Emotion Classification Using Emotion Distribution Vectors and BERT

Shayna Gaulden
CS274 — Dr. Teng Moh
San Jose State University
San Jose, USA
shayna.gaulden@gmail.com

***Abstract*—This project is an empirical study of emotion classification using Bidirectional Encoder Representations from Transformers (BERT). There are two main focuses of this study. The first is exploring existing preprocessing solutions to determine their impact on classification performance. Choices made during preprocessing have a direct and large impact on the downstream classification task. The second focus is to consider how the class memberships to each instance in a data set are represented. This project will take a data set with single-label emotion classes and transform the single-label binary membership vectors into continuous valued emotion distribution vectors. The findings of this study will show that when using a transformer model the data should not be over-processed with steps such as stop word removal. The conversion from single-label to a distribution vector does not improve overall classification quality, although it has special interpretations regarding a richer representation of the data, and can be used to increase the classification accuracy for specific emotion classes. The link to the code used in this project can be found in a shared Google Drive https://drive.google.com/drive/folders/16RTafl9lsX2TTBURwin2vYHvBFaGzhjn?usp=share_link.**

***Index Terms*—classification, natural language processing, emotion detection, bert, preprocessing, NLP, emotion distribution vectors**

I. Introduction

There is a large increase in interactions that occur online through written communication. Whether it is on social media posts and comments, writing reviews, chatting with customer service, emails, blog posts, news articles, or chat applications, there is a need to automatically classify emotions on a large scale. People can usually recognize the emotion behind the text, but this is a complex task for even state-of-the-art algorithms and models. Emotion recognition assigns an example to a specific emotion class such as 'Joy' or 'Anger'. On the other hand, sentiment analysis only focuses on whether the example is more positive or negative. Emotion and sentiment recognition of text can be used for tracking emotions throughout a chat [1], incorporating a user's emotions into a recommender system [2], and it is often used on data sets from social media posts or comments.

How data is represented is an important and sometimes overlooked element of classification models. Emotion classification is a natural language processing task. Segments of text, referred to as documents, must be represented in a way that retains information from the raw data but can also be processed by a classification model. The way the documents are represented has a big impact on the downstream classification task because it directly affects the features fed into the model. There are also different ways to represent class membership. Single-label allows a document to be a member of only one class, multi-label allows a document to be a member of several classes simultaneously, but class members can also be represented by a membership grade that is a continuous value representing a document's belonging to each class. This project is aimed at exploring the effect of data representation in both the preprocessing task and the class membership representation on the overall classification quality. F1 score, accuracy, precision, and recall will all be used to determine a model's classification quality.

II. Related Study

A. Preprocessing

Emotion classification of text data requires preprocessing and cleaning to get the data set into something usable for classification. During this step decisions and assumptions are made that will have a direct impact on the performance of the downstream tasks and it is crucial to know whether this impact will be positive or negative. V. S. Kodiyala and R. E. Mercer [3] perform extensive preprocessing to maximize the information contained in the short pieces of text usually used in emotion classification. The most notable improvement was finding ways to get information from emojis and emoticons, as well as extracting more information from slang words and abbreviations. The extent of their preprocessing steps is described below.

- Removing any URLs.
- Reducing the maximum length of repeated characters to three.
- Replacing emojis and emoticons with a verbal description.
- Translating slang into proper English.
- Expanding word contractions and abbreviations to their full form.
- Removing all punctuation except for "!" and "?".
- Removing a customer list of stop words.
- Performing lemmatization.

