Ayushi Patel
A20407392

Sentiment Analysis on Enron emails

ABSTRACT

Sentiment analysis on emails can be a tough task for quite a few reasons. But the main reasons are their availability in unstructured format and unavailability of labelled email datasets. There are many tools available for pre-processing and data cleansing to obtain raw text to perform analysis on. The major issue although arises when Machine Learning methods are used instead of orthodox Lexicon based analysis. This project attempts to bridge the gap between unavailability of labelled emails and training set for a classifier model. An inter-domain supervised learning approach can help perform sentiment classification on email dataset. We train the model on labelled dataset of item reviews, improve the model by tuning the parameters and eventually run the model on the Enron dataset. With this approach we eliminate the need for email dataset for training and in turn generate labelled dataset for other sentiment analysis projects which might need labelled emails. We discuss the selection techniques and performance of the models and conclude with the final results and future works.

INTRODUCTION

The project revolves around predicting sentiment running across the Enron Corp amongst the employees before the event of filing for bankruptcy. The task will be performed by analyzing the mails shared across the organization. Two data sources will be used to complete the project, first will be the sentiment labelled data from 3 websites (Amazon, IMDB and Yelp) to train the model. These 3 data files will be used in combinations to train and test the model. Our accuracy check would be to see how many reviews from these websites are correctly identified as positive or negative. Second data source will be the Enron mail dataset which would include mails of employees and the model will be run on this data to predict the sentiment.

DATA

Dataset	Size of Data	Data Type	File Format	Source
Amazon	61 KB	Text	Text	https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences
IMDb	90 KB	Text	Text	https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences
Yelp	66 KB	Text	Text	https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences
Enron Emails	458 MB	Text	Text	https://www.cs.cmu.edu/~./enron/

Labelled Datasets

Amazon: The dataset consists of reviews from Amazon.com. Reviews include product and user information, ratings and a plaintext review. There are 1000 reviews of which 500 are positive and 500 are negative. The data is relatively clean, i.e. it is free of timestamps, location data, URLs, etc.

so there are a fewer pre-processing steps involved. We are only required to separate the text from its label and read the text separately for the model training.

IMDb: IMDb.com is a website devoted to collecting movie data supplied by studios and fans. It claims that it is the biggest movie database on the web and is run by Amazon. The dataset used for has 748 movie reviews selected at random and have a sentiment label assigned to it. There are 386 positive reviews and 362 negative reviews.

Yelp: The dataset was created for a research paper by Dimitros Kotzias. The file from Yelp contains 1000 reviews out of which 500 are positive and 500 are negative. The data is clean, so the only preprocessing step required is to separate the review from its label.

Dataset	# of Positive Reviews	# of Negative Reviews
Amazon	500	500
IMDb	386	362
Yelp	500	500

Overview of the datasets and their distribution

Review (plain text)	Score (0 or 1)

Format of the training sets

2. Enron Emails

The dataset contains data from 150 users, mostly from the senior management of Enron, organized into folders. The corpus contains 500,000 emails exchanged amongst employees. The mails are available from their inbox, sent items and even deleted items folder.

APPROACHES

Sentiment classification can be performed by two methods:

1. Lexicon based classification

2. Machine Learning method

Lexicon based classification compares each word in an instance against a compiled set of sentiment words. Each sentiment word has an individual score, which are then summed up to determine if an instance is positive or negative. The scope of Lexicon based method is limited due to the usage of sentiment words compiled by other researchers. Given the human related aspect, we cannot be certain the word lists are exhaustive. Thus, we decide to go with machine learning approach, where we implement classification from a training set and apply those learnings on the testing set and this approach.

Machine Learning approach offers a few popular implementations for classifications such as Naïve Bayes, Maximum Entropy and Support Vector Machine(SVM). We implement our project using

SVM, more specifically LinearSVC(Linear Support Vector Classification) which is the best approach as our data is classified to either 1 (positive sentiment) or 0 (negative sentiment).

As discussed, we use SVC for this project because it carries a lot of advantages in terms of flexibility on choosing the parameters. LinearSVC implements a *one-vs-all* technique which means the model trains one classifier per class (positive and negative). A simple SVM implements *one-vs-one* technique would have half the number of classifiers but would require to be trained by each set of class which would require more computational effort.

Programming Language, packages: Python is used for this project as there are variety of sentiment analysis packages available. The packages to be used are Pandas, Numpy and Matplotlib2. Also, sklearn will be used for vectorization and modelling.

In this project, the model is trained using the review data from Amazon, IMDb and Yelp. Results from the testing data are evaluated using *accuracy_score, precision_score, recall_score and f_score* methods provided by the *sklearn* package in Python.

