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.

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 show 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.

3. Dataset and Evaluation

We have used We Feel Fine 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.
For Feature Extraction, we have used the following two techniques:

- 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.

- 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.

Evaluation Metric
We are using accuracy as evaluation metric. We will use accuracy over all test data instead of individual emotions.

4. Analysis and Progress

We have implemented three models i.e LR, LDA and SVM using the features from BOW_ADJ. These three models are also repeated with Doc2Vec feature extraction. Firstly, the preprocessing is done on the text. In preprocessing, all the sentences are stemmed, stopwords are removed, punctuations are removed and converted to lowercase.

4.1. Challenges

4.1.1 Model Challenges

As the model is based on the Unigrams Bag of words approach, the classifier is not able to detect the difference between sentences like I am happy and I am not happy. This will be rectified if we take n-grams as features. If the sentence is not containing any emotion (for e.g I am a boy.), even then the classifier will output some emotion. Hence, we need to add a emotion NEUTRAL also in the model.

4.1.2 Design Choices

Learning Method
We have used supervised learning method. We have initially chosen 4 models for our problem.These are : Logistic Regression, Linear Discriminant Analysis, Support Vector Machine, and Deep Learning. We will choose the final model on the basis of accuracy. Till now, we have implemented first three models and Logistic Regression has performed best in case of BOW_ ADJ. For Doc2Vec, Logistic Regression, Linear Discriminant Analysis(LDA) and Support Vector Machine models are applied. LDA is performing better than the others. For best hyperparameter

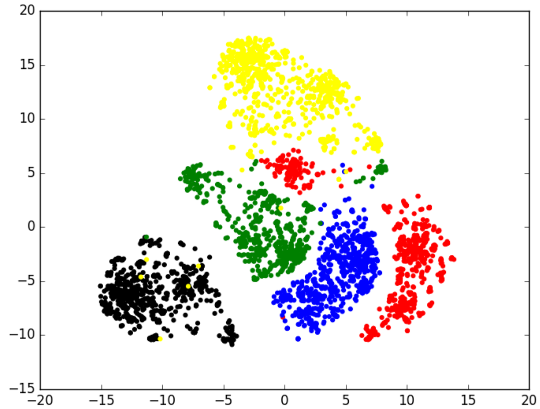

Figure 1. We feel fine Dataset

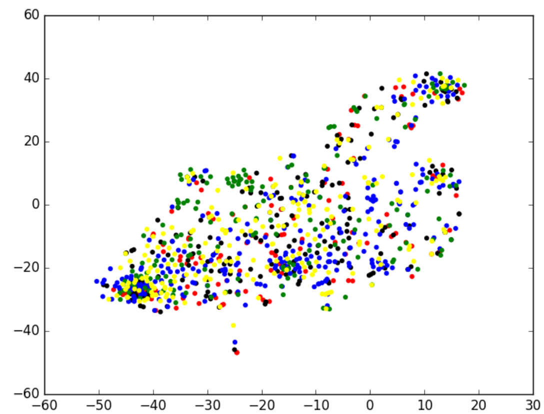

Figure 2. Affective Dataset

selection, grid search is used.

Data Visualization

1. We Feel Fine Dataset
   We see that wefeelfine data visualization, Figure 1 is not linearly separable. But the different emotions are clustered separately from each other. 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.

2. Affective_ text Dataset
   Affective text contains news headlines. News headlines contain very limited and important information only i.e the use of adjective words is rare. Hence this data was also not suitable for emotion classification.