All classification tasks of text data involve some preprocessing, but it is not always considered as part of the model, or

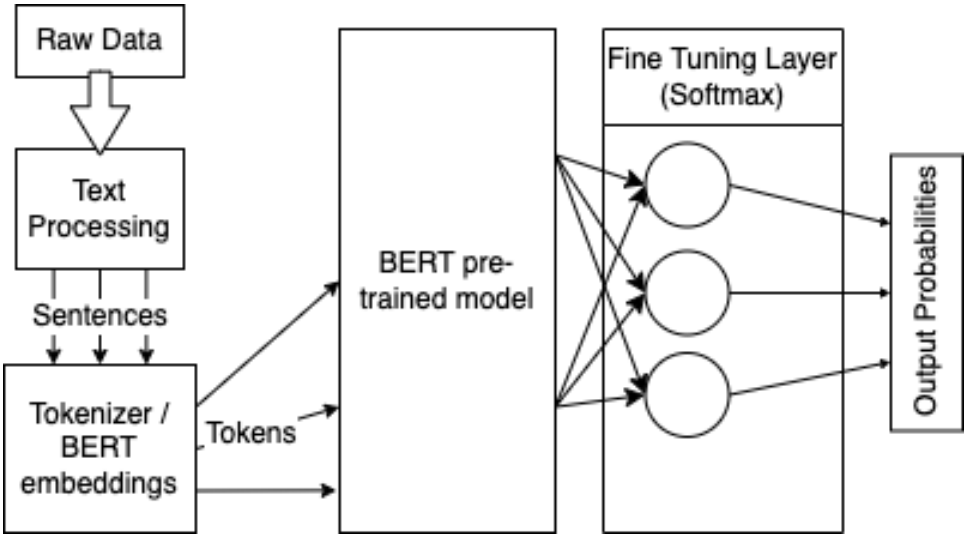

Fig. 1. A diagram of the baseline model to be reproduced on the CBET dataset.

described in detail as much as V. S. Kodiyala and R. E. Mercer describe what has been done to the dataset. Another researcher who aims to get more information from emoji use is C.-H. Chen et al. [1]. They analyze chat data that involves heavy usage of emojis, so to avoid losing the information they assign a vector related to emotions to each emoji and incorporate this into their classification model.

Classification can learn bias when associations are created to certain words indicating membership to a specific emotion class. To address this D. Demszky et al. choose to mask proper names with [NAME] and religious terms with [RELIGION] [4]. this prevents specific names or religions from being assigned emotional values.

B. Emotion Representation

To assign an emotion to an example the emotion must be represented in some way. The most natural representation is to assign that emotion a label like "happy" or "sad". There are three main ways to assign an emotion to a text using single-label, multi-label, or a distribution of emotion representation. Single-label emotion assignment restricts an example to be assigned only one label from a set of multiple different emotion classes. This is seen in the research by V. S. Kodiyala and R. E. Mercer [3], and A. G. Shahraki and O. R. Zaiane [5]. Single-label emotion assignment is not always realistic because a given text can express multiple emotions at the same time. Multi-label emotion assignment fixes this issue by allowing a given example to be assigned multiple emotions at once. The drawback to the multi-label approach is it further complicates the classification process. X. Guo et al. [6] use an approach that is not single or multi-label by considering each unique label combination as its own class. This transforms the multi-label problem into a single-label problem that can then be solved with single-label classification models. For example, if a given text displays 'anger' and 'sadness' the label will be 'anger + sad'. D. Demszky et al. [4], uses the full multi-label approach allowing examples to be assigned multiple labels at once.

The third representation of emotions is to represent them as a distribution. This technique is the most realistic so far, but also the most complex. The multi-label classification does not capture the depth of human emotions. Someone can be very happy because they just won the lottery or they can be happy that it has stopped raining. These two examples both express happiness but one is much stronger than the other. To account for this Z. Li et al. [7], D. Zhou et al. [8], and C.-H. Chen et al. [1] use numeric vectors to represent emotions. The elements of the vectors typically are between 0-1 and in some way represent the strength of the emotion.

Emotion categories are often based on P. Ekman's [9] work identifying six primary emotions–anger, surprise, disgust, enjoyment, fear, and sadness–or R. Plutchiks' [10] wheel of emotions that has eight primary emotions organized on a wheel based on similarity to each other. A. G. Shahraki and O. R. Zaiane [5] have based the categories in their data set on a combination of Ekmans and Plutchiks' work. Emotion distribution vectors are typically based on the three-factor theory, an idea extended by J. A. Russell and A. Mehrabian [11], that there are three main dimensions to describe an emotion. The dimensions are shortened to VAD, V for valence measuring pleasure to displeasure, A for arousal measuring active to passive, and D for dominance measuring dominant to submissive. Within these dimensions, categories of emotions can be mapped out. Z. Li et al. [12], D. Zhou et al. [8], and C.-H. Chen et al. [1] all base their emotion distributions on VAD vectors.

