A Comparative Study of Multimodal Toolkit with BERT and Random Forest for Multiclass Classification

RAHUL K JOHNY, Dublin City University, Ireland

Multiclass classification is a difficult task in natural language processing, however recent developments in deep learning models have improved performance dramatically. In this paper we investigate the application of BERT, a state-of-the-art transformer-based model, for multiclass classification and prediction tasks in this study. On Etsy's dataset, we compare its performance to random forest, and our results show that while BERT performed well for two of the three labels, random forest performed well for one.

CCS Concepts: • **Computing methodologies → Machine learning algorithms; Multiclass classification**.

Additional Key Words and Phrases: BERT, transformers, multimodal, multiclass classification, random forest

ACM Reference Format:
Rahul K Johny. 2023. A Comparative Study of Multimodal Toolkit with BERT and Random Forest for Multiclass Classification. 1, 1 (April 2023), 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Multiclass classification is a fundamental problem in natural language processing, which involves assigning a given input to one of multiple possible classes. In this study, we focus on multiclass classification for three different labels using a dataset given by Etsy. To accomplish this objective, we use BERT, a pre-trained transformer-based model, and a package developed for learning with transformers on tabular and text data. We specifically use the BERT implementation from the paper [Gu and Budhkar 2021]. This package provides an effective and scalable method for integrating text and tabular data for multimodal learning, which has been shown to increase text classification task performance .Our aim is to assess the effectiveness of BERT and the package in multiclass classification and compare their performance against other baseline models, including random forest. The evaluation is based on the F1 score, a commonly used metric for measuring the performance of classification models.

2 RELATED WORK

[Kowsari et al. 2019] categorizes text classification algorithms into five major sections: feature extraction methods, dimensionality reduction methods, existing classification algorithms, evaluation methods, and critical limits of each component. Methods for extracting features, such as Term Frequency-Inverse Document Frequency (TF-IDF), word embedding, and text cleaning, are frequently utilised in both academic and commercial settings. Dimensionality reduction approaches, such as principle component analysis (PCA), linear discriminant analysis (LDA), and t-distributed Stochastic Neighbour Embedding (t-SNE), can help existing text classification algorithms reduce their time and memory complexity.

[Kowsari et al. 2019] also discusses about existing classification algorithms such as the Rocchio algorithm, bagging and boosting, logistic regression (LR), Naïve Bayes Classifier (NBC), k-nearest Neighbor (KNN), Support Vector Machine (SVM), decision tree classifier (DTC), random forest, conditional random field (CRF), and deep learning. Furthermore, evaluation methods such as accuracy, Matthew correlation coefficient (MCC), receiver operating characteristics (ROC), and area under curve (AUC) are discussed that can be used to evaluate the text classification algorithms. The paper covers the critical limitations of each component of the text classification pipeline and compares the most common text classification algorithms. Finally, the authors discuss the usage of text classification in various fields such as medicine, law, and other majors. In summary, the paper provides a comprehensive survey of various text classification algorithms, their strengths and weaknesses, and their suitability for different applications.

[Chaudhary et al. 2016] describes a new Random Forest Classifier (RFC) technique for multi-class illness classification issues. To improve the efficiency of the RFC algorithm, the methodology combines the Random Forest machine learning algorithm with an attribute evaluator method and an instance filter method. The authors put the improved-RFC approach to the test on five benchmark datasets and the groundnut disease diagnostic multi-class classification problem.

The results reveal that the suggested approach outperforms the traditional RFC algorithm, with disease classification accuracy increasing up to 97.80% for the groundnut disease dataset. When compared to the RFC method, the improved-RFC technique with CFS, SU, and Gain Ratio improves disease classification accuracy. The improved-RFC approach is also tested for classification accuracy, F-measure, and sensitivity values using 10-fold cross-validation on five benchmark datasets from the UCI machine learning library, and it outperforms the RFC algorithm.

The authors conclude that the improved-RFC method is a viable alternative for dealing with computer-aided diagnostic and multi-class classification issues. This study makes an important contribution to disease detection by demonstrating the efficacy of combining attribute evaluator and instance filter methods with the RFC algorithm to improve classification performance.

