CLEF 2024 JOKER Task 2: Using BERT and Random Forest Classifier for Humor Classification According to Genre and Technique*

M Saipranav[1], Jaswanth Sridharan[1], Gautham Narayan G[1], Angel Deborah S[1], Rajalakshmi S[1], Mirnalinee T T[1] and Samyuktaa Sivakumar[1]

[1]*Sri Sivasubramaniya Nadar College Of Engineering, Chennai*

Abstract

In this paper, we present our work for the Automatic Humour Analysis (JOKER) Lab at CLEF 2024. The objective of the JOKER Lab is to research the automated processing of humour that includes tasks such as retrieval, classification, and interpretation of various forms of humorous texts. Our task involved the classification of humorous texts into different genres for which we undertook two different approaches. These approaches involved the usage of BERT (a transformer architecture) and a traditional machine learning model such as a Random Forest classifier. Out of the two models, BERT had a higher accuracy score of 0.6731. From this, we concluded BERT is better for most Natural Language Processes. We showcase our experiments on the training data and the results on the provided test dataset are presented in the forthcoming pages.

Keywords

Humor, Genre Classification, BERT, TF-IDF Vectors, Sentence Embedding, Random Forest

1. Introduction

Humor plays a crucial role in human communication and social interaction. However, it is multifaceted and elicits different types of responses from various types of audiences. Accurate classification of humor not only enhances our understanding of its various forms but also has practical application in fields such as sentiment analysis, human-computer interaction and social media content moderation.

Traditional humor classification techniques can be labor and time consuming. Automating this process through NLP and ML techniques can improve the efficiency and accuracy of humor classification, benefiting academic research. With the proliferation of digital media, humor is more pervasive and varied than ever, presenting a challenge to even state of the art models to discern the differences between various genres of humor.

The CLEF 2024 JOKER [1][2][3] Track comprised of 3 tasks, which were: Task 1- Humor-aware information retrieval[1], Task 2- Humour classification according to genre and technique[1] and Task 3- Translation of puns from English to French[1]. We participated in task 2.

By leveraging some advanced natural language processing techniques and fine-tuning some of the well-known pre-trained models, this study for the chosen task - 2 aims to develop a system capable of accurately classifying text into the following humor categories

- **IR** - Irony relies on a gap between the literal meaning and the intended meaning, creating a humorous twist or reversal.
- **SC** - Sarcasm involves using irony to mock, criticize, or convey contempt.
- **EX** - Exaggeration involves magnifying or overstating something beyond its normal or realistic proportions.

CLEF 2024: Conference and Labs of the Evaluation Forum, September 09–12, 2024, Grenoble, France
saipranav2310324@ssn.edu.in (M. Saipranav); jaswanth2310325@ssn.edu.in (J. Sridharan); gauthamnarayan2310332@ssn.edu.in (G. N. G); angeldeborahs@ssn.edu.in (A. D. S); rajalakshmis@ssn.edu.in (R. S); mirnalineett@ssn.edu.in (M. T. T); samyuktaa2210189@ssn.edu.in (S. Sivakumar)

- **AID** - Incongruity refers to the unexpected or contradictory elements that are combined in a humorous way and Absurdity involves presenting situations, events, or ideas that are inherently illogical, irrational, or nonsensical.
- **SD** - Self-deprecating humor involves making fun of oneself or highlighting one's own flaws, weaknesses, or embarrassing situations in a lighthearted manner.
- **WS** - Wit refers to clever, quick, and intelligent humor and Surprise in humor involves introducing unexpected elements, twists, or punchlines that catch the audience off guard.

This automated approach significantly benefits various fields by providing deeper insights into the mechanics of humor and enhancing the way machines understand and respond to human emotions.

2. Approach

We took up 2 approaches for the humor classification task: multiclass classification using BERT base uncased and classification using Random Forest classifier. Preprocessing of the data was done differently for both methods.

