News Articles Topic Hierarchical Classification -SI 630 Final Project Report

Xiaoyang Sheng
shengxy@umich.edu

Abstract

The hierarchical classification has been an important topic in Natural Language Processing (NLP), and it is challenging due to its hierarchical and sparse labels. In this project, we try to come up with a model to classify the news articles topic with double hierarchical labels. We use the pre-trained BERT models, together with other methods including additional weights of titles, reweighing of loss function and different thresholds for the hierarchical labels. Finally, our method reaches 95.6% of Top 3 Accuracy for level 1, 0.867 F1 Score for level 1 and 0.793 F1 Score for 2 levels, which are way more better than the baseline results.

1 Introductions

This project aims to develop a classifier for hierarchical news article data, encompassing numerous classes and subclasses. For instance, given the title and content of a news article, the classifier seeks to identify the primary class, such as weather, and its corresponding subclass, such as weather forecast. Our method is designed to achieve high performance in both single-level and two-level predictions. In other words, the classifier should accurately determine the class at the first level and subsequently provide correct results for the second level, leveraging its strong performance at the initial classification stage.

We have read through some literature regarding this topic and found that many people had come up methods regarding either the long text classification like news articles or the hierarchical classification but on relatively short text, like customer service complaints. They proposed some special embeddings or presentation of tokens, as well as some additional layer structure to the original model, to improve performance on either of these two main fields. However, few of them combine them together and our work is to solve the hierarchical classification on the long text like news articles in a handy and efficient way.

After experimenting with various techniques, we settled on employing a pre-trained BERT model, augmented with additional weights on titles, a refined loss function considering hierarchical structures and label sparsity, and varying thresholds for scores of two labels. This refined approach achieved notable results: a Top 3 Accuracy of 95.6% for level 1, an F1 Score of 0.867 for level 1, and an F1 Score of 0.793 for 2 levels. These outcomes represent significant improvements over the baseline results.

Solving this problem is crucial in NLP, where text classification is vital across diverse domains, from information retrieval to email filtering and data organization in legal, healthcare, and news sectors. Real-world scenarios often involve complex hierarchical structures, demanding nuanced strategies beyond simple multi-label classification. Successful implementation of hierarchical classification, particularly in news media, can revolutionize content management, empowering platforms and agencies to operate more efficiently. Moreover, its relevance extends to text-intensive domains like academic search. In essence, this pursuit holds broad practical significance, warranting continued exploration and experimentation.

Our project shed light on these challenges through several key augmentations to the modeling process, such as text tokenization, refining the loss function, and adjusting thresholds. These enhancements specifically target the complexities of hierarchical and sparse multilabel classification. Further investigation and refinement in this area hold promise for future improvements.

2 Data

The data for this project is an open-source dataset on Zenodo, titled with "MN-DS: A Multilabeled

News Dataset for News Articles Hierarchical Classification"(Petukhova and Fachada, 2022). It contains the data column of id, data id, date, source, title, content, author, url, published date, published utc time, collection utc time, category level 1 and category level 2. See Table 1 for one of the records in the dataset.

The dataset has some collection problem that when we first did a quick view of the data, we found that some of the records have exact identical text for feature "title" and "content", and many records miss the value for feature "author". Therefore, we mainly used features "title" and "content" for the text input.

We try to get some rough statistics about our data. The data set has **10917** instances overall and **17** level-1 labels and **109** level-2 labels, the plot distribution of the hierarchical 2-level labels is shown in Figure 1, note that since there are a lot of level-2 labels, only top 20 level-2 labels are shown on the plot.

3 Related Work

There are many research papers working on the text classification, and we found some related papers regarding news classification and hierarchical classification respectively. Zhao, Lin, et al once proposed a improved word embedding methods on the news text classification, where an improved TF-IDF algorithm is proposed to weight word vectors (Weidong Zhao and Zhang, 2022). It introduces LDA topic generation model to enhance the topic semantic information of TF-IDF values, and with the algorithm the CNN could reach a higher accuracy on their dataset. We ended give up this method after we found that the performance of CNN is much worse than the BERT on our long text materials.

