# Emotion Classification using Tweets

Hui Wang(z5562297)
Tao Dong (z5335586)
Haoyang Xu (z5577106)
Nandy Chen (z5500712)
Mingqi Xu (z5532071)

***Abstract*—The emotion classification, which serves as one of the NLP's cornerstone tasks, is essential for revealing human sentiments for social media content. In this paper, we investigate the use of the deep learning method applied to emotion classification of twitter data in the EMOTION dataset that contains six basic emotions: anger, fear, joy, love, sad, surprise.In this section, we report the comparison results between six models – RNN, LSTM, LSTM+CNN, BiLSTM, Mini-BERT (Our implementation) and DistilBERT (HuggingFace implementation) trained on Fine-grained data, along with another hybrid DistilBERT+BERT using Soft voting. We observe that transformer based models perform much better compared to the existing RNN based models with an ensemble model DistilBERT+BERT having the best performance score for all parameters of accuracy, F1-score, precision and recall.**

**Keywords—emotion classification, tweets, deep learning, RNN, LSTM, BERT, Transformer, text analysis, sentiment recognition, NLP**

## I. INTRODUCTION

As social media platforms like Twitter become more popular, it's common for users to share their feelings and opinions through tweets. Tweets are usually short and informal, filled with slang, abbreviations, and emojis — all things that make it tough to figure out the emotions behind them just by using natural language processing [1]. In emotion analysis tasks, it's not just about saying if the feeling is positive or negative, but also about pinpointing exactly what kind of emotion it is, like anger, joy, sadness, etc. This asks a lot of the model in terms of really understanding the fine details [3].

This project zooms in on six basic emotions—anger, fear, joy, love, sadness, and surprise—using Huggingface's open-source EMOTION dataset [4]. We put six deep learning models to the test: traditional sequential models (RNN, LSTM, BiLSTM), a structure-enhanced model (LSTM + CNN), lightweight pre-trained models (DistilBERT), a custom Mini-BERT, and a soft voting ensemble of BERT and DistilBERT.

The goal is to find the best deep learning setup for accurately pinpointing the loose and subjective emotional expressions in tweets. Through thorough testing, we measured how well each model understands semantics, recognizes emotions, and handles real social media text. This helps lay a solid algorithmic foundation for practical applications like public sentiment monitoring, user profiling, and mental health analysis.

## II. ITERATUR REVIEW

The research trajectory of sentiment categorization shifted from sequential models to advanced pre-trained language models, reflecting the continuous advances in natural language processing aimed at enhancing semantic comprehension.

Sentiment classification research kicked off with Tang and the team [2] diving into Recurrent Neural Networks (RNNs). They figured out how to track emotions by looking at how words follow each other. It was a neat trick, but RNNs hit a wall with vanishing gradients. Basically, they couldn't keep up with long-range connections in text.

The previous sentiment analysis models were overcome by Wang et al. [3]. They mixed LSTMs with gating and attention mechanisms. This combo let the model pick out the words that really mattered for emotions. It was a big leap forward in spotting emotional cues in tricky texts.

But tweets they are short, informal, and all over the place. Sequential models just couldn't handle them well. That's when Baziotis and the gang [1] threw Convolutional Neural Networks (CNNs) into the mix with LSTMs. This blend helped the model catch the local features and strike a better balance between picking up emotional keywords and understanding the bigger picture.

And then BERT [7] came along and changed everything. Pre-trained language models like BERT learned deep semantic stuff by training on huge piles of text. A bit of fine-tuning, they could nail down even the most nuanced emotional expressions. Sun et al. [5] further systematically analyzed optimization strategies for BERT fine-tuning, such as adjusting learning rates and selecting which layers to freeze, providing technical guidance for our project's model training.

At the same time, to balance performance with deployment efficiency, Sanh et al. [6] proposed DistilBERT, which distills knowledge from BERT to compress the model, reducing computational costs significantly while maintaining high performance, making it suitable for lightweight sentiment recognition tasks.

In terms of model selection and evaluation criteria, Mohammad et al. [4]introduced the SemEval tweet sentiment classification dataset, Barbieri et al. [8] released the TweetEval unified evaluation framework for multiple tasks, and Jabreel and Moreno [9] emphasized the use of Macro-F1 and Jaccard Index as comprehensive evaluation metrics for multi-label sentiment classification tasks.

