Multi-Label Text Classification of Research Articles: Evaluating the Performance of Pre-Trained and Transformer Models

Anonymous ACL submission

1 Project Objective

This study examines the use of text mining techniques for multi-label classification of research articles, aiming to automate the assignment of relevant categories based on content. We analyze a dataset of research paper abstracts using logistic regression, Naive Bayes, SVM, and RoBERTa. Preprocessing steps such as tokenization, stemming, and stopword removal refine the data, enhancing input quality for model training. RoBERTa, a robust transformer-based model, is additionally employed for its advanced capabilities in handling contextual nuances and complex dependencies within text. Model performance is evaluated using metrics like accuracy, precision, recall, F1 score, Hamming loss, and Jaccard score. By integrating traditional machine learning techniques with advanced neural network architectures, this research contributes significantly to the field of automatic text classification, enhancing the discoverability and categorization of academic publications.

2 Descriptive Statistics

In our examination of the dataset from Analytics Vidhya Hackathon, which consists of 20,972 entries, each record is uniquely identified by its ID, title, and abstract, ensuring no repetition in these primary attributes. The dataset categorizes entries under several academic disciplines. Computer Science, Physics, Mathematics, and Statistics are notably more represented compared to the more niche fields of Quantitative Biology and Quantitative Finance. Notably, the dataset contains no null values, thus eliminating the need for initial data cleaning related to missing entries. This distribution indicates a skew towards traditional scientific and mathematical domains, revealing the dataset's broad academic scope. It serves as a resource for multidisciplinary research while highlighting the need for analytical strategies that can adapt to the complexities and intersections present across different scientific disciplines. This compositional variety is crucial for developing balanced approaches in subsequent analyses and understanding potential biases towards the more heavily represented fields.