C. Classification Techniques

Text classification can be done with a variety of different techniques. For single-label classification, X. Guo et al. [6] use a single-label representation and can compare the results using Support Vector Machines (SVM), logistic regression,

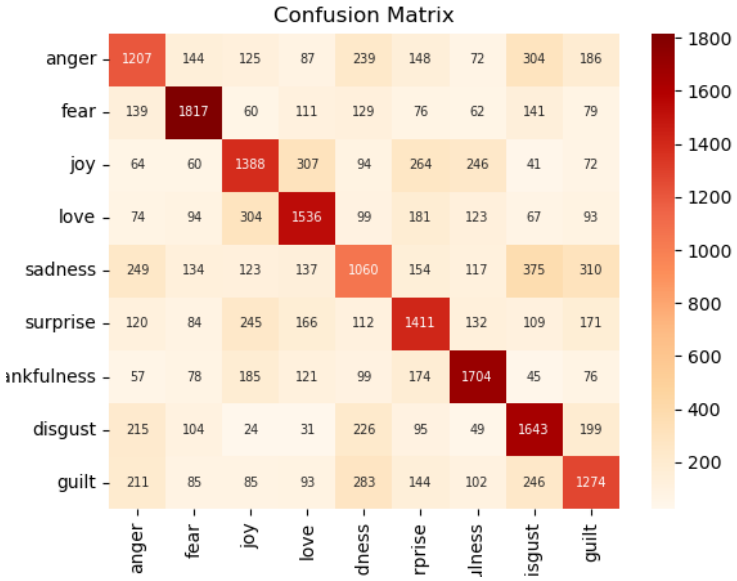

Fig. 2. A confusion matrix showing the results from the reproduction of the baseline.

and random forest to classify emotions. A. G. Shahraki and O. R. Zaiane [5] perform emotion classification by using nine distinct SVM classifiers trained on each individual nine emotions. A. Breitfuss et al. [2] implement emotion classification using a Bayesian classifier to incorporate emotion information in a movie recommender system. M.-L Zhang and Z.-H. Zhou [13] propose a new loss function for neural networks that is designed specifically for multi-label text categorization. D. Demszky et al. [4] and V. S. Kodiyala and R. E. Mercer [3] both use a pre-trained model called Bidirectional Encoder Representations from Transformers (BERT) to classify emotions. In [8] the dataset is multilabel, while in [3] the data is single-label and is implemented on both sentiment and emotion classification data. Z. Li et al. [12] find a Word-level Emotion Distribution (WED) vector for each text and then use a Convolution Neural Network (CNN) to assign and group text into sentiments. C.-H. Chen [1] cluster each text based on a two-dimensional emotion distribution vector and then assign the emotions to each cluster. The choice of classification technique can be limited by how the emotions are represented in the data.

III. Existing Solution

A. Architecture

The process to clean the text data proposed by V. S. Kodiyala and R. E. Mercer [3] is used as a baseline for this project. In their work, they compare their technique to seven different data sets to show that it is robust in classifying a variety of data. Three of these data sets out of the seven are sentiment classification specific and four are emotion classification specific. In this project, only the emotion classification data will be considered. First, V. S. Kodiyala and R. E. Mercer [3] provide extensive preprocessing to prepare the data and maximize the usefulness of the lexical information contained in the text. Next, they use BERT for emotion classification and sentiment analysis. They have added an output layer using softmax activation that produces probabilities of each example belonging to each emotion class. Then predictions are made by choosing the emotion that maximizes the output probabilities. Refer to Figure 1 to see a visual representation of this architecture. Their results show improvement over previous research for a variety of data by using their preprocessing framework and applying a BERT model with fine-tuning layer for classification. They found that BERT was able to perform much better than a typical convolution neural network.

