Emotion Detection in Social Media Posts

John Lee[1], Isabella Tang[1], Bea Lu[2], Holly Sederlund[2], Gabriella Stickney[2]

[1]Data Science, College of Natural Science
[2]Computational Data Science, College of Engineering
leejoh18, tangisab, lubangji, sederlu6, stickn21

March 12, 2024

Abstract

In this project, we are aiming to create a machine learning model that can accurately recognize emotions conveyed in social media messages. With digital conversation becoming more and more popular, it is important that the messages are properly understood emotionally by the recipient. Our model will use a dataset of social media messages labeled with different emotions, such as happiness, sadness, anger, and fear. We will process the text data using natural language techniques to extract important features. Then, we will test different machine learning models to predict the writer's intended emotion. Through this process, we aim to improve communication through social media posts.

1 Motivation

Social media is on the rise as a form of communication for the younger generations, including ourselves. As these populations grow, the presence of these applications will only become more prominent. As a result, there will only be more posts that can be emotionally misinterpreted by the readers. We have found studies and efforts with machine learning to gain more insight into this problem. However, we believe it will be valuable for young people, such as ourselves, to continue these efforts with our own experiences and perspectives.

2 Datasets

We found two datasets sourced from Kaggle, a platform with public datasets, that include Twitter posts and their labels. The "Emotion Detection from Text"[1] and "Emotions"[2] datasets each consist of three columns: an ID, a message, and a classification label. We will test for duplicate messages from these two sources to ensure unique inputs. From Google, we found the "GoEmotions"[3] dataset of Reddit posts that consists of 37 columns containing the text, subreddit, and their labels. All three datasets have different emotion labels. To maintain consistency, we hope to refine the data points to contain of the labels determined by us. If necessary, we will exclude the data points with labels outside the scope of our project. The final goal will be to aggregate the three datasets into one dataset. We currently plan to split the data into a 70-30 ratio, allocating 70 percent for training and reserving 30 percent for testing. If we run into issues with this plan, there appears to be other datasets that we can use as substitutes.

3 Methodology

Since we are working with text data, there will be pre-processing that has to be done for the models to be usable. Two methods that we will try to test are Term Frequency - Inverse Document Frequency (TF-IDF) and word embedding. TF-IDF can return values for the words in the text by considering their frequency and uniqueness to the corpus. Word embedding works further to include semantic meaning in the values. It works by mapping semantically similar words to nearby points in the embedding space, other methods could be considered if needed.

For our models, we plan to use Support Vector Machines (SVM) as a baseline model. SVM has been proven in literature to work well with text classification tasks[4]. It has the ability to create boundary lines, or hyper-planes, to maximally separate and classify hyper-dimensional data. The model was implemented with the Scikit-Learn library. We chose to use LinearSVC and SGDClassifier(with hinge loss) for shorter computation time as our dataset is quite large. In addition, we also used StandardScaler() to help with quicker convergence.

Following Professor Johnson's suggestion, we investigated Recurrent Neural Networks (RNNs) and Transformer-based Neural Networks. Based on our research, we decided to use Bidirectional Encoder Representations from Transformers (BERT) as a Transformer-based Neural Network. Compared to SVM, this model can pick up more detail to help with their classification as it has been pre-trained on Wikipedia and BookCorpus datasets[5]. It also gains further insight due to its ability to use bidirectional context. In our code, we obtained the 'distilbert-base-uncased' model from the HuggingFace library. We chose this condensed version of BERT as an initial test to achieve a faster result knowing that it retains 97% of its functionality[6].

Finally, we plan to report the results for all the models with the four basic metrics: accuracy, precision, F1, and recall score.

4 Preliminary Results

The performance of three different models, SVM using TF-IDF features, SVM using word embeddings, and BERT, was evaluated on the test set. The results reveal significant variations in accuracy, recall, precision, and F1 score across the models. The SVM model utilizing TF-IDF features achieved an accuracy of 0.86, recall of 0.81, precision of 0.82, and an F1 score of 0.81. The SVM model using word embeddings showed a much lower performance with an accuracy of 0.31, recall of 0.17, precision of 0.24, and F1 score of 0.11. When comparing the SVM with TF-IDF features and SVM with word embeddings, it is apparent that SVM with TF-IDF outperforms SVM with word embeddings. This could be related to the nature of the feature representations they utilize. TF-IDF represents words based on their frequency and their importance in the corpus of the social media posts. On the other hand, word embeddings represent words as dense vectors in a continuous vector space. While word embeddings can capture some semantic relationships between words, it is known to not perform well when the dataset is relatively small or lacks enough diverse context for effective learning[7]. In contrast, the BERT model showcased remarkable performance, achieving an accuracy of 0.97, recall of 0.92, precision of 0.93, and F1 score of 0.92. A possible reason for BERT's significantly better performance would be due to its approach to text representation and modeling. BERT utilizes Transformer-based architectures, which are capable of capturing complex dependencies and contextual relationships within sentences and posts. Another advantage is that it is pre-trained on text data measuring 3.3 billion words. A summary is provided in Table 1.

