Emotion Mining from Text

Megha Tyagi
Indraprastha Institute of Information Technology, Delhi
megha16033@iiitd.ac.in

Neha Jhamb
Indraprastha Institute of Information Technology, Delhi
neha16037@iiitd.ac.in

Nitish Srivastava
Indraprastha Institute of Information Technology, Delhi
nitish16038@iiitd.ac.in

Raveena Gupta
Indraprastha Institute of Information Technology, Delhi
raveena16046@iiitd.ac.in

Abstract

Emotions are a vital part of any human interaction. The ability to discern and understand human emotions is crucial for making interactive computer agents like chatbots, search engines, etc more human-like. This project focuses on detecting emotions from text. Automatic emotion detection from text has attracted growing attention due to its potentially useful applications.

1. Introduction

This section explains the problem statement and motivation for the project.

1.1. Problem Statement

There are basically 8 types of basic emotions that are commonly seen in human expressed texts - joy, trust, fear, surprise, sadness, disgust, anger, anticipate. We aim to recognize only five emotions - anger, fear, joy, surprise and sadness from text. However, we have added a neutral emotion, in case our system does not detect any emotion.

1.2. Motivation

We as humans can recognize emotions by reading texts easily but for a machine it is very difficult. Machine or computers need exact algorithm for recognizing emotions from text. Detecting and recognizing emotions from text is a recent field of study now a days.
Text based emotion recognition is useful for psychologists. Emotion mining can also be used for finding how a customer feels about a product, making response of chat-bot much more user-specific, understanding anxiety, etc.

2. Related Work

According to a Microsoft research article[1] in which Deep Learning is used, they have collected their own dataset with 784,349 samples of informal short English messages (i.e. a collection of English tweets), with 5 emotion classes: anger, sadness, fear, happiness and excitement. The unweighted accuracy they got is 64.47% while the weighted accuracy is 60.60 %. However, they have not shared the dataset publicly.
There is also a project called TEXEMO, they have collected data from We Feel Fine API. This was a keyword based extractor. They have used N-Gram Probabilistic Model, Naive Bayes Classifier, K-means and SVM. They have specified accuracies for individual emotions and have got maximum accuracy of 80% for Joy. They have shared the dataset publicly which we have referred.
According to [2], [3], [4] word2vec shows good performances in english text classification. However, for identifying emotions from text, word2vec has very limited use and it does not provide good results[5]. Generally, word2vec has been mostly used for sentiment analysis not for emotion mining.
Convolution Neural Network (CNN) is generally used in classifying images.[6] explains how to use CNN for natural language processing.

3. Dataset and Evaluation

3.1. Datasets

We have used following two datasets :

1. We Feel Fine (WFF) Dataset
   We have used a total of 50,000 English sentences. There are 10,000 sentences for each emotion - Anger, Joy, Sadness, Surprise and Fear. We have used 60%

dataset for training, 20% for validation and 20% for testing.

2. Tweets data from Crowdflower (TE)
   TE contains Twitter tweets labeled with emotions [7]. This dataset was released on Crowdflower website in November, 2016. This dataset contains around 9000 sentences. Same split of training, validation and testing data is used as used in WFF.

3.2. Feature Extraction

For Feature Extraction, we have used the following five techniques:

1. BOW_ADJ - Simple bag of words combined with some special adjective words. In this approach, first important words are extracted using TFIDF technique. Then a list of words is extracted from text which are adjectives, verbs and adverbs. Then the words common in both of these lists are considered for features.

2. Doc2Vec - Word2Vec (W2V) is an algorithm that takes every word in the vocabulary that is, the text we are classifying and turns it into a unique vector that can be added, subtracted, and manipulated in other ways just like a vector in space. Doc2Vec is an application of Word2Vec that takes the tool and expands it to be used on entire document, in our case a sentence. In the simplest form, naive Doc2Vec takes the Word2Vec vectors of every word in the text and aggregates them together by taking a normalized sum or arithmetic mean of the terms. Here, we have sentences for each emotion and Doc2Vec representation of sentences are used as features.

3. CNN Feature Extraction CNN is normally used for extracting features from images. According to [6], we reshaped our feature vector and took the sum of count of ones according to the size specified and thus reducing the number of features.

4. Feature Selection - Select k-best We select features on the basis of k-highest scores. For the dataset WFF, we took k=1000 and for dataset TE we used k= 500.

5. Unigram and Bigram - We noticed that if we use only unigrams as feature vector in our text classification task, we get some of the predictions wrong. For example, for the sentences like: I am not happy, the predicted emotion is Happy. While the actual emotion is Sad. To handle such cases, we used both bigram and unigram features and did not include the negative words like not, doesnt, wont etc. in the stop words. Thus the features like (not, happy) are now being captured and the predictions for such sentences as above are accurate.

3.3. Evaluation Metric

We are using precision, recall, F1-score, confusion matrix and accuracy over all test data instead of individual emotions.

4. Methodology

4.1. Visualization

The datasets are visualized as shown in figures in 2-dimensional and 3-dimensional 1 , 2 and 3.
We see that wefeelfine as seen from Figure 1, data is not linearly separable. But the different emotions are clustered separately from each other.
The colors in the visualisation of datasets represents their respective emotion as follows: Anger-Red, Sadness-Black, Joy-Blue, Fear-Green and Surprise-Yellow. This dataset has been created for emotion mining and the sentences are similar due to which the features can be clearly extracted and thus the data assigned with different emotions can be visualized as different clusters. We see that TE dataset as seen from 3 is not linearly separable and also there is overlapping among different emotion classes. Firstly, the preprocessing is done on the text. In preprocessing, all the sentences are converted to lowercase, punctuations are removed and then stopwords are removed. Lastly, stemming is done. As stated in the interim report, we have applied LR , SVM and LDA . This time we implemented them by using above mentioned feature extraction techniques.