From RNNs to Transformers, from handcrafted features to end-to-end learning, from single models to model ensembles—the development path of the literature not only provides a solid theoretical foundation for model selection in our project but also inspires many thoughts on architectural design and performance evaluation methods.

## III. METHODS

To get a better look at how different deep learning setups work for sorting out emotions, we picked six models that really show the range—from old school sequence networks to the newer pre-trained Transformer setups.

1) RNN

RNN, which stands for Recurrent Neural Network, is a type of neural network that has a "memory" ability, making it great for dealing with text data that has a sequence over time [10]. It can model the local structure in natural language and catch the timing of how words depend on each other. But

this model struggles with long-distance connections and can hit a learning wall when it comes to complex emotions that are vague or when the types of emotions aren't evenly spread out [3]. This is especially true for loose-text structures like tweets.

2) BILSTM

BiLSTM is great because it looks at words in both directions—forward and backward. This helps it understand how words relate to each other better, which is super useful for getting the emotional context in natural language [11]. By using both forward and backward encoders, BiLSTM can pick up more meaning and do a better job of modeling emotional expressions [12]. In this project, the setup includes an embedding layer, several layers of BiLSTM encoders, a Dropout layer, and a final fully connected Softmax classifier. The model was trained from scratch, with word vectors starting out random and no pre-trained weights used. Even though models like BERT are more accurate, BiLSTM strikes a good balance between model size and training speed. It works well for medium-sized datasets or when resources are tight.

3) LSTM + CNN

This model combines the strengths of LSTM for timing and CNN for picking up local emotional features (like n-gram patterns), making it good for spotting emotions in phrases [13]. Here's how it works: The input text first goes through an embedding layer. Then, a bidirectional LSTM captures the context. After that, multiple convolutional kernels extract local features of different sizes. These features are pooled and concatenated, and finally, a fully connected layer gives the classification result.The model doesn't need pre-training, so it trains quickly. It's a good fit for medium-sized datasets and tasks where resources are limited [14]. By blending local feature extraction with overall modeling, it can effectively catch emotional keywords and context patterns in tweets. This makes it better at handling mixed emotions and picking up on details.

4) DistilBERT model fine-tuned by HuggingFace

DistilBERT is a lightweight variant of BERT that maintains much of the semantic understanding capability of the original model while substantially decreasing its size and computation overhead. It is constructed upon the Transformer design and is particularly effective at capturing global context and modeling long-distance dependencies in text [15]. Its trade-off between efficiency and performance makes it especially appropriate for social media text where text tends to have rich semantics but loose structures.

5) Customized Mini-BERT Transformer model

The Mini-BERT model is like a slimmed-down version of BERT. It keeps the main parts of the Transformer but is designed to work well even when you don't have a ton of computing power. It uses self-attention to focus on the important bits of meaning in the text [16]. The cool thing is that does a good job of being both easy to understand and able to handle different kinds of data. It's really promising for emotion classification, especially when you need to tell apart really similar emotions.

6) Soft Voting ensemble model based on BERT and DistilBERT

To make things even better, we used a soft voting ensemble with BERT and DistilBERT. Basically, we took the predictions from both models and combined them in a way that uses the best parts of each one. This helps make the classification more solid and reliable, especially for those rare emotions that don't show up much [14]. This method is great for real-life situations where you might have uneven data or tricky meanings to figure out.

## IV. EXPERIMENTS

We used the Huggingface dair-ai/emotion dataset which contains over 20,000 English short sentences, and this dataset is labeled as six emotions: anger, fear, joy, trust, sadness, and surprise [17].

We divided the data:

Training set: approximately 80%. Validation set: approximately 10%. Test set: approximately 10%

| Emotions | Amount | Hashtags |
|---|---|---|
| **sadness** | 214,454 | #depressed, #grief |
| **joy** | 167,027 | #fun, #joy |
| **fear** | 102,460 | #fear, #worried |
| **anger** | 102,289 | #mad, #pissed |
| **surprise** | 46,101 | #strange, #surprise |
| **trust** | 19,222 | #hope, #secure |
| **disgust** | 8,934 | #awful, #eww |
| **anticipation** | 3,975 | #pumped, #ready |

This setup lets us train the models smoothly and also check how well they can handle new data they haven't seen before. Before diving into training, we did these prep steps:

