CS4765/CS6765
Recognizing Emotions in Poems using BERT

Rohith Kumar Saravanan
rohithkumar.s@unb.ca
Student ID: 3756386

ABSTRACT

Poetry is often difficult to understand because of the use of complex language, metaphors, and idioms. This project makes use of NLP to identify predominant emotions in poems by treating the problem as a multi-class classification problem wherein poems are classified into seven emotion labels: Anger, Disgust, Fear, Joy, Neutral, Surprise, or Sadness. We fine-tuned BERT on the Kaggle Poem Dataset, which consisted of 450 poems labeled with these emotions. The model performance was compared to that obtained with simpler baselines consisting of the most-frequent class and Naive Bayes classifiers. Although fine-tuned BERT slightly outperforms the baseline models, it falls short for this task of emotion recognition in poetry compared to emotion recognition in simple texts.

KEYWORDS

Poem; Emotion Detection; Natural Language Processing; Multi-class Classification; Transformers; BERT

1 INTRODUCTION

Poetry often expresses emotions using complex and abstract languages, such as metaphors and idioms. This kind of creative writing poses a very special challenge for both human readers and automated systems in understanding the underlying emotions conveyed through the text. Traditional methods of emotion detection in text mostly fail when applied to poetry because of the figurative nature of the language used.

This project is focused on the application of Natural Language Processing (NLP) techniques for automatically detecting the predominant emotions in poems. We approach this problem as a multi-class classification task, where the goal is to classify each poem into one of seven emotion categories: *Anger, Disgust, Fear, Joy, Neutral, Surprise,* and *Sadness*.

For this, we leverage BERT (Bidirectional Encoder Representations from Transformers), a state-of-the-art NLP model which is known for its deep contextual information capture in text. BERT is pre-trained on large amounts of data and has shown massive performance improvements on a variety of

NLP tasks. We fine-tune this pre-trained model on the Kaggle Poem Dataset, which contains 450 poems labeled with these seven emotions.

In addition to fine-tuning BERT, we also compare its performance with simpler baseline models. Compared with the most-frequent class classifier and Naive Bayes, these comparisons can give a good proof for BERT's effectiveness in catching such nuances of emotional expression in poetry. Models are evaluated using the accuracy of the test set due to class imbalance.

This project proposes ways to improve emotion detection performance in poetic texts and contributes to the wider application of NLP in literary analysis and creative text understanding.

2 BACKGROUND

Emotion detection in text [1] is a challenging task in NLP because understanding emotional text requires not only an understanding of words but also their emotional contexts. The challenge is more extended in poetry [2], where the use of language is generally figurative and metaphorical. Classical machine learning models like logistic regression, SVM, and Naive Bayes rely on hand-engineered features like the Bag-of-Words model or TF-IDF; therefore, they fail to capture the subtlety in emotional expressiveness due to poetic language.

With the rise of deep learning, models such as RNNs, LSTMs, and transformer models like BERT [3], superior performance has been observed across many NLP tasks. BERT, which is pre-trained on large text corpora, captures the contextual relationships in a sentence, making it particularly effective for emotion recognition.

The project applies BERT to emotion detection in poetry, which is relatively new ground compared to previous research focused on prose or social media text. This work uses the Kaggle Poem Dataset [4], comprising 450 poems labeled with seven emotions: Anger, Disgust, Fear, Joy, Neutral, Surprise, and Sadness. We fine-tune BERT on this dataset to improve emotion recognition in poetic language. The most-frequent class, Logistic Regression and Multinomial Naive Bayes are used as baseline models to evaluate BERT's performance.

3 METHODOLOGY

The project applies supervised learning to identify emotions in poems. In this approach, the model is trained on labeled data and tested for its performance. The methodology adopted in this work is described below:

3.1 Data Preprocessing

The text data is preprocessed to standardize it for analysis:

- **Lowercasing:** All text is converted to lowercase.
- **Removing Punctuation and Digits:** Special characters and digits are removed.
- **Whitespace Removal:** Extra spaces are removed.

These steps ensure that the models focus on the emotional content of the poems rather than formatting or noise.