Li, et al. also proposed a model for text classification which is called Bi-LSTM-CNN, composed of a BiLSTM layer with a word vector and a left and right context, a local feature, a global feature and a softmax layer (Li et al., 2018). It utilizes the loop structure to obtain the context information, which is more accurately expressed the semantics of the text and greatly improve the performance. This is relative early paper and their proposed layer of left-right context is very similar to the transformer structure and therefore inspired us to try the BERT pretrained model.

Regarding hierarchical classification, Song, et al. proposed a Matrix Factorization (MF) and Recursive-Attention (RA) Approach, which is called MF-RA, to handle Hierarchical Multi-label Text Classification (HMTC) tasks (Song et al., 2022). However, they used the data from customer complaint texts, which are typically short and oral compared with news articles. We found that MF-RA approach trains really slow on our long text due to complexity, and the memory limit of the Great Lakes did not support the training.

Huang, et al. also proposed a framework called Hierarchical Attention-based Recurrent Neural Network (HARNN) for classifying such documents (Huang et al., 2019). They applied a documentation representing layer for obtaining the representation of texts and the hierarchical structure, and developed an hierarchical attention-based recurrent layer to model the dependencies among different levels of the hierarchical structure. Moreover, a hierarchical attention strategy and a hybrid method are proposed, which are capable of predicting the categories of each level while classifying all categories in the entire hierarchical structure precisely. They did some research on the layer and attention strategy, which could be a good reference for us to try out in our case. This paper added additional layer and attention strategy, which inspires us to add additional weights to the text input and change loss function to make full use of the BERT model.

Gargiulo et al. presented a methodology named Hierarchical Label Set Expansion (HLSE), used to regularize the data labels, and an analysis of the impact of different Word Embedding (WE) models that explicitly incorporate grammatical and syntactic features (Gargiulo et al., 2019). It is more like a preprocessing methodology regarding the classification modeling and we could try their work regarding the data label regularization before diving into the modeling part. The method proposed by the paper did not work quite where on our data, but its idea of label regularization inspired us to consider differentiating the thresholds for two levels of labels, therefore it could better distinguish the hierarchical structures.

4 Methods

We proposed a series of method to solve the long text hierarchical classification including featuring, reweighing and so on. The overall work flow is shown in Figure 2.

feature	data
id	7038
data id	ageofautism–2019-04-12–Physician Father and Caretaker of 29 Year Old Autistic Man Found Brutally Murdered
date	2019-04-12
source	ageofautism
title	Physician, Father and Caretaker of 29 Year Old Autistic Man Found Brutally Murdered
content	"One family member said Derek "can be violent and has attacked Rex in the past," court documents stated. The family member also said Derek's mother had died years ago and Rex was...
author	Age of Autism
url	http://feedproxy.google.com/ r/ageofautism/ 3/zc44EG7xWFM/physician-father-and-caretaker-of-29-year-old-autistic-man-found-brutally-murdered.html
published	2019-04-12 09:00:00+00:00
published utc	1555074000
collection utc	1567543083
category level 1	crime, law and justice
category level 2	crime

Table 1: One of the record in the news article dataset.

4.1 Features and Cleaning

After conducting some exploratory data analysis (EDA), we decided to focus solely on the title and content features for our analysis. Features such as date and published time were deemed inadequate for characterizing news article categories due to their uniformity across all categories. Additionally, we cannot use url feature as well, and it will be a trick using it since sometimes it contains the string of category inside the link, as we can see from the example in Table 1. While source and author could potentially provide insights, the abundance of null values led us to ultimately discard them from consideration. Thus, our analysis solely relies on the title and content features.

Furthermore, we use one-hot encoding for the two-level of labels, with the format of level1_xxxx and level2_xxxx with value 0 or 1.

