Analyzing Emotion, Sentiment, and Sarcasm: A Comprehensive Comparison of Baseline, Deep Neural Network, and Fine-tuned BERT Models

Chaimae Jallouli, Ayyoub Manssouri

Abstract

This project delves into the dynamic landscape of natural language processing by systematically experimenting with diverse models for the prediction of emotion, sentiment, and sarcasm in English text. Employing a variety of approaches, our exploration aims to comprehensively understand the diverse ways these models contribute to a nuanced understanding of emotions, sentiments, and sarcasm in the realm of English-language text analysis.

Keywords: Natural language processing, Emotion detection, Sentiment analysis, Sarcasm detection, Text classification

1 Introduction

1.1 Introduction

In the field of Natural Language Processing (NLP), sentiment, emotion, and sarcasm detection are essential foundations that have a big influence on a variety of applications.

These attributes are essential for the right interpretation of textual subtleties, averting misunderstandings and promoting more accurate assessments, they're extremely useful for tracking social media environments and offer up-to-date information on trends and public opinion.

These methods are essential to content management since they help platforms remove offensive or dangerous information depending on the emotional context of the user, and refine the user experience by getting feedback.

Beyond the technological, the incorporation of mood, emotion, and sarcasm detection into natural language processing (NLP) models has positive social effects by raising awareness of mental health issues by identifying pertinent signals in textual communication.

These characteristics essentially enable NLP to decipher the complexity of human language, leading to breakthroughs in market knowledge, tailored experiences, and social well-being.

1.2 Project scope

This project's main goals are to do an extensive comparative study of baseline models (e.g. Naive Bayes), deep neural networks (NN), and Fine-tuned BERT models for text interpretation.

One of the main objectives is to assess how well each model category performs in managing different components of textual data, such as mood, emotion, and sarcasm recognition and to shed light on these various methods' advantages, disadvantages, and suitability for various text analysis tasks by contrasting them.

The main goal is to advance a deep comprehension of the advantages and disadvantages of baseline models, deep neural networks, and optimized BERT models. This will help guide future choices regarding the creation and application of text analysis solutions.

2 Methodology

The approach to text analysis used in this study is methodical and multifaceted, comparing baseline models, deep neural networks (NN), and optimized

BERT models on a range of variables, including sentiment, emotion, and sarcasm detection. The following is an outline of the comprehensive methodology:

Through the use of this complex technique, the project seeks to provide a full knowledge of the capabilities and trade-offs related to deep neural networks, optimized BERT models, and baseline models in the field of comprehensive text analysis.

2.1 Emotion detection

Emotion detection is the recognition and interpretation of human emotions as they are conveyed in text.The incorporation of emotion detection methods into natural language processing (NLP) applications is becoming more sophisticated and profound as technology progresses, allowing us to better understand and interact with written emotions.

2.1.1 Baseline model

For emotion detection, we workd with a Multinomial Naive Bayes classifier to classify text in order to predict emotions. It is especially made for text classification tasks, in which the objective is to predict the category or class of a document based on word frequencies represented by characteristics in the document. In our case, the Multinomial Naive Bayes model was trained on TF-IDF vectorized data.

This model demonstrates a basic method of text classification, in which the model assessment metrics, vectorization methodology, and classifier selection are crucial elements in predicting emotions from textual material.

2.1.2 LSTM-Based model

In the next sections, we examine the architecture of the LSTM model used for emotion detection, examining its constituent parts and explaining how these parts enable the model to perform well in tasks that need comprehension of sequential input.

The model we used in this part has a total of 13,886,625 parameters, with 7,014,625 trainable parameters, and is a combination of Bidirectional LSTM layers, Convolutional Neural Network (CNN) layers, and Dense layers. Let's break down the summary:

- **Input layer**: the model takes an input sequence of length 30.
- **Embedding layer**: Two seperate embedding layers convert input sequences into dense vectors of dimension 200.
- **Bidirectional LSTM layers**: Two layers that process the embedded sequence, and capture dependencies in both forward and backward directions.

