COMP551: Mini 3

Edwin Zhou (260988798), Harry MacFarlane (260991258), Yongru Pan (261001758)

November 2023

Abstract

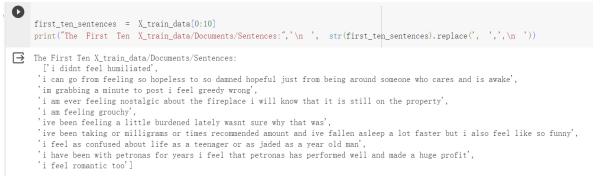
Emotions are different are hard to quantify. Therefore, for a machine to truly understand them, the use of deep learning is highly recommended. Our results support the fact that we can achieve better accuracy when trying to predict emotions of text when using deep learning models versus using tradition machine learning methods.

1 Introduction

.

1.1 Dataset Origin

The follow information is presented and fully detailed in the introduction of *Saravia and Others, 2018*(ref 1). Data was found through twitter's API, and was mixed by combining subjective tweets from normal user and objective tweets from news accounts. It was then tokenized by removing extra white spaces, turning all the letters into lowercase as well as pre-processed removing usernames and urls to reduce bias. Using an graph methods and pattern extraction, we were left with processed text documents with emotion labels, which were assigned loosely based on the hashtags used with the documents as well as a sentiment score based on the text.



```
first_ten_sentences = X_train_data[0:10]
print("The First Ten X_train_data/Documents/Sentences:","\n ", str(first_ten_sentences).replace(', ',',\n '))
```

```
The First Ten X_train_data/Documents/Sentences:
 ['i didnt feel humiliated',
 'i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake',
 'im grabbing a minute to post i feel greedy wrong',
 'i am ever feeling nostalgic about the fireplace i will know that it is still on the property',
 'i am feeling grouchy',
 'ive been feeling a little burdened lately wasnt sure why that was',
 'ive been taking or milligrams or times recommended amount and ive fallen asleep a lot faster but i also feel like so funny',
 'i feel as confused about life as a teenager or as jaded as a year old man',
 'i have been with petronas for years i feel that petronas has performed well and made a huge profit',
 'i feel romantic too']
```

Figure 1: Text Examples

Emotions (label)	**Amount**	**Hashtags**
Sadness (0)	214,454	depressed, grief
Joy (1)	167,027	fun, joy
Fear (2)	102,460	fear, worried
Anger (4)	102,289	mad, pissed
Surprise (5)	46,101	strange, surprise
Trust (NA)	19,222	hope, secure
Disgust (NA)	8,934	awful, eww
Anticipation (NA)	3,975	pumped, ready

Table 1: Table 3 Data Statistics from CARER Paper with label

*Love was not included in this table, it was given the label 3

1.2 Goals

In our paper, we choose to instead implement a naive bayes and BERT model instead of a CNN to predict emotion. For our purposes we used methods like **countvetorize1** from *scikit-learn* to make numerical features for the naive bayes model and use transformers package from *pytorch* to tokenize the input and transform it into numerical features as well for the BERT models.

2 System Models

We trained a total of 3 models.

2.1 Naive Bayes Model

Using Bayes rule for classification, we can train an AI model that uses the probability distribution of its features in order to calculate probabilities for different classes. Among these types of models there are two basic models that can do text classification, **Guassian** and **Multinomial**.
We chose to use **Multinomial** Naive Bayes as it is most often used for text classification and is more reliable compared to **Guassian** Naive Bayes, which is limited by restrictions of its means and standard deviations.
However, it should be mentioned that the independent feature likelihood assumption that Naive Bayes relies on is quite weak in this case due to the obvious possible correlation between word inclusions in documents, words whose inclusion suggest a certain emotion are more likely to accompanied with words that also suggest that same emotion. The pattern matching work done in the pre-processing in the CARER paper helps offset this, but it is still a source of variance. However, *Domingos and Pazzani 1997* (ref 2) implies that for classification problems, the dependency of likelihood for features doesn't significantly compromise the results of the classification.

2.2 Pre-Trained BERT

Another model we can use for classification are transformers such as the BERT model (Bidirectional Encoder Representations from Transformers). The latter is composed of a neural network architecture that processes text in a bidirectional manner. As a result, unlike a Bayes model, it can be used to learn from sequential data and therefore keep context of words within a sentence.

For this classification experiment, the pre-trained BERT model bert-base-uncased-emotion was used.
This model was chosen as it had a high accuracy on the emotion dataset without tuning (92 percent). It also had a classification head of 6 emotions which meant that the model's layers would not need to be processed before being used and tuned.