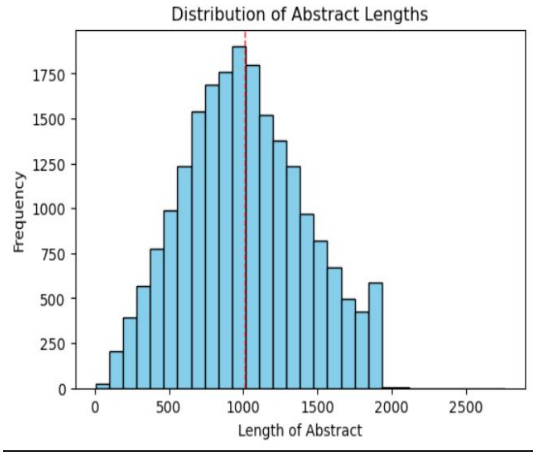

Figure 1: Distribution of Abstract Length

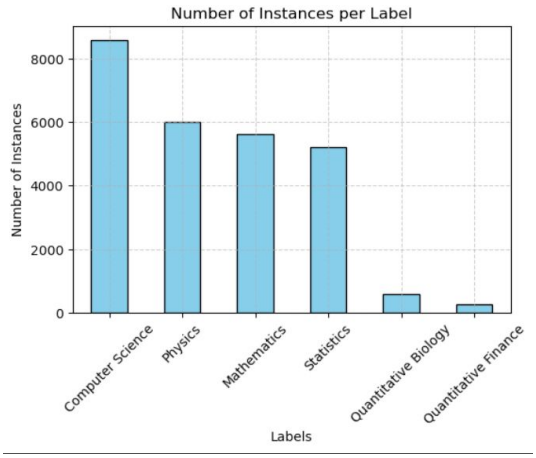

Figure 2: Observing Label Imbalance

ID	TITLE	ABSTRACT	Computer Science	Physics	Mathematics	Statistics	Quantitative Biology	Quantitative Finance
2	Rotation Invariance Neural Network	Rotation invariance and translation invariance have great values in image recognition tasks. In this paper, we bring a new architecture in convolutional neural network (CNN) named cyclic convolutional layer to achieve rotation invariance in 2-D symbol recognition. We can also get the position and orientation of the 2-D symbol by the network to achieve detection purpose for multiple non-overlap target. Last but not least, this architecture can achieve one-shot learning in some cases using those invariance.	1	0	0	0	0	0
5	Comparative study of Discrete Wavelet Transforms and Wavelet Tensor Train decomposition to feature extraction of FTIR data of medicinal plants	Fourier-transform infra-red (FTIR) spectra of samples from 7 plant species were used to explore the influence of preprocessing and feature extraction on efficiency of machine learning algorithms. Wavelet Tensor Train (WTT) and Discrete Wavelet Transforms (DWT) were compared as feature extraction techniques for FTIR data of medicinal plants. Various combinations of signal processing steps showed different behavior when applied to classification and clustering tasks. Best results for WTT and DWT found through grid search were similar, significantly improving quality of clustering as well as classification accuracy for tuned logistic regression in comparison to original spectra. Unlike DWT, WTT has only one parameter to be tuned (rank), making it a more versatile and easier to use as a data processing tool in various signal processing applications.	1	0	0	1	0	0

Figure 3: Dataset Example

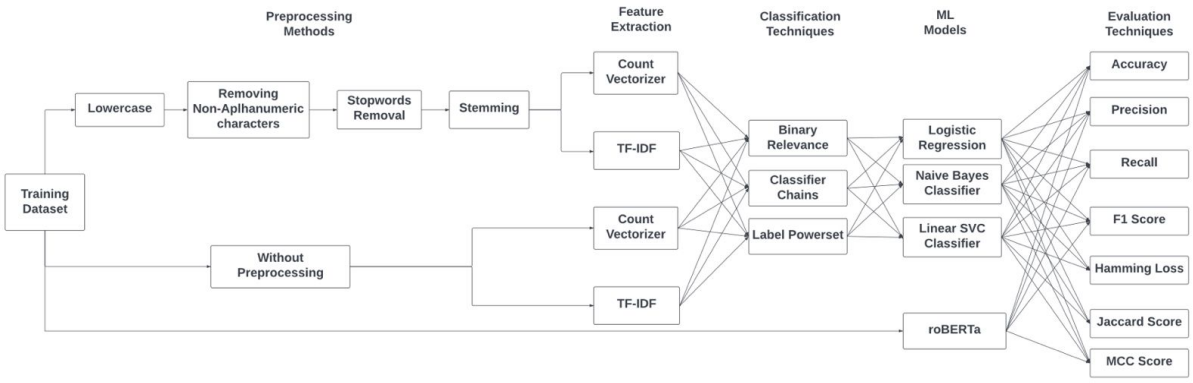

Figure 4: Architecture

3 Approach 1: Integrating Feature Extraction, Multilabel Classification Techniques, and Machine Learning Models

3.1 Pre Processing Steps

In our data preprocessing for the project, we implemented several steps to clean and standardize the text data to ensure refined and consistent input for our models. Initially, we combined the title and abstract columns to enrich the textual data, providing a more comprehensive context for analysis. Following this, we removed stopwords using a predefined list from the NLTK library to eliminate common words that add little semantic value. The text was then standardized by converting it to lowercase and removing non-alphabetic characters. Additionally, we applied the Snowball Stemmer from NLTK, reducing words to their root form. Finally, to adapt the data for our BERT model, we performed word piece tokenization, which effectively handles rare words by breaking them into meaningful sub-units.

3.2 Feature Extraction Methods

In the text processing phase of our project, we utilized both Count Vectorizer and TF-IDF (Term Frequency-Inverse Document Frequency) methods to transform the text data into numerical representations that are suitable for machine learning models. Count Vectorizer converts the text into a matrix of token counts, effectively measuring the presence of each word within the documents, which helps in understanding the frequency distribution of terms across the corpus. On the other hand, TF-IDF goes a step further by not only counting the frequency of words but also weighing these counts by inversely scaling them with the word's occurrence across multiple documents. This technique helps to highlight words that are more interesting, i.e., frequent in specific documents but rare overall, thereby enabling models to capture more nuanced information about the text's content. Employing both methods allows us to leverage the simplicity of raw counts and the advanced insights provided by TF-IDF, making our analysis robust and nuanced.

3.3 Classification Techniques

3.3.1 Binary Relevance

Binary Relevance tackles multi-label classification by breaking it into independent binary tasks, each addressing a single label. Each classifier predicts if a label is present for a given input. This approach simplifies the problem but ignores label dependencies, leading to suboptimal performance.