4.2. Hyperparameters Selection

For BOW_ ADJ

1. In case of Logistic regression, hyperparameters (penalty, C,max_ iter) are chosen for grid search and maximum accuracy can be seen from Figure 4.

2. In case of Linear Discriminant analysis, hyperparameters (solver, n_ components) are chosen for grid search and maximum accuracy can be seen from Figure 5.

3. The hyperparameters chosen for grid search for SVM are Penalty (l1 or l2) , C (penalty parameter of the error term) and Loss (squared-hinge or hinge)
   The accuracy vs hyper parameters plot is shown in Figure 6.

For **Doc2Vec**, different parameters for model creation are tried and best parameters are chosen for each model i.e. Linear Discriminant Analysis(LDA), Logistic Regression(LR) and SVM. The parameters used for model

creation grid search are min_count and size of vector. Here, we have tried for min_count = [1,4] and size[100,200,1000]. For SVM, we used linear and rbf kernels and linear kernel is performing better. Figure 7, 8, 9 show the accuracies vs hyperparameters graph for LR, LDA and SVM respectively. The best parameters come out to be min_count = 1 and size = 100 for Doc2Vec.

Also, for neural network using BOW_ADJ and Doc2Vec, grid search parameters chosen were Hidden layer nodes [Number of hidden layers and number of nodes in each hidden layer] and plots are shown for different sized subsets of WFF in figures 11 and 12.
Similarly, for TE dataset, neural network trained on Doc2vec, grid search plot is shown in 15.

CNN features, Feature Selection - Select k best The parameter 'stride' is chosen using grid search for LR, SVM and LDA classifiers as shown in figure 10.

4.3. Overfitting and Underfitting

The training has been done correctly. We have checked this by seeing if the model is overfitted or underfitted. We have calculated accuracy on training and validation data both.
For BoW_ADJ, neural network is trained on 25000 size subset of WFF dataset where each emotion has 5000 sentences as well as on 50000 sentences of WFF dataset [10000 for each emotion]. The training and testing accuracies for each is shown in the figures 11 and 12 .
Similarly, for Doc2vec, neural network is trained and accuracies are shown in Figure 13 and 14.
For TE dataset, neural network for Doc2vec features is trained and training and validation accuracies are shown in figure 15

We can infer from these graphs and plots that there is no overfitting and underfitting hence, training is correctly done.

5. Results and Analysis

5.1. Results

Accuracies We have implemented four models i.e LR (Logistic Regression) , LDA (Linear Discriminant Analysis) , SVM (Support-Vector Machine) and Neural Network using the features extracted from different techniques stated before. LR, LDA and SVM have been implemented earlier and the challenge was to get above 98 percent accuracy which we have got through CNN features and feature reduction. The test accuracies are shown in 16. Doc2vec with neural network got 80 percent accuracy on WFF test set and hence not included in the figure.

The final best accuracies for WFF and TE dataset are shown in 17 and 18 respectively.

Precision, Recall and F1-score The precision, recall and F1-score for WFF 25k dataset and WFF 50k dataset are shown in figures 19 and 20 respectively.
The precision, recall and F1-score for TE dataset are shown in figure 21.

Confusion Matrices The confusion matrix for WFF dataset [25000 sentences] [5k for each sentence] using BOW_ADJ, Feature Selection and Unigram+Bigram and neural network is shown in 22 , 23 and 24 respectively.

The confusion matrix for WFF dataset [50000 sentences] [10k for each sentence using Doc2vec, Feature Selection and Unigram+Bigram and neural network are shown in 25 , 26 and 27 respectively.

The confusion matrix for TE dataset using feature selection, CNN and Doc2vec and neural network are shown in 28, 29 and 30 respectively.

5.2. Analysis

We see that Logistic Regression having accuracy of 90 percent is best out of these (LR, LDA and SVM) models when trained on **BOW_ ADJ** features. In case of **Doc2Vec**, there is not much difference in accuracies for different models [LR, LDA and SVM]. All models are getting accuracies of around 60 68 percent. But LDA is still performing better.

Clearly, the BOW_ADJ feature extraction technique is performing better than Doc2Vec on the we feel fine dataset for the models [LR, LDA and SVM] .

Then we used other feature extraction techniques like CNN, Feature Selection and Unigram and Bigram and a new model Neural Network.
CNN has been used for extracting features from images but by referring [6] , we extracted features from text. We got 86 percent as the highest accuracy from Neural Network model.
Feature Selection technique gave us its best accuracy of 88 percent using Neural Network model.
Unigram and Bigram performs better than the other approach since it takes into account some semantic knowledge as explained in the earlier section.

For neural network model, as the number of layers are increased, the training accuracy either increases or remains more or less the same. While the validation accuracy decreases, hereby showing that the model starts to overfit

as we increase the number of layers.

For WFF dataset, we choose the model which gives the best validation accuracy. So, the best model turns out to be Neural Network with Unigram and Bigram features [97 percent] as seen from the plots 11 , 12, 13, 14 and 17.
The neural network with only 1 hidden layer and 50 neurons in that layer is the best model for WFF dataset as seen so far because there are non-linearities involved in the dataset and neural network works best for non-linear relationships in data.

For TE dataset, till now no work has been done except multinomial naive bayes classifier but it performed poorly.