- Concatenation layer: The outputs of the Bidirectional LSTM layers are concatenated.
- **Convolutional Layers (Conv1D):** Multiple Conv1D layers with different kernel sizes, and filters, are applied to the concatenated output. These convolutional layers likely aim to capture local patterns in the sequence.
- **Dropout Layers:** Dropout layers are applied after each Conv1D layer to prevent overfitting.
- **MaxPooling1D Layer:** Reduces the sequence length
- **Flatten Layer**: reshapes the tensor into a 1D tensor.
- **Dense Layers**: Two Dense layers with 26 and 5 units, respectively, are applied for classification. The first Dense layer uses a Rectified Linear Unit (ReLU) activation function. The final Dense layer employs a softmax activation function for multiclass classification (5 classes).

The architecture combines the strengths of Bidirectional LSTMs and Convolutional Layers, demonstrating a hybrid approach to sequence processing, with the goal of capturing both short and long-range dependencies in the input sequences. The model is designed for multiclass classification with five output classes.

2.1.3 Fine-tuned Bert model

In this part, we used the Hugging Face Transformers library to fine-tune a BERT model named 'bert-base-uncased' for emotion classification.

We employed the BERT tokenizer to process and tokenize the textual data, ensuring proper formatting for model input.

Training parameters, including batch size, number of epochs, and learning rate, are set, and the data is loaded into PyTorch DataLoaders. The model is trained using the AdamW optimizer and a step-wise learning rate scheduler.

The fine-tuned model with the corresponding files was pushed into a huggingface reppository [1] for furteher mentioning or usage.

2.2 Sentiment detection

Sentiment detection involves discerning and understanding emotions and opinions expressed in text or speech. As technology advances in natural language processing (NLP), incorporating sophisticated sentiment detection methods becomes crucial. This progress enhances our ability to analyze sentiments in textual content, providing valuable insights into public opinions, customer feedback, and overall sentiment trends.

[1] https://huggingface.co/Shaimae22/bertemotion

2.2.1 Baseline model

The sentiment detection process begins with tokenizing the tweets using CountVectorizer. This tokenized data is then transformed into TF-IDF (term-frequency times inverse document-frequency) representations. A Multinomial Naive Bayes classifier is selected and trained using the TF-IDF vectorized data. The model's predictions are generated and evaluated using metrics such as accuracy, precision, recall, and F1 score. This Naive Bayes model provides a fundamental method for text classification, particularly in the context of sentiment analysis, and serves as a baseline for comparison with more sophisticated models.

2.2.2 Bi-LSTM based model

In this section, we delve into the architecture of the Bi-LSTM model utilized for sentiment analysis, elucidating its components and detailing how these components contribute to the model's effectiveness in tasks requiring sequential input comprehension.

The employed model encompasses a total of 179,939 parameters, with 179,939 being trainable. It is a fusion of Bidirectional LSTM layers, a Convolutional Neural Network (CNN) layer, and Dense layers. Let's break down the summary:

- **Input layer:** The model processes an input sequence of length 50.
- **Embedding layer:** An embedding layer transforms input sequences into dense vectors of dimension 32.
- **Convolutional Layer (Conv1D):** A single Conv1D layer with 32 filters and a kernel size of 3, employing 'same' padding and ReLU activation.
- **MaxPooling1D Layer:** Reduces the sequence length through max pooling.
- **Bidirectional LSTM layer:** A Bidirectional LSTM layer with 32 units, capturing dependencies in both forward and backward directions.
- **Dropout Layer:** A dropout layer with a rate of 0.4 to prevent overfitting.
- **Dense Layer:** The final Dense layer with 3 units and a softmax activation function for multiclass classification (Negative, Neutral, Positive).

The architecture combines the strengths of Bidirectional LSTMs and Convolutional Layers, presenting a hybrid approach to sequence processing. This design aims to capture both short and long-range dependencies in the input sequences, specifically tailored for sentiment classification into three classes.

2.2.3 Fine-tuned Bert model

In the BERT Sentiment Analysis section, we utilized the 'bert-base-uncased' model from Hugging Face Transformers. The process involved defining a custom tokenizer function for formatting and tokenizing textual data. This

function was applied to the training, validation, and test sets to generate input IDs and attention masks.

Next, the 'bert-base-uncased' model was imported, and a custom function, create_model, was established to host the pretrained BERT model. It included a three-neuron output layer for classifying three emotion classes.

The model was fine-tuned over four epochs, and its architecture and training details were outlined.