X	y1	y2	y3	y4
x1	0	1	0	1
x2	1	1	1	0
x3	0	0	1	0
x4	0	1	0	1
x5	0	0	1	0
x6	1	1	1	0

Figure 5: Binary Relevance 1

X	y1
x1	0
x2	1
x3	0
x4	0
x5	0
x6	1

X	y2
x1	1
x2	1
x3	0
x4	1
x5	0
x6	1

X	y3
x1	0
x2	1
x3	1
x4	0
x5	1
x6	1

X	y4
x1	1
x2	0
x3	0
x4	1
x5	0
x6	0

Figure 6: Binary Relevance 2

In this method, the final multi-label output is simply the union of predictions from all classifiers. However, one key limitation is that this approach doesn't take into account the potential relationships between different labels.

3.3.2 Classifier Chains

Classifier Chains builds on Binary Relevance by incorporating label dependencies. Instead of treating each label independently, classifiers are linked in a chain, where each classifier uses the predictions of the previous classifiers as additional inputs. This allows the method to capture relationships between labels.

X	y1
x1	0
x2	1
x3	0
x4	0
x5	0
x6	1

X	y1	y2
x1	0	1
x2	1	1
x3	0	0
x4	0	1
x5	0	0
x6	1	1

X	y1	y2	y3
x1	0	1	0
x2	1	1	1
x3	0	0	1
x4	0	1	0
x5	0	0	1
x6	1	1	1

X	y1	y2	y3	y4
x1	0	1	0	1
x2	1	1	1	0
x3	0	0	1	0
x4	0	1	0	1
x5	0	0	1	0
x6	1	1	1	0

Figure 7: Classifier Chains

Since the predictions of each classifier depend on the previous ones, Classifier Chains can often provide better results than Binary Relevance by leveraging label correlations.

3.3.3 Label Powerset

Label Powerset tackles the problem by transforming a multi-label classification problem into a multi-class classification problem. Each unique combination of labels is treated as a distinct class, and the classifier is trained on these combinations.

X	y1	y2	y3	y4
x1	0	1	0	1
x2	1	1	1	0
x3	0	0	1	0
x4	0	1	0	1
x5	0	0	1	0
x6	1	1	1	0

X	Class
x1	1
x2	2
x3	3
x4	1
x5	3
x6	2

Figure 8: Label Powerset

This approach, however, becomes computationally expensive as the number of unique label combinations grows with more labels. Another limitation is that it only predicts label combinations seen during training, which can be restrictive in real-world scenarios.

For the ML models that we used, please refer to the architecture figure.

3.4 Evaluation Metrics

In addition to common evaluation metrics like accuracy, precision, recall, and F1 score, we have incorporated Hamming Loss and Jaccard Score into the assessment of the multi-label text classification model. These metrics are particularly insightful for this context as they provide a deeper understanding of model performance in scenarios involving complex label structures and interactions.

3.4.1 Hamming Loss

Hamming Loss is the fraction of labels that are incorrectly predicted, i.e., the fraction of the wrong labels to the total number of labels. Lower Hamming Loss is desirable as it indicates fewer mistakes in the label predictions, reflecting higher accuracy and effectiveness of a classifier across multiple labels. Higher Hamming Loss indicates more widespread

errors in the label predictions, suggesting issues with the model's accuracy across its outputs, which can significantly impact the overall utility of a classifier in real-world applications.

3.4.2 Jaccard Score

Jaccard Score, or Jaccard index, measures the percentage of the intersection over union for predicted and true binary label sets. It is a measure of similarity between the two sets. A higher Jaccard Score indicates a greater similarity between the predicted labels and the actual labels, suggesting more accurate model predictions. Conversely, a lower Jaccard Score reflects poorer agreement between the predicted and true labels, pointing to less effective model predictions.