4.2 Tokenization

Given the relatively lengthy nature of our text, we conducted exploratory data analysis (EDA) on its overall length. With a median length of 466, it's reasonable to set the max_length of the pretrained BERT tokenizer as 512. This choice is reasonable since most news articles convey their crucial information at the very beginning. Thus, our tokenizer configuration includes a max_length of 512, with truncation enabled, ensuring uniform token length within each batch.

4.3 Additional Weights on Title

Drawing from previous research on additional layers and specialized embeddings, we opted to integrate both the title and content in our model, with a focus on highlighting the significance of the title text. This decision stems from the common practice in media to convey key information upfront, potentially indicating the article's category. While initially considering handling multiple inputs with varying weights in the input layer, we encountered limitations with pretrained models from Hugging Face lacking such direct features. To address this, we devised an alternative approach: repeating the title a fixed number of times (referred to as title_weights in our code) before the content, thus combining them for tokenization. To ensure coherence, we carefully managed spacing between the repeated titles and content to prevent bad tokens from direct concatenation. Experimenting with different title_weights, we settled on a value of 2, balancing the need to emphasize the title while minimizing potential information loss from content truncation.

4.4 Reweighing of Loss Function

A common approach to hierarchical classification involves customizing loss functions to integrate hierarchical constraints, penalizing predictions that deviate from these constraints. Additionally, following one-hot encoding, the resulting label matrix

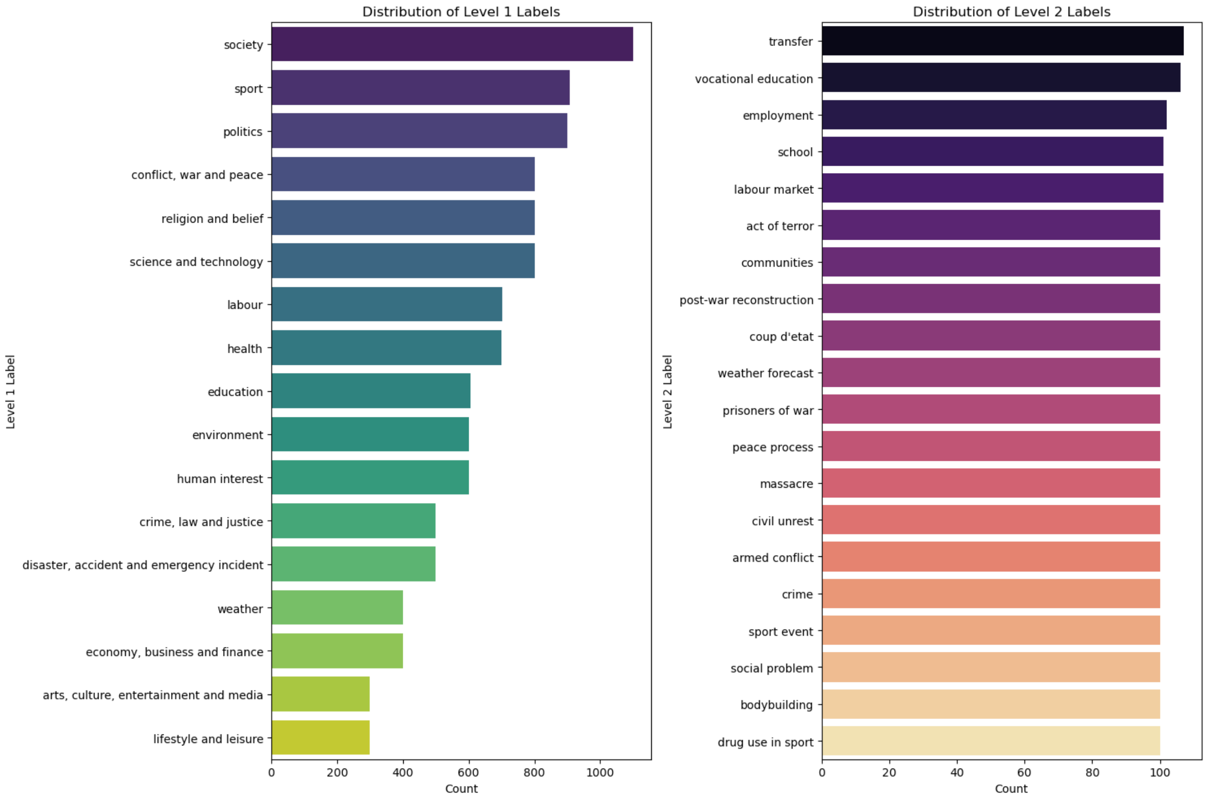

Figure 1: the plot of distribution of two-level labels (only top 20 out of 109 for level 2)