BERT is a popular choice for text classification and other machine learning tasks because it is freely available and performs well over a variety of different tasks. A team of Google researchers, including Devlin et al., [14] developed BERT in 2019. BERT is a pre-trained model that is able to learn contextual relationships between words. It can be used for a variety of tasks including classification by adding a new output layer in the BERT model. Because BERT is pre-trained, only the weights for the output layer are learned from scratch during training and all other weights are fine-tuned.

BERT can be broken down into the following key steps: tokenizing and embedding input, transformer layers, and the fine-tuning component. First, a sequence of embeddings for each document is generated. Sentences are tokenized with WordPiece embedded, which breaks words into sub-pieces that allow for the model to handle words that do not exist in the pre-trained vocabulary. Each sentence will be prepended and appended with a special token to indicate the beginning and ending of a sentence respectively. Then there are a series of transformer encoder layers. Transformer models come from the research by Vaswani et al. [15]. These layers use a

mechanism called attention which allows the model to learn relationships between words and capture information about the position of the word in the sentence. The transformer encoder layers are the pre-trained part of the model. After pre-training, a fine-tuning layer can be used for classification. The architecture of the fine-tuning layer used in this paper includes a fully connected neural network with 120 neurons and RelU activation, then a preceding softmax output layer. The size of the output must match the number of classes. The length of the tokens is chosen to be of size 120.

B. Reproduction of Baseline

Out of the seven data sets used by V. S. Kodiyala and R. E. Mercer [3] this project will be using the Clean Balance Emotional Tweet (CBET) data set [5]. This data set was chosen because it has classes of emotions while some of the other data sets only have sentiment information. CBET data set is also the only emotion data set that was found publicly online. To evaluate the quality of the model predictions Precision, recall, F1-score, and accuracy are used.

Table I shows the results between the baseline and the reproduction. The table is color-coded red and green to indicate a lower or higher difference in the comparisons. The reproduction is similar to the baseline but has a low performance. Sadness and surprise had much higher metrics in the baseline model over the reproduction. The differences in the reproduction may be due to some significant differences in preprocessing. The preprocessing and cleaning steps used by V. S. Kodiyala and R. E. Mercer [3] have been replicated as closely as possible. The reproduction most likely differs from the results in [3] because the code is not publicly available and the methods were reproduced only by following what is explicitly stated in the article. The Python libraries and exact implementation used for the preprocessing steps may be different. V. S. Kodiyala and R. E. Mercer [3] use a custom dictionary to translate slang words and extend abbreviations to their full form. This dictionary is linked to a Github that does not appear to exist anymore. A custom abbreviation dictionary was also implemented in this project, but there is no guarantee it is similar. Additionally, results could be different due to the randomness when splitting the data set into a training, testing, and validation data. It is very unlikely that in the reproduction of the baseline, the data has been split in the same way.

The results in the original baseline and the reproduction may seem low when compared to other classification tasks but in emotion classification, it is common to see results similar to this due to the complexity of the task. Emotion classification often involves a large number of classes and emotions that are interrelated causing problems during training and prediction. Figure 2 uses a confusion matrix to demonstrate that in general most emotions have been classified correctly in the reproduced baseline, despite a large number of misclassifications.

TABLE I
A COMPARISON SHOWING THE RESULTS FROM THE BASELINE MODEL AND THE REPRODUCTION OF THE BASELINE PERFORMED FOR THIS PROJECT ON THE CBET DATA.

	V. S. Kodiyala and R. E. Mercer [3]			Reproduction		
Emotion	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Joy	54.4	51.2	52.7	48.0	51.7	49.8
Sadness	70.3	74.0	72.1	69.5	69.9	69.7
Anger	57.7	55.6	56.6	54.7	54.7	54.7
Love	62.7	65.4	64.0	59.7	59.3	59.5
Thankfulness	49.9	49.9	49.9	39.9	45.3	42.4
Fear	62.3	58.8	60.5	55.3	53.3	54.3
Surprise	71.3	74.4	72.8	67.1	65.4	66.2
Guilt	63.2	61.6	62.4	63.5	55.3	59.1
Disgust	55.9	59.1	57.4	56.5	56.3	56.3
Average	60.8	61.1	60.9	56.5	56.3	56.3

IV. New Solution

A. Architecture

The idea proposed in this project can be broken down into the following.