4 Evaluation Results and Discussion of Approach 1

4.1 Before Pre Processing

The comparative analysis of different classifiers using Count Vectorizer (BOW) and TF-IDF feature extraction methods reveals key insights into their performance across various classification techniques. For BOW, the Logistic Classifier with Label Powerset achieves the highest accuracy at 0.669, while Naive Bayes under the same technique scores best in precision at 0.833, and also leads in recall using Binary Relevance at 0.855, highlighting its effectiveness in identifying positive instances. Logistic Regression also records the best F1 score with Classifier Chains at 0.801. In the TF-IDF setup, the Logistic Regression Classifier with Classifier Chains tops in accuracy at 0.672, and shows the highest precision at 0.931 in Binary Relevance. SVM with Label Powerset shows the best recall at 0.777, while achieving the highest F1 score at 0.821 under Classifier Chains. Logistic repeatedly shows lower Hamming losses, indicating fewer average errors, and achieves the highest Jaccard Score with Label Powerset at 0.782, demonstrating superior performance in matching predicted and actual label sets. Generally, TF-IDF outperforms BOW in precision, suggesting its efficiency in highlighting relevant terms for classification. Logistic Regression's consistent performance across metrics, especially in precision and Jaccard Score, establishes it as a reliable option for multi-label classification, with techniques like Classifier Chains and Label Powerset enhancing performance by considering label dependencies, thereby optimizing classification effectiveness across various setups.

Feature Extraction	Classification Technique	Classifier	Accuracy	Precision	Recall	F1 Score	Hamming Loss	Jaccard Score
Count Vectorizer	Binary Relevance	Logistic	0.603	0.8	0.7611	0.778	0.089	0.731
		Naive Bayes	0.61	0.772	0.856	0.802	0.082	0.762
		SVM	0.556	0.76	0.748	0.753	0.102	0.699
	Classifier Chains	Logistic	0.613	0.794	0.762	0.775	0.091	0.738
		Naive Bayes	0.608	0.772	0.855	0.801	0.087	0.761
		SVM	0.569	0.753	0.748	0.75	0.103	0.708
	Label Powerset	Logistic	0.652	0.809	0.768	0.786	0.086	0.764
		Naive Bayes	0.669	0.833	0.773	0.793	0.08	0.775
		SVM	0.628	0.783	0.757	0.769	0.094	0.745

Feature Extraction	Classification Technique	Classifier	Accuracy	Precision	Recall	F1 Score	Hamming Loss	Jaccard Score
TF-IDF	Binary Relevance	Logistic	0.641	0.861	0.743	0.788	0.078	0.743
		Naive Bayes	0.517	0.885	0.544	0.631	0.11	0.594
		SVM	0.646	0.834	0.774	0.8	0.079	0.757
	Classifier Chains	Logistic	0.672	0.931	0.771	0.799	0.08	0.776
		Naive Bayes	0.539	0.883	0.569	0.657	0.105	0.62
		SVM	0.661	0.816	0.781	0.798	0.082	0.772
	Label Powerset	Logistic	0.676	0.845	0.76	0.789	0.079	0.777
		Naive Bayes	0.59	0.833	0.625	0.615	0.113	0.688
		SVM	0.676	0.83	0.777	0.8	0.079	0.782

Figure 9: Before PreProcessing

4.2 After Pre Processing

The updated performance metrics, post-preprocessing, using both Count Vectorizer and TF-IDF feature extraction techniques reveal notable improvements and highlights across various classification techniques and classifiers. In the Count Vectorizer setup, Logistic Regression with Label Powerset excels in precision and Jaccard Score, reaching 0.791 and 0.777 respectively, indicating effective precision and set similarity. Naive Bayes with Binary Relevance achieves excellent recall at 0.861, though its precision at 0.762 suggests a trade-off. Logistic Regression stands out with Classifier Chains, offering the best balance with an F1 score of 0.801. With TF-IDF, Logistic Regression again leads in most metrics across techniques, particularly under Classifier Chains with the highest F1 score at 0.797 and under Label Powerset with both high accuracy and Jaccard Score at 0.674 and 0.778, respectively. SVM also shows strong performance, especially in precision and F1 under Classifier Chains. Despite these advancements, some challenges persist, such as lower recall in some setups and the balance between recall and precision, particularly evident in Naive Bayes' results. The overall enhancements suggest that preprocessing significantly boosts the effectiveness of feature extraction and classification techniques, optimizing the model's performance across different metrics and setups.