tends to be highly sparse, as each record may only have two positives (one for level 1 and one for level 2) across a total of 126 labels. Consequently, it's imperative to account for both the hierarchical structure and sparsity of the data when designing the loss function.

Therefore, we devised our loss function using BCEWithLogitsLoss() from Hugging Face along with a mixed weight vector. This vector comprises two components: the first part consists of weights for different levels, with level-1 labels initialized as 1.0 and level-2 labels as 0.5. This emphasis on level-1 classification aims to improve overall performance, as correct predictions at this level are crucial for accurate level-2 predictions. The second part incorporates sparsity weights, calculated for each label as $2 \times \frac{\text{negative samples}}{\text{positive samples}}$. Combining these components through element-wise multiplication yields the mixed weight vector.

4.5 Different Thresholds for Two Levels

Another common approach involves setting varying thresholds for each label, which can be beneficial when classes exhibit different frequencies or importance levels. Despite implementing different weights for the loss function, setting diverse thresholds for the final prediction based on the model's logits output can further optimize performance. We conducted experiments with different threshold values on both the validation and training sets to achieve the best 2-level F1 score. Ultimately, we settled on a threshold of 0.5 for level 1 and 0.4 for level 2. This means that if the score of a level-2 label is greater than or equal to 0.4, it is considered positive for that label, and the same holds for level 1.

5 Evaluation and Results

5.1 Evaluation Metrics

Like other classification problem, we will use the common metrics including Precision, recall and F1-score. But since our problem is hierarchical and multi-labeled, we will also use the top k accuracy for the level-1 prediction results. The final evaluation would be in the following way: for the level-1 label, we will use normal precision, recall and f1-score for its own prediction, together with a top 3 accuracy (we made a mistake for the project update, it should be accuracy not precision) for level 1 which is calculated by looking at the proportion of times the correct level-1 label is in the top 3 predictions. For the level-2 label will use the nor-

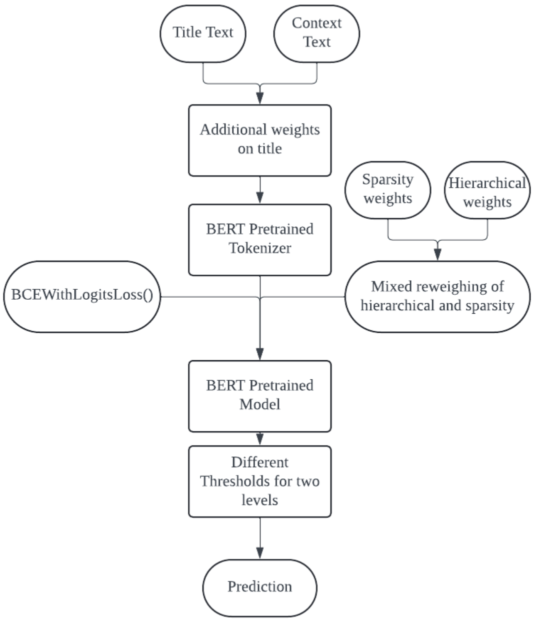
Title Text
Context Text
Additional weights on title
Sparsity weights
Hierarchical weights
BERT Pretrained Tokenizer
BCEWithLogitsLoss()
Mixed reweighing of hierarchical and sparsity
BERT Pretrained Model
Different Thresholds for two levels
Prediction

Figure 2: The overall structure and flow of the method.

mal precision, recall and f1-score, but only when both level 1 and level 2 are predicted correctly, it is then considered as true positive. To clarify, for the situation when there is no prediction for certain level1-level2 labels, we discard that results. Therefore, the level 2 metrics could possibly significantly lower than level 1 since it is more difficult to predict both labels correctly.