We used LR, LDA, SVM and Neural Network models on this dataset using previously mentioned feature extraction techniques. The best accuracy although poor comes out to be 39 percent with Doc2vec feature vector and Neural Network model.
Our models didn't perform well on TE dataset because the feature extraction techniques we have used so far are purely based out of formal English rules while this dataset contains informal sentences since they are from Twitter which is a social networking site and people here don't care to use formal language.
So, we used two simple approaches for TE dataset which are performing better than the above mention approaches. These are stated as followed:

1. **Count Vectorizer** In this approach, we have feature vector for each of the emotion class. For a given input test sentence, we count the number of words common in feature vector of each class and the sentence. Then the class having the highest count is the predicted emotion for that sentence.

2. **Tf-Idf Vectorizer** In this approach, we have feature vector for each of the emotion class. For a given input test sentence, we calculate tf-idf score for each word of the the sentence. Then the class having the highest normalized sum of tf-idf is the predicted emotion for that sentence.

These approaches give accuracy of around 65 percent much better than the previous approaches.

In all cases, Neural Network model performed better than the other models.
The precision, recall and F1-score evaluation metrics have also been used to evaluate the performance of models and it can be concluded that precision and recall of all models are comparable to each other which shows that our models are generalized and correctly classifying the data.

For WFF 25k dataset, it can be concluded from confusion matrix that Neural Network Model classify the 'JOY' and 'SURPRISE' emotions with the highest accuracy using BOW_ ADJ and Feature Selection. For Unigram and Bigram, neural network is classifying 'JOY' and 'SADNESS' emotions with the highest accuracy.
For WFF 50k dataset, Doc2vec and Feature Selection classifies 'FEAR' and 'SURPRISE' correctly whereas Unigram and Bigram does classify 'JOY' and 'ANGER' with highest accuracy.

For TE dataset, Feature Selection and CNN predicts 'JOY' with high accuracy whereas Doc2vec classifies 'FEAR' with high accuracy.

6. Developed System

We have developed a web application which provides an interface to the developed classification system for emotion mining as shown in 31.

7. Contributions

7.1. Deliverables

The deliverables promised and actually delivered by each team member are shown in the below table 32.

7.2. References and Citations

1. http://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html

2. http://www.wildml.com/2015/11/understanding-convolutional-neural -networks-for-nlp/

3. http://linanqiu.github.io/2015/10/07/word2vec-sentiment/

4. https://radimrehurek.com/gensim/models/doc2vec

5. Sklearn documentation

6. nltk documentation

7.3. Individual Contributions

1. Megha Tyagi Preprocessing data for LDA model and training LDA model using hyper-parameter tuning,extracting features using CNN.Feature Reduction using select k best technique. Training LDA using reduced features with hyper-parameter tuning.
   Contribution in code: All the files in listed folders : LDA, CNN_features, Preprocess, Vectorizer/countvect_te.py, Explore

2. Neha Jhamb Preprocessing data for Logistic Regression model and training LR model using hyper-parameter tuning, training SVM (linear and rbf kernel)

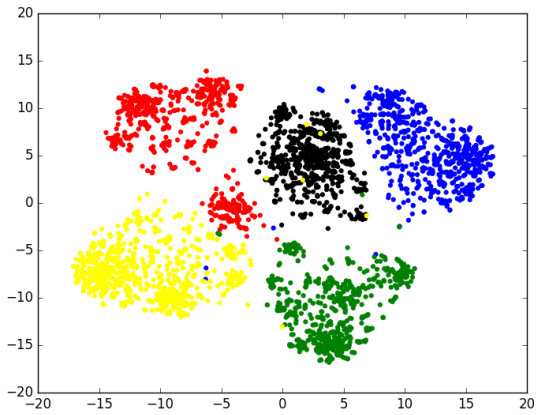

Figure 1. WFF 2-d

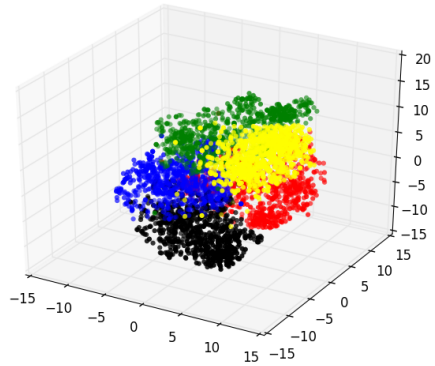

Figure 2. WFF 3-d Visualization

, LR using features extracted from CNN. Feature Reduction using unigram and bigram technique.
Contribution in code: All the files in listed folders : Logistic, Vectorizer/countvect.py, Feature_reduction, Uni_bigram

3. Nitish Srivastava Preprocessing data for Neural Network model and training Neural Network model using hyper-parameter tuning, training neural network on the extracted features using CNN, training NN on reduced features, developing web interface for interacting with the system.
Contribution in code: All the files in listed folders : Analysis, WebApp, NeuralNet, Visualise

4. Raveena Gupta Preprocessing data for SVM model and training SVM model using hyper-parameter tuning,implementing DOC2VEC feature extraction technique, Lemmatization technique on DOC2VEC and then training neural networks using them.
Contribution in code: All the files in listed folders : Doc2Vec, SVM, lemmatization