Feature Extraction	Classification Technique	Classifier	Accuracy	Precision	Recall	F1 Score	Hamming Loss	Jaccard Score
Count Vectorizer	Binary Relevance	Logistic	0.641	0.862	0.87	0.788	0.078	0.743
		Naive Bayes	0.601	0.763	0.862	0.804	0.088	0.76
		SVM	0.543	0.743	0.74	0.741	0.108	0.687
	Classifier Chains	Logistic	0.672	0.831	0.79	0.799	0.08	0.776
		Naive Bayes	0.6	0.762	0.861	0.802	0.08	0.759
		SVM	0.554	0.744	0.741	0.742	0.107	0.697
	Label Powerset	Logistic	0.635	0.791	0.759	0.774	0.092	0.75
		Naive Bayes	0.671	0.828	0.782	0.8	0.08	0.777
		SVM	0.603	0.757	0.744	0.75	0.103	0.724

Feature Extraction	Classification Technique	Classifier	Accuracy	Precision	Recall	F1 Score	Hamming Loss	Jaccard Score
TF-IDF	Binary Relevance	Logistic	0.64	0.858	0.853	0.789	0.079	0.744
		Naive Bayes	0.55	0.877	0.587	0.675	0.103	0.631
		SVM	0.64	0.823	0.773	0.795	0.082	0.753
	Classifier Chains	Logistic	0.67	0.828	0.828	0.797	0.081	0.774
		Naive Bayes	0.573	0.868	0.619	0.702	0.097	0.66
		SVM	0.66	0.815	0.782	0.798	0.0831	0.771
	Label Powerset	Logistic	0.674	0.853	0.853	0.795	0.0796	0.778
		Naive Bayes	0.587	0.806	0.626	0.619	0.114	0.686
		SVM	0.668	0.821	0.776	0.797	0.082	0.776

Figure 10: After PreProcessing

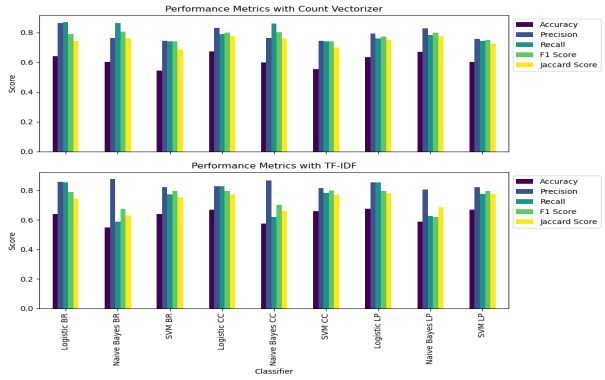

Figure 11: Performance Metrics

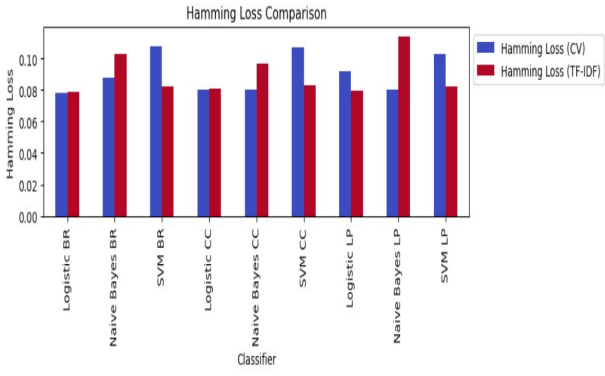

Figure 12: Hamming Loss Comparison

5 Approach 2: Transformer Model

RoBERTa (Robustly Optimized BERT Approach) is an enhanced version of the well-known BERT model, developed by researchers at Facebook AI. As a transformer-based language model, RoBERTa leverages self-attention mechanisms to analyze input sequences and create contextualized word representations within sentences. Unlike BERT, RoBERTa was trained on a substantially larger dataset—160GB of text, over ten times the size of BERT's training dataset. This extensive training, combined with dynamic masking—a method that alters the mask for each training instance—allows RoBERTa to develop more robust and generalizable word representations, significantly advancing its effectiveness across various NLP tasks.

In refining BERT's architecture, the developers of RoBERTa made several critical adjustments to optimize performance. They eliminated the Next Sentence Prediction (NSP) task, which initially aimed to determine if two text segments were from the same document. This modification proved to enhance or maintain performance on downstream applications. Furthermore, RoBERTa was trained using larger batch sizes and longer sequences than BERT, which not only improved the model's perplexity on language modeling tasks but also its accuracy on applied tasks. The dynamic approach to changing the masking pattern during training, as opposed to the static mask used in BERT, also contributed to its enhanced learning capability, making RoBERTa a more powerful tool in the field of NLP.