We tried two baseline, one is just random guessing, where each label has the same chance to be picked, both for the level-1 and level-2. The second one is guessed by frequency of labels in each level, i.e. the random guess by the probability calculated by the frequency of each label. Meanwhile, since we are dealing with multi-class problem and the labels are a little bit imbalanced, all the metrics above are used with weighted average to get the final single metric of all the classes.

The metrics for our results and two baselines are shown in Table 2.

5.2 Ablation Study

Though we tried several hyper parameters, we only recorded some training outcomes for different learning rates and original loss function. Meanwhile, it is interesting to find that for certain large learning rates, the training will never converge and the evaluation f1 score is always close to zero under original loss function. However, with our new reweighed loss function, the situation will be much better. Table 3 is metric of level-2 F1 score under some settings we recorded.

	level-1 precision	level-1 recall	level-1 F1	level-1 top-3 accuracy
Baseline random	0.061	0.056	0.057	0.058
Baseline frequency	0.068	0.068	0.068	0.0675
Our method	**0.863**	**0.793**	**0.867**	**0.956**
	level-2 precision	level-2 recall	level-2 F1	
Baseline random	0.011	0.0007	0.0014	
Baseline frequency	0.010	0.0006	0.0009	
Our method	**0.7469**	**0.863**	**0.793**	

Table 2: Comparison of evaluation metrics between baseline and our method

learning rate	loss function	addition weight on title	Evaluation set level-2 F1 score
0.01	original	No	0
0.001	original	No	0
0.0001	original	No	0.6029
0.0001	mixed weight	No	0.6913
0.0001	mixed weight	Yes	0.8039

Table 3: Metrics under different settings trained for 30 epochs

Meanwhile, for smaller learning rate, the converge speed is too slow and will exceed the time limit on Great Lakes server. Therefore, finally we choose the setting of 0.0001 learning rate, additional double weights on title, reweighed loss function and 30 epochs of training.

6 Discussion

Our method achieves good performance metrics: a Top 3 Accuracy of 95.6% for level 1, an F1 Score of 0.867 for level 1, and an F1 Score of 0.793 for 2 levels. Comparing these results with two baseline guessing scenarios, our method significantly improves the F1 score for two levels from less than 0.1 to approximately 0.8, alongside a robust 95.6% top 3 accuracy for level 1 labels. Moreover, given the sparsity and large number of labels across both levels, these outcomes underscore the potential of our approach in addressing hierarchical classification in lengthy text data. This suggests practical utility for end-users seeking to leverage our model for similar NLP tasks.

The achieved performance closely aligns with our expectations. Baseline guessing would be inefficient due to the sparse labeling and numerous classes and subclasses. If the baseline incorrectly predicts level 1, it will never achieve accuracy at level 2. Our model addresses these challenges by incorporating a reweighed loss function, customized thresholds, and additional weights on title information to account for sparsity and hierarchical structure. This approach encourages balanced label learning and adherence to hierarchical constraints. Coupled with the strong performance of the pre-trained BERT model across various NLP tasks, these enhancements significantly improve the effectiveness of our model.

7 Conclusion

In summary, we introduced a method leveraging BERT, supplemented with extra weights on title information, a weighted loss function, and customized thresholds. This approach accounts for the hierarchical structure, effectively addressing the task of hierarchical classification in news article topics. We have uploaded our source code to GitHub and you may also check them by the link: GitHub Source Code Link

8 Other Things We Tried

Besides other hyper parameters settings we tried in the Ablation Study section, we also tried other structures or models mentioned before.

We tried an TF-IDF improved embeddings on CNN, but its evaluation set level-2 F1 score is only around 0.21, which is way worse than BERT we are using, even if without any augmentations. Therefore, normal CNN is not enough for our task scenario, even with improved word embeddings.