- **Preprocessing** is done the same as baseline.
- **Translate** the labels to emotion distribution vectors
- **BERT** used as the classifier with the emotion distirbutionn vectors used as targets.
- **Translate** the predicted emotion distribution vector back into single-label data, for evaluation.

The idea to translate the single-labels into emotion distribution vectors comes from Z. Li et al. [7]. This solution was expected to improve the results of the baseline model because it represents the emotions more realistically while still keeping most of the original architecture of Kodiyala and R. E. Mercer's model. It is also thought that this representation of the label space allows more flexibility and retains more information about each document. The architecture of the proposed solution and how it is structured with the original architecture can be seen in Figure 3. Detailed steps of the new solution are outlined below.

1) **Step one:** First the mean and standard deviation of the VAD vectors for each of the original nine emotion labels are found and fit to a multivariate Gaussian distribution following Z. Li et al. [7]. This is done by extending the nine emotions to over 200 similar emotions. For example, 'anger' becomes associated with a list of words including 'anger', 'rage', 'mad', 'hatred', and so on. A custom dictionary was created for this purpose. Once similar emotions have been found they are used to query the NRC-VAD dictionary and the average and standard deviation for the VAD vectors of all similar emotions are calculated. The NRC-VAD dictionary comes from S. Mohammad and has been made available for non-commercial research and education [16].
2) **Step 2:** Sentence-level VAD vectors are found for each cleaned and preprocessed text example by finding the average of the VAD vectors of each word in the sentence.
3) **Step 3:** The emotion distribution vector for each example is found using the multivariate Gaussian models

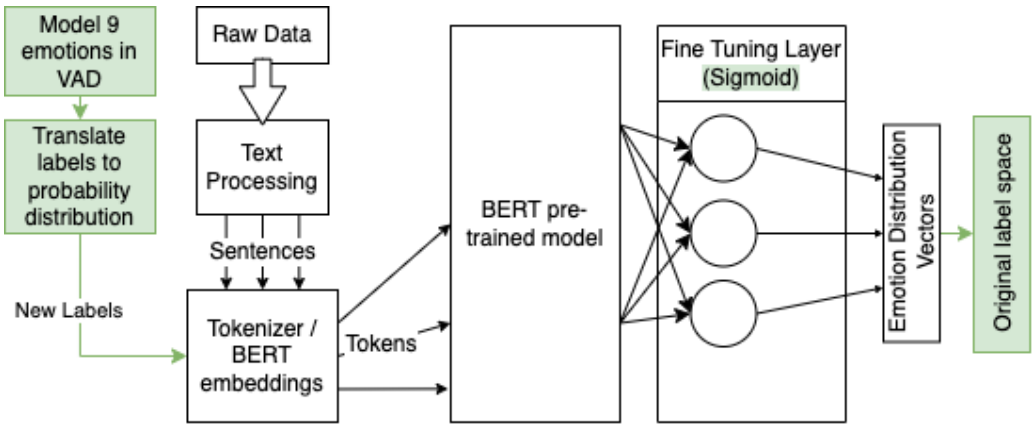

Fig. 3. A diagram showing the structure of the new method and highlighting the changes of how it fits into the existing solution by V. S. Kodiyala and R. E. Mercer [3] by coloring the changes in green.

created for each emotion. Each element of the emotion distribution vector represents instances belonging to the corresponding class. This value is the probability density of that emotion's multivariate Gaussian for the VAD vector. This gives the intensity of each emotion.