2.3 Fine-Tuned BERT

Possible changes we could make include chaging the number of hidden layers, the number of attention heads, activation functions, hidden size of encoder, pooler and intermediate layers, dropout rate of attention layers, etc.
To tune the model, the pretrained BERT model was loaded as is, and fed the same processed emotion dataset used to train the Naive Bayes model for 3 more epochs using the following hyperparameters:

Parameter	Value
Number of Training Epochs	3
Per Device Train Batch Size	16
Per Device Evaluation Batch Size	16
Weight Decay	0.01
Evaluation Strategy	steps
Load Best Model at End	True
Learning Rate (AdamW)	5e-5

Table 2: Tuning Hyper-parameters for BERT Model

There were a few reasons behind these choices of parameters. Since the model already had an accuracy of 92, the goal was only to improve it by a few percent. By looking at the original training parameters, we can see that the model was trained with batch size 64 and learning rate 2e-5 over 8 epochs. Since there were some memory issues for training with larger batches, we have decided to use a batch size of 16, but train for more than 1 epoch. Then, the learning rate was increased in order to not have the model stagnate at the same accuracy. This higher learning rate was counterbalanced with the introduction of a small weight decay to make sure none of the weights become too high. In addition, we did not need to worry about a high learning rate as much since the best performing model over the validation dataset would be saved and loaded at the end of training.
As a result, the BERT model testing accuracy was improved by 1 percent.

3 Experiments

3.1 Dataset Analaysis

As outlined in the introduction, the data is text data with labels denoting different emotions. The dataset contains documents expressing 6 emotions: Surprise, Fear, Anger, Love, Joy and Sadness. Looking at the distribution of classes, we have a pretty skewed data set. (See Figure 2)

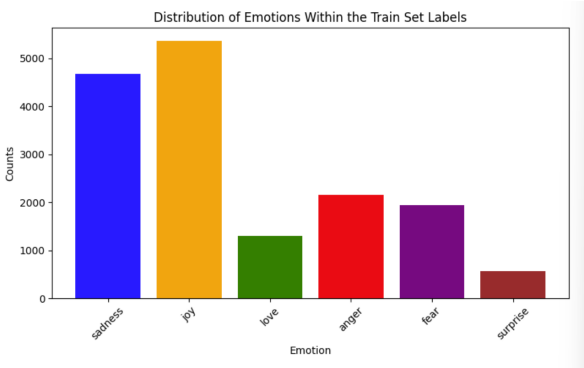


Figure 2: Distribution of Emotions (Classes)

There seems to be a larger amount of joy and sadness documents, 33.5 and 29.2 percent respectively. On the other hand, there are relatively smaller amounts of surprise and love documents, 8.2 and 3.9 percent. Unfortunately, without an accurate measure of general frequency of emotion related to what emotion an individual expresses over text, there is not much information to be gleamed from this. It probably will become a source of slight bias for classification perhaps, something to keep an eye on.
However, there does seem to be an outlier as far as frequency of individual words. (See Figure 3)

We can see that "like" is by far the most common word in the documents, which makes sense as it can be used for comparisons and to express joy in our case. This could lead to bias towards the joy class, which we should look out for.
To use '**countvectorize1**', we need a **stopwords list**, words that shouldn't be used in our analysis, obtained from the NLT library of words as it contains more stop words for english than the default

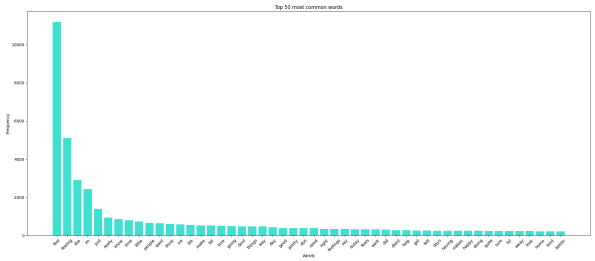


Figure 3: Word Frequency in Data set

countvectorize1's **stopword list**. We also added our own custom words. Once we used *countvectorize1*, we transformed the data into presentation of whether or not a certain word among the list of feature words created was present or not in each of the documents.

```
first_ten_sentences = X_train_data[0:10]
print("The First Ten X_train_data/Documents/Sentences:","\n ", str(first_ten_sentences).replace(', ',',\n '))
```

```
The First Ten X_train_data/Documents/Sentences:
 ['i didnt feel humiliated',
'i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake',
'im grabbing a minute to post i feel greedy wrong',
'i am ever feeling nostalgic about the fireplace i will know that it is still on the property',
'i am feeling grouchy',
'ive been feeling a little burdened lately wasnt sure why that was',
'ive been taking or milligrams or times recommended amount and ive fallen asleep a lot faster but i also feel like so funny',
'i feel as confused about life as a teenager or as jaded as a year old man',
'i have been with petronas for years i feel that petronas has performed well and made a huge profit',
'i feel romantic too']
```