[González-Carvajal and Garrido-Merchán 2020] presents an empirical study of the BERT (Bidirectional Encoder Representations from Transformers) model's performance versus traditional NLP (Natural Language Processing) techniques, such as TF-IDF (Term Frequency-Inverse Document Frequency) vocabulary fed to machine learning algorithms. The authors hope to contribute empirical

Author's address: Rahul K Johny, rahul.kuruppasseryjohny2@mail.dcu.ie, Dublin City University, Ballymun Road, Dublin, Co. Dublin, Ireland, D09W6Y4.

XXXX-XXXX/2023/4-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

evidence to support the usage of BERT as the default approach for NLP tasks. The study offers four different NLP scenarios in which BERT outperformed the typical NLP methodology, and the authors point out that applying BERT is simpler than implementing old methods. The study also emphasises the significance of transfer learning and pre-training in reaching these outcomes.

The authors acknowledge that the BERT model has limitations, and its results can be improved. They suggest researching a hyper-parameter auto-tuned BERT model using Bayesian Optimization for any new NLP task. The authors also suggest using an auto-tuned BERT model to enable classification of language messages for robots, showing consciousness correlated behaviors.

Overall, the paper provides empirical evidence to support the use of BERT as a default technique for NLP tasks. It also highlights the importance of transfer learning and pre-training in achieving good performance.

3 METHODOLOGY

In this study, both the Random Forest and BERT models will be used to classify different labels on the dataset provided by Etsy. The BERT model will be trained using a multimodal approach that incorporates both text and tabular data. On the other hand, the Random Forest model will solely rely on text data for analysis. By comparing the performance of these two models, the study aims to determine the effectiveness of using a multimodal approach with BERT in mutliclass classfication

3.1 Random Forest

Random forest is a popular machine learning method used to develop prediction models. It was first introduced by [Breiman 2001] as a collection of classification and regression trees, which use binary splits on predictor variables to determine outcome predictions. While decision trees offer an intuitive method for predicting outcomes, they often provide poor accuracy for complex datasets. In the random forest setting, many trees are constructed using randomly selected training datasets and random subsets of predictor variables for modeling outcomes. Results from each tree are combined to give a prediction for each observation, resulting in higher accuracy compared to a single decision tree model. Additionally, random forests maintain some of the beneficial qualities of tree models, such as the ability to interpret relationships between predictors and outcomes. Therefore, random forests are a popular choice for classification tasks and consistently offer among the highest prediction accuracy compared to other models. Random forest is advantageous in modeling predictions because it can effectively manage large datasets containing a vast amount of predictor variables [Speiser et al. 2019]. In this report, we will be using random forest methodology as the baseline classification model to compare with BERT.

3.2 Multimodal Package with BERT

For our study, we are utilizing the multimodal package from [Gu and Budhkar 2021] in combination with BERT. Our approach involves using both text and tabular data. Specifically, the text data includes features such as 'title', 'description', and 'tags', while the 'type' column serves as our tabular data. The model's framework comprises a data processing module that generates processed text, numerical, and categorical features. These features are then fed as inputs to our Transformer With Tabular module, which consists of a Hugging Face Transformer and a combining module.

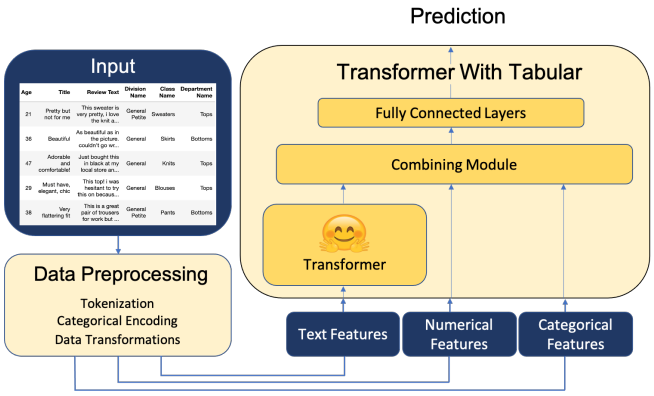

Fig. 1. The framework of Multimodal-Toolkit. Figure from [Gu and Budhkar 2021] A data processing module generates processed text, numerical, and categorical features, which are subsequently utilized as input to our Transformer With Tabular module. The latter comprises a Hugging Face Transformer and a combining module.