8. Figures and Tables

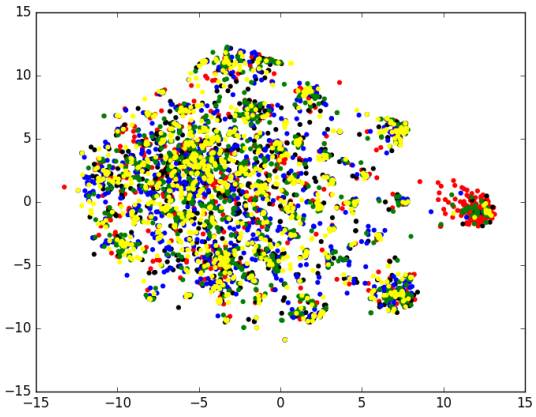

Figure 3. TE 2-d Visualization

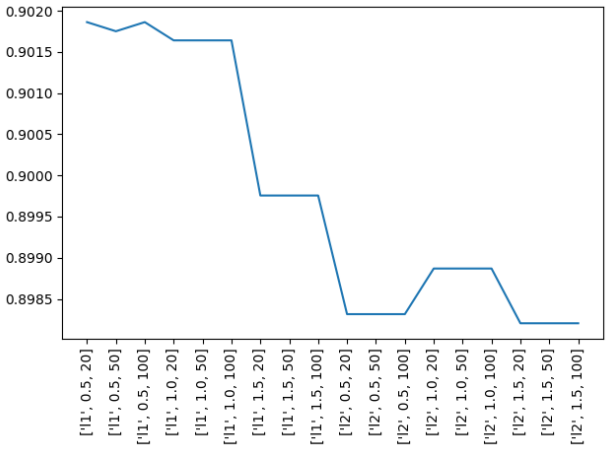

Figure 4. Logistic Regression Grid Search

References

[1] Chew-Yean Yam *Emotion Detection and Recognition from Text Using Deep Learning* https://www.microsoft.com/developerblog/2015/11/29/emotion-detection-and-recognition-from-text-using-deep-learning/

[2] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). *Efficient estimation of word representations in vector space.* In Proceedings of workshop at ICLR.

[3] Mikolov, T., Sutskever, I., Chen, K., Greg, C., & Jeffrey, D. (2013b). *Distributed representations of words and phrases and their compositionality.* In Proceedings of neural information processing systems.

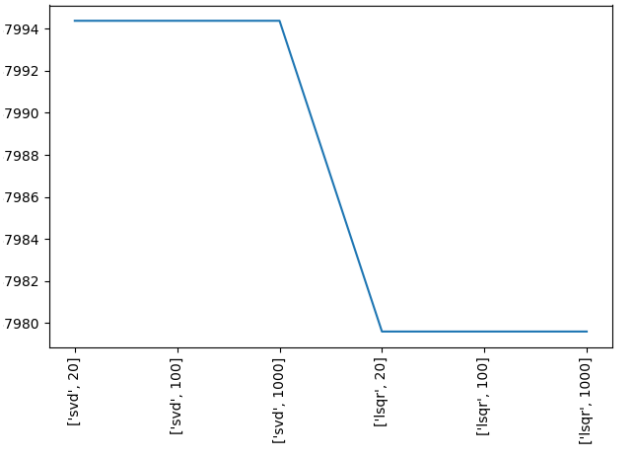

Figure 5. LDA Grid Search

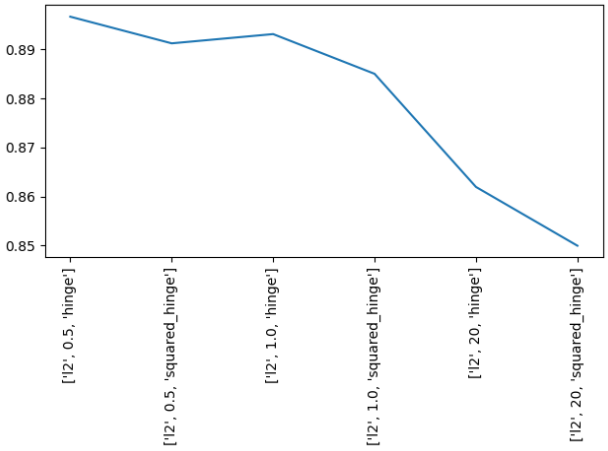

Figure 6. SVM Grid Search

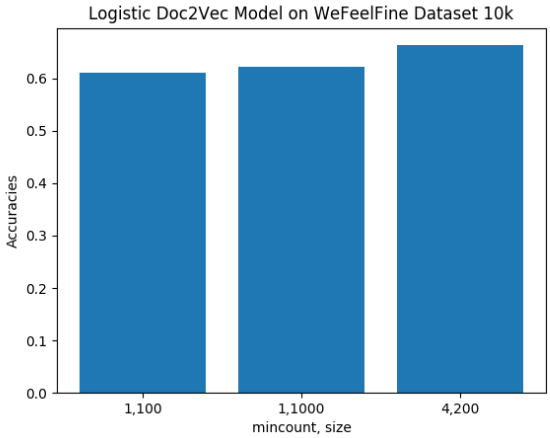

Figure 7. LR Grid Search (Doc2Vec)

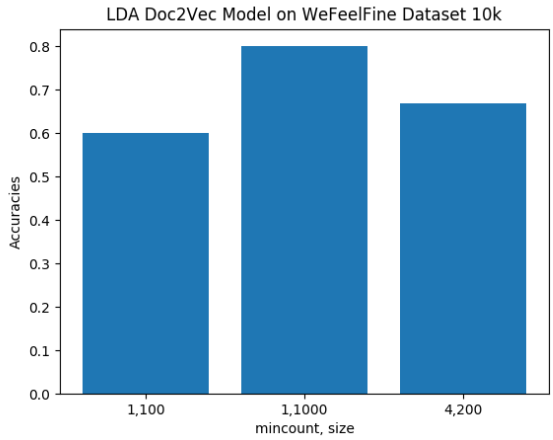

Figure 8. LDA Grid Search (Doc2Vec)