EXPERIMENTS

Pre- Processing**:** The pre-processing step includes splitting the score from its review to ensure only text is read for our model training. The model then goes through multiple iteration of training and testing. Enron emails, which is the target dataset, is tested after multiple training steps and the accuracy is checked by comparing against the labels.

The following table gives a brief description:

Iterations	1	2	3	4	5	6	7
Amazon	training	Test	test	training	test	training	training
IMDb	Test	training	test	training	training	test	training
Yelp	Test	Test	training	test	training	training	training
Enron	-	-	-	-	-	-	test

Table 1: Iterations and combinations for a potential classifier model

We test each of the above models derived by different combinations. These models derive their accuracy and we cross check the data points where we can evaluate the agreement among the models. Then we select the models where data points are common. The new selected model is tested on a sample Enron emails and evaluated for accuracy, as explained in detail further in this report. The model with highest accuracy would be our best model for our project.

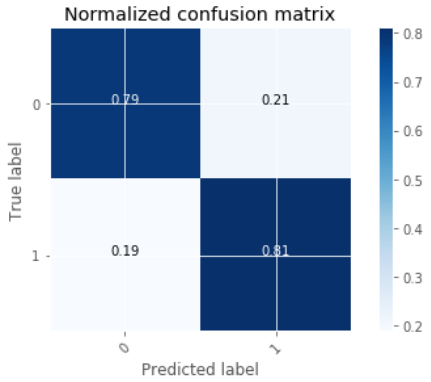
The experiment is performed by training the model with following seven combinations of datasets yeilding different scores:

Model	Accuracy	Precision	Recall	F1 Score
Amazon	84.4%	79.8%	83.7%	81.7%

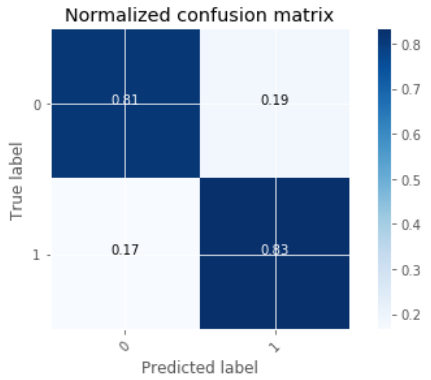
Sentiment Analysis on Enron emails

Amazon + IMDb	82.4%	83.3%	83.3%	83.3%
Amazon + IMDb + Yelp	80.0%	81.8%	81.4%	81.6%
Amazon + Yelp	82.8%	84.7%	79.2%	81.9%
Yelp	79.2%	77.2%	81.0%	79.0%
IMDb +Yelp	80.2%	83.0%	77.5%	80.2%
IMDb	80.4%	82.1%	78.9%	80.5%

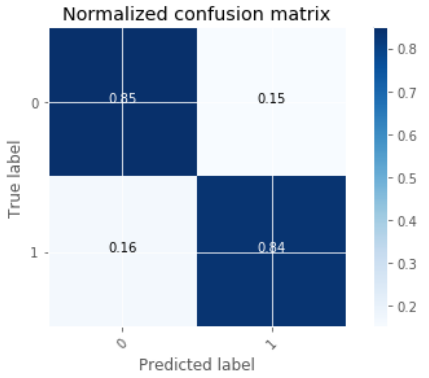
Table 2: Models with their respective classification score



Confusion Matrix 1.1: Amazon + Yelp + IMDb



Confusion Matrix 1.2: Amazon + IMDb



Confusion Matrix 1.3: Amazon

Normalized Confusion Matrices for the models (further matrices available in teammate's report)

The confusion matrix gives us an idea to how good our model is in classifying. Confusion matrix can be used to calculate attributes such as accuracy, precision, recall and f score(see *Table 2*). It is constructed using the labels predicted by the model and the true label of the data. This divides the graph in 4 quadrants, namely –

- True Negatives(TN:0,0) – Reviews predicted negative and contained negative
- False Positives(FP:1,0) – Reviews predicted positive but had negative sentiment
- True Positives(TP:1,1) – Reviews predicted positive and are positive review
- False Negatives(FN:0,1) – Reviews predicted as negative but was a positive review

Precision: When the prediction is positive, shows how often it is correct

Sentiment Analysis on Enron emails

$$Precision = \frac{TP}{TP + FP}$$

Recall: When the review is positive, shows the number of positive reviews predicted correctly

$$Recall = \frac{TP}{TP + FN}$$

F Score: Weighted average of *Recall* and *Precision* to combine the two attributes

$$F\ Score = 2.\frac{Precision\ .Recall}{Precision + Recall}$$

Accuracy: The number of classifications done correctly for positive and negative reviews

$$Accuracy = \frac{TP + TN}{Total}$$

To select a model for predicting sentiments of Enron emails, we look for a balanced score of *Accuracy* and *Precision* (in *Table 2*), as we would like the model to predict both positive and negative sentiments correctly without having an overfitting issue.