3.3 F1-Score

In the case of multiclass classification, precision, recall, and F1 score can be calculated for each individual class, as well as for the overall classification model. In this scenario, precision refers to the proportion of true positives for a specific class out of all predicted positives for that class, while recall refers to the proportion of true positives for that class out of all actual positives for that class. The F1 score for a specific class is the harmonic mean of the precision and recall for that class [Grandini et al. 2020].

$$F1\text{-}Score = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (1)$$

The equation is from [Grandini et al. 2020]. The overall F1 score for the multiclass classification model can be calculated by taking the weighted average of the F1 scores for each class, with the weights being the proportion of samples in each class. It is important to consider the performance of each individual class in addition to the overall model performance, as certain classes may be more difficult to classify than others. By considering both precision and recall, the F1 score provides a balanced evaluation of a classification model's performance. Our models will be evaluated and assessed using the F1 score metric

4 EXPERIMENTS AND RESULTS

We utilized the datasets provided by Etsy for our experiments. Our objective is to optimize the F1 score for the labels 'top_category_id', 'bottom_category_id', and 'color_id'. Our initial exploratory data analysis (EDA) of the dataset revealed that there are 15 distinct classes for top category, 20 classes for color, and a challenging 2,782

classes for bottom category. The large number of classes for bottom category makes it the most difficult label to predict.

Further analysis of the dataset revealed that one of the 20 different color classes (color_id: 8 or color_text :'Grey') only had one entry. This means that it would be nearly impossible to separate it into training and validation sets without oversampling, but it didn't make sense to oversample for just one entry as it could cause unwanted bias. Therefore, we have dropped that row to ensure that our models do not predict color_id 8. It is essential to have more than one row to accurately predict or classify.

4.1 Random Forest

4.1.1 Data Cleaning and Preprocessing. For the random forest model, we conducted some initial data cleaning and preprocessing, which included normalizing the text data, removing unnecessary characters and symbols that are unlikely to help our model, as well as stemming and tokenization [Mihalcea and Tarau 2004]. In addition to these basic cleaning and preprocessing steps, we also conducted exploratory data analysis to gain further insights into the data and identify any potential issues that may impact the performance of our model. This included identifying and addressing class imbalance, checking for missing data, and analyzing the distribution of our target variables. Overall, our goal was to ensure that our data was as clean and well-prepared as possible before training our random forest model.

4.1.2 Results and Evaluation. We utilized the 'description', 'title', and 'tags' columns from the Etsy dataset as input variables for our model. We ran the model for three different labels: 'top_category_text', 'bottom_category_text', and 'color_text'. The columns' text and ID namesakes have the same mapping, making it easy to convert our predictions to ID if necessary. The pipeline consists of three main components:

- CountVectorizer: This is used to convert the raw text data into a matrix of token counts. It creates a dictionary of all the words in the input text and counts the number of times each word appears in each document [Patel and Meehan 2021].
- TfidfTransformer: This is used to transform the count matrix into a matrix of TF-IDF (term frequency-inverse document frequency) features. TF-IDF is a measure that takes into account the frequency of a word in a document and the frequency of the word in the entire corpus. It helps to normalize the data and reduces the weight of common words [Zhao et al. 2018].
- The RandomForestClassifier is our classification algorithm. It creates multiple decision trees and aggregates their predictions to produce a final prediction. In this context, the algorithm is being used with specific parameters such as verbose=100, which instructs the algorithm to display progress every 100th tree, n_jobs=-1, which indicates that the algorithm will utilize all available CPUs to parallelize computation, and max_depth=100, which sets the maximum depth of decision trees to 100.

We achieved F1 scores of 0.70, 0.45, and 0.40 for the respective labels 'top_category_id', 'bottom_category_id', and 'color_id'.

Table 1. F1 scores for the required labels with Random Forest

Label	F1 Score
top_category_id	0.70
bottom_category_id	0.45
color_id	0.40

Table 2. F1 scores for the required labels with BERT

Label	F1 Score
top_category_id	0.88
bottom_category_id	0.24
color_id	0.60

4.2 BERT with Multimodal-Toolkit

4.2.1 Data Cleaning and Preprocessing. For our multimodal toolkit using BERT, we are using the same dataset as before. Our approach for this model is similar to the previous one, where we have dropped the color_id 8 to make the predictions for this label more accurate. Large Language models like BERT do not require extensive data cleaning and preprocessing [Devlin et al. 2018]. Therefore, in our approach, we have solely tokenized the data and not performed any other data cleaning or preprocessing [Uysal and Gunal 2014]. This decision was made to prevent the loss of any valuable information during the cleaning process [Mohammad 2018].