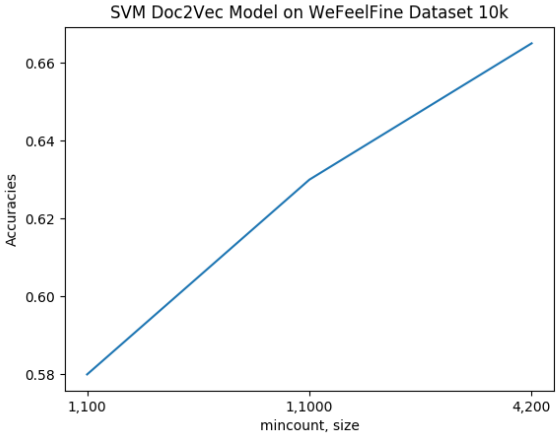

Figure 9. SVM Grid Search (Doc2Vec)

Technique	Normal features	CNN features				Features Reduction
		Stride = 8	Stride = 16	Stride = 2	Stride = 4	1000 feats.
Logistic	0.774657172996	0.596650843882	0.496703586498	0.766745780591	0.699235232068	0.792457805907
LDA	0.606276371308	0.588212025316	0.525712025316	0.587025316456	0.651503164557	0.812368144346
SVM	0.769910337553	0.587816455696	0.506856540084	0.746967299578	0.699235232068	0.802478902954

Figure 10. CNN Grid Search

[4] T. Mikolov, W.-t. Yih, G. Zweig *Linguistic regularities in continuous space word representations* Proceedings of the 2013 conference of the North American chapter of the association for computational linguistics: Human language technologies, Association for Computational Linguistics (2013).

[5] Armin Seyeditabari and Wlodek Zadrozny *Can Word Embeddings help find latent emotions in text? Preliminary Results* Proceeding of the Thirtieth International Florida Artificial International Research Society Conference 2017

[6] Denny Britz *Understanding Convolutional Neural Networks for NLP* http://www.wildml.com/2015/11/understanding-