Parameter Tuning:

Parameters play an important role in improving the accuracy of any classifier. Our parameter setting should ensure that the classifier has minimum bias and minimum variance, i.e. no overfitting of the training data and no over-generalization of data points respectively. This characteristic can be ensured by the setting the parameters for loss function, penalty and regularization parameter by trial and error method. The cost function derived from the terms mentioned above is given as:

```
V(f(X, y) + lambda*R(f)
```

Loss Function Penalty Function

The loss function measures the quality of the solution, meaning how well the model maps the data to the labels. LinearSVC uses *hinge loss* by default.
The penalty function is used to impose some constraints R f on the solution f. As mentioned earlier this would avoid overfitting of data and we use *L2 norm* as penalty for our SVC.
We then have *C* which is the Regularization parameter used to attain the width from the hyperplane between the support vector. *C* is calculated from the range of 0.01 to 100 using *GridSearchCV* function.

Sentiment Analysis on Enron emails

The primary aim of any model is to achieve highest accuracy possible. Through multiple trials of mixing and matching the parameters, we can decide which right parameters for the model which would eventually predict on the test set.
The tables below clearly present a picture of what values should be set for the parameters of the model. The peak values observed for accuracy are the optimum values for the model.

C	Accuracy
0.01	50.4%
0.1	78.3%
1	80.0%
10	78.3%
100	77.2%

ngrams	Accuracy
Unigrams	79.5%
Bigrams	62.8%
Trigrams	50.9%
4-grams	49.3%

Loss Function	Accuracy
hinge loss	80.0%
squared hinge loss	78.8%

Table 3.1: Accuracy for different values of parameters – Amazon + IMDb + Yelp

C	Accuracy
0.01	47.4%
0.1	64.0%
1	82.4%
10	80.0%
100	79.6%

ngrams	Accuracy
Unigrams	79.2%
Bigrams	58.6%
Trigrams	47.8%
4-grams	47.2%

Loss Function	Accuracy
hinge loss	82.4%
squared hinge loss	82.0%

Table 3.2: Accuracy for different values of parameters – Amazon + IMDb

C	Accuracy
0.01	41.6%
0.1	59.6%
1	84.4%
10	83.2%
100	84.0%

ngrams	Accuracy
Unigrams	79.6%
Bigrams	69.2%
Trigrams	43.6%
4-grams	42.0%

Loss Function	Accuracy
hinge loss	84.4%
squared hinge loss	81.6%

Table 3.3: Accuracy for different values of parameters – Amazon

As we can see from the tables above, the following parameter values produce best results:
C = 1 *n-grams=Unigrams* *Loss function=hinge loss*

Sentiment Analysis on Enron emails

Model Agreement:

After initial attribute check and parameter tuning, the final step for the selection of model involves checking for model agreement. In this step we manually tag 6 random emails from Enron dataset, run all the 7 classifiers on these mails and look for two things:

- Prediction similarity to the manual tagging
- Prediction similarity to amongst the models

Mails	Manual	All	Amazon + IMDb	Amazon	Amazon + Yelp	IMDb + Yelp	IMDb	Yelp
sorry i've taken so long...just been trying to fend off the chicks. life issooooo hard sometimes.MONKEY !!!! !!!!!!!!!!!!	1	0	0	1	0	0	0	0
Monkey;Hey you little bastard what the fuck are you doing in a picture inE-Company??? What do you think that should help you score women. How doyou say B A L A N C E SHEET!!!!!!!!!!!!!!!!!!!! !!!!!!!!Yeah MonkeyB	0	0	0	0	0	0	0	0
You are hereby invited to the tenth annual Spectron/Enron Celebrity Tony'sdinner featuring Brian Tracy John Arnold and Mike Maggi.Regrets only	1	1	1	0	1	1	1	1
please schedule a round of interviews with john griffith with scott hunterphillip and tom asap (today if possible).thx	1	0	0	0	0	1	1	0
what's this about?	0	0	0	0	0	0	0	0
Ken Lay and Jeff Skilling were interviewed on CNNfn to discuss the successionof Jeff to CEO of Enron. We have put the interview on IPTV for your viewingpleasure.	1	1	1	0	0	1	1	0

Table 4: Highlighting the models with highest agreement with manual tags and amongst themselves

Sentiment Analysis on Enron emails

The models predicted the sentiments of a small set of emails (in *Table 4*). The table contains emails, a column manually tagged, and 7 predictions made by different models. We see the models IMDb + Yelp and IMDb have correctly guessed 5 out of the 6 emails. The predictions from these two models are closest to our tagging. The IMDb component of the IMDb + Yelp model is better suited for predicting sentiment in Enron emails, but we choose IMDb + Yelp for a broader range of words.
The final part of choosing our model is based on how good the models have been in predicting on their own data. After performing parameter tuning of the models, we take the approach from the research paper Thumbs Up? and add a pre-processing step of negation tagging. We implemented a negation detection function, that prefix a 'not_' tag to every word between a negation word and a punctuation symbol.