4.2.2 Results and Evaluation. This model utilizes a multi-modal learning approach, with text data including features such as 'title', 'description', and 'tags', and the 'type' column serving as tabular data. The Multimodal_Toolkit framework includes a data processing module that generates processed text, numerical, and categorical features. These features are then used as input to the Transformer With Tabular module, which consists of a Hugging Face Transformer and a combining module. The combining module in this model is responsible for merging the tokenized text data with the tabular data that passes through to the Transformer With Tabular module. By combining these two types of data, the model can better utilize the information contained within each feature, leading to improved performance and accuracy [Gu and Budhkar 2021]. We achieved F1 scores of 0.88, 0.24, and 0.60 for the respective labels 'top_category_id', 'bottom_category_id', and 'color_id'.

5 CONCLUSION AND FUTURE WORK

We compared the performance of BERT and random forest for multiclass classification using Etsy's dataset in this study. The results reveal that BERT outperformed random forest for two of the three labels, namely 'top_Category_id' and 'color_id,' while random forest outperformed BERT for the third, namely 'bottom_category_id'. While extra training and hyperparameter adjustment may improve BERT's performance, the computational cost may not be worth it.

Future work on this project will be looking into various other transformer-based models that are supported by the multimodal toolbox, as well as fine-tune random forest hyperparameters. Testing

other alternative ensembles, such as XGBoost is also an option. The growing popularity of machine learning has resulted in a large increase in computer energy usage. This issue is growing increasingly prominent as energy consumption has a substantial environmental impact. As a result, when creating novel techniques to machine learning, researchers must take the energy consumption of their models in mind. We need to build more energy-efficient algorithms and models that optimise the usage of processing power as we move forward. Excessive computer energy will very certainly be frowned upon in the future, and we must address this issue if machine learning is to be sustainable and good to society. Ultimately, optimizing the energy consumption of our models will not only help preserve the environment but also make machine learning more accessible and cost-effective for everyone.

REFERENCES

Leo Breiman. 2001. Random forests. *Machine learning* 45 (2001), 5–32.

Archana Chaudhary, Savita Kolhe, and Raj Kamal. 2016. An improved random forest classifier for multi-class classification. *Information Processing in Agriculture* 3, 4 (2016), 215–222.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

Santiago González-Carvajal and Eduardo C Garrido-Merchán. 2020. Comparing BERT against traditional machine learning text classification. *arXiv preprint arXiv:2005.13012* (2020).

Margherita Grandini, Enrico Bagli, and Giorgio Visani. 2020. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756* (2020).

Ken Gu and Akshay Budhkar. 2021. A package for learning on tabular and text data with transformers. In *Proceedings of the Third Workshop on Multimodal Artificial Intelligence*. 69–73.

Kamran Kowsari, Kiana Jafari Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura Barnes, and Donald Brown. 2019. Text classification algorithms: A survey. *Information* 10, 4 (2019), 150.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*. 404–411.

Fahim Mohammad. 2018. Is preprocessing of text really worth your time for online comment classification? *arXiv preprint arXiv:1806.02908* (2018).

Ankitkumar Patel and Kevin Meehan. 2021. Fake News Detection on Reddit Utilising CountVectorizer and Term Frequency-Inverse Document Frequency with Logistic Regression, MultinominalNB and Support Vector Machine. In *2021 32nd Irish Signals and Systems Conference (ISSC)*. IEEE, 1–6.

Jaime Lynn Speiser, Michael E Miller, Janet Tooze, and Edward Ip. 2019. A comparison of random forest variable selection methods for classification prediction modeling. *Expert systems with applications* 134 (2019), 93–101.

Alper Kursat Uysal and Serkan Gunal. 2014. The impact of preprocessing on text classification. *Information processing & management* 50, 1 (2014), 104–112.

Guifen Zhao, Yanjun Liu, Wei Zhang, and Yiou Wang. 2018. TFIDF based feature words extraction and topic modeling for short text. In *Proceedings of the 2018 2nd international conference on management engineering, software engineering and service sciences*. 188–191.