Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Querétaro

School of Engineering and Sciences

A comprehensive comparison of emotion detection methods

A thesis presented by

Daniel Sebastián Cajas Morales (Student)

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Bachelors of Science

in

Computer Science

Monterrey, Querétaro, Dec, 2024

A comprehensive comparison of emotion detection methods

by

Daniel Sebastián Cajas Morales (Student)

Abstract

Sentiment analysis and emotion detection have emerged as prominent research topics in machine learning and natural language processing (NLP). Significant advancements in these fields over recent years have driven the adoption of emotion detection models for various commercial applications. However, the associated computational costs have become a critical factor. This paper aims to evaluate and compare traditional emotion detection methods, such as rule-based systems, with neural network-based approaches, including vanilla neural networks and LSTMs. The evaluation considers training and inference time, F1 score, precision, and recall. Our experiments reveal that LSTMs although more computationally demanding provide a significant enough performance increase, specially with bigger datasets and GPU acceleration.

List of Figures

List of Tables

Contents

1. Introduction

1.1 Motivation

Emotion analysis is one of the fastest growing research areas of the last couple decades. More than 99% of the papers regarding sentiment analysis are from after 2004 [7]. There have been multiple articles published comparing different approaches for emotion detection. The most recent and comprehensive one I found being [9]. However this article not only focuses in emotion detection but on text classification in general. Since it is a review and the models compared were not run with equivalent hardware, time comparison for training and inference is impossible. It also does not compare rule based systems.

This research is motivated by the lack of direct comparison in both performance (F1 score, precision, accuracy, recall) and computation time of rule based emotion detection, vanilla neural networks (NN) and more complex NN architectures (LSTM, RoBERTa, etc). The goal is to provide new insights into the costs of training and running these models so that better decisions can be made when trying to deploy them for commercial applications. That is why this study asks, *Are the performance improvements of more complex and computationally expensive models worth the cost?*

1.2 Emotion detection approaches

This study will focus on three specific approaches to emotion detection. These being rule based systems, vanilla neural networks and more complex neural network architectures, specifically Long Short Term Memory (LSTM) networks.

1.2.1 Rule based systems

Rule based systems are the oldest approach to emotion detection. They are based on a set of rules that are used to determine the emotion of a given text. These rules are usually based on the presence of specific words or phrases that are associated with a specific emotion. The most famous model representing these relationships is the OCC model [8] where the different dimensions such as tense, direction, polarity, etc. are used to determine the emotion of a given text. Rule based systems are usually very fast and can be easily implemented. However, they are limited by the fact that they rely on a predefined set of rules and are not able to learn from new data. This means that they are not able to adapt to new types of text or new emotions that may not be covered by the rules.

1.2.2 Neural Networks

Vanilla neural networks are a type of machine learning model that is inspired by the human brain. They consist of nodes (neurons) that are connected to each other in a network. Each node takes an input, processes it, and passes the output to the next node in the network. The output of the final node in the network is the prediction made by the model.

Neural networks are considered to be universal function approximators [5]. They are able to learn any function given enough data and computational power. Neural networks are able to learn from new data and adapt to new types of text or new emotions. However, they are usually slower than rule based systems and require more computational power to train.

Vanilla neural networks have some disadvantages when processing text.They have fixed length inputs and outputs, meaning the same model would either be unable to process long sequences or waste resources processing smaller ones. There are ways around this problem like pooling results but this looses the order of the words in the text.

1.2.3 Beyond vanilla neural networks

Recurrent Neural Networks (RNN) are a type of neural network that is able to process sequences of data. They are able to remember information from previous time steps in the sequence and use it to make predictions about future time steps (fig 1.1). This makes a good fit for tasks like emotion detection where the order and correlation of the words in the text is important. However, they still have limitations as back propagation through time can lead to vanishing or exploding gradients. Effectively making long term dependencies impossible to learn [4].