2.1. Data Preparation

The provided dataset [3] consisted of 1742 examples of text that must be classified into the above-mentioned 7 genres of humor. We partitioned the dataset into an 80% training dataset and a 20% validation dataset. The content from the dataset was of the following format

Table 1
Different Classes of Humor from the given Train Dataset

Id	Text	Class	Number of texts available per class
1112	Honesty may be the best policy, but insanity is the best defense.	SC	356
782	no more instagram. we must all return to scrapbooks.	IR	212
484	The answer is going to a grocery store during a pandemic . That's what I'd do for a Klondike bar	EX	125
1613	Knock knock. Who's there? Tank. Tank who? You're welcome.	AID	232
167	All my imaginary friends tell me that I need therapy.	SD	169
2140	The winter drive - by shooting was a slay ride.	WS	537

Basic text preprocessing was done to the provided dataset. Firstly, the class identifiers for each humorous text were mapped with respective numerical values. All texts were stripped of punctuation, stop words, and other special characters. These texts were then lemmatized. This preprocessed dataset was directly used for BERT (see figure 1)

For the approach involving the use of the Random Forest classifier, the preprocessed text data were further prepared by combining Sentence Transformer, a pre-trained model, and TfidfVectorizer, a scikit-learn tool, to generate sentence embeddings and TF-IDF feature vectors, respectively (see figure 2)

SentenceTransformer: This pre-trained model (multi-qa-mpnet-base-dot-v1) from the sentence-transformers library is utilized to generate sentence embeddings. This model captures the semantic meaning of text at the sentence level, effectively embedding the contextual nuances and relationships between words within sentences.

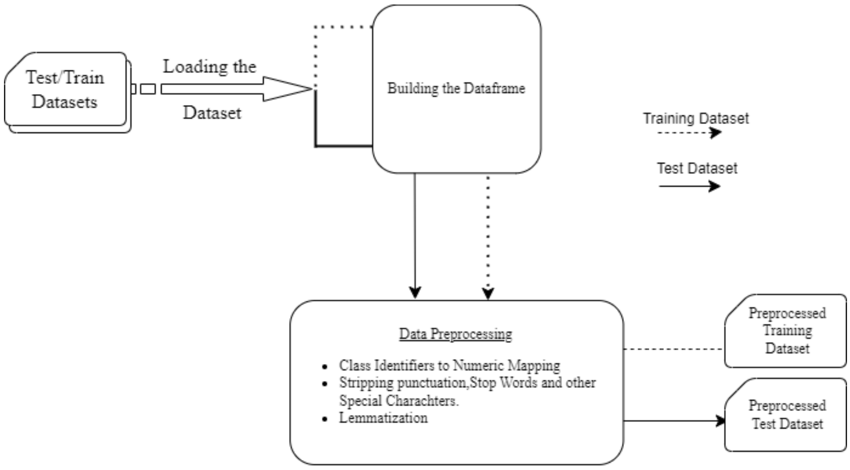

Figure 1: Data Preprocessing

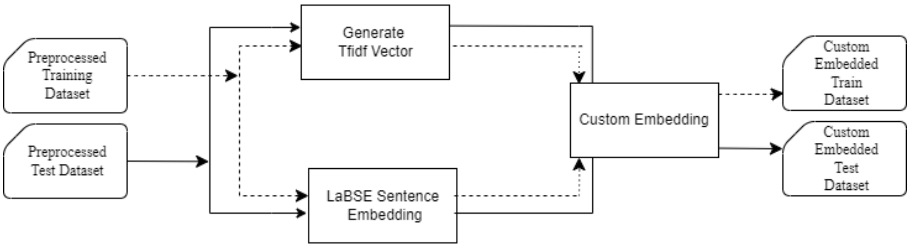

Figure 2: Sentence Embedding/TFidf Vectorisation of Preprocessed Data

TfidfVectorizer : This is a tool from scikit-learn that converts textual data into TF-IDF feature vectors. TF-IDF vectors highlight the importance of words within a document relative to the entire corpus, thus providing a measure of the significance of terms.