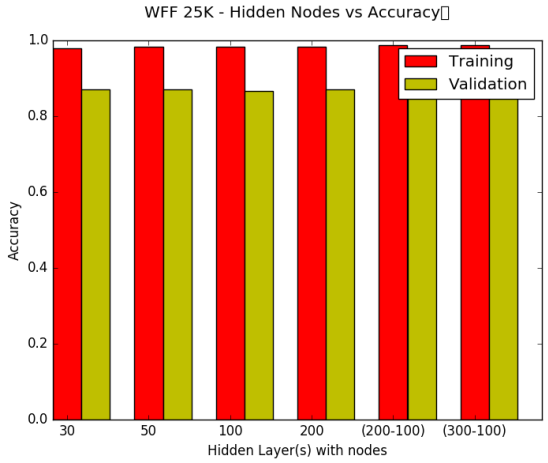

Figure 11. WFF 25k Grid Search

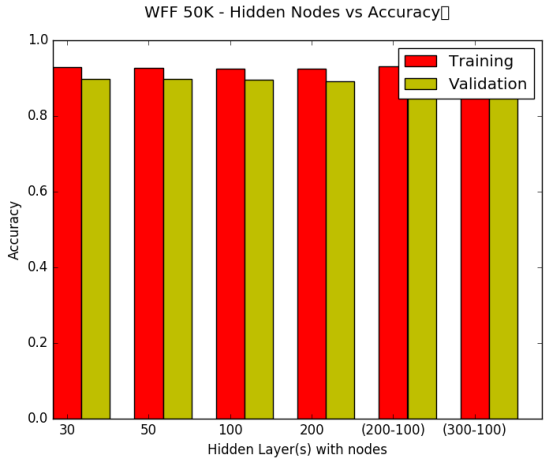

Figure 12. WFF 50k Grid Search

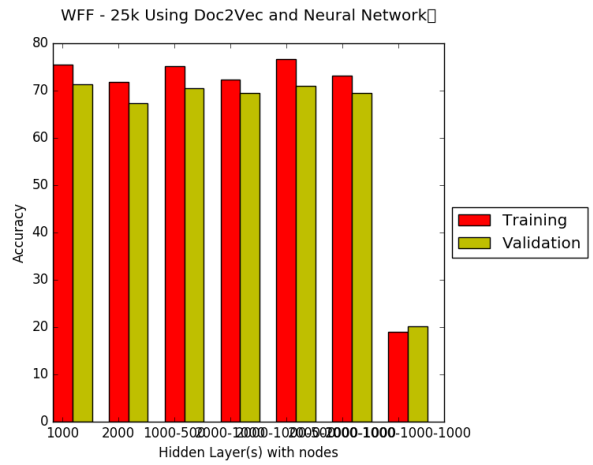

Figure 13. WFF 25k Grid Search

convolutional-neural-networks-for-nlp

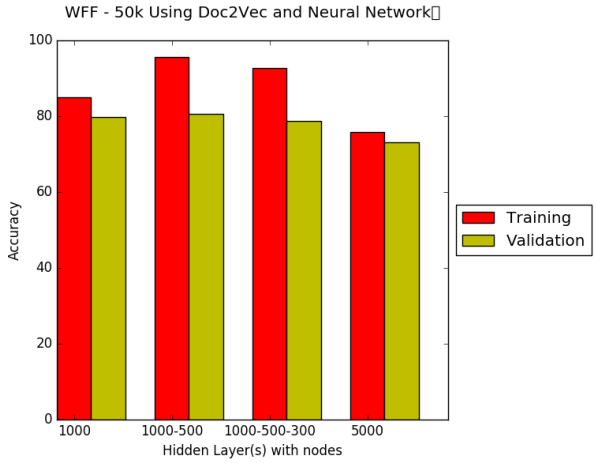

Figure 14. WFF 50k Grid Search

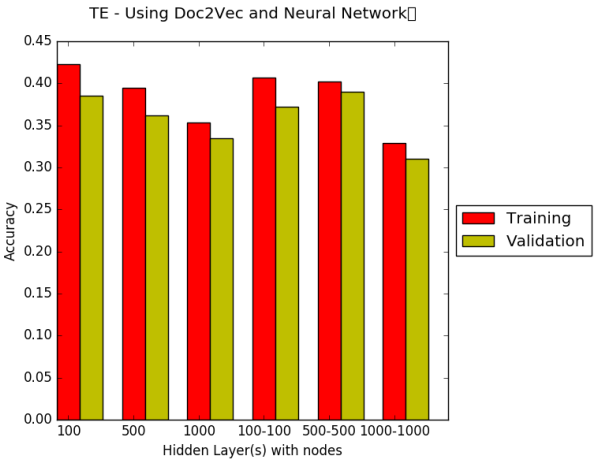

Figure 15. TE Grid Search

Technique	BOW_ADJ	CNN features	Features Reduction
Logistic	0.9979	0.9909	0.9976
LDA	0.7724	0.9868	0.8556
SVM	0.9973	0.9912	0.9982

Figure 16. LR, SVM, LDA Results

[7] Twitter Emotion *Link for dataset* https://www.crowdflower.com/wp-content/uploads/2016/07/text_emotion.csv