Example: "I am not happy today, and I am not feeling well."
When the above line is passed through the negation detection function, it returns the following:
I am not not_happy not_today and I am not not_feeling not_well

We have observed marginal changes in the accuracy after successfully implementing the above function.

Comparison between accuracies for the models using negation methods

We now combine our learnings from above above conducted experiments and by other teammates to choose the final model. The choice is the model trained on IMDb+Yelp data for the following reasons:

- Precision and Accuracy of 83% suggests correct classification on most occasions but also maintaining generalisation and avoiding overfitting of data
- Closer to manual tagging when compared to other models

Ayushi Patel
A20407392

Sentiment Analysis on Enron emails

RESULTS

The test was run on a total of 40k mails on 3 machines with the content ranging from personal to official to informative. We manually analyzed nearly 500 mails each to understand how the model performed on a data never seen by it. We observed at least 3-4 mails tagged correctly in every 10 mails viewed.
Words containing outright positive words were correctly classified as positive. The following list of top 15 positive and negative words with their coefficients makes it easier to understand how the model may have classified the mails:

Feature	Coefficient
great	2.77102
good	2.719699
amazing	2.053379
delicious	1.936535
best	1.865834
love	1.82636
beautiful	1.751627
wonderful	1.679543
sweet	1.646791
loved	1.564066
awesome	1.560378
cool	1.560316
perfect	1.558635
funny	1.52985
nice	1.527561

Feature	Coefficient
bad	-2.62754
not	-2.51333
worst	-2.22542
no	-1.97377
never	-1.90758
disappointed	-1.82183
bland	-1.81952
terrible	-1.79709
fails	-1.71277
average	-1.69196
awful	-1.63724
poor	-1.58604
stupid	-1.58362
off	-1.57653
slow	-1.4547

Following are the positive and negative mails classified correctly:

'Can you also approve Mike Maggi to trade crude as well. Thanks for your help.'	1
'other question and reason i dont do anything with jan/feb is whats gona makethemkt bearish the feb? perception is stx get titire so inverses grow.. onlythingi can think of is will they get concerned over this industrial slowdown goingforward and weather going above-i struggle generally tho is weather was stillsowarm last year hard to get overly bearish rest of the winter from a y on ystandpoint'	0

Sentiment Analysis on Enron emails

The words appearing in the above two examples clearly depict that the model can classify mails with positive or negative words. This can be attributed to the use of unigrams in creating the vectorizer of the corpus.

There are also instances of misclassification by the model as shown below:

Your words of encouragement are greatly appreciated. I've certainly had sometroubles this quarter. I do appreciate your offer but I don't want to takeaway from the amazing year you've had so far. Maybe you should come tradethis...	0
I am starting options on EOL in about two weeks. As we discussed earlier, Idon't have the appropriate manpower to run this in certain circumstances,such as when I'm out of the office. As such, I'd like to bring in JohnGriffith for anohter round of interviews for an options trading role withyour permission.	1

The above misclassifications point to the shortage of vocabulary of model to be able to classify such instances as well.

CONCLUSION

Based on the above observation we can see a clear need of increasing in training data. The model currently classifies 30-40% of unseen data correctly. The misclassification can be reduced by training the model with more labelled data. Training the model on just 2000 labelled data points and with an output of correct classifications for 4 out of every 10 mails is a positive pointer. Addition of more training data from websites like Amazon.com, IMDb and many more would have a visible impact on accuracy of the model and we can expect better classifications of email. In an office setting, a very plain and precise language is used to communicate ideas which can be perceived as positive or negative in a very subtle manner. Our model is trained on data which has loud choice of words and extremities at both end. To enable our model in classifying those subtle messages, training with a labelled dataset of emails can improve the performance. The approach of using a linear classifier model alone can achieve good results. The more data it is trained with, better results can be expected.

FUTURE WORKS

As mentioned in the conclusion, a direct effect would be noticed on model performance if it is trained with mode data to improve its vocabulary. Another approach could be to bootstrap the pre-compiled lexicon to the model and then train it further with more labelled datasets. This should show marked improvement in performance as well as reduce misclassification as well.

Team Contribution Summary:

Team Member	Datasets
Vaibhav Sharma	Amazon / Amazon + IMDb / Enron Emails
Ayushi Patel	IMDb / IMDb + Yelp / Enron Emails
Chandana Ravindra Prasad	Yelp / Yelp + Amazon / Enron Emails
The model Amazon + IMDb + Yelp was tested by all the team members	