To generate TF-IDF vectors for the test and training data, the TF-IDF vectorizer is first fitted to the text data within the training set. This fitting process involves learning the vocabulary and the inverse document frequency (IDF) values from the training corpus. After fitting, the text data is transformed into TF-IDF vectors, resulting in a sparse matrix representation of the documents where each entry reflects the importance of a term within a document. The SentenceTransformer model encodes the training text data into sentence embeddings, which capture the semantic content of the text. The concatenation of TF-IDF vectors and sentence embeddings in each document creates a comprehensive feature set that considers both local word importance and sentence semantic meaning.

The target labels (classes) are extracted from the data frame to prepare the target variable for model training and evaluation. This extraction isolates the dependent variable, which the machine learning model will learn to predict based on the input feature set which is a combination of the TF-IDF vectors and sentence embeddings.

2.2. Methodology

2.2.1. BERT

BERT [4] stands for Bidirectional Encoder Representations from Transformers. It is faster and is better at capturing context than normal Long Short Term Memory or other traditional models. BERT is pre-trained on a large corpus of text using two unsupervised learning tasks namely Masked Language Model (MLM) and Next Sentence Prediction(NSP). In MLM, a percentage of the input tokens are randomly masked, and the model is trained to predict the original tokens based on the context of the surrounding words. This bidirectional context allows BERT to learn representations that capture deeper semantic meaning. For NSP, pairs of sentences are sampled from the corpus, and the model is trained to predict

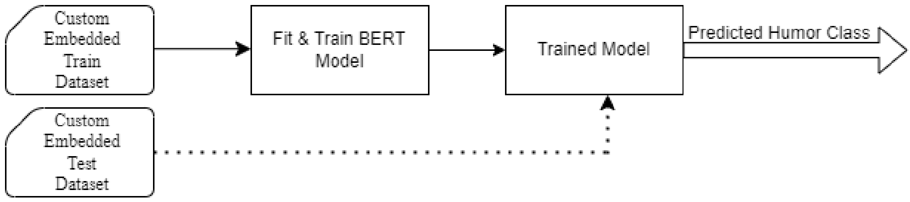

Figure 3: BERT Classification Process

whether the second sentence follows the first one. This exercise helps BERT to understand relationships between sentences and improves its ability to handle tasks like question answering and natural language inference.

BERT [5] consists of a stack of Transformer encoder layers. In the case of BERT Base Uncased, it has 12 such layers. Each layer contains self-attention mechanisms and feedforward neural networks.

At every layer, BERT calculates the attention scores for each token in the input sequence, indicating the importance of other tokens about it. This allows BERT to understand contextual information by trying to understand all tokens in the input sequence simultaneously, in both directions. After self-attention, the output is passed through a feedforward neural network, typically with a ReLU activation function. This network helps find complex patterns in the data and further improves the representations learned by the self-attention mechanism (see figure 3).

Before inputting text into BERT, it undergoes tokenization into subword units using WordPiece tokenization. This allows BERT to handle out-of-vocabulary words effectively. Each input sequence is then represented as a combination of three types of embeddings namely token, segment, and positional embedding. Token Embedding represents the identity of each token in the input sequence. These embeddings are learned during the pre-training stage and understand the semantic meaning of individual words. Segment Embedding indicates whether a token belongs to the first sentence or the second sentence in a pair of sentences. This helps BERT understand the relationship between sentences, especially in tasks like question answering and natural language processing. Positional Embedding encodes the position of each token in the input sequence allowing BERT to capture sequential information and understand the order of words in a sentence.

After pre-training, BERT can be fine-tuned on specific tasks using task-specific labeled data[6]. During fine-tuning, the pre-trained parameters are adjusted to optimize performance on the task. Fine-tuning BERT on specific tasks enables it to achieve state-of-the-art results across various natural language processing tasks.