5.1 Workflow and Hyper Parameters used in BERT

The machine learning model training process starts with importing the dataset, which is then split into training, validation, and testing segments with respective proportions of 70, 10, and 20 percent. The setup involves tuning various hyperparameters such as maximum sequence length, batch sizes for training, testing, and validation, and the number of epochs. Key elements such as the tokenizer and data loader prepare the data for processing, and the BERT model class is utilized for the core computational architecture. Important training parameters like dropout rate, learning rate, weight decay, and class weights are meticulously adjusted to optimize performance. The training routine is powered by a selected loss function and optimizer, with specific train, validation, and test functions to

streamline the evaluation phases. Finally, the process concludes with the visualization of results to assess the effectiveness and efficiency of the model in handling the task at hand.

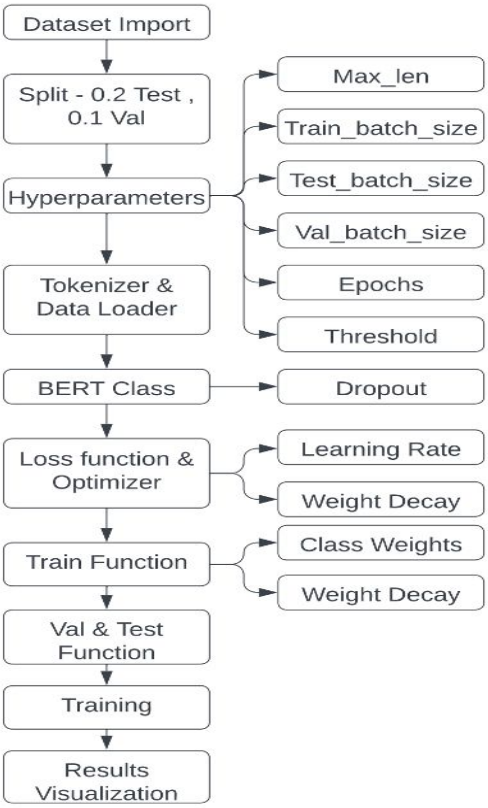

Figure 13: RoBERTa Workflow

5.2 Evaluation Metric - MCC Score

In addition to common evaluation metrics like accuracy, precision, recall, and F1 score, we have incorporated MCC Score into the assessment of the multi-label text classification model.

The Matthews Correlation Coefficient (MCC) is a robust statistical measure used to evaluate the quality of binary and multi-class classifications in machine learning. It considers true and false positives and negatives, making it a balanced measure suitable for datasets with varying class sizes. In multi-label classification, MCC is crucial as it provides a reliable, comprehensive score of model performance across all labels, unlike accuracy, which can be misleading in imbalanced datasets. This makes MCC invaluable for ensuring that predictions are accurate and meaningful across diverse categories.

6 Evaluation Results and Discussion of Approach 2

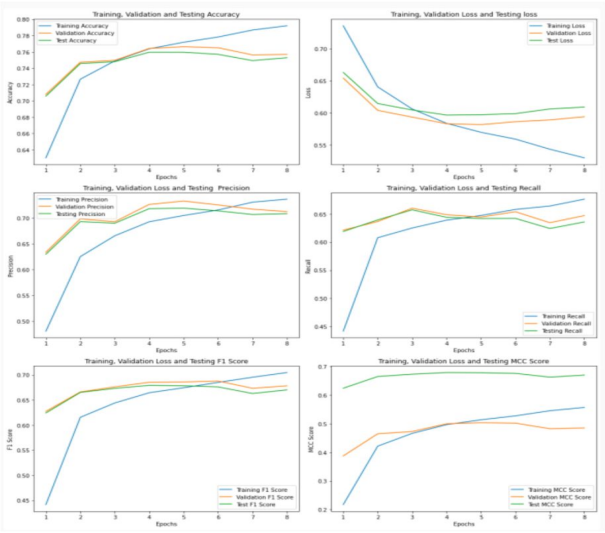

Figure 14: Results of Baseline Bert Model

6.1 Improvement Strategies for Overfitting and Label Imbalance

In the pursuit of optimizing model performance and ensuring robustness in machine learning, several advanced techniques are employed to counter issues like label imbalance and overfitting. The use of dropout serves as a regularization method to prevent overfitting by randomly omitting a fraction of input units during training, thus enabling the model to generalize better on unseen data. Additionally, to address class imbalance, a function automatically adjusts weights inversely proportional to class frequencies, thereby giving greater emphasis to smaller classes and ensuring a fair influence on model learning.