(a) Before

```
Document 0 word counts:
  create: 1
  easy: 1
  giving: 1
Document 1 word counts:
  eat: 1
  gave: 1
  general: 1
  hard: 1
Document 2 word counts:
  easy: 1
  fat: 1
  gonna: 1
  issues: 1
  lucky: 1
  stunned: 1
Document 3 word counts:
  eat: 1
  healthy: 1
  laugh: 1
Document 4 word counts:
  eat: 1
  fault: 1
Document 5 word counts:
  blog: 1
  eat: 1
  happen: 1
  held: 1
  hoping: 1
  really: 1
  soo: 1
Document 6 word counts:
  easy: 1
  excited: 1
  happen: 2
  honest: 1
  idiotic: 1
  recently: 1
  run: 1
```

(b) After

Figure 4: Before & After Vectorization

As shown by the figures above, there is a noticeable difference between the data fed into the BERT model and the unprocessed data obtained from the Emotion dataset. The tokenized sentence is composed of 3 tensors. The input ids tensor is of a fixed size where each word in the sentence is represented by a unique numerical identifier. To differentiate between an actual token and padding in the input ids tensor, the attention mask element is used. The third element of the tokenized sentence is the label which stays the same. This way, the BERT model can process any sentence up to a maximum length while encoding the sentence into a tensor keeps the spacial context of the words in the sentence.

```
Original Dataset:
Text: i didnt feel humiliated
Label: 0
```

Figure 5: Example in Emotion Dataset

3.2 Results

As we can see, the fine tuned Bert model clearly has the best performance of all the models. This makes sense as our Multinomial Naive Bayes model while good, still suffers from a dependency in its likelihood features, the effect may not be major, but it is still present (Domingos and Pazzani, 1997). On the other hand, our pre-trained BERT model has better accuracy than the Naive Bayes model,

```
Tokenized Dataset:
Input IDs: tensor([  101,  1045,  2134,  2102,  2514, 26608,   102,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0,     0,     0,     0,     0,     0,     0,     0,     0,
            0,     0])
Attention Mask: tensor([1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
```

Figure 6: Example in tokenized Emotion Dataset

Model	**Accuracy**
Multinomial Naive Bayes	86
Pretrained BERT	92
Fine Tuned BERT	93.9

Table 3: Comparing Accuracies of different models

but fine tuning BERT for our specific data set obviously leads it to be better at classification than it would be without modification.