5 Project Limitations and Dangers

One limitation that we have is based on our datasets obtained. Each dataset had varying classifications of emotions that were not shown in the others. To accommodate for this, we had to create a final dataset using a select subset of emotions: joy, sadness, love, anger, fear, and surprise. Based on these select emotions, we had to eliminate other emotions such as curiosity, disappointment, disgust, grief, and nervousness. Therefore, in our models we are only predicting six emotions, in reality, there are multiple other emotions that an individual could express. This is a limitation of our model due to the select emotions that we are modeling as individuals can feel otherwise in their tweets. Additionally, we had another limitation based on our datasets. We were only able to find emotions that were expressed through Twitter and Reddit posts. In reality, there are more social media interfaces that users express emotions on such as Instagram, Snapchat, Facebook, and YouTube. These social media platforms unfortunately did not have datasets with labeled messages. This can be another limitation of our project because some individuals can express emotions in different ways based on the platform. For example, someone could express more anger in their posts on Twitter versus more joy and love in their posts on Instagram. This leads to the danger of misinterpreting emotions. Having more variety in social media datasets could allow our model to better understand emotions behind all different kinds of posts, instead of being confined to the scope of Reddit and Twitter. Finally, another limitation is the processing power allotted to our project. Our best resource available is university run servers as they are more powerful than our personal devices. However, there are still better resources that we do not have access to. This limits the amount of epochs that we can allow the models to run, as well as how much data we can train the models on. The result of this may be a less robust model which may classify messages with less confidence and accuracy.

6 Addressing Feedback

To address the feedback we were given on our project proposal, we were given the suggestion of using a sliding scale model. We discussed this option for our model, but we could not find a labeled dataset to accomplish this task. We further discussed the possibility for our model to output the certainty of each data point's prediction. Additionally, we will look to find other datasets to test our models on other than from Kaggle. We unfortunately did not have the opportunity to do this due to time limitations. In

	SVM (TF-IDF)	SVM (Word Embeddings)	BERT
Accuracy	0.86	0.31	0.97
Recall	0.81	0.17	0.91
Precision	0.82	0.24	0.93
F1	0.81	0.11	0.92

Table 1: Preliminary Metrics of All Models

the case where we can't find other datasets, we believe the Kaggle datasets are organized and in-depth enough to complete our task. Moreover, our dataset is not purely from Kaggle and consists of messages collected by Google Research.

7 Related Works

There have been many studies on emotion analysis on online social media platforms. Many of them compare the performance of different models on the same dataset. A study done by researchers at UNC Charlotte and the University of Virginia argued that current methods for doing emotion analysis on social media, which are often based on conventional machine learning models, cannot grasp the complexity of emotional language[8]. The team used a model that was based on bidirectional Recurrent Neural Networks (RNN), specifically a bidirectional Gated Recurrent Unit network to analyze a dataset of tweets and improve on previous models. They compared their results to two other papers which used multinomial Naive Bayes, LIBLINEAR, and maximum entropy classifier with a bag of words model. The result was an improvement in classification using the RNN. This was the paper whose future work included the comment of testing with BERT for improvements. Heavily referenced in the GoEmotions paper[3] was a work by psychologist Paul Ekman[9], which proposes a theory on six basic emotions which are: anger, fear, disgust, joy, sadness, and surprise. We believe we can also use this grouping in model training as a way to group the more specific emotions from the GoEmotions dataset.

8 Collaboration Efforts

We will continue to meet up as a group to work on the project for each part of the process. We should all be coding, analyzing the dataset, and researching, so it seems unfair to be splitting up work when it could be unequal. We will collaborate on the coding, write-ups, and the presentation slides during our weekly sessions. The weekly meetings we have set are Tuesdays from 4:00-6:00 P.M and if that does not work, we will do Thursdays from 4:00-6:00 P.M. If we do not make enough progress in these meetings, we will then assign individuals to finish certain parts of the assignment to meet our deadlines.

9 Future Work

In the future we will consider using MSU's High-Performance Computing Cluster (HPCC) or GPU to train our model with more epochs. We will also search for datasets containing messages outside of Kaggle to test our model on. Also, we could look into hyper parameters for our models as a possibility. Further, we could look into utilizing DLSTA (Deep Learning Assisted semantic text analysis), which utilizes natural language processing concepts. In comparison to the BERT model, DLSTA utilizes sequence tagging that will label each token in a sequence with a tag, in this case an emotion. By using DLSTA, we will have a more comparable model to BERT to analyze which model is better at predicting emotions from text input. We will also attempt to investigate why the SVM model with word embeddings performed so poorly. We will read more papers to look for explanations. Additionally, we will also look at different packages or methods of coding to see if there are improvements. For further research, we plan to use a subset of each post to train and test the models to see how much of a text is needed in order to classify the emotion being displayed. We will remove text with multiple emotions to prevent the model from needing to determine emotions from messages that switches throughout the post. Using the first word or n words, we will compare how well the models predict using varying lengths of text.

References

[1] Pashupati Gupta. Emotion Detection from Text. Kaggle.

[2] Nelgiri Withana. Emotions. Kaggle.

[3] Google Research Blog. GoEmotions: A Dataset for Fine-Grained Emotion.

[4] Thomas Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features.

[5] Raman Kumar. Introduction to BERT and Segment Embeddings.

[6] Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

[7] Peng Wang, Bo Xu, Jiaming Xu, Guanhua Tian, Cheng-Lin Liu, Hongwei Hao. Semantic expansion using word embedding clustering and convolutional neural network for improving short text classification.

[8] Armin Seyeditabari, Narges Tabari, Shafie Gholizadeh, Wlodek Zadrozny. Emotion Detection in Text: Focusing on Latent Representation. 2019.

[9] Paul Ekman. Are There Basic Emotions? 1992.