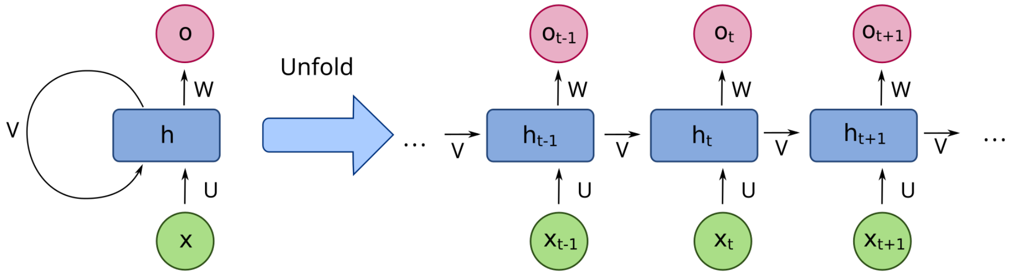

Figure 1.1: Unfolded RNN [3]

Here is where Long Short-Term Memory (LSTM) networks come in. They are a type of RNN that is able to learn long term dependencies in the data. They are able to remember information from previous time steps and use it to make predictions about future time steps. This is done by using a series of gates (fig 1.2) that guarantee that the gradients do not vanish or explode. This makes them well suited for tasks such as emotion detection, where the order of the words in the text is important.

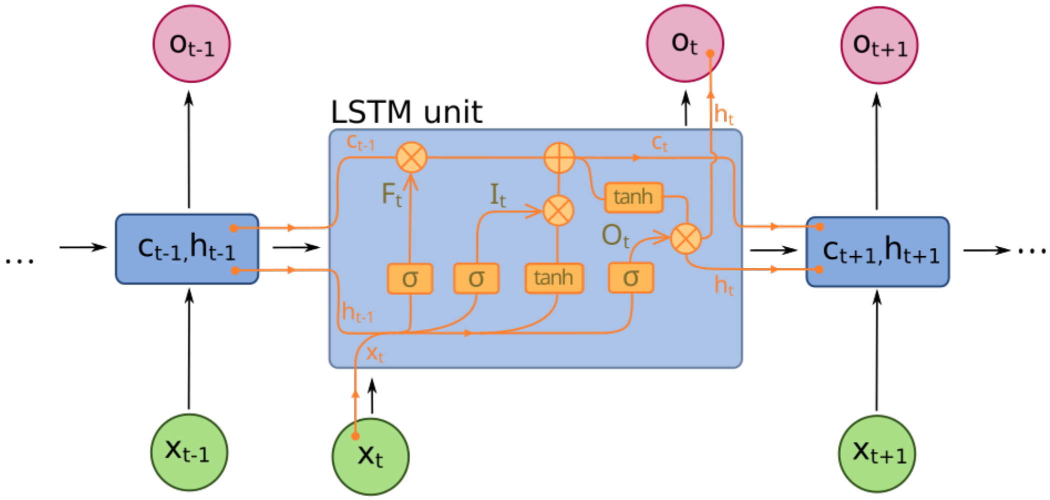

Figure 1.2: LSMT unit [2]

2. Experimentation

2.1 Models

Hyperparameter optimization was not performed in depth for any of the models as this was outside of teh scope of this study. The hyperparameters were chosen based on the best practices for each model and the results of the initial tests.

2.1.1 Rule based

For the rule based system lexicon based approach was tested. In this model, for a given dataset a lexicon is built where we simply keep track of the words used in the texts for each emotion label, this part can be tough of as the training step. When all the words have been counted they are truncated in order to keep the n most common words for each emotion. This is the only hyperparameter for this model which we chose to be 350 words.

Then when we want to classify a new text we simply count the number of words in the text that are in the lexicon for each emotion and assign the label with the most words in the text, this can me tough of as inference.

For preprocessing punctuation and stop words were removed from the text and all words were lowercased.

2.1.2 Neural Network

For the neural network a simple feed forward neural network with an embedding layer, a hidden layer and an output layer was tested. The embedding layer is just a linear layer that maps each token in the input sequence into a length 100 vector (embedding dimensions). After this, mean pooling is applied to the output tensor to collapse the sequence into a single vector of size 100 (embedding dimensions).

At the end of the network a hidden linear layer with the number of emotions as its output size is used. When inferring the argmax of the output tensor is taken as the predicted emotion.

