantic, and sentiment features. Given only limited amounts of training data, statistical sentiment analysis systems often benefit from the use of manually or automatically built sentiment lexicons. Sentiment lexicons are lists of words (and phrases) with prior associations to positive and negative sentiments. Some lexicons can additionally provide a sentiment score for a term to indicate its strength of evaluative intensity. Higher scores indicate greater intensity. For example, an entry great (positive, 1.2) states that the word great has positive polarity with the sentiment score of 1.2. An entry acceptable (positive, 0.1) specifies that the word acceptable has positive polarity and its intensity is lower than that of the word great.

Thus we can select source domains to label which will give low target domain error. In the future, we wish to include some of the more recent advances in sentiment classification, as well as addressing the more realistic problem of ranking. We are also actively searching for a larger and more varied set of domains on which to test our techniques. we got our classifier efficiency **~93%(92.83%)**

IX. Acknowledgement

We thank Mr**. P.Srikanth** for helpful advice throughout the course of this work. This material is based upon work supported by the **University of Petroleum and Energy Studies (UPES)** in **2017** Any opinions, findings, and conclusions or recommendations expressed in this material are from the faithful support of the organization.

Sentiment lexicons are lists of words (and phrases) with prior associations to positive and negative sentiments. Some lexicons can additionally provide a sentiment score for a term to indicate its strength of evaluative intensity. Higher scores indicate greater intensity. For example, an entry great states that the word great has positive polarity with the sentiments.

Thus we can select source domains to label which will give low target domain error. In the future, we wish to include some of the more recent advances in sentiment classification, as well as addressing the more realistic problem of ranking. We are also actively searching for a larger and more varied set of domains on which to test our techniques. we got our classifier efficiency.

X. References

[1] "Shiva Kumar, Vaithyanathan IBM Almaden Research Center 650 Harry Rd."(2002) Bo Pang and Lillian Lee

[2] "Recent Improvements in the Sentiment Analysis of Tweets" (2009) Xiaodan Zhu

[3] Saif M. Mohammad (2013) NRC-Canada: Building the State-of-the-Art in Sentiment Analysis of Tweets"

[4] Raj Kumar Verma, Dr. Ritu Tiwari (2016) "Sentiment Analysis of Social Web Data: A Review"

[5] Federico Alberto Pozzi, Elisabetta Fersini (2106) "Sentiment Analysis in Social n Network".

[6] Alexandra Balahur (2013) "Sentiment analysis in social media tasks" (2013).

[7] Rie Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. JMLR, 6:1817–1853.

[8] Sanjiv Das and Mike Chen. 2001. Yahoo! for amazon: Extracting market sentiment from stock message boards. In Proceedings of Athe Asia Pacific Financ Association Annual Conference.

[9] Andrew Goldberg and Xiaojin Zhu. 2004. Seeing stars when there aren't many stars: Graph-based semisupervised learning for sentiment categorization. In HLT-NAACL 2006 Workshop on Textgraphs: Graphbased Algorithms for Natural Language Processing.

[10] Matt Thomas, Bo Pang, and Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. In Empirical Methods in Natural Language Processing (EMNLP).

[11] Anthony Aue and Michael Gamon. 2005. Customizing sentiment classifiers to new domains: a case study.http://research.microsoft.com/ anthaue/.

[12] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. 2006. Analysis of representations for domain adaptation. In Neural Information Processing Systems (NIPS).

[13] Ciprian Chelba and Alex Acero. 2004. Adaptation of maximum entropy capitalizer: Little data can help lot. In EMNLP.

[14] Peter Turney. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. In Proceedings of Association for Computational Linguistics.

[15] G. B. Loghmani and M. Ahmadinia, “Numerical Solution of Third-Order Boundary Va *Journal of Science & Technology*, Vol. 30, No. A3, 2006, pp. 291-295.

[16] E. A. Al-Said, M. A. Noor and A. A. Al-Shejari, “Nu- merical Solutions for System of Second Order Boundary Value Problems,” *The Korean Journal of Computational & Applied Mathematics*, Vol. 5,No. 3, 1998, pp. 659-667.

[17] D. Kinderlehrer and G. Stampacchia, “An Introduction to Variational Inequalities and Their Applications,” New York Academic Press, New York, 1980.

Approved by: -

(Name & Sign)

Project Guide

(Name & Sign)

Program Head