2.2.2. Random Forest

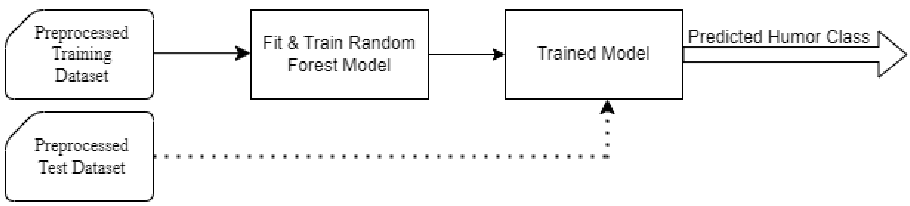

Figure 4: Random Forest Classification Process

Random Forest [7] is an ensemble classifier that contains several decision trees. Instead of using a single decision tree, this ensemble method leverages the decision-making ability of multiple decision trees and based on the majority number of predictions, the final output is predicted. The prepared input feature set is passed to the Random Forest classifier comprising 1500 decision trees. (see figure 5) The use of out-of-bag samples is also enabled to estimate the generalization accuracy of the model. This provides an internal cross-validation measure of the model performance. Decision trees make

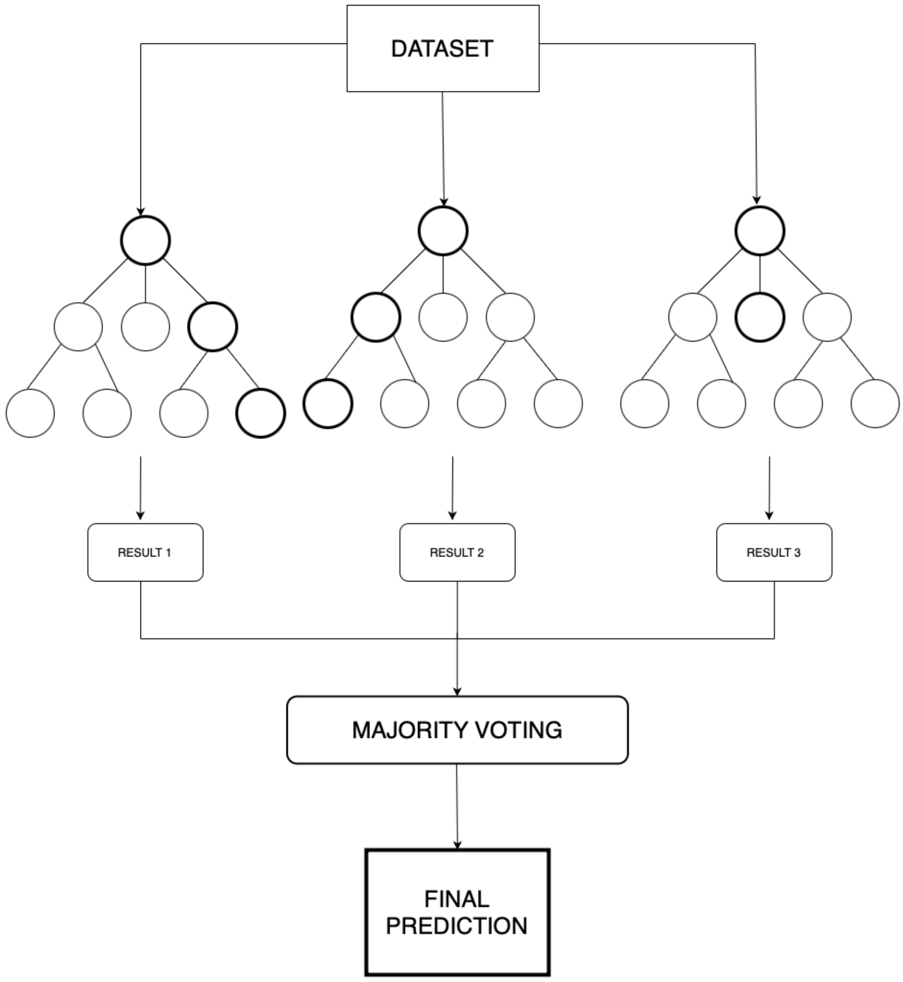

Figure 5: A Random Forest comprising of 3 decision trees.

up the most fundamental component of the Random Forest classifier. Each decision tree works to find the best split to divide the data into multiple subsets and is trained through the Classification and Regression Tree (CART) algorithm. Gini impurity, information gain, or mean square error are some of the commonly used metrics to evaluate the quality of the split. A single decision tree can be prone to bias and over-fitting, hence an ensemble classifier consisting of multiple decision trees is used to improve the accuracy of the predictions. Random Forest algorithm (see figure 4) makes use of bagging and feature randomness to create an uncorrelated forest of decision trees. Each tree in the ensemble comprises of data sample drawn from the provided training data set with replacement. One-third of it is set as the out-of-bag sample. The diversity of the dataset is increased and correlation among the decision trees is reduced through feature bagging. For a classification task, such as the one performed, the most frequent categorical variable will yield the predicted class. Finally, the out-of-bag sample is used for cross-validation.