Further refining the training process, gradient clipping is implemented to prevent the explosion of gradient values during backpropagation, maintaining stable training dynamics and safeguarding against numerical instability. Weight decay is also set to decouple regularization from the learning rate, applying consistent weight penalties across all parameters. This technique not only helps in preventing overfitting but also contributes to a more disciplined optimization process, allowing for smoother convergence and improved model accuracy across diverse datasets.

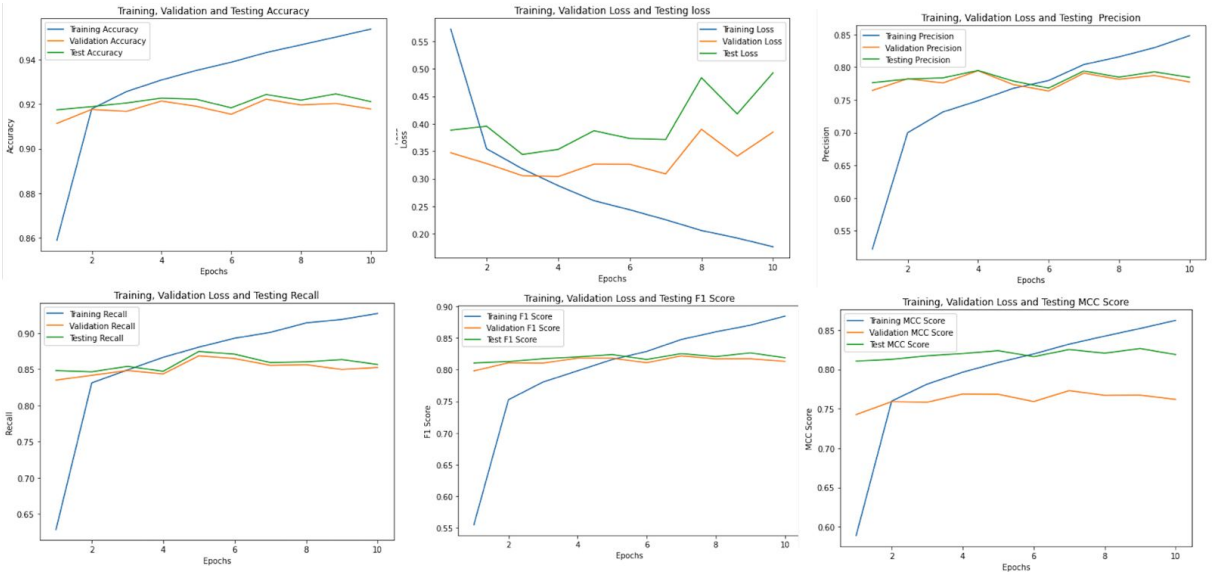

Figure 15: Bert results after applying strategies

7 Error Analysis, Insights, and Interpretation

7.1 Machine Learning Models

In evaluating the performance of Logistic Regression, Naive Bayes, and SVM classifiers in our multi-label text classification framework, several sources of error have been identified. Logistic Regression showed significant improvements in accuracy, precision, and F1 score with preprocessing, suggesting it effectively handles cleaner, noise-reduced data. However, the model may still risk overfitting, particularly in complex or imbalanced datasets where label interdependencies are prevalent. Naive Bayes, while exhibiting high recall, indicated potential overfitting to the positive class and struggled with precision, particularly when preprocessing and TF-IDF were applied. This might be attributed to the model's foundational assumption of feature independence, which often does not hold in processed text data. SVM displayed variable performance; while it excelled in some configurations, possibly benefiting from specific kernel settings, it showed inconsistencies, particularly in configurations requiring robust generalization.