   As can be seen from the Figure Affective Dataset the

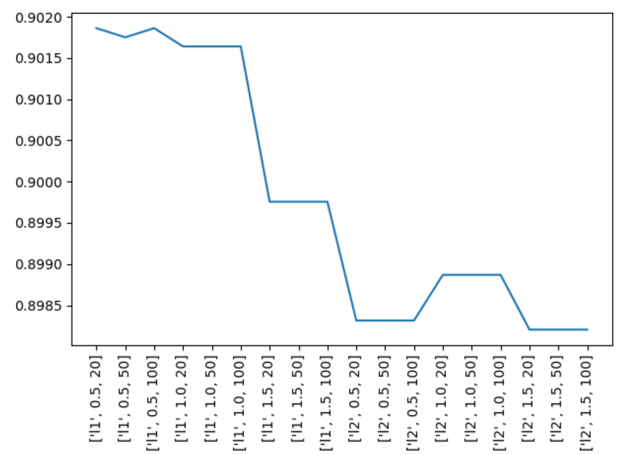

Figure 3. Logistic Regression Grid Search

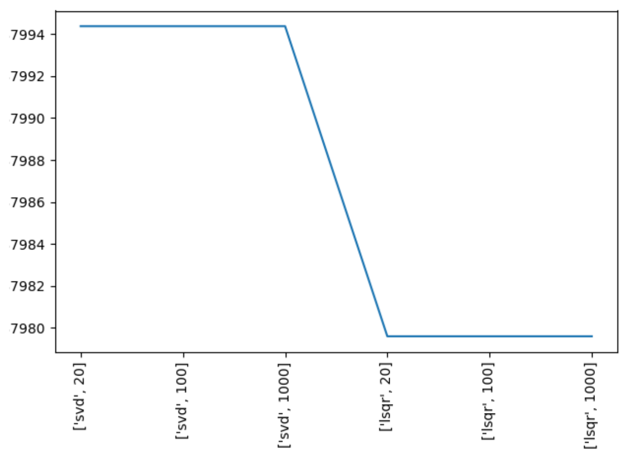

Figure 4. LDA Grid Search

dataset is not separable.

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 3.

2. In case of Linear Discriminant analysis, hyperparameters (solver, n_ components) are chosen for grid search and maximum accuracy can be seen from Figure 4.

3. The hyperparameters chosen for grid search for SVM are: Penalty (l1 or l2) C (penalty parameter of the error term) Loss (squared-hinge or hinge)
   The accuracy vs hyper parameters plot is shown in Figure 5.

For **Doc2Vec**, different parameters for model creation are tried and best parameters are chosen for each model

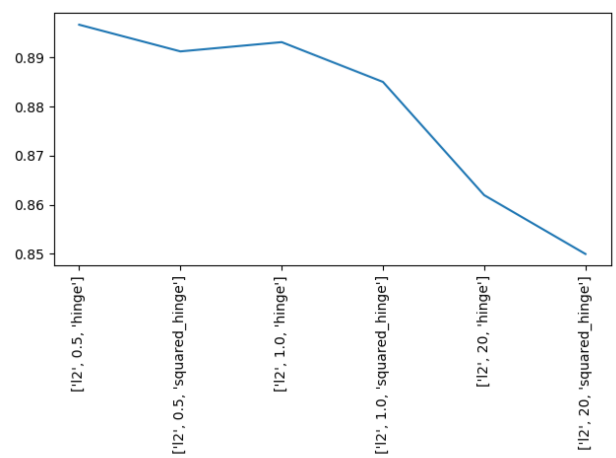

Figure 5. SVM Grid Search

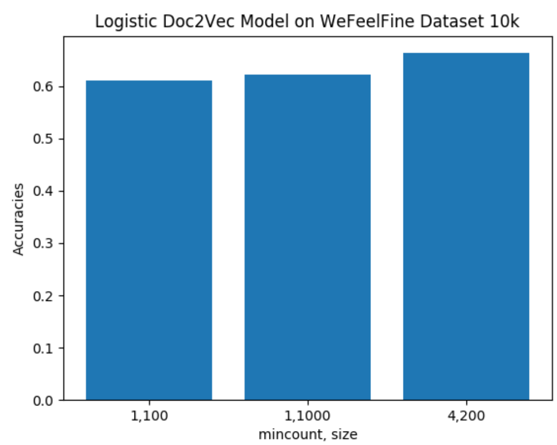

Figure 6. LR Grid Search (Doc2Vec)

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 6, 7, 8 shows the accuracies vs hyperparameters graph for LR, LDA and SVM respectively. The best parameters comes out to be min_count = 1 and size = 100 for Doc2Vec.

Overfitting/Underfitting

Training is done correctly. We have checked this by checking if the model is overfitted or underfitted. We have calculated accuracy on training data and testing data both. The values are shown in Table 1 We can infer from the table that there is no overfitting and underfitting i.e training is correctly done.