For preprocessing punctuation and stop words were removed from the text and all words were lowercased. The sequences were also truncated to a maximum length of 25 tokens. If shorter than 25 tokens the sequence was padded with the special token '<PAD>'.

The hyperparameter for this model are the embedding dimensions, the max sequence length, the batch size, the learning rate and the number of epochs. These were chosen to be 100, 25, 128, 0.001 and 100 respectively.

2.1.3 LSMT

These model has an almost identical architecture to the neural network but instead of the pooling layer a LSTM layer with a hidden size of 128. Preprocessing was also almost identical, with the only difference being that sequences were not truncated. The hyperparameters were also the same as the NN.

2.2 Metrics

The metrics used to evaluate the models were accuracy, precision, recall and F1 score. These were chosen as they are the most common metrics used to evaluate classification models stated by [6].

the following acronyms will be used: TP (True Positives), TN (True Negatives), FP (False Positives), FN (False Negatives).

2.2.1 Accuracy

Accuracy simply represents the percentage of correctly classified samples, meaning the true positives divided by the total number of samples.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

2.2.2 Precision

Precision represents the percentage of correctly classified positive samples of all the samples classified as positive. Precision prioritizes reducing false positives.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.2}$$

2.2.3 Recall

Recall represents the percentage of correctly classified positive samples of all the positive samples. Recall prioritizes reducing false negatives, even if it might lead to more false positives.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.3}$$

2.2.4 F1 score

The F1 score is the harmonic mean of precision and recall. This means that it gives equal weight to both precision and recall. It is calculated as follows:

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.4}$$

2.2.5 Training and inference time

The training and inference time were also measured for each model. This was done by averaging the time it taken to process 1000 samples. For training the time was measured from the start of the training loop to the end of the training loop.

2.3 Datasets

The first dataset used for this study were dair-ai/emotion [10], the most popular dataset for emotion detection in huggingface which was the dataset provided for the research stay. The dair-ai/emotion dataset is a collection of tweets tagged with 6 different emotions: anger, fear, joy, love, sadness, and surprise.

The second dataset used was the Emotions dataset [1] from Kaggle. A highly up-voted dataset for emotion detection. It has the same 6 emotion labels as the dair-ai/emotion dataset.

2.4 Hardware

All the tests were run on a zephirus g14 laptop (GA401QM) with an AMD Ryzen 9 5900HS and a Nvidia RTX 3060. The tests were run on both on the cpu and with gpu acceleration for the neural network and LSTM models.

3. Results

3.1 Experiment results

After training and testing the models on the two datasets, across all metrics the LSTM model outperformed the NN, which in turn outperformed the rule based model. Training took the longest on the LSTM, followed by the NN and the fastest was the rule based system.

When comparing between datasets, both the LSTM and rule based models performed better with the bigger (20x) Emotions dataset however training times did go up proportionally.

Table 3.1: Accuracy %

Model	dair_ai_emotion	Emotions
rules_based	60.20	65.27
NN_gpu	86.50	83.81
NN	86.05	83.85
LSTM_gpu	**89.00**	**91.30**
LSTM	86.95	91.25

Table 3.2: Precision %

Model	dair_ai_emotion	Emotions
rules_based	60.19	66.15
NN_gpu	84.29	79.26
NN	83.94	79.37
LSTM_gpu	**84.94**	86.60
LSTM	81.52	**87.03**

Table 3.3: Recall %

Model	dair_ai_emotion	Emotions
rules_based	63.27	70.34
NN_gpu	81.23	78.28
NN	80.76	78.05
LSTM_gpu	**86.29**	**87.86**
LSTM	84.39	85.78

Table 3.4: F1 %

Model	dair_ai_emotion	Emotions
rules_based	55.49	62.38
NN_gpu	82.59	78.75
NN	82.17	78.69
LSTM_gpu	**85.58**	**87.12**
LSTM	82.77	86.32

Table 3.5: Training time (s)

Model	dair_ai_emotion	Emotions
rules_based	**0.08**	**1.78**
NN_gpu	7.71	159.38
NN	20.58	2063.97
LSTM_gpu	25.50	645.32
LSTM	195.69	5413.62

Table 3.6: Inference time [1000] (ms)

Model	dair_ai_emotion	Emotions
rules_based	8.23	7.53
NN_gpu	4.47	4.36
NN	**4.06**	**3.65**
LSTM_gpu	19.76	17.29
LSTM	43.54	47.64

3.2 Discussion

The rule-based system exhibited the poorest performance, with a deficit exceeding 20 percent in nearly all metrics. Its inference time was unremarkable, as it was slower than the vanilla neural network. The primary strength of this approach was its minimal training time, which was negligible compared to the other models. However, given that training is a one-time cost, this advantage is not particularly significant.