4) **Step 4:** The emotion distribution vector is then updated by multiplying weights to reduce the noise of the emotion distribution. This idea comes from D. Zhou et al. [8] who quantifies the closeness of emotions using Plutchik's wheel [10]. If emotions are at a 90-degree angle from each other on the wheel they have no relationship, emotions opposite from each other have a negative relationship, and any emotions close together have a positive relationship. By using this idea each emotion can be assigned a set of weights for how close it is to every other emotion in the label set. The weighting scheme is then based on a combination of emotions proximity in Plutchik's wheel of emotions and the original labels. Emotions that occur close in the wheel to the original emotion label will receive higher weights, and emotions more than 90 degrees of distance on the wheel will receive weights of 0. The reason weights need to be applied is because averaging the word-level VAD vector causes a significant amount of noise. Consider, the VAD vector that might be produced from the word 'birthday', this should intuitively support positive emotions. Many sentences like "Happy birthday!" would confirm this but a sentence such as "My cat died on her birthday" should not be associated with positive emotions.
5) **Step 5:** The emotion distribution is now used as the target in the BERT model. The output layer activation is changed to sigmoid so that each emotion can have a strength independent of the other emotions.
6) **Step 6:** The output of BERT should correspond to the original emotion distribution. To check this, the text is classified into a single-label class by finding the label that maximizes the predicted emotion distribution and seeing if it matches the emotion that maximizes the target emotion distribution.

Steps 1-4 show how to translate the single-label data into emotion distribution vectors. Step 5 explains using the emotion distribution with BERT and step 6 describes how to evaluate the results.

B. Results

The major change being made is transforming the labels from multi-class into an emotion distribution vector. The emotion distribution vector should still represent the original labels. To check that the emotion distribution is still a good representation of the original data set, the label distribution can be converted back into categorical labels by finding the label for each example that maximized the emotion distribution. Looking at this showed some discrepancies between the original labels and the new ones. Sometimes this change made sense such as sentences labeled as 'disgust' in the old label space might now be labeled as 'anger' and if it is a sentence that could be considered acceptable under both labels then this is an acceptable difference. But some labels did not make sense. The new and old label sets matched each other with 66.39% accuracy. The frequency of labels was examined and it became apparent that the emotion distribution vector was favoring some emotions such as "thankfulness" which suddenly occurred with disproportionate frequency. To fix this the weights being applied were altered by reducing weights for emotions that were being represented with too high of an intensity. Updating the weights changed the new label distribution so it could be matching with the old labels with 76.49% accuracy. The now updated emotion distribution vectors were used and compared to the previous iteration.

Overall results can be seen in Table II which has been color-coded with red boxes indicating the metric is lower than the alternative version, and green indicating the metric is higher than the alternative version. The emotions 'joy', 'sadness', and 'guilt' have actually improved when compared to the baseline method. All other emotions were not classified as well in the new method when compared to the baseline results. By

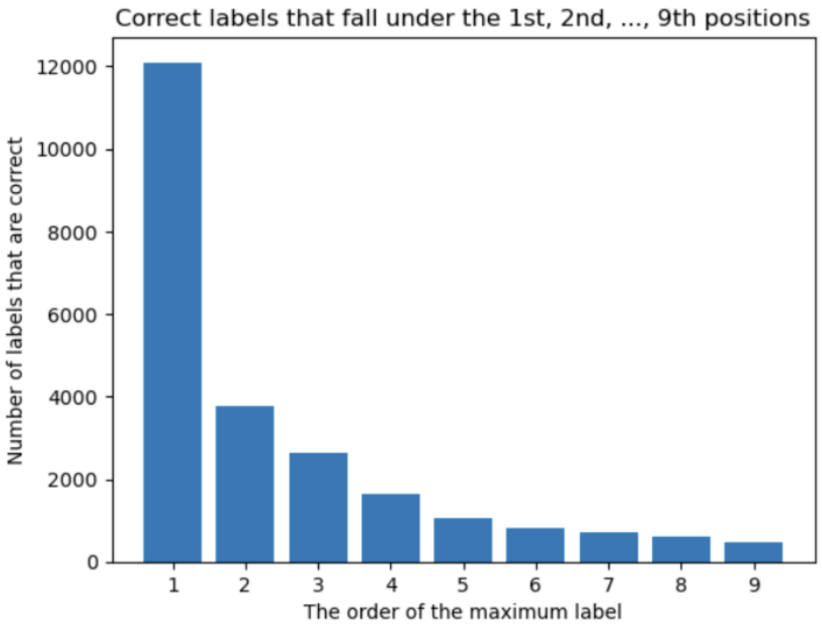

Fig. 4. This chart shows the number of correct labels that occurred in the 2nd highest probability, the 3rd, 4th, … and all the way to the 9th highest probability.