2.3 Sarcasm detection

The goal of sarcasm detection is to identify sarcastic sequences in text by creating models that can interpret language's nonliteral and frequently ironic characteristics. In order to do this, algorithms must be trained to recognize sarcastic expression-related linguistic clues, word usage patterns, and contextual subtleties. The goal of the task is to improve computational comprehension of intricate language dynamics so that systems can distinguish between true feelings and satirical comments in textual data. Sarcasm detection contributes to more nuanced and context-aware natural language processing applications.

2.3.1 Baseline model

In the context of sarcasm detection, a Multinomial Naive Bayes model was employed as the baseline approach. The process began with the cleaning and tokenization of headlines, aiming to preprocess textual data effectively. Subsequently, the dataset was divided into training and testing sets. The CountVectorizer was utilized to transform the tokenized headlines into a bag-of-words representation, capturing the frequency of words in each document. The Multinomial Naive Bayes classifier was then instantiated and trained on the vectorized training data. Finally, the model's performance was evaluated on the test set, providing insights into its effectiveness in discerning sarcastic from non-sarcastic headlines.

2.3.2 LSTM-based model

In this section, we elucidate the architecture of the LSTM-based model employed for sarcasm detection. The model encompasses a total of 5,799,505 parameters, with 67,905 being trainable. Here is a breakdown of the summary:

Input Layer: The model processes input sequences with a length of 25. Embedding Layer (GloVe): An embedding layer utilizes pre-trained GloVe word embeddings of dimension 200. GloVe, or Global Vectors for Word Representation, is a method that captures semantic relationships between words based on their co-occurrence in large text corpora. LSTM Layer: A single LSTM layer with 64 units captures sequential dependencies, promoting a

nuanced understanding of contextual information. Dropout Layer: A dropout layer with a rate of 0.2 is integrated to mitigate overfitting during training. Dense Layer: The final Dense layer with 1 unit and a sigmoid activation function facilitates binary classification (Sarcasm, Not Sarcasm).

The architecture leverages pre-trained word embeddings from GloVe and LSTM units to comprehend the contextual intricacies of input sequences, aiming to discern the presence of sarcasm effectively.

3 Datasets

3.1 Emotion detection

The dataset we used for emotion detection is comprised of 55,774 tweets. This dataset was basically pre-processed (no lemmarization, no removal of stopwords).

In order to train a multi-class emotion detection on this dataset, it contains labelles emotions of five classes:

[Neutral, Happy, Sad, Love, Anger]

The text extracted from the dataset was pre-processed as follows:

- **Tokenization:** Converts a list of texts into sequences of integers. Each word in the text is assigned a unique integer.
- **Padding Sequences:** used to ensure that all sequences have the same length.
- **Building Emebdding matrix:** Loads a pre-trained word embedding, where each line in the file represents a word and its corresponding embedding vector. These embedding are used to build the embedding matrix.

3.2 Sentiment analysis

The sentiment analysis dataset is a combination of three datasets: Twitter and Reddit Sentimental Analysis Dataset (162,980 records), Apple Twitter Sentiment Texts (1,630 records), and Twitter US Airline Sentiment (14,640 records), resulting in a total of 179,239 records.

Preprocessing Steps

1. **Concatenation:** Combined the three datasets into a single dataframe.
2. **Handling Missing Data:** Checked for missing values and dropped rows with null entries.
3. **Mapping Categories:** Mapped sentiment categories to standard values: Negative (-1), Neutral (0), Positive (1).

4. **Data Visualization:** Explored the distribution of sentiments through bar plots and pie charts.
5. **Text Cleaning:**
   - Removed HTML tags, brackets, and special characters.
   - Converted text to lowercase.
   - Applied stemming to reduce words to their root forms.
6. **Removing Stopwords:** Eliminated common English stopwords to focus on meaningful content.
7. **Train-Validation-Test Split:** Split the dataset into training (60

3.3 Sarcasm detection

The sarcasm detection dataset comprises headlines from various sources, including news and web content. It includes instances labeled as sarcastic (1) or non-sarcastic (0).

The headlines undergo preprocessing, involving text cleaning, tokenization, and removal of stopwords. Exploratory data analysis reveals a balanced distribution between sarcastic and non-sarcastic instances, providing a diverse and representative set for training and evaluating the sarcasm detection model.