The LSTM model outperformed the NN model in all metrics,while taking longer to train and infer. The LSTM model outperformed the NN model by 1 to 6 percentage points in the dair_ai_emotion dataset. This difference is accentuated on the Emotions dataset where the difference is between 7 and 9 percent.

In terms of training time, when using gpu acceleration the LSTM model was around 4x slower than the NN model. This is due to the multiple calls to the LSTM cells compared to the mean-pooling dome by the NN model. The LSTM model was also slower when inferring, taking around 5x longer than the NN model.

When only the cpu was used these relationships get more complex. The LSTM model is affected disproportionately as the CPU cash was not able to store the full model. This can be seen in the near 9x increase in training time with the Emotions dataset.

With these results in consideration I argue that the LSTM model provides a significant enough performance increase to justify the increase in training and inference time when GPU acceleration is available. If that is not the case, then vanilla NN model should be considered if training resources are limited. The rule based systems should be avoided as they provide the worst performance across all metrics.

With accuracy above 90% and F1 scores above 85%, the LSTM model demonstrated impressive performance despite not being fully optimized. This raises an important question:

How can the performance of emotion detection models be further enhanced, and what impact would these new developments have on training and inference times, particularly in resource-constrained environments?

3.3 Conclusion

In this study, we evaluated the performance of three different models: LSTM, NN, and rule-based on two datasets for emotion detection.

In conclusion, the LSTM model provides a significant performance advantage for emotion detection, justifying its use when computational resources allow. However, in scenarios where resources are limited, the NN model serves as a practical alternative. The ongoing challenge remains to balance accuracy and efficiency, paving the way for future advancements in this field. These findings highlight the importance of considering both performance and computational requirements when selecting models for emotion detection tasks.

A. Appendix

The source code for the models and tests run can be found in the following repository `https://github.com/DanielSebasCM/research-stay-emotion-classification`

Bibliography

[1] ELGIRIYEWITHANA, N. Emotions, 2024.

[2] FDELOCHE. Long short-term memory, 2017. [Online; accessed 29-november-2024].

[3] FDELOCHE. Recurrent neural network unfold, 2017. [Online; accessed 29-november-2024].

[4] HOCHREITER, S. Long short-term memory. *Neural Computation MIT-Press* (1997).

[5] HORNIK, K., STINCHCOMBE, M., AND WHITE, H. Multilayer feedforward networks are universal approximators. *Neural networks 2*, 5 (1989), 359–366.

[6] LIESKOVSKÁ, E., JAKUBEC, M., JARINA, R., AND CHMULÍK, M. A review on speech emotion recognition using deep learning and attention mechanism. *Electronics 10*, 10 (2021), 1163.

[7] MÄNTYLÄ, M. V., GRAZIOTIN, D., AND KUUTILA, M. The evolution of sentiment analysis—a review of research topics, venues, and top cited papers. *Computer Science Review 27* (2018), 16–32.

[8] ORTONY, A., CLORE, G. L., AND COLLINS, A. *The cognitive structure of emotions*. Cambridge university press, 2022.

[9] REUSENS, M., STEVENS, A., TONGLET, J., DE SMEDT, J., VERBEKE, W., VANDEN BROUCKE, S., AND BAESENS, B. Evaluating text classification: A benchmark study. *Expert Systems with Applications 254* (2024), 124302.

[10] SARAVIA, E., LIU, H.-C. T., HUANG, Y.-H., WU, J., AND CHEN, Y.-S. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, Oct.-Nov. 2018), Association for Computational Linguistics, pp. 3687–3697.