3.2 Model Selection and Training

We utilize several models for emotion detection and compare their performances as follows:

- **Baseline Model: Most Frequent Classifier** - Assigns the most frequent emotion label to all poems.
- **Logistic Regression with BoW** - Changes text into fixed-length vectors using Bag-of-Words, followed by logistic regression for classification.
- **Multinomial Naive Bayes:** A popular probabilistic model for discrete data such as word counts.
- **BERT:** Pre-trained deep model fine-tuned on the dataset to learn contextual meanings of poetry.
- **DistilBERT:** Distilled version of BERT. This is more light weight and fast compared to BERT base model.

Kaggle Poem Dataset is used with 450 poems labeled across seven emotions. This will be split into 80% training and 20% test sets. Now, fine-tune BERT as follows:

- **Tokenization:** Split text into subwords and map each subword to a token ID.
- **Input Formatting:** Tokenized text is padded or truncated to a uniform length.
- **Fine-Tuning:** BERT is fine-tuned for better emotion classification.

3.3 Evaluation

Model performance is evaluated with the following metrics [5]:

- **Accuracy:** The ratio of correctly classified poems to the total number of poems.
- **Macro-Average F1 Score:** Measures class balance by averaging F1 scores across classes.
- **Precision, Recall, and F1-Score:** Give insights into the model's ability to detect emotions correctly and minimize false positives/negatives.

The evaluation accuracy of the fine-tuned BERT model is compared with baseline models to determine its effectiveness.

3.4 Results and Comparison

The trained models are tested on the test set, and their results are compared using the above-mentioned evaluation metrics to demonstrate the effectiveness of BERT in emotion detection.

4 IMPLEMENTATION DETAILS

This section provides an overview of the tools, libraries, dataset and workflow used to implement the emotion detection system for poems.

4.1 Libraries and Frameworks

To build the emotion detection system, the following libraries and frameworks were used:

- **Hugging Face Transformers**: This library provides pre-trained models for NLP tasks, including BERT. It simplifies the process of fine-tuning BERT on the specific dataset for emotion recognition.
- **scikit-learn**: A popular Python library for machine learning that was used to implement traditional models like Logistic Regression and Naive Bayes. It also provided utilities for data preprocessing, splitting datasets, and evaluating model performance.
- **pandas**: A data manipulation and analysis library used to load and process the dataset. It helps in reading the Kaggle Poem Dataset and performing initial data transformations.
- **NumPy**: A library for numerical computing, used for handling arrays and performing mathematical operations during model training and evaluation.
- **PyTorch**: A deep learning framework that provides tools to work with neural networks, used to fine-tune BERT and other models.
- **Datasets by Hugging Face**: Used for creating dataset splits and managing the data in a structured format compatible with transformer models.

4.2 Dataset

The Kaggle Poem Dataset, which consists of 450 poems, is used for training and evaluating the emotion detection models. Each poem is labeled with one of seven emotions: Anger, Disgust, Fear, Joy, Neutral, Surprise, and Sadness. The dataset is well-suited for this task as it provides a diverse range of poetic expressions, which can help in training the models to generalize to various types of poetic language.

The dataset is split into three parts:

- **Training Set**: 80% of the dataset is used for training the models.
- **Validation Set**: 10% of the dataset is used for validating the model during training.
- **Test Set**: The remaining 10% is reserved for evaluating the models' performance.

4.3 Model Design and Workflow

The following steps outline the workflow used to implement the emotion detection system:

1. **Data Loading**: The Kaggle Poem Dataset is loaded into a pandas DataFrame. Each poem is paired with its corresponding emotion label, which is encoded numerically for model training.
2. **Text Preprocessing**: The text data is pre-processed as described in the methodology section (lowercasing, punctuation and digit removal, whitespace removal).