adjusting the weights in small amounts the results for the new solution could be improved or made worse. This indicates that there might be some ideal weighting based on the relationship between emotions.

TABLE II
RESULTS FROM THE BASELINE COMPARED TO THE NEW SOLUTION.

	Original Method (Baseline)			New Solution		
Emotion	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Anger	48.0	51.7	49.8	44.4	56.4	49.7
Fear	69.5	69.9	69.7	60.0	59.5	59.8
Joy	54.7	54.7	54.7	59.6	75.2	66.5
Love	59.7	59.3	59.5	43.3	55.8	48.7
Sadness	39.9	45.3	42.4	62.6	46.1	53.1
Surprise	55.3	53.3	54.3	55.8	42.8	48.4
Thankful	67.1	65.4	66.2	30.5	49.4	37.7
Disgust	63.5	55.3	59.1	41.0	60.7	48.9
Guilt	56.5	56.3	56.3	57.6	58.3	57.9
Average	56.5	56.3	56.3	50.5	56.0	52.3

An experiment was done to test the effect of stop word removal on the classification results. Originally a Python dictionary was used to remove all common English stop words. In the newer implementation, the stop words are left in. This is because BERT can learn the context in a sentence and the relationship between words, so it does better when stop words remain in the text data. Refer to Table III to compare the solution with and without stop word removal. Leaving stop words in increased performance of all metrics for all emotion classes. This strongly indicates that transformer models might perform better with preprocessing steps that add more information instead of taking information away such as stop word removal or lemmatization.

TABLE III
NEW SOLUTION COMPARING STOP WORD REMOVAL IN PREPROCESSING.

	New Solution Stop word Removal			New Solution Stop words Stay		
Emotion	**Precision**	**Recall**	**F1**	**Precision**	**Recall**	**F1**
Anger	41.3	53.6	46.7	44.4	56.4	49.7
Fear	55.3	57	56.1	60	59.5	59.8
Joy	59.1	74.3	55.8	59.6	75.2	66.5
Love	36.4	52.9	43.1	43.3	55.8	48.7
Sadness	60.7	44.9	51.7	62.6	46.1	53.1
Surprise	54.2	40.3	46.2	55.8	42.8	48.4
Thankful	29.7	47.8	36.6	30.5	49.4	37.7
Disgust	38.5	58.2	46.5	41	60.7	48.9
Guilt	55.3	54.2	54.8	57.6	58.3	57.9
Average	47.8	53.7	49.7	50.5	56	52.3

V. CONCLUSION AND FUTURE WORK

The results of this report lead to several conclusions. It was discovered that adjusting the weights affected the results and even affected which classes performed better than the baseline. There might be a better way to find an ideal set of weights. It can also be noted that the weights can be adjusted to improve all metrics of specific emotion classes. More research is needed to discover a repeatable process that can reliably increase the classification accuracy of only a few important emotions. This could be useful for applications of emotion detection if only a few classes of emotions are important to identify.

The emotion distribution vector might also be more accurate at classifying emotions but some information is lost when the emotion vectors are translated back into a single label representation. For example, a tweet could express 'joy' and 'surprise' which can be captured by an emotion distribution vector if both of the values for 'joy' and 'surprise' are close to

one. But when this is translated back into a single label if the original label is 'surprise' but in the emotion distribution vector 'joy' is slightly higher than 'surprise' this will be considered as a misclassification. Recall that the emotion distribution vector is translated back into a single-label multi-class set by choosing the label with the maximum element in the vector. Figure 4 demonstrates how many times the correct label is in the maximum, then 2nd highest, then 3rd highest, and so on. From Figure 4, the correct label is usually within the top three emotions in the distribution vector. This information is lost during the translation back to single-label. Perhaps in the future an evaluation metric that considers the ranking of top emotions could be used.

