Institut Polytechnique des Sciences Avancées

Mathematical Foundations for Statistical Learning Project:

Multi-Label Classification of Scientific Literature Using the NASA SciX Corpus

---

BLAISE Lucas

Année universitaire 2024-2025

Introduction

The classification of scientific texts is essential for efficient navigation, search, and categorization of the vast corpus of academic literature. This project addresses the task of **multi-label classification**, where each scientific document can be associated with multiple relevant keywords (labels). Unlike traditional single-label classification, this approach reflects the multifaceted nature of scientific writing, where a paper may be about multiple themes or research areas simultaneously.

We use the *NASA SciX* corpus, provided by the Hugging Face dataset hub, which contains thousands of documents with associated title, abstract, and keyword labels. The objective is to build a machine learning model that learns to predict the correct set of labels (keywords) based solely on a document's title and abstract.

1 Dataset Description

The dataset `adsabs/SciX_UAT_keywords` includes fields such as the title of the publication, the abstract, and a set of human-verified labels called `verified_uat_labels`. These labels are sparse but of high quality, making them ideal for evaluation. The dataset is rich in domain-specific language, making it well-suited for training a contextual language model.

However, a significant challenge comes from the dataset's sparsity. Many documents lack labels, and the distribution of labels is extremely unbalanced. Some keywords appear frequently, while many appear only once or twice. This imbalance can cause the model to be biased toward more common classes and ignore rare but meaningful ones.

2 Preprocessing

The preprocessing part consists of several steps. First, we filter the dataset to retain only those samples that include at least one verified label. This ensures that every training example has a valid target for learning. Next, we concatenate the title and abstract into a single string for each document, forming the model's input.

We then use `MultiLabelBinarizer` to transform the list of labels into a binary vector format. Each entry in the vector corresponds to a keyword, with 1 indicating presence and 0 indicating absence. Finally, the text is tokenized using the BERT tokenizer, truncated to a maximum length of 128 tokens to manage memory usage, and attention masks are generated to distinguish padded tokens.

These steps allow the data to be fed into a BERT-based model while preserving the structure required for multi-label learning.

3 Model

We use the `bert-base-uncased` model from the Hugging Face Transformers library. This model is pre-trained on large general-domain corpora such as Wikipedia and BookCorpus. On top of BERT, we add a classification head—a single linear layer followed by a sigmoid activation function.

The sigmoid function allows the model to output a probability for each class independently, which is essential for multi-label tasks where multiple labels can be active simultaneously. We use Binary Cross Entropy (BCE) loss as the objective function, which calculates the error independently for each label.

The AdamW optimizer is used to update the model weights, as it incorporates both momentum and weight decay, which helps prevent overfitting. The model was trained for 1 epoch with a batch size of 8 and a learning rate of 1×10^{-5}.

4 Training Details

Training was performed using PyTorch. Each training step involves passing a batch of examples through the model in training mode. The model's predictions are compared to the true labels using the BCE loss. The gradients of this loss with respect to the model's parameters are then computed using backpropagation, and the parameters are updated via the optimizer.

The training was limited to one epoch to minimize overfitting and reduce runtime. Additionally, the maximum input length was restricted to 128 tokens. While this reduces context, it enables faster training and avoids memory issues on limited hardware.

5 Evaluation

Evaluation metrics include macro F1-score, sample-based accuracy, and a complete classification report. The sigmoid outputs were thresholded at 0.2 instead of the default 0.5. This decision was based on the observation that the model's confidence scores were typically low due to the sparsity of labels. By lowering the threshold, we encourage the model to make more predictions and reduce false negatives.

Despite these adjustments, the initial results yielded very low scores across all metrics. This outcome suggests that the model failed to generalize, likely due to class imbalance, limited training time, and label sparsity.

6 Results and Observations

The classification report revealed that many classes were never predicted correctly. Even the micro and macro F1-scores were close to zero. One contributing factor was the high number of rare labels, which makes it difficult for the model to learn meaningful patterns in a single epoch.

To address these issues, one could consider filtering out extremely rare labels, implementing oversampling for underrepresented classes, or using weighted loss functions. Another possibility is to switch to a smaller, faster model such as DistilBERT to allow more epochs and experimentation.

Conclusion

This project shows both the promise and limitations of transformer-based models for scientific multi-label classification. The BERT architecture provides a strong foundation due to its deep contextual understanding of language. Using it with a sigmoid-activated classification head and binary cross-entropy loss is an effective design choice for multi-label problems, as it treats each class independently and handles overlapping labels naturally.

Several parameter choices were made to balance training time with feasibility. A batch size of 8 allows reasonable GPU memory usage while maintaining enough examples per update to stabilize training. Limiting the maximum sequence length to 128 tokens reduces computational load and training time, though it can omit context from longer abstracts. Training for only one epoch was chosen to keep the project runtime short; however, this significantly limits the model's ability to learn meaningful patterns, especially in a high-dimensional, imbalanced label space.

A lower threshold of 0.2 was applied to the sigmoid outputs during prediction to compensate for the model's low confidence due to sparse and infrequent labels. This decision increases recall but can reduce precision, often resulting in a trade-off between the two. Although this approach encouraged more predictions, the classification report revealed that the model still failed to generalize effectively. Most metrics remained close to zero, reflecting the model's struggle with extreme class imbalance and insufficient training time.

In summary, while the overall pipeline is functional and demonstrates the process of multi-label classification using transformers, its effectiveness is currently limited by short training duration, label sparsity, and imbalanced class distributions. Future work should explore longer training schedules, label-aware loss functions, threshold optimization, and possibly a more aggressive filtering of rare labels to improve both performance and robustness.