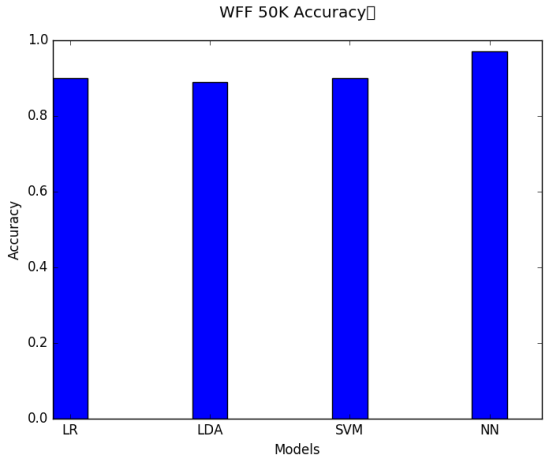

Figure 17. WFF Final Test Accuracies

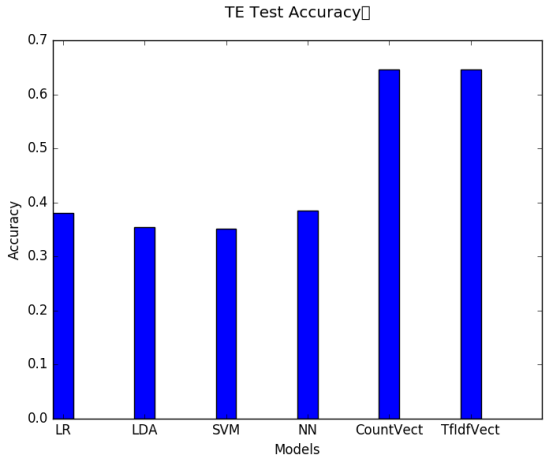

Figure 18. TE Final Test Accuracies

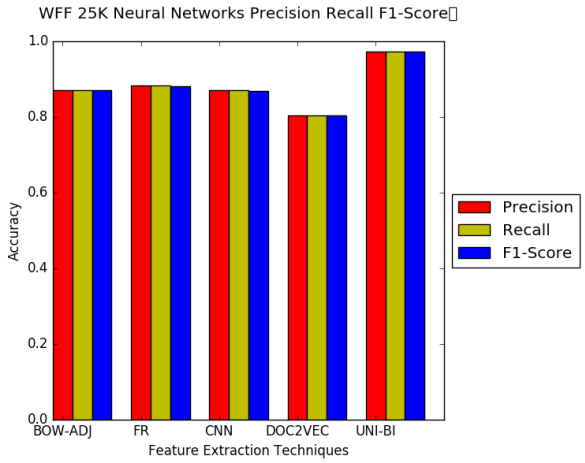

Figure 19. WFF 25k Precision, Recall and F1-Score

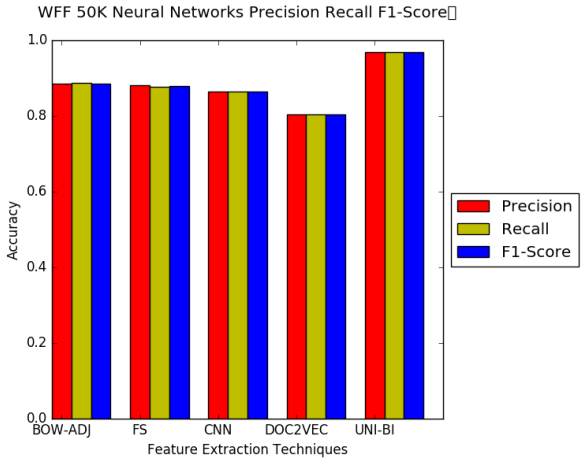

Figure 20. WFF 50k Precision, Recall and F1-Score

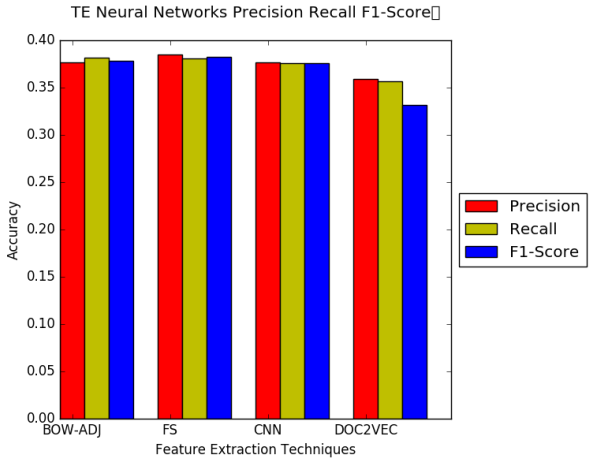

Figure 21. TE Precision, Recall and F1-Score

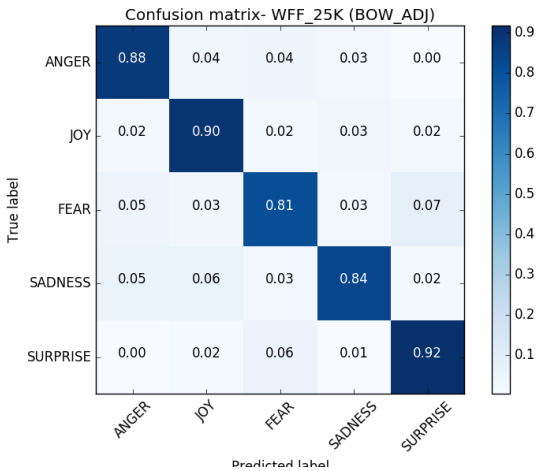

Figure 22. WFF 25k BOW_ADJ

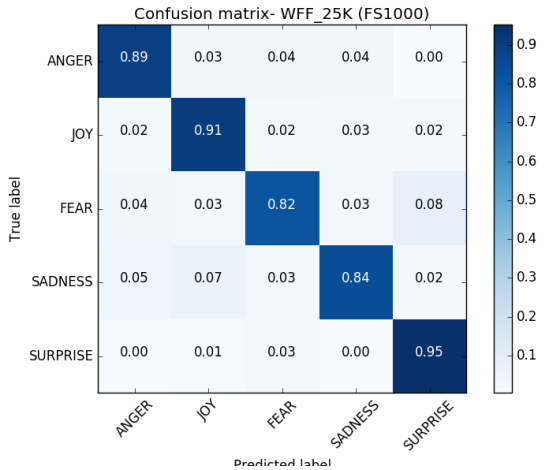

Figure 23. WFF 25k Feature Selection

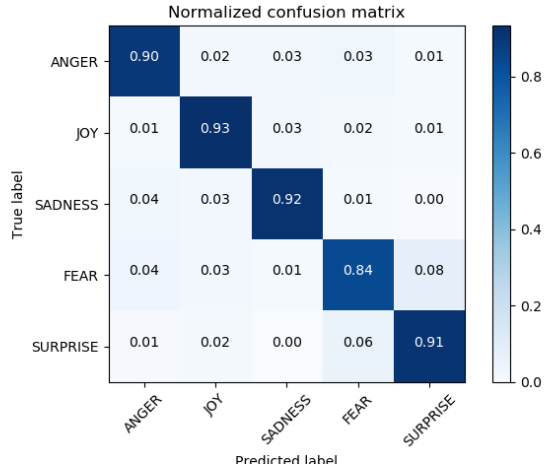

Figure 24. WFF 25k Unigram and Bigram

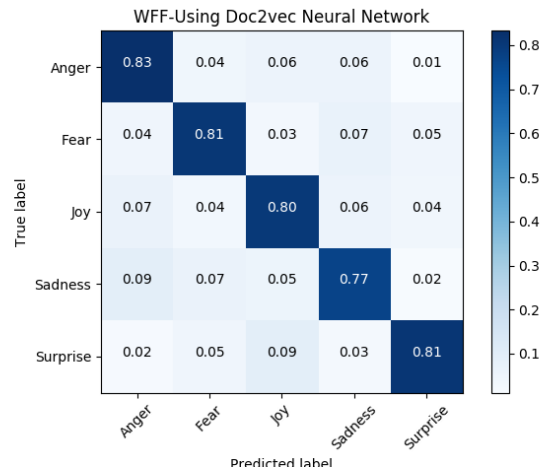

Figure 25. WFF 50k Doc2vec

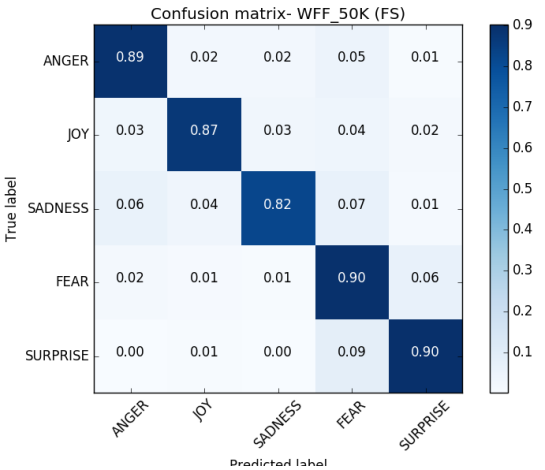

Figure 26. WFF 50k Feature Selection

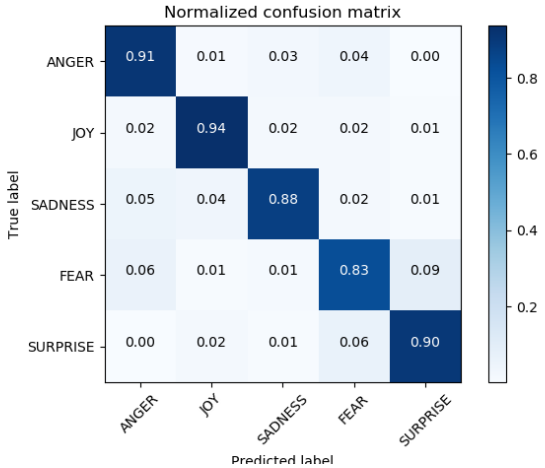

Figure 27. WFF 50k Unigram + Bigram

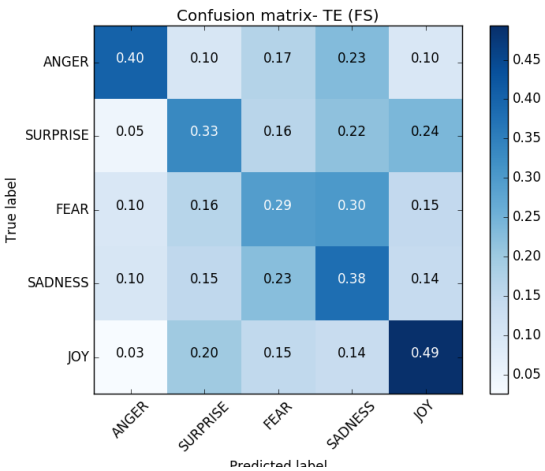

Figure 28. TE Feature Selection

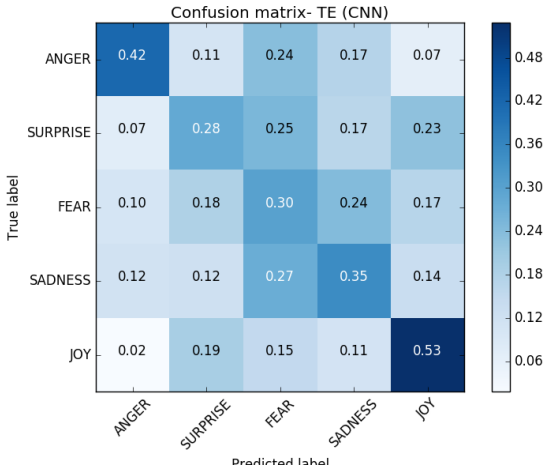

Figure 29. TE CNN

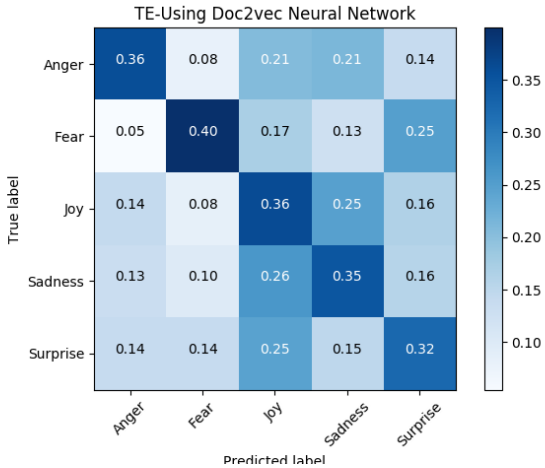

Figure 30. TE Doc2vec

Team Member	Task Proposed	Task Delivered