In conclusion, more work can be done to find an ideal set of weights, or possibly create a better way to translate the emotion distribution back to a single label. In addition, the removal of stop words has been show to be unnecessary when using an advanced transformer-based classification model such as BERT.

Acknowledgment

I would like to express my gratitude to Dr. Teng Moh for instructing the CS274 class and providing constructive feedback that was crucial to the implementation of this project. I am also very appreciative to the students in this class whose questions and feedback shaped the direction of this project and lead me to improving the results of my original solution.

References

[1] C.-H. Chen, W.-P. Lee, and J.-Y. Huang, "Tracking and recognizing emotions in short text messages from online chatting services," *Information Processing & Management*, vol. 54, no. 6, pp. 1325–1344, 2018, ISSN: 0306-4573. DOI: 10.1016/j.ipm.2018.05.008.

[2] A. Breitfuss, K. Errou, A. Kurteva, and A. Fensel, "Representing emotions with knowledge graphs for movie recommendations," *Future Generation Computer Systems*, vol. 125, pp. 715–725, Dec. 2021. DOI: 10.1016/j.future.2021.06.001.

[3] V. S. Kodiyala and R. E. Mercer, "Emotion recognition and sentiment classification using bert with data augmentation and emotion lexicon enrichment," in *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2021, pp. 191–198. DOI: 10.1109/ICMLA52953.2021.00037.

[4] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, "Goemotions: A dataset of fine-grained emotions," *arXiv*, Jun. 2020. eprint: 2005.00547. [Online]. Available: http://arxiv.org/abs/2005.00547.

[5] A. G. Shahraki and O. R. Zaiane, "Lexical and learning-based emotion mining from text," in *Proc. of the International Conference on Computational Linguistics and Intelligent Text Processing*, 2017, pp. 24–55.

[6] X. Guo, Y. Sun, and S. Vosoughi, "Emotion-based modeling of mental disorders on social media," in *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Apr. 2022, pp. 8–16. DOI: 10.1145/3486622.3493916.

[7] Z. Li, X. Li, H. Xie, Q. Li, and X. Tao, "A label extension schema for improved text emotion classification," in *IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, New York, NY, USA, Apr. 2022, pp. 32–39. DOI: 10.1145/3486622.3493935.

[8] D. Zhou, X. Zhang, Y. Zhou, Q. Zhao, and X. Geng, "Emotion distribution learning from texts," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas, Nov. 2016, pp. 638–647. DOI: 10.18653/v1/D16-1061.

[9] P. Ekman, "Basic emotions," in *Handbook of Cognition and Emotion*, T. Dalgleish and M. J. Power, Eds., Wiley, 1999, ch. 3, pp. 45–60.

[10] R. Plutchik, "Emotions: A general psychoevolutionary theory," in *Approaches to Emotion*, K. R. Scherer and P. Ekman, Eds., Psychology Press, Taylor & Francis Group, 1984, ch. 8, pp. 197–219.

[11] J. A. Russell and A. Mehrabian, "Evidence for a three-factor theory of emotions," *Journal of Research in Personality*, vol. 11, no. 3, pp. 273–294, 1977. DOI: 10.1016/0092-6566(77)90037-X.

[12] Z. Li, H. Xie, G. Cheng, and Q. Li, "Word-level emotion distribution with two schemas for short text emotion classification," *Knowledge-based systems*, vol. 227, 2021. DOI: 10.1016/j.knosys.2021.107163.

[13] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006. DOI: 10.1109/TKDE.2006.162.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of deep bidirectional transformers for language understanding*, May 24, 2019. arXiv: 1810.04805[cs]. [Online]. Available: http://arxiv.org/abs/1810.04805 (visited on 01/24/2023).

[15] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[16] S. Mohammad, "Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 english words," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 174–184. DOI: 10.18653/v1/P18-1017.