4 Results & Comparative analysis

4.1 Evaluation metrics

In this study, the performance of the proposed framework is rigorously evaluated using standard metrics to provide a comprehensive benchmark. The chosen metrics, outlined below, offer a nuanced understanding of the model's efficacy:

Recall: This metric measures the model's ability to detect positives in each sentence, also known as sensitivity. It is calculated as the ratio of true positives (TP) to the sum of true positives and false negatives (FN), as expressed in Equation 1.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{1}$$

Precision: Precision gauges the accuracy of positive predictions and is defined as the ratio of true positives to the total predicted positives, encompassing both true positives (TP) and false positives (FP), as articulated in Equation 2.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

F1-score: This metric serves as a comprehensive measure to compare two classifiers, considering both recall and precision. It is calculated using Equation

3.

$$F1\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

The accuracy metric, commonly used but deemed unreliable in this context, is defined as the ratio of total correct predictions to total predictions. However, it does not account for false negatives/positives, making it less suitable for evaluation.

In our experiments, we prioritize the F1-score as the primary metric for evaluating the proposed model's performance. This metric ensures a balanced assessment, especially when dealing with imbalanced datasets and emphasizes the importance of both precision and recall in sentiment, emotion, and sarcasm detection.

4.2 Models performance

To assess the performance of the proposed models, we employed several evaluation metrics, including accuracy, precision, recall, and F1-score. These metrics provide a comprehensive overview of the models' effectiveness in handling sentiment, emotion, and sarcasm detection.

4.2.1 Emotion detection

Naive Bayes

The table 1, presents the Accuracy, Precision, Recall and F1 Score of the Naive Bayes model used for Emotion detection.

Metric	Value
Accuracy	0.5405
Precision	0.5648
Recall	0.5405
F1 Score	0.5168

Table 1 Naive Bayes Emotion Detection Metrics

LSTM Model

The table 2, presents the Accuracy, Precision, Recall and F1 Score of theLSTM model used for emotion detection.

Metric	Value
Accuracy	0.4444
Precision	0.5694
Recall	0.4444
F1 Score	0.4296

Table 2 LSTM Model Emotion Detetion Metrics

Fine-tuned Bert

The table 3, presents the Accuracy Score of Fine-tuned BERT model used for emotion detection.

Metric	Value
Accuracy	0.90

Table 3 BERT Emotion Detection Metrics

Comparison

The baseline model exhibited reasonable performance, achieving competitive accuracy. However, they showed limitations in capturing nuanced patterns.

The LSTM model, an integral component of our comprehensive text analysis, demonstrates a noteworthy performance. The precision of 0.569 indicates that the model is relatively accurate in its positive predictions, while the recall of 0.444 suggests that it captures less than half of the actual positive instances. The F1 score, a harmonized metric considering both precision and recall, is 0.430, reflecting a balance between these two aspects.

The Bert model achieved a remarkable accuracy of 90%, which demonstrates excellent ability in discerning emotion in text. The outstanding accuracy of 90% positions the BERT model as a robust and reliable tool for comprehensive text analysis, showcasing its potential in real-world applications.

These results provide valuable insights into the strengths and areas for enhancement in our models. Further experimentation, potentially with different architectures or hyperparameters, can refine the model's capabilities and contribute to a more robust emotion detection.

4.2.2 Sentiment Analysis

Naive Bayes

The table 4, presents the Accuracy, Precision, Recall and F1 Score of the Naive Bayes model used for sentiment analysis.

Metric	Value
Accuracy	0.5910
Precision	0.7068
Recall	0.5910
F1 Score	0.5528

Table 4 Naive Bayes Sentiment Analysis Metrics

The table 5, presents the classification report of the Naive Bayes model used for sentiment analysis.

Class	Precision	Recall	F1-Score	Support
Negative	0.80	0.32	0.46	8999
Neutral	0.87	0.32	0.47	11814
Positive	0.52	0.97	0.68	15035
Accuracy			0.59	35848
Macro Avg	0.73	0.54	0.53	35848
Weighted Avg	0.71	0.59	0.55	35848

Table 5 Classification Report for Naive Bayes

LSTM Model

The table 6, presents the Accuracy, Precision, Recall and F1 Score of the LSTM model used for sentiment analysis.

Metric	Value
Accuracy	0.8706
Precision	0.8799
Recall	0.8590
F1 Score	0.8693

Table 6 LSTM Model Sentiment Analysis Metrics

The table 7, presents the classification report of the LSTM model used for sentiment analysis.