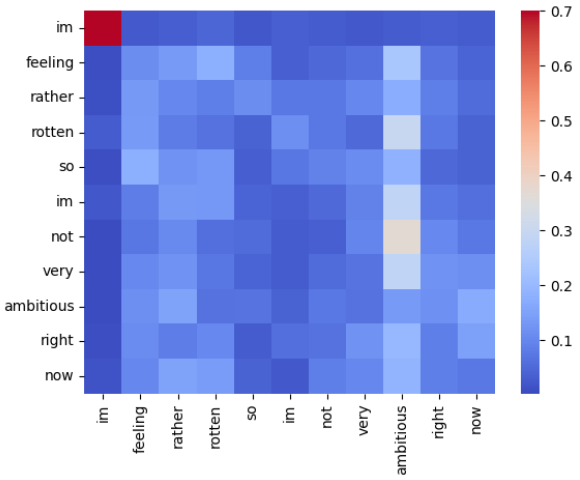


Figure 7: Correctly labeled sadness

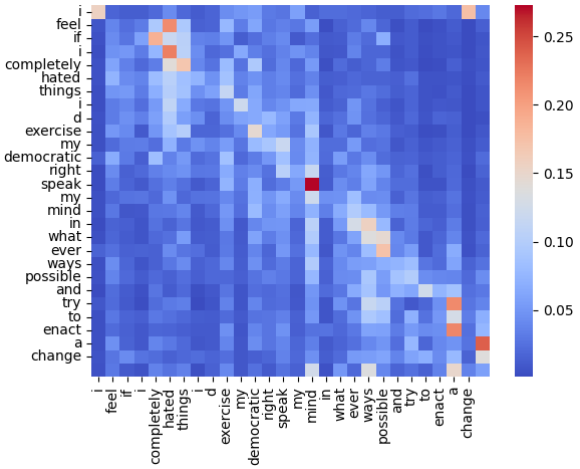


Figure 8: Incorrectly labeled sadness when is anger

Sadness	Joy	Love	Anger	Fear	Surprise
0.5158	0.0040	0.0036	0.4744	0.0008	0.0014

Table 4: Probabilities of Incorrectly Labeled Example

When looking at the attention matrix of the first transformer block and last attention head for two different predictions, it helps us understand how the BERT model made its decisions. For the correct label, it can be seen that there was a big emphasis on the term "I", but also on the relation between "not" and "ambitious". These key terms and relationships immediately evokes sadness which the model accurately labeled. Looking at the matrix of the incorrect labeling, we see that it is not immediately clear which term and relations are most important. There are many red points on the figure meaning that there were many "attention hotspots" within the sentence. Another thing to notice is that the hotspots on the matrix seems to form an identity matrix, where each word is mainly correlated to itself or its neighbors. This could be explained by the fact that the model wasn't successful at finding the context of the different words in the sentence, thus not being able to find an obvious emotion class to label it to. In fact, we see that it was a quite ambiguous decision by the model as the difference between labelling as anger vs sadness was 0.04.

It would be interesting to see however if we could possibly catch up to Bert's effectiveness without

over-fitting our Multinomial Naive Bayes model. After setting up the model, we tried increasing the number of features and seeing the effects it had on our validation sets.

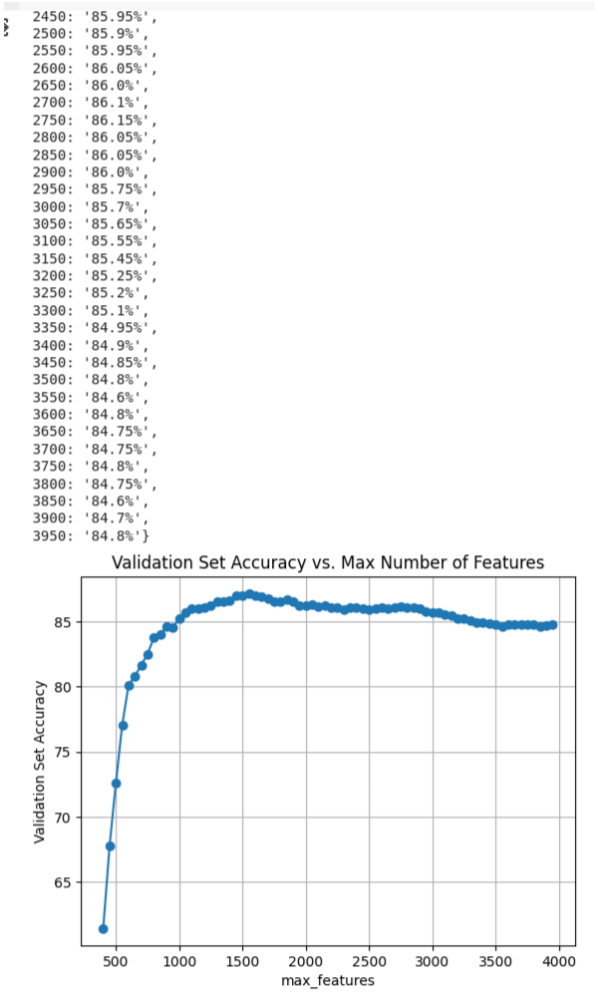


Figure 9: Evolution of Validation Set Accuracy with Max Features

Our accuracy continues to climb until 85% accuracy, but it still cannot catch up to BERT's effectiveness, even if we add as many features as we can. (See Figure 9)

4 Conclusion

For the BERT model, we conclude that using the pre-trained model is a good approach for this dataset, primarily because the accuracy was above 90 percent out of the box, which allowed us to minimize our efforts for fine-tuning. This leads to quicker training time as we only needed 3 epochs compared to the 8 that was used for the pre-training. Additionally, it doesn't require as much data for fine-tuning since the model was already pre-trained on a large corpus, enabling it to understand language context and semantics before it is further trained on the specific dataset in question.

We can see from our results that deep learning (BERT) gives us better results, helping us classify emotions more accurately than the traditional machine learning technique used in our Naive Bayes model.

5 References

1) Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized Affect Representations for Emotion Recognition. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 3687–3697, Brussels, Belgium. Association for Computational Linguistics.
2) Domingos, P., Pazzani, M. On the Optimality of the Simple Bayesian Classifier under Zero-One Loss. Machine Learning 29, 103–130 (1997). https://doi.org/10.1023/A:1007413511361