Emotion Detection in Tweets using Natural Language Processing

Lukas Marović

Faculty of Information and Communication Technology
Wrocław University of Science and Technology
293855@student.pwr.edu.pl

1 Introduction

Emotion detection in text has become increasingly important as social media continues to serve as a dominant platform for public expression, opinion sharing, and real-time communication. Platforms such as Twitter generate massive amounts of user-generated content every day, and understanding the emotional states conveyed in these messages is essential for a wide range of applications. These include content moderation, mental health monitoring, crisis detection, brand reputation analysis, and even financial forecasting, as fluctuations in public mood have been shown to correlate with stock market behavior [1]. As a result, automated emotion detection systems have become a critical research topic within the broader field of Natural Language Processing (NLP).

Emotion detection differs from traditional sentiment analysis, although the two are often related. Sentiment analysis typically categorizes text into coarse-grained polarity labels such as positive, negative, or neutral. While useful, such labels fail to capture the rich spectrum of human emotions present in natural language. Emotion detection expands beyond polarity and aims to classify specific affective states such as joy, anger, fear, sadness, surprise, and others. This finer granularity enables more accurate interpretation of user intent and more actionable insights, especially in contexts where the type of emotion is important. For example, both fear and anger may be considered negative sentiments, yet they imply very different psychological states and may require different responses from automated moderation or decision-support systems. Natural language processing provides the computational methodologies for analyzing and interpreting human language. [5]

Modern NLP relies heavily on machine learning and deep learning techniques, particularly neural network architectures capable of modeling complex linguistic patterns. Among the most prominent model families used in emotion detection tasks are Long Short-Term Memory (LSTM) networks and Transformer-based architectures. LSTMs, a type of recurrent neural network, are designed to capture long-range dependencies in text and have been widely used in early affective computing research. More recently, Transformer-based models such as the BERT

family have achieved state-of-the-art performance across numerous NLP tasks due to their ability to capture contextual meaning more effectively than recurrent models. The use of different BERT models will be achieved using the transfer learning technique to repurpose a model trained on one task for another related task. [4]

The goal of this project is to implement and compare the performance of multiple neural network models. Specifically, the study contrasts one or more LSTM models with several pre-trained BERT-based architectures fine-tuned for emotion classification. Through systematic experimentation, the project aims to determine how these models differ in accuracy, generalization ability, and behavior on noisy, real-world Twitter data. This comparison provides insights into the strengths and limitations of traditional recurrent methods versus modern Transformer-based approaches.

2 Literature Review

Emotion detection in text has evolved through several methodological phases. The first approaches were traditional machine learning-based, but were replaced with the rise of deep learning methods and transformer-based models. Each stage improved performance by better handling context, noise, and informal language typical for social media. [4]

Traditional machine learning classifiers (e.g., SVM, Logistic Regression, Naïve Bayes) require feature engineering to convert the text into a numerical format that the algorithms can interpret. The feature engineering methods used are Bag of Words (BOW) and TF-IDF, with TF-IDF also preserving some semantic nature of the original sequence. These methods were computationally efficient and interpretable, and often produced reasonable results for simple sentiment detection. [3] However, due to their inability to capture semantic context or handle informal language, their limitations started to become clear. [7]

To overcome those limitations, neural networks began to be applied, notably Long Short-Term Memory (LSTM) architectures, often using dense word embeddings such as Word2Vec and GloVe. Word embeddings are numerical representations of words as vectors in a multi-dimensional space, where these vectors can be used as features for natural language processing tasks. [8] These models marked a clear improvement over previous approaches in overall accuracy and per-class performance. [7]

The next evolution in NLP tasks came in the form of transformer-based models, especially BERT (Bidirectional Encoder Representations from Transformers) and its variants. Multiple recent studies show that fine-tuned BERT models significantly outperform both classical ML and LSTM-based methods. For example, a BERT-based classifier for Twitter sentiment/emotion classification reported

92% accuracy for sentiment analysis and around 90% for emotion analysis. [2] Another study that compared several transformer models and an LSTM model reported accuracies as high as 92%, but noted that preprocessing the data did not improve the performance of the best BERT model and led to a loss of accuracy. [8]

Preprocessing is an important step in NLP tasks to transform text into a format that can be easily analyzed by models. Common preprocessing steps include removing HTML tags, punctuation characters, links, emails, special characters like emojis, and changing all letters to lowercase. However, for transformer-based models like BERT, aggressive preprocessing can degrade performance. Transformers are designed to work with raw text, including sub-word tokenization mechanisms that handle rare words, misspellings, and informal constructs. Heavy preprocessing may lead to the loss of emotional signals and context and lead to poorer performance. [6]

3 Method and Dataset

3.1 Dataset

The experiments will be conducted on the dair-ai emotion dataset. [9] The class contains six class labels representing the emotions of joy, sadness, anger, fear, love, and surprise. The dataset consists of 20000 tweets split into 80% for training, 10% for validation, and 10% for testing.