3. Results

The metrics of precision, recall, accuracy, and f1-score are reported for the two models that were used to complete the given task. Precision is calculated mathematically as the ratio of true positives and the sum of true and false positives. Accuracy is the ratio of the number of correct predictions to the total number of data points. Recall is calculated as the ratio of true positive and the sum of true positive and false negative. The F1 score is calculated from the values of precision and recall. It mathematically, is equal to twice the ratio of the product of precision and recall to the sum of precision and recall.

Tables 2 and 3 summarise the results of our runs as sent by the Joker lab for the fore mentioned approaches. These were carried out on the provided test dataset. Using a transformer architecture model such as BERT gave a higher accuracy of 0.6731 compared to a traditional machine learning model such as the Random Forest Classifier which only gave a accuracy of 0.5235.

Table 2
Accuracy Metrics

Model	Accuracy
BERT	0.6731
Random Forest	0.5235

Table 3
Precision, Recall and F1 scores

Model	Type	Precision	Recall	F1
BERT	macro	0.6024	0.6027	0.6006
	weighted	0.6662	0.6731	0.6687
Random Forest	macro	0.5353	0.3736	0.3742
	weighted	0.5278	0.5223	0.4583

4. Conclusions

As mentioned before two different approaches were used to solve the given task. The first approach involved using a transformer architecture such as BERT. The second approach involved using a traditional machine learning model such as a Random Forest classifier. Higher accuracy (0.6731) of BERT suggests that using transformer architecture like BERT for classification proves to be more accurate than traditional and feature-dependent machine learning models that are commonly used for classification. Overall, it can be concluded that BERT's deep contextual and language understanding with its ability to leverage transfer learning, makes it better suited for the nuanced task of humor classification according to genre.

References

[1] L. Ermakova, A.-G. Bosser, T. Miller, V. M. Palma Preciado, G. Sidorov, A. Jatowt, Overview of JOKER @ CLEF-2024: Automatic humour analysis, in: L. Goeuriot, P. Mulhem, G. Quénot, D. Schwab, L. Soulier, G. M. D. Nunzio, P. Galuščáková, A. G. S. de Herrera, G. Faggioli, N. Ferro (Eds.), Experimental IR Meets Multilinguality, Multimodality, and Interaction: Proceedings of the Fifteenth International Conference of the CLEF Association (CLEF 2024), Lecture Notes in Computer Science, Springer, Cham, 2024. To appear.

[2] L. Ermakova, A.-G. Bosser, A. Jatowt, T. Miller, The joker corpus: English-french parallel data for multilingual wordplay recognition, in: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '23, Association for Computing Machinery, New York, NY, USA, 2023, p. 2796–2806. URL: https://doi.org/10.1145/3539618.3591885. doi:10.1145/3539618.3591885.

[3] L. Ermakova, A.-G. Bosser, T. Miller, T. Thomas-Young, V. Preciado, G. Sidorov, A. Jatowt, CLEF 2024 JOKER Lab: Automatic Humour Analysis, 2024, pp. 36–43. doi:10.1007/978-3-031-56072-9_5.

[4] M. V. Koroteev, Bert: A review of applications in natural language processing and understanding, 2021. arXiv:2103.11943.

[5] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).

[6] S. Prabhu, M. Mohamed, H. Misra, Multi-class text classification using bert-based active learning, arXiv preprint arXiv:2104.14289 (2021).
[7] G. Biau, E. Scornet, A random forest guided tour, Test 25 (2016) 197–227.