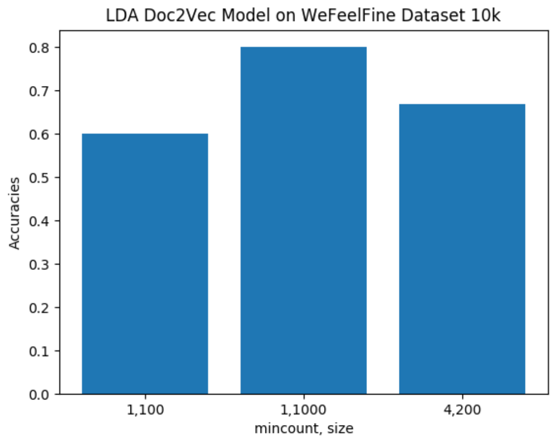

Figure 7. LDA Grid Search (Doc2Vec)

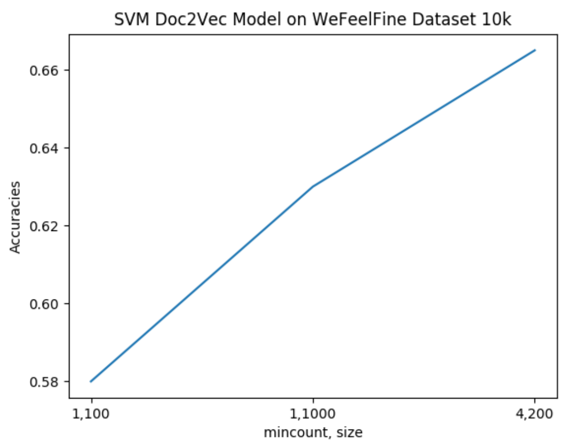

Figure 8. SVM Grid Search (Doc2Vec)

Table 1. Training and Testing Accuracies

Model	Training Accuracy	Testing Accuracy
Logistic Regression	0.917	0.903
LDA	0.951	0.889
SVM	0.961	0.895

5. Results

The results are displayed in the table 1. We see that Logistic Regression is best out of these three when trained on BOW_ ADJ features.

1. LR : The best accuracy is obtained for the following values of hyper parameters: Penalty : l1, C=0.5 and max_ iter = 100 As can be seen from Figure 3, there is no effect of max_ iter on accuracy, but l1 is giving more accuracy than l2 norm and also it is maximum with the less regularization parameter i.e. C=0.5.

2. LDA : The best accuracy is obtained for the following values of hyper parameters: Solver : svd, n_ components : 20 As we see from the Figure 4, n_ components is not having any effect on accuracy. SVD solver is giving more accuracy than LSPR solver.

3. SVM: The best accuracy is obtained for the following values of hyper parameters: [penalty, C, loss] : [l2, 0.5, hinge]
Penalty: l2 norm is better as we have sparse vectors here in this case. Loss: As can be seen from the Figure 5, accuracy decreases for squared-hinge loss as the hinge loss penalizes predictions y ¡ 1.
C : Small C gives better accuracy as the dataset is linearly separable.

In case of **Doc2Vec**, the Doc2Vec model was built on 10000 sentences of each emotion. However, 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 if the Doc2Vec model is built with minimum count of word as 1 and size of the output vector is 1000 as seen from the figures 7, 6 and 8.
The size of output vector determines the size of array in which each word of the vocabulary is represented in the form of float values. The low value of min_ count ensures that even if the word is appearing lesser number of times, it would still be present in vocabulary. The overall accuracy is still low since we have used only this vector representation as feature vector and have not added any other feature like tfidf, etc. As told in [5], word2vec embeddings alone are not enough for classifying emotion from the text. These results may improve if the size of training set is increased or if the model is built on other rich vocabulary datasets.

Thus, the BOW_ADJ is performing better than Doc2Vec on the we feel fine datasets.

6. Future Work

1. The new technique that we are going to use in future will be Deep Learning.

2. We are going to use one other dataset also Emotion in Text data set by CrowdFlower.

3. We are not going to use any other evaluation metrics, we will stick to accuracy.

4. We are going to analyse the models on negative sentences such as not happy, etc. According to the analysis, we will be changing the features to get more accuracy. Also we will apply features reduction algorithm to reduce the redundant features.

Team Member	Task
Megha Tyagi	Extracting features using CNN.
Neha Jhamb	Training SVM, LDA and LR on the above extracted features.
Nitish Srivastava	Training Deep Neural Networks on the above extracted features.
Raveena Gupta	Lemmatization during preprocessing then Training on LR, LDA and SVM.

Figure 9. Individual Task Allocation

5. Individual team member roles are shown in the Figure 9

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.

[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