7.2 Transformer Models - RoBERTa (Before applying improvement strategies)

Before implementing dropout, class weights, gradient clipping, and weight decay, the model displayed potential overfitting as indicated by higher training accuracy compared to validation and test accuracies, along with more significant variability in validation and test losses. This overfitting suggested that the model was too closely attuned to the training data's nuances, failing to generalize effectively to new, unseen datasets. Precision and recall metrics showed gradual improvement, but the test precision lagged behind training, suggesting issues with generalization likely due to the model capturing noise rather than the underlying patterns. The F1 scores and MCC values also reflected a disparity between training and other datasets, with F1 scores plateauing and MCC values remaining relatively flat for validation and test data. This highlighted difficulties in balancing precision and recall, potentially caused by label imbalance where the model was biased towards more frequently occurring labels, and managing complex label interdependencies, indicating challenges in accurately capturing the relationships between different labels in the multi-label setting.

7.3 Transformer Models - RoBERTa (After applying improvement strategies)

Implementing dropout likely mitigated overfitting by randomly deactivating neurons, thereby forcing the model to learn more generalizable features, which would enhance accuracy and stabilize F1 scores across all data sets. Class weights helped correct imbalances, improving the model's recall and F1 scores by enhancing sensitivity to minority classes. Gradient clipping controlled potentially unstable updates, contributing to more consistent

loss reductions and improved precision and MCC in validation and test sets. Lastly, weight decay promoted simpler models that generalize better, reducing overfitting and enhancing MCC scores, aligning training and testing performance more closely. These enhancements likely led to a more robust model with improved generalization capabilities across unseen datasets.

Despite implementing improvement strategies such as dropout, class weights, gradient clipping, and weight decay, the BERT model still exhibits some challenges that could be attributed to residual issues of model training and generalization. Even after these adjustments, there remains a noticeable variability in validation and test loss and precision, particularly seen as fluctuations in later epochs. This continued variability suggests that while the model has become more robust against overfitting, it may still be sensitive to certain features or noise within the training data that do not generalize well across unseen datasets. Moreover, the mild divergence between training and validation/test accuracy persists, hinting at possible subtle overfitting or the model's conservative nature in class predictions. These outcomes underscore the potential need for further refinements in model architecture or training approach, such as exploring more sophisticated regularization techniques, experimenting with different model architectures that might capture the data complexities better, or further tuning the balance between model complexity and training data characteristics to enhance overall model performance and reliability across diverse data conditions.

8 Future Scope

In addressing the challenge of label imbalance in our datasets, we've devised a novel strategy that leverages the capabilities of advanced language models like Llama or GPT-4. Our approach involves using these models to systematically rephrase text data within columns that exhibit label imbalance. By generating linguistically diverse versions of the original text while retaining the same labels, we aim to enrich our dataset with a broader spectrum of linguistic expressions associated with underrepresented labels. This augmentation method not only enhances the dataset's diversity but also aids in reducing label imbalance, thereby potentially improving the accuracy and generalization ability of our machine learning models. This technique promises to refine our model's training process by providing a more balanced and comprehensive dataset, ensuring that all categories are adequately represented and learned.

References

[1] N. K. Mishra and P. K. Singh, "Feature construction and SMOTE-based imbalance handling for multi-label learning," *Journal of Computational Science*, vol. 54, p. 101468, 2021. doi: 10.1016/j.jocs.2021.101468.

[2] B. R. Bhamare and J. Prabhu, "A multilabel classifier for text classification and enhanced BERT system," *ResearchGate*, 2021. https://www.researchgate.net/publication/351880052_A_Multilabel_Classifier_for_Text_Classification_and_Enhanced_BERT_System.

[3] J. Nam, J. Kim, E. Loza Menc, I. Gurevych, and J. Furnkranz, "Large-scale multi-label text classification — Revisiting neural networks," *ResearchGate*, 2013. https://www.researchgate.net/publication/259367487_Large-Scale_Multi-label_Text_Classification_-_Revisiting_Neural_Networks.

[4] A. Y. Taha, S. Tiun, A. H. Abd Rahman, and A. Sabah, "Multilabel over-sampling and under-sampling with class alignment for imbalanced multilabel text classification," *ResearchGate*, 2021. https://www.researchgate.net/publication/352563832_Multilabel_Over-sampling_and_Under-sampling_with_Class_Alignment_for_Imbalanced_Multilabel_Text_Classification.

[5] S. M. Liu and J. H. Chen, "A multi-label classification based approach for sentiment classification," *Expert Systems with Applications*, vol. 41, no. 5, pp. 2526-2535, 2015.