It's important to note that the dataset is imbalanced with the joy and sadness classes making up over 60% of entries. To prevent bias toward majority classes, we will use the weighted cross-entropy loss function to assign greater weights to minority classes. The Macro-F1 score could also be used as a primary evaluation metric because it gives equal importance to all classes.

Preprocessing of text usually consists of removing punctuation, special characters, urls, and changing the text to lowercase. However, the dataset that is being used already implemented all of those preprocessing methods of text manipulation, so we can skip that step. The downside of the data already being preprocessed is that we won't be able to benchmark how the models perform when trained on non-preprocessed text, which could heavily impact the BERT models since some of them work better when trained with it.

As the final step of preprocessing, we need to tokenize the text using a pre-existing tokenizer like WordPiece and BPE. Tokenization is a fundamental step in NLP and it is the process of breaking down text into smaller units called tokens, which can be words, subwords, or characters. It essentially breaks down text into discrete units and transforms messy language into a structured, numerical format that machines can process and understand.

3.2 Method

The performance of four models will be benchmarked in the experiment. First, we will be benchmarking a Bidirectional LSTM with GloVe used for word embedding. A hidden size of 128 will be used and the selected loss function is the weighted cross-entropy function. A small dropout value will be set, which will randomly zero parts of the input with a given probability to prevent overfitting.

The next three models are BERT models and we will be testing the BERT-base model, DistilBERT, and RoBERTa-base model. The BERT-base model is the standard pretrained transformer, DistilBERT is a faster lightweight version, and RoBERTA-base is a stronger pretrained variant. With that we will cover a wide range of BERT-based transformers, which is important for benchmarking.

4 Experimental Protocol

4.1 Setup

The goal is to compare the performance of LSTM and BERT models in emotion detection on the dair-ai emotion dataset and to see if BERT models are the better option for NLP tasks despite their high computational cost.

The dataset came pre-split into training, validation, and test sets so that will not be necessary to do in the experimental loop. When it comes to cross-validation, BERT models usually don't use it due to the high computational cost of their training, so we will leave it out and work only with the split dataset, which should be enough considering the large size of the dataset.

The text will be preprocessed as described in the previous section, with GloVe being used for word embedding in the LSTM model.

For experimentation, we will be using PyTorch, alongside the HuggingFace transformers, scikit-learn, and matplotlib libraries. The experimental loop will consist of training the models on the training set and applying early stopping based on the macro F1-score of the validation set. We do this, alongside having a dropout value, to prevent overfitting in our models. Additionally, the LSTM model will use the Adam optimizer, while the BERT model will use the AdamW optimizer to adjust their internal parameters. The loop for the LSTM training will be explicit, while the BERT loop will be abstracted in the HuggingFace Trainer class. After the models are trained, we will evaluate each model on the test set, recording the selected metrics. Finally, we will compare model performance using statistical testing and interpret the results.

4.2 Metrics and statistical tests

The metrics that will be used to analyze the results are precision, recall, and macro F1-score. Macro F1-score is used due to the imbalance of the dataset to give equal importance to each class. For statistical analysis, the Wilcoxon singed-rank test will be used to test if one model performs significantly better than the other.

References

1. Bollen, J., Mao, H., Zeng, X.: Twitter mood predicts the stock market. Journal of computational science **2**(1), 1–8 (2011)
2. Chiorrini, A., Diamantini, C., Mircoli, A., Potena, D., et al.: Emotion and sentiment analysis of tweets using bert. In: Edbt/icdt workshops. vol. 3, pp. 1–7 (2021)
3. Kher, D.: Multi-label emotion classification using machine learning and deep learning methods. Ph.D. thesis (2021)
4. Khurana, D., Koli, A., Khatter, K., Singh, S.: Natural language processing: state of the art, current trends and challenges. Multimedia tools and applications **82**(3), 3713–3744 (2023)
5. Nandwani, P., Verma, R.: A review on sentiment analysis and emotion detection from text. Social network analysis and mining **11**(1), 81 (2021)
6. Pota, M., Ventura, M., Fujita, H., Esposito, M.: Multilingual evaluation of preprocessing for bert-based sentiment analysis of tweets. Expert Systems with Applications **181**, 115119 (2021)
7. Rahman, M.M., Shova, S.: Emotion detection from social media posts. arXiv preprint arXiv:2302.05610 (2023)
8. Rezapour, M.: Emotion detection with transformers: A comparative study. arXiv preprint arXiv:2403.15454 (2024)
9. Saravia, E., Liu, H.C.T., Huang, Y.H., Wu, J., Chen, Y.S.: CARER: Contextualized affect representations for emotion recognition. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3687–3697. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018). https://doi.org/10.18653/v1/D18-1404, https://www.aclweb.org/anthology/D18-1404