Megha Tyagi	Preprocessing data for LDA model and training LDA model using hyper-parameter tuning,extracting features using CNN, training LDA using extracted features.Feature Reduction using 'select k best' technique. Training LDA using reduced features with hyper-parameter tuning.	All the tasks proposed have been delivered.
Neha Jhamb	Preprocessing data for Logistic Regression model and training LR model using hyper-parameter tuning, training SVM, LR using features extracted from CNN. Feature Reduction using 'unigram' and 'bigram' technique.	All the tasks proposed have been delivered.
Nitish Srivastava	Preprocessing data for Neural Network model and training Neural Network model using hyper-parameter tuning, training neural network on the extracted features using CNN, training NN on reduced features, developing web interface for interacting with the system.	All the tasks proposed have been delivered.
Raveena Gupta	Preprocessing data for SVM model and training SVM model using hyper-parameter tuning,implementing DOC2VEC feature extraction technique, Lemmatization technique on DOC2VEC and then training neural networks using them.	All the tasks proposed have been delivered.

Figure 32. Deliverables

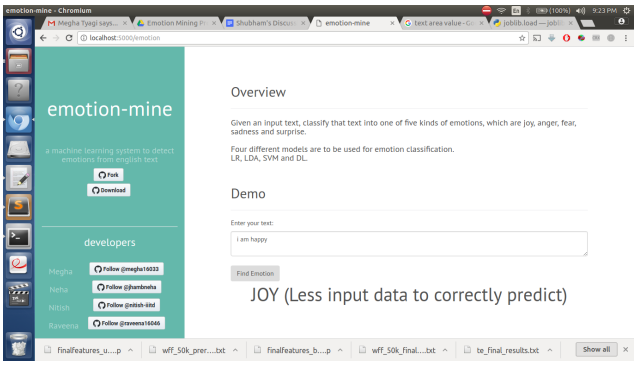

Figure 31. Web Application