Then we tried the Matrix Factorization (MF) and Recursive-Attention (RA) approach, which is a traditional machine learning model typically applied in Bayesian modeling. The training speed is too slow, and the requirement of memory is huge, since it often needs to construct a large matrix for learning. Our news articles are long, and it is almost impossible for us to train them on the current limited 32GB memory source on Great Lakes servers.

9 Future Improvement

During training, we observed that there may be further room for the training loss to decrease, and certain evaluation metrics continued to improve even after 30 epochs of training. However, constrained by the time limit of Great Lakes, we could only train for 30 epochs. If conditions permit, we would explore longer training times in future iterations.

Regarding addressing sparsity and hierarchical structure, our approach primarily focuses on aspects such as text weighting, loss function weighting, and thresholds. However, alternative methods such as modifying the overall model architecture using a local-global mixed classifier or implementing different training strategies like top-down or bottom-up for the tree structures of the labels could also be explored. Due to time constraints and the need to modify the source code of the original BERT model for these methods, we were unable to investigate them thoroughly. We may explore these avenues in future iterations.

References

Francesco Gargiulo, Stefano Silvestri, Mario Ciampi, and Giuseppe De Pietro. 2019. Deep neural network for hierarchical extreme multi-label text classification. *Applied Soft Computing*, 79:125–138.

Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. 2019. Hierarchical multi-label text classification: An attention-based recurrent network approach. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1051–1060.

Chenbin Li, Guohua Zhan, and Zhihua Li. 2018. News text classification based on improved bi-lstm-cnn. *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*, pages 890–893.

Alina Petukhova and Nuno Fachada. 2022. MN-DS: A Multilabeled News Dataset for News Articles Hierarchical Classification.

Yong Song, Zhiwei Yan, Yukun Qin, Dongming Zhao, Xiaozhou Ye, Yuanyuan Chai, and Ye Ouyang. 2022. Hierarchical multi-label text classification based on a matrix factorization and recursive-attention approach. In *2022 7th International Conference on Big Data Analytics (ICBDA)*, pages 170–176.

Ming Wang Xiliang Zhang Weidong Zhao, Lin Zhu and Jinming Zhang. 2022. Wtl-cnn: a news text classification method of convolutional neural network based on weighted word embedding. *Connection Science*, 34(1):2291–2312.

A GitHub Link

`https://github.com/xiaoyang-sheng/News-Articles-Topic-Hierarchical-Classification-Weigh`