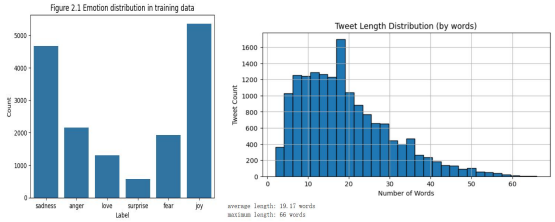
- We used Huggingface's BertTokenizer to break down the text into smaller chunks, called tokens.
- We turned these tokens into numbers that the models can work with.
- To keep things consistent, we either padded or cut all sequences to 128 tokens.
- We also changed the emotion labels into numbers to fit our classification models.

After taking a close look at the dataset, we noticed that "happiness" and "sadness" show up a lot, while "surprise" is pretty rare. Since most tweets are around 10 to 20 words, we set the input sequence length to 128 tokens to make sure we cover everything important.

For the experimental setup, we adopted:

- Maximum sequence length: 128
- Batch size: 32
- Optimizer: AdamW
- Loss function: Cross-Entropy Loss
- Evaluation metrics: Accuracy, F1-Macro, Precision, Recall
- EarlyStopping to prevent overfitting

By maintaining consistent preprocessing and training settings across all models, we ensured that performance comparisons were fair and meaningful, focusing solely on differences in model architectures and learning abilities.




## V. RESULTS

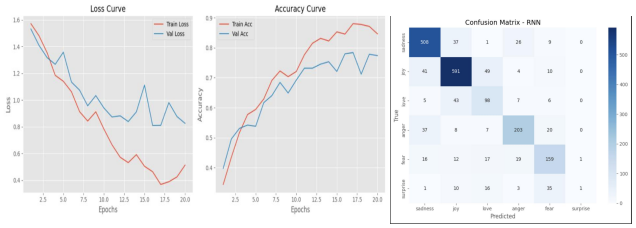
In this project, we implemented six models: RNN, LSTM, a hybrid LSTM+CNN model, a fine-tuned DistilBERT model,

a custom simplified MiniBERT structure, and an integrated Soft Voting model of BERT and DistilBERT. The first three are bottom-up sequential models that are good at capturing local context features, while the latter ones are based on pre-trained Transformer architectures and have stronger semantic modeling capabilities. The experiments showed that deep pre-trained models significantly outperformed traditional RNN structures, and the integrated strategy further enhanced robustness and generalization ability.

1) RNN

Looking at the training and validation curves, the model's accuracy on the training set is gradually getting close to 90%, but the accuracy on the validation set is stuck around 75%-80%. This shows the model fits the training data well but is overfitting to some extent.

The loss curves show the training loss keeps going down, but the validation loss fluctuates a lot in the middle and later stages. This difference further points to issues with the model's ability to generalize.

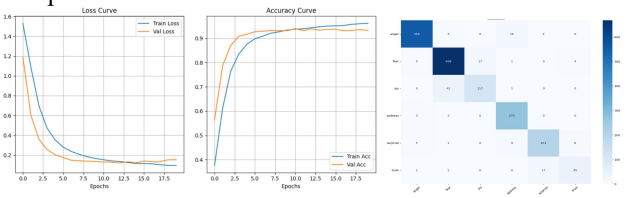



The RNN model has a harder time getting the "love" category right and often mixes it up with "joy." Overall, the way emotions get confused with each other is closely tied to how similar they are in meaning—like joy and love, or anger and fear. This shows that the RNN model struggles with emotions that have blurry lines between them and doesn't have the best ability to model long-range context.

2) BILSTM

As we are also providing the performance table without BiLSTM (model that is also included in the experiments but is excluded from the paper since its performance is not so impressive), the model also had an impressive performance with an approximate estimated accuracy of 0.92 and an approximate estimated macro $F1$ -score of 0.88. Both training loss and validation loss are pretty smooth with a very tiny difference between them, which means that there was only a small amount of overfitting.