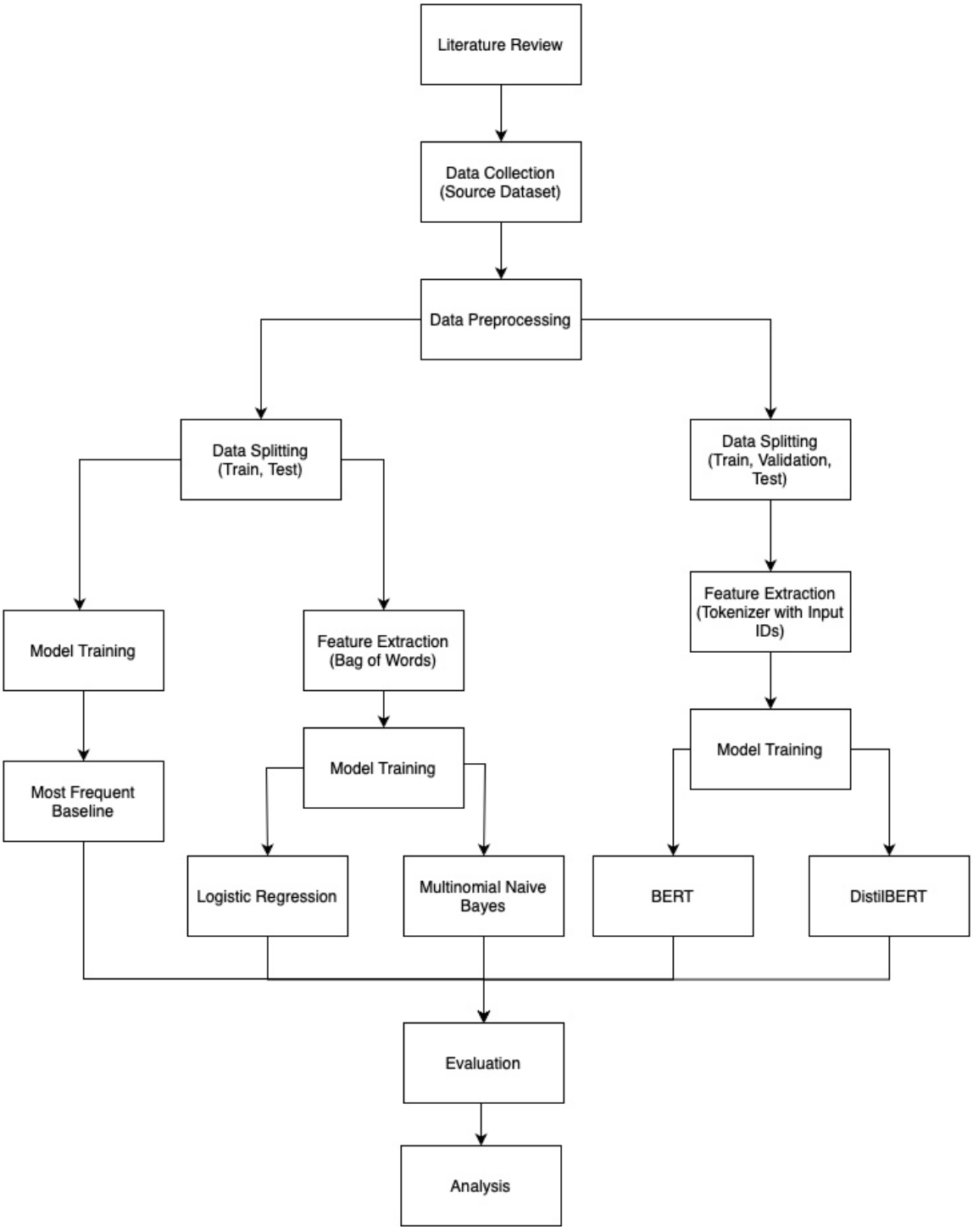

Figure 1: Workflow for recognizing emotions in poems

3. **Feature Extraction**: For the classical models (Logistic Regression, Naive Bayes), the text data is converted into numerical features using the Bag-of-Words (BoW) model. For BERT, the text is tokenized and converted into input IDs compatible with the model.

4. **Model Training**: The models are trained using the training set. For traditional models, the data is vectorized, and classifiers are trained. For BERT, the model is fine-tuned on the training data to adjust its weights.

5. **Model Evaluation**: After training, the models are evaluated on the test set using metrics like accuracy, macro-average F1 score, precision, recall, and accuracy is used to compare performance across different models.

4.4 Fine-Tuning BERT

BERT is fine-tuned for the emotion recognition task as follows:

- **Tokenizer**: The BERT tokenizer is used to convert the raw text into tokens. These tokens are then mapped to token IDs that are compatible with BERT's pre-trained model.
- **Input Length Management**: The tokenized input is padded or truncated to ensure all inputs have the same length, which is necessary for feeding data into the BERT model.
- **Optimization and Loss Function**: BERT is fine-tuned using cross-entropy loss and the Adam optimizer, both of which are commonly used in classification tasks. The model's parameters are adjusted during training to minimize the loss and improve the classification accuracy.
- **Evaluation**: After fine-tuning, BERT is evaluated on the test set, where it predicts the emotion for each poem and is compared to the actual labels to compute performance metrics.