B Wandb Plot

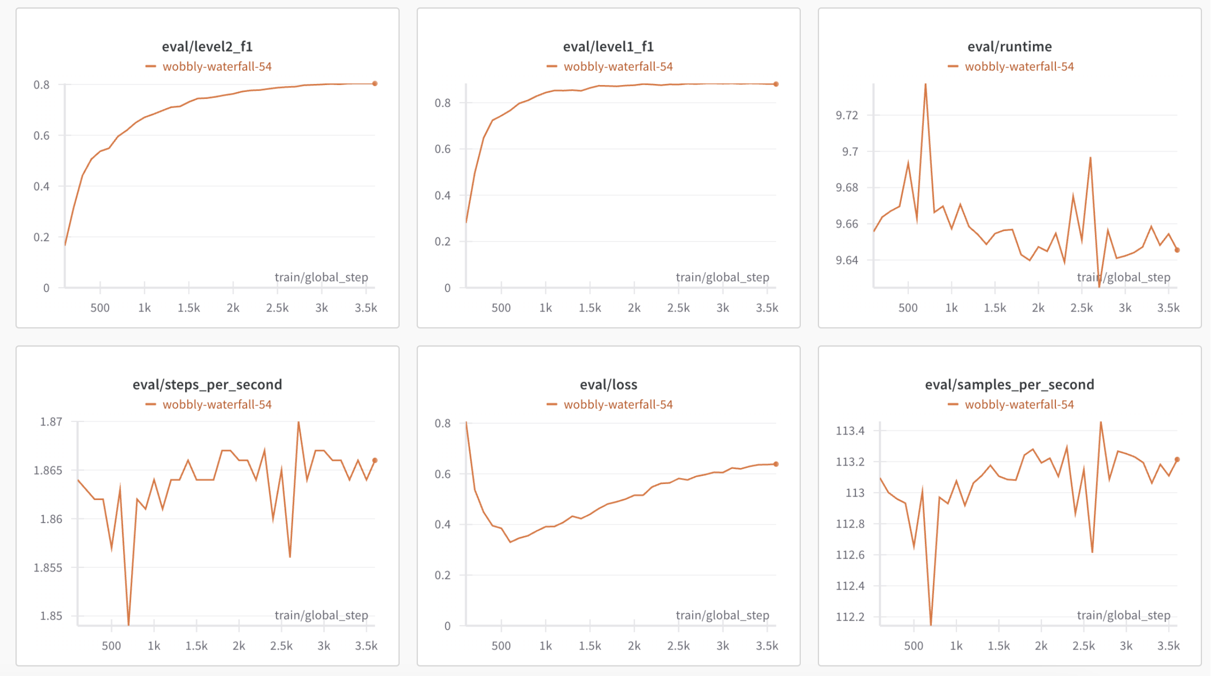

Figure 3: the wandb plot of evaluation metrics during training

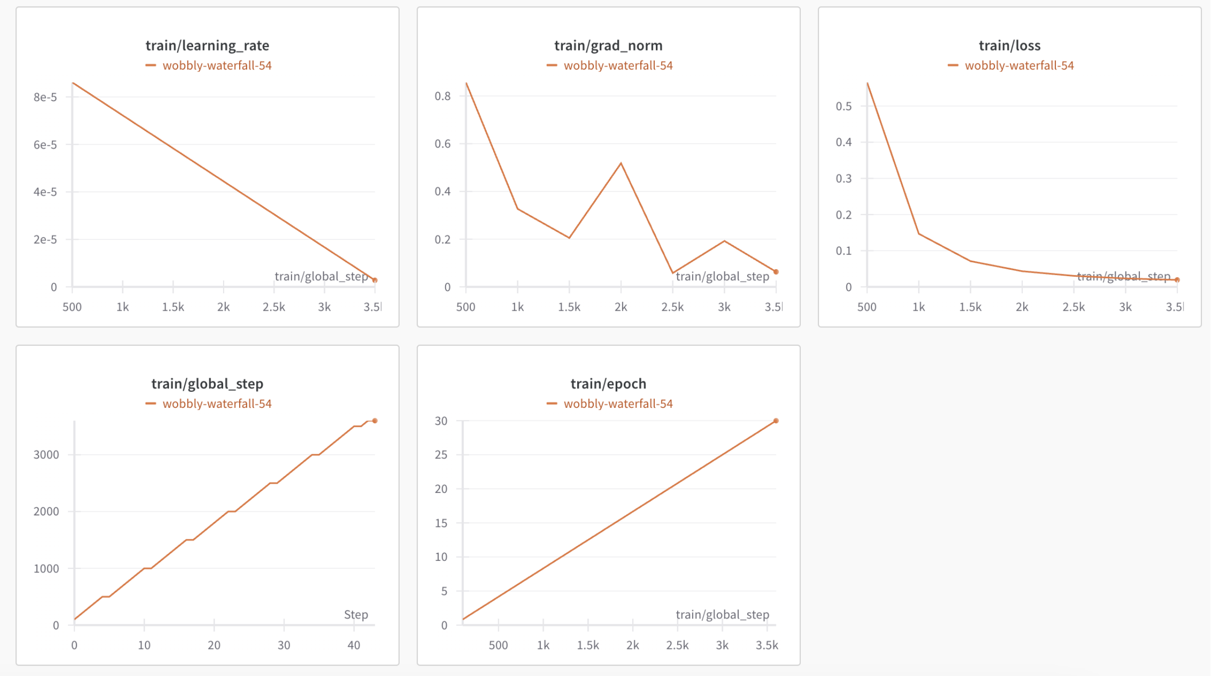

Figure 4: the wandb plot of training metrics during training