Class	Precision	Recall	F1-Score	Support
Negative	0.82	0.81	0.81	9226
Neutral	0.87	0.91	0.89	11863
Positive	0.90	0.87	0.89	14759
Accuracy			0.87	35848
Macro Avg	0.86	0.87	0.86	35848
Weighted Avg	0.87	0.87	0.87	35848

Table 7 Classification Report for LSTM Model

BERT

The table 8, presents the Accuracy, Precision, Recall and F1 Score of the Fine-tuned BERT model used for sentiment analysis.

Metric	Value
Accuracy	0.88

Table 8 BERT Sentiment Analysis Metrics

The table 9, presents the classification report of Fine-tuned BERT model used for sentiment analysis.

Class	Precision	Recall	F1-Score	Support
Negative	0.85	0.81	0.83	9226
Neutral	0.86	0.94	0.90	11863
Positive	0.91	0.87	0.89	14759
Micro Avg	0.88	0.88	0.88	35848
Macro Avg	0.87	0.87	0.87	35848
Weighted Avg	0.88	0.88	0.88	35848
Samples Avg	0.88	0.88	0.88	35848

Table 9 Classification Report for BERT

Comparison

In comparing the sentiment analysis models, Naive Bayes demonstrated a moderate performance with an accuracy of 59.10%, showing lower precision and recall for negative and neutral sentiments. The LSTM model outperformed Naive Bayes significantly, achieving an accuracy of 87.06% with higher precision, recall, and F1 scores across all sentiment classes.

BERT, a state-of-the-art transformer-based model, exhibited superior performance with an accuracy of 88%. BERT showed balanced precision, recall, and F1 scores for all sentiment categories, indicating its robustness in capturing nuanced sentiment patterns.

Overall, BERT stands out as the most effective model among the three, showcasing its capability to handle complex sentiment analysis tasks with higher accuracy and generalization.

4.2.3 Sarcasm detection

Naive Bayes

The table 10, presents the Accuracy, Precision, Recall and F1 Score of the Naive Bayes model for Sarcasm detection.

Table 10 Sarcasm detection Performance - Naive Bayes

Metric	Value
Accuracy	0.87
Precision	0.87
Recall	0.85
F1 Score	0.86

Glove + LSTM

The table 11, presents the Accuracy, Precision, Recall and F1 Score of the Glove + LSTM model for Sarcasm detection.

Table 11 Sarcasm detection Performance - Glove + LSTM Model

Metric	Value
Accuracy	0.9529
Precision	0.9522
Recall	0.9453
F1 Score	0.9487

Comparison

This comparison highlights the superior performance of the Glove + LSTM Model, which achieved higher accuracy, precision, recall, and F1 score compared to Naive Bayes in sarcasm detection.

5 Conclusion & Perspectives

5.1 Conclusion

This project's thorough examination of sentiment, sarcasm, and emotion recognition in English text has provided insightful information about the relative merits of several approaches.

We have compared deep neural network topologies, optimized BERT models, and baseline methods to assess their performance on various tasks. The project expands its effect by creating a user-friendly Streamlit application in addition to exploring the technical elements of model construction, training, and assessment.

This program makes it easy to upload CSV files for batch processing and to enter single words for mood, emotion, or sarcasm prediction. In addition, the research examines the importance of mood, emotion, and sarcasm analysis in natural language processing (NLP), highlighting the usefulness of such efforts in real-world settings.

The results open the door for more study and advancements in the field of text analysis and natural language processing by offering a detailed grasp of the advantages and disadvantages of each model.

5.2 Perspectives

The process of working on this project, opend our eyes to new perspectives and future work possibilities, some of which are:

- **Pseudo-Labelling:** This technique is used when dealing with limited data and uses existing trained models to predict labels for unlabeled data and incorporate this data into your training set.
  It improves model generalization, making it exposed to a more diverse range of unseen examples. Eventually, this technique acts as a form of regularization that leads to a robust model that is more confident in its predictions.
- **Multi-Task Learning:** involves training a model to perform multiple tasks simultaneously. In our case, a multi-task model should be able to predict sentiment, emotion and sarcasm.
  This method allows the model to learn shared representations across tasks, in our case, that would be informations and contexts relevant for sentiment, emotion, and sarcasm analysis.
- **Integration:** The pseudo-labeled data can be used later in the multi-task learning, to enhance its ability to handle diverse scenarios.
  It's also important to experiment with different fine-tuning strategies, and regularly monitor and evaluate the results given by the model.