5 EVALUATION

5.1 Experimental Setup

The experiments were run on google colab notebooks with T4 GPU to accelerate the training process for transformer models.

5.2 Evaluation Metrics

Accuracy has been chosen as the primary metric due to strong class imbalance. Accuracy on the test set is considered from all the models for comparison.

5.3 Experimental Results

The comparison of the performance of baseline and transformer-based models is shown in Table 1.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Most-Frequent Baseline	27.78	8.00	28.00	12.00
Logistic Regression (BoW)	37.04	38.00	37.00	31.00
Multinomial Naive Bayes	41.48	23.00	41.00	30.00
BERT (Fine-Tuned)	44.44	-	-	-
DistilBERT (Fine-Tuned)	33.33	-	-	-

Table 1: Performance comparison of baseline and transformer-based models.

5.4 Analysis

The Most-Frequent Baseline performed poorly, reflecting its simplistic approach. Logistic Regression and Multinomial Naive Bayes provided moderate improvements but lacked the ability to capture complex textual relationships. Fine-tuned transformer models—BERT and DistilBERT—did not provide any significant enhancements. BERT only slightly outperforms Naive Bayes and DistilBERT falls below Logistic Regression. However, the contextual understanding of poems by transformer models can be noteworthy, especially in creative fields like poetry. This demonstrates the difficulty of task in hand and how even state-of-the-art models like BERT and DistilBERT fail to rightly classify emotions in poetry.

6 CHALLENGES

This project had several challenges in emotion detection within poems:

- **Complexity of Poetic Language:** Poetry is rich in metaphorical and figurative language, which makes it hard to map words to emotions directly. The subtlety and ambiguity in poetic expression added a layer of complexity to the emotion recognition task.
- **Class Imbalance:** The Kaggle Poem Dataset has an imbalanced distribution of emotions, which may result in poor performance of the model on under-represented classes. The data was heavily imbalanced with sadness and fear comprising about 60% of the total dataset. The other 5 emotions accounted for 40% of the total data which shows the dominance of the other two emotions.
- **Fine-Tuning BERT:** Although BERT is a powerful model, the fine-tuning required much computational resource and hyper-parameters tuning like batch size, learning rate, and number of epochs to reach the optimal point in detecting emotions in poetry.
- **Data Preprocessing:** The text preprocessing for BERT itself included handling variable poem lengths, tokenization, and padding/truncation, which introduced an additional layer of complexity in the implementation.
- **Low Accuracy Despite Multiple Models:** Despite using a range of models, from traditional machine learning classifiers to BERT, the accuracy was relatively low at the start. Even fine-tuning and hyperparameter tuning did not significantly improve this issue, reflecting the challenge of emotion detection in poetry due to its intrinsic complexity.

7 CONCLUSION

This project explores the use of Natural Language Processing (NLP) for detecting emotions in poetry, a task made challenging by the figurative language in poems. It frames the problem as a multi-class classification task, categorizing poems into seven emotions: Anger, Disgust, Fear, Joy, Neutral, Surprise, and Sadness. The project fine-tunes the BERT model on a Kaggle Poem Dataset, comparing its performance with simpler baseline models. After training and evaluating different models, it was observed that, the best accuracy achieved was with BERT (44.44%) which was only a 3% increase over multinomial naive bayes. In conclusion, the project demonstrated how recognizing emotions in complex textual data can be a difficult task even for state-of-the-art models like BERT and fall short significantly in comparison to simple textual data.

8 ETHICS STATEMENT

Ethical consideration in emotion detection tasks revolves around biases and potential misuse. Lack of diversity in training might cause biases, leading to either inaccurate or unfair prediction, especially across different demographics such as culture, gender, or age.

Moreover, this work can be misused in a way against the poets by blaming them for instilling certain emotions in the audience. In its use, fairness, transparency, this work shall be guided by moral and ethical considerations.

REFERENCES

[1] Alm, C. O., Roth, D., & Sproat, R. (2005). Emotions from text: Machine learning for text-based emotion prediction. *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 579–586.

[2] Haider, I., Ghazal, M., Arif, M., & Yousaf, M. (2020). Challenges in Sentiment and Emotion Detection in Poetry: A Survey. *Journal of Intelligent Information Systems*, 55(3), 541–563.

[3] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186.

[4] Kaggle Poem Dataset. Available at: `https://www.kaggle.com/datasets/mexwell/poem-dataset`

[5] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427–437.