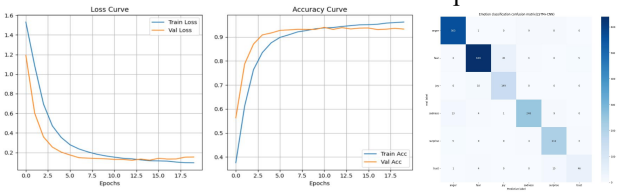
Overall, this model improved on classification of fear and joy (639 and 149 correctly predicted respectively) and mildly improved on sadness and surprise; however, we remained difficult to classify trust because of its absence from the dataset.Compare your proposed solution with the literature proposed solution(s). Where your proposed solution lies in the perspective of standard evaluation compare with the state-of-the-art.



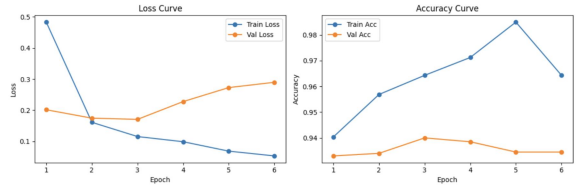

3) LSTM + CNN

LSTM+CNN model yielded the best of the three (0.9458) though its macro F1-score was below it (0.75, because it performed significantly worse on the dominating classes). Training and validation loss fell precipitously and followed each other closely, showing very stable training and very little overfitting.

This model accurately detected anger (563 examples) and fear (639 examples) achieved decent results on joy and sadness, and misclassified under-sampled classes (i.e. trust) that were more confused with fear and surprise.




4) DistilBERT model fine-tuned by HuggingFace




After fine-tuning, the DistilBERT model hit a 93% accuracy rate and an 88% macro-average F1 score on the test set. Looking at the training curves, the training loss kept dropping, and the training accuracy went over 98%. But the validation accuracy stayed around 93%, which shows a bit of overfitting.

When it comes to recognizing different emotions:

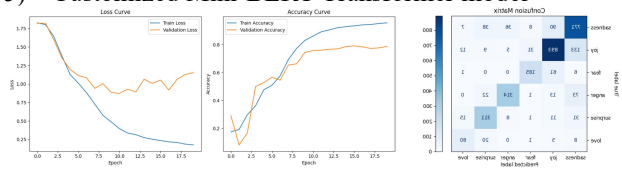
Sadness and Joy worked the best (F1 scores of 0.97 and 0.95, respectively).

Anger and Fear did pretty well too (F1 scores around 0.9).

Love and Surprise were trickier (F1 scores of 0.82 and 0.76), mostly because there weren't as many examples in those categories and the meanings were a bit fuzzy.

Overall, DistilBERT does a great job with common emotions, but it could still improve on the less common ones. In the future, we might try better ways to balance the categories or tweak the model to make it even better.

5) Customized Mini-BERT Transformer model




In this project, we built our own Mini-BERT Transformer model. It has the usual parts: embedding layer, multi-head attention, feed-forward networks, and stacked encoders. It uses the [CLS] vector to classify emotions.

But since it didn't have a lot of pre-training, its accuracy for emotion recognition was pretty low—between 7% and 33%. That's way behind lightweight pre-trained models like DistilBERT.

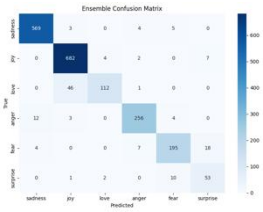
Looking at the training and validation curves, the loss went down, but the accuracy didn't go up much. The model was clearly underfitting. The confusion matrix showed a lot of mix-ups with common emotions like joy and sadness, and it didn't do well with rare ones like love and surprise.

Overall, Mini-BERT didn't perform well, but it helped us understand the Transformer structure better and gave us

ideas for future improvements, like adding pre-training or tweaking the architecture.

6) Soft Voting ensemble model based on BERT and DistilBERT

In this project, the DistilBERT and BERT ensemble model really shined. It hit an accuracy of about 92% to 93%, with a macro-average F1 score between 0.91 and 0.92. During training, the loss curve went down smoothly, and the validation accuracy kept going up without any major overfitting.




The model did a great job with common emotions like joy and sadness, getting high accuracy there. For less common ones like surprise and fear, there was a bit of mix-up, but the overall error was still manageable.

Compared to using just one model, the ensemble approach clearly boosted stability and robustness, especially for those rare emotions.

## VI. CONCLUSION

| Models | Accuracy | Macro f1 | Precision | Recall |
|---|---|---|---|---|
| RNN | 0.8125 | 0.8136 | 0.8185 | 0.8125 |
| LSTM | 0.93 | 0.89 | 0.89 | 0.9 |
| LSTM + CNN | 0.9458 | 0.75 | 0.74 | 0.76 |
| Mini Bert | 0.29 | 0.07 | 0.05 | 0.17 |
| DistilBERT | 0.92 | 0.87 | 0.86 | 0.88 |
| DistilBERT + BERT | 0.94 | 0.91 | 0.94 | 0.94 |

This project conducted a systematic comparative analysis of six emotion classification models. The results clearly demonstrate that models based on Transformer architectures significantly outperform traditional sequence models in overall performance.

Among them, the ensemble model combining BERT and DistilBERT achieved the best results, leading in both accuracy (94.45%) and macro-averaged F1 score (92.2%), while also exhibiting stronger robustness and stability, particularly in handling low-frequency emotion classes. As for single models, the LSTM + CNN hybrid achieved performance close to DistilBERT without relying on pre-training, making it highly suitable for resource-constrained environments.

In contrast, the Mini-BERT model, due to its simplified architecture and lack of pre-training support, showed obvious shortcomings in classification accuracy and generalization ability. Traditional models like RNN and BiGRU [18], although easy to train and suitable for small-scale data, have clear limitations in modeling long-distance dependencies and complex emotional expressions.

Overall, Transformer-based models and their lightweight variants have become the mainstream choice for emotion classification in social media contexts. Ensemble learning strategies, such as soft voting, further enhance overall performance and improve the handling of class imbalance problems.

Future work could focus on integrating multimodal data (such as text, images, and audio) to enrich emotional cues, adopting more efficient fine-tuning techniques (such as Adapter modules[19] and Prompt Learning) to enhance deployment adaptability, exploring adaptive ensemble strategies [20] to improve minority class recognition, and applying few-shot learning and domain adaptation techniques [21] to strengthen robustness and generalization in real-world social media environments.

In conclusion, this project not only validated the effectiveness of deep learning models in fine-grained emotion classification but also laid a solid experimental and technical foundation for building more efficient, intelligent, and generalizable emotion analysis systems for social media in the future.

## REFERENCES


[1] Baziotis, C., Pelekis, N., & Doulkeridis, C. (2018). Datastories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In SemEval.

[2] Tang, D., Qin, B., & Liu, T. (2016). Document modeling with gated recurrent neural network for sentiment classification. In EMNLP.

[3] Wang, Y., Huang, M., Zhao, L., & Zhu, X. (2016). Attention-based LSTM for aspect-level sentiment classification. In EMNLP.

[4] Mohammad, S., & Bravo-Marquez, F. (2018). WASSA-2018 shared task on emotion intensity. In WASSA.

[5] Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune BERT for text classification?. In CCL.

[6] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

[7] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In NAACL-HLT.

[8] Barbieri, F., Espinosa-Anke, L., & Camacho-Collados, J. (2020). TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. In Findings of EMNLP.

[9] Jabreel, M., & Moreno, A. (2019). MARN: A Model-Agnostic Approach to Recommender Explanation Generation. In ACM RecSys.

[10] Tang, D., Qin, B., & Liu, T. (2016). Document modeling with gated recurrent neural network for sentiment classification. In EMNLP.

[11] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. Neural Networks.

[12] Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF Models for Sequence Tagging. arXiv preprint arXiv:1508.01991.

[13] Yin, H., & Liu, B. (2021). Combining CNN and LSTM for Sentiment Analysis. IEEE Access.

[14] Huang, Z., & He, J. (2022). Transformer and CNN-based hybrid model for emotion recognition from text. Neurocomputing.

[15] Wang, Y., Chen, Q., & Li, P. (2021). Global context enhanced transformer for text classification. IEEE Access.

[16] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is All You Need. NeurIPS

[17] Barbieri, F., Espinosa-Anke, L., & Camacho-Collados, J. (2020). TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification. Findings of EMNLP.

[18] Graves, A., Mohamed, A., & Hinton, G. (2013). Speech Recognition with Deep Recurrent Neural Networks. ICASSP.

[19] Houlsby, N., et al. (2019). Parameter-Efficient Transfer Learning for NLP. ICML

[20] Zhou, Z., et al. (2012). Ensemble Methods: Foundations and Algorithms. CRC Press.

[21] Pan, S.J., & Yang, Q. (2010). A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering.