Enhancing Multi-Label Classification via Dynamic Label-Order Learning

Jiangnan Li[1,3†] , **Yice Zhang**[1,3†] , **Shiwei Chen**[1,2], **Ruifeng Xu**[1,2,3*]

[1]Harbin Institute of Technology, Shenzhen, China
[2]Peng Cheng Laboratory, Shenzhen, China
[3]Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies
lijiangnan@stu.hit.edu.cn, xuruifeng@hit.edu.cn

Abstract

Generative methods tackle Multi-Label Classification (MLC) by autoregressively generating label sequences. These methods excel at modeling label correlations and have achieved outstanding performance. However, a key challenge is determining the order of labels, as empirical findings indicate the significant impact of different orders on model learning and inference. Previous works adopt static label-ordering methods, assigning a unified label order for all samples based on label frequencies or co-occurrences. Nonetheless, such static methods neglect the unique semantics of each sample. More critically, these methods can cause the model to rigidly memorize training order, resulting in missing labels during inference. In light of these limitations, this paper proposes a dynamic label-order learning approach that adaptively learns a label order for each sample. Specifically, our approach adopts a difficulty-prioritized principle and iteratively constructs the label sequence based on the sample's semantics. To reduce the additional cost incurred by label-order learning, we use the same SEQ2SEQ model for label-order learning and MLC learning and introduce a unified loss function for joint optimization. Extensive experiments on public datasets reveal that our approach greatly outperforms previous methods. We will release our code at https://github.com/KagamiBaka/DLOL.

Introduction

Multi-Label Classification (MLC) aims to assign multiple labels to a single sample (Zhang and Zhou 2014). It holds substantial significance within the field of natural language processing and machine learning, finding applications in various real-world scenarios, including text categorization (Hayes and Weinstein 1990; Agrawal et al. 2013), information retrieval (Prabhu et al. 2018), and fine-grained emotion classification (Demszky et al. 2020; Huang et al. 2021).

Prior efforts have applied SEQ2SEQ models to MLC tasks (Nam et al. 2017; Yang et al. 2018, 2019; Madaan et al. 2022). Through autoregressive modeling, these methods exhibit a notable capability to capture label correlations, thereby yielding remarkable performance. One key challenge in these methods lies in determining the order of labels during training, as empirical findings indicate the significant impact of different orders on label generation (Vinyals, Bengio, and Kudlur 2016; Jin and Nakayama 2016; Chen et al. 2018). Early works primarily rely on alphabetical order or label frequencies to establish this order (Jin and Nakayama 2016; Nam et al. 2017; Yang et al. 2018). More recent work exploits label co-occurrences to determine a more appropriate order (Madaan et al. 2022). These methods are categorized as static label-ordering approaches since the relative order of labels within each sample remains constant.

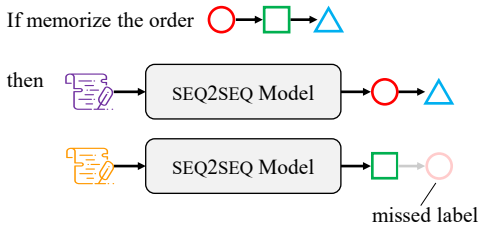

Figure 1: Training with a static order results in the model rigidly memorizing the order itself and further causes the model to miss labels during inference.

However, static label-ordering approaches suffer from two significant limitations. Firstly, the optimal order varies greatly across different samples as it is closely tied to the semantics of each sample. Imposing a static order for all samples disregards the unique characteristics inherent in each sample. Secondly, training with a static order will result in the model rigidly memorizing the order itself, as discussed in previous studies (Tsai and Lee 2020; Yazici et al. 2020). The rote memorization of the training order restricts the model's ability to generate label sequences that differ from the training order. This limitation further leads to the model's increased likelihood of missing labels since certain labels can be overlooked if they are not generated at specific positions, as illustrated in Figure 1.

Moreover, we observe a notable trend wherein labels positioned later in the label sequence tend to result in lower performance. The trend can be attributed to the concept of shortcut learning (Geirhos et al. 2020), wherein the model heavily relies on earlier gold labels provided during training to infer the subsequent labels. This reliance serves as a shortcut and reduces the model's dependence on the inher-

† These authors contributed equally to this work.
* Corresponding author.

ent semantics of the sample itself. Based on these analyses, we propose a difficulty-prioritized principle, which prioritizes placing difficult labels at the beginning of the label sequence. The underlying objective of this principle is to prioritize the learning of difficult labels, thereby ultimately improving the overall performance of the MLC model.

Motivated by the observations above and analyses, this paper proposes a dynamic label-order learning approach for MLC. This approach adopts the difficulty-prioritized principle to adaptively learn a label order for each sample. Specifically, we employ perplexity as a measure of label difficulty. Given a set of labels for a particular sample, we iteratively select a label with the highest perplexity and sequentially concatenate these selected labels to form a label sequence. Furthermore, to mitigate the additional cost associated with label-order learning, we integrate it into the MLC model and jointly optimize them with a unified loss function.

Our contributions can be summarized as follows:

- We investigate the effects of label orders in MLC tasks and introduce the difficulty-prioritized principle for label-order learning, which prioritizes placing difficult labels at the beginning of the label sequence.
- We propose a dynamic label-order learning approach for MLC. Unlike static label-order determining approaches, this approach adaptively learns a label order for each sample based on its unique characteristics and semantics. To the best of our knowledge, our approach is the first to learn a dynamic label order for MLC samples.
- We conduct extensive experiments on public MLC datasets, and the results consistently demonstrate that our approach achieves state-of-the-art performance. Besides, our experiments reveal that incorporating our approach into large language models for in-context learning significantly enhances their overall performance in MLC tasks.

Preliminary and Pilot Study

SEQ2SEQ Models for Multi-Label Classification

Given a text sequence $\boldsymbol{x}$, the MLC task is to assign a set of labels $L = \{l_1, \cdots, l_n\}$ to $\boldsymbol{x}$. Nam et al. (2017); Yang et al. (2018) apply SEQ2SEQ models to this task. In their approach, the text sequence serves as the source sequence, and the target sequence $\boldsymbol{y}$ is formed by concatenating the label set in a predetermined order. To adapt this approach to existing pre-trained models, we additionally introduce separator tokens to disaggregate the labels and include a <sos> token at the beginning and an <eos> token at the end of the target sequence. As such, the probability $p(L|\boldsymbol{x})$ is approximated by $p(\boldsymbol{y}|\boldsymbol{x})$. SEQ2SEQ model further decomposes $p(\boldsymbol{y}|\boldsymbol{x})$ autoregressively using the chain rule as follows:

$$p_\theta(\boldsymbol{y}|\boldsymbol{x}) = p_\theta(y_1 y_2 \cdots y_T|\boldsymbol{x}) \quad (1)$$
$$= p_\theta(y_1|\boldsymbol{x}) \prod_{t=2}^{T} p_\theta(y_t|y_1 y_2 \cdots y_{t-1}, \boldsymbol{x}),$$

where θ denotes the parameters of the SEQ2SEQ model, which are optimized through the following objective:

$$\hat{\theta} = \operatorname*{argmax}_\theta \sum_{\boldsymbol{x},\boldsymbol{y}} \log p_\theta(\boldsymbol{y}|\boldsymbol{x}). \quad (2)$$

Ordering Method		ρ_{differ}	Prec	Rec	F1
Static	Alphabetical	1.1	69.8	59.0	62.6
	Freq-first	1.0	69.6	57.7	61.7
	Rare-first	0.4	**74.1**	60.3	64.3
Dynamic	Random(ind)	-	72.9	**62.4**	**65.2**

Table 1: Comparison of static and dynamic label-ordering methods (%). ρ_{differ} denotes the proportion of samples differing from training order. Samples with only one label are ignored when calculating ρ_{differ}. *Alphabetical* refers to arranging the labels in alphabetical order. *Freq-first* and *rare-first* indicate sorting the labels based on their frequencies in the training set, with high or low-frequency labels placed at the beginning, respectively. *Random(ind)* means randomly assigning an individual label order for each sample.

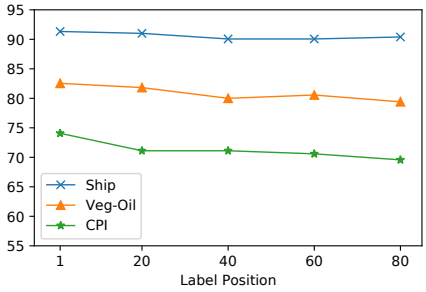

Figure 2: Label performance under different positions (F_1-score, %).

Pilot Study

We conduct pilot experiments to investigate label orders in the SEQ2SEQ model. The SEQ2SEQ model used for these experiments is BART-base (Lewis et al. 2020), and the dataset used is `Reuters-21578` (Hayes and Weinstein 1990).

Rote Memorization of Training Order. We train the SEQ2SEQ model using three static label-ordering methods and one dynamic method. We then count the proportion of samples differing from the training order (ρ_{differ}) on the test set, present the results in Table 1. The results show that the static methods exhibit a ρ_{differ} of only around 1%, indicating that *the model rigidly memorizes the training order*. This rote memorization of the training order increases the likelihood of missing labels since once a label is not generated at a specific position, it will not be generated again. Table 1 reveals that the static methods have significantly lower recall than the dynamic method, providing further evidence supporting this claim.

Effect of Label Position. To explore the impact of label position, we randomly choose three labels and alter their positions within the label sequence. We then observe the changes in the performance of these labels. According to Figure 2, label performance tends to decline as their positions move further back. Notably, this decline is particularly evident for difficult labels (those that underperform). These observations imply that the position substantially affects la-

Ordering Method	Difficult-labels	Easy-labels	All-labels
Random(uni)	44.9	84.4	64.7
Difficulty-prioritized	**47.9 (+3.0)**	84.2 (-0.2)	**66.1 (+1.4)**
Ease-prioritized	40.4 (-4.5)	**85.1 (+0.7)**	62.8 (-1.9)

Table 2: The performances of the three static label-order methods on difficult, easy, and all labels (F_1-score, %). *Random(uni)* means randomly assigning a uniform label order to all samples.

bel performance, especially for difficult ones, and placing labels at the beginning can improve their performance.

The above observations motivate us to prioritize placing difficult labels at the beginning of the label sequence. To validate this approach, we conduct a pilot experiment. Initially, we assign a random label order to all samples, train the MLC model, and roughly measure the label difficulty based on label performance on the development set. Next, we experiment with difficulty-prioritized and ease-prioritized label-ordering principles, with their respective performances detailed in Table 2. The results demonstrate that the difficulty-prioritized principle considerably improves the performance of difficult labels, resulting in a performance boost of 3%. Although this principle has a minor negative effect on easy labels, with a decrease of 0.2%, the overall impact remains positive, leading to a performance gain of 1.4%. In contrast, the ease-prioritized principle shows a noticeable negative impact on overall performance.

In summary, these pilot experiments indicate that (1) static label-ordering methods lead to rote memorization of the training order, resulting in a higher likelihood of missing labels during inference, and (2) the difficulty-prioritized principle enhances overall performance. Therefore, it is crucial to develop a dynamic label-order learning approach based on the difficulty-prioritized principle for MLC.

Our Approach

Overview

Our objective is to learn an appropriate label order for a given sample $\boldsymbol{x}$ with a label set L. As outlined in Algorithm 1, we iteratively select the most difficult label from the label set and sequentially concatenate these labels to form the label sequence. The core of our approach lies in estimating the difficulty of labels. To accomplish this, we construct multiple candidate label sequences based on both the ordered and unordered labels. Subsequently, we estimate the difficulty of labels by calculating the perplexity of these sequences using a SEQ2SEQ model. Furthermore, to enhance the accuracy of this estimation, we optimize the SEQ2SEQ model on these sequences through a loss function.

Perplexity-Based Estimation of Label Difficulty

While it is feasible to employ label performance on the development set as an estimate of label difficulty, it is challenging to scale this approach on individual samples. In this paper, we utilize perplexity as a measure of label difficulty, as it quantifies the model's uncertainty associated with a given sequence.

Algorithm 1: The proposed label-order learning algorithm

Input: the text sequence $\boldsymbol{x}$, the label set L, and a SEQ2SEQ model with parameter θ
Output: the ordered label sequence $\boldsymbol{y}_{\text{ordered}}$

Let $\boldsymbol{y}_{\text{ordered}} \leftarrow [\texttt{<sos>}]$.
Let $L_{\text{unordered}} \leftarrow L$.
while $\text{len}(L_{\text{unordered}}) > 0$ **do**
  Update θ by the loss function in Eq. 9.
  Estimate the difficulty of unordered labels by Eq. 3-4.
  $l^* = \underset{l \in L_{\text{unordered}}}{\text{argmax}}\ \text{difficulty}(l)$.
  $L_{\text{unordered}} \leftarrow L_{\text{unordered}} \setminus \{l^*\}$.
  $\boldsymbol{y}_{\text{ordered}} \leftarrow \boldsymbol{y}_{\text{ordered}} \oplus [l^*, \texttt{<sep>}]$.
end while
$\boldsymbol{y}_{\text{ordered}}[-1] \leftarrow \texttt{<eos>}$.
return $\boldsymbol{y}_{\text{ordered}}$.

Given the ordered labels $\boldsymbol{y}_{\text{ordered}}$ and the unordered labels $L_{\text{unordered}}$, we construct multiple candidate label sequences by individually appending each unordered label to the ordered label sequence. We then calculate the perplexity of each label sequence and use it as a measure of the difficulty of the unordered labels. This process can be formulated as follows:

$$\boldsymbol{y}^{(l)} = \boldsymbol{y}_{\text{ordered}} \oplus [l, \texttt{<tbc>}], \tag{3}$$

$$\text{difficulty}(l) = \text{PPL}(\boldsymbol{y}^{(l)}|\boldsymbol{x}; \theta), \tag{4}$$

where the token <tbc> signifies that the sequence is incomplete, and $\text{PPL}(\cdot|\boldsymbol{x}; \theta)$ denotes the perplexity of a sequence, which is calculated using a SEQ2SEQ model with parameters θ. The perplexity calculation is performed as follows:

$$\text{PPL}(\boldsymbol{y}|\boldsymbol{x}; \theta) = p_\theta(\boldsymbol{y}|\boldsymbol{x})^{-1/|\boldsymbol{y}|}. \tag{5}$$

Joint Optimization of Label-Order&MLC Learning

As illustrated in Algorithm 1, before estimating the difficulty of unordered labels, we optimize the SEQ2SEQ model on the candidate label sequences. This optimization step serves to enhance the accuracy of difficulty estimation. Another benefit is that it equips the SEQ2SEQ model with the capability to handle the MLC task. This implies that we can use the same SEQ2SEQ model for both label-order learning and MLC learning, streamlining the overall process. We detail the loss function for joint optimization below.

Multi-Reference Training. Vinyals, Bengio, and Kudlur (2016) propose a loss function for optimizing multiple label sequences simultaneously. However, Qin et al. (2019) point out that this loss function suffers from the uniformity issue, as it treats all label sequences equally, and thus they propose a new loss function to circumvent this issue. The formulations of these two loss functions are as follows:

$$\mathcal{L}_{\text{vinyal}} = -\sum_{l \in L_{\text{unordered}}} \log p_\theta(\boldsymbol{y}^{(l)}|\boldsymbol{x}), \tag{6}$$

$$\mathcal{L}_{\text{qin}} = -\log \sum_{l \in L_{\text{unordered}}} p_\theta(\boldsymbol{y}^{(l)}|\boldsymbol{x}). \tag{7}$$

Dataset	Samples			Labels Per Sample			Label Sets	Words Per Sample
	Train	Dev	Test	Avg	Min	Max		
Reuters-21578 (Hayes and Weinstein 1990)	7,587	1,000	2,005	1.2	1	15	90	152.2
RCV1-V2 (Lewis et al. 2004)	775,220	21,510	1,191	3.2	1	17	103	123.9
Slashdot (Qin et al. 2019)	19,258	2,000	2,814	4.2	2	12	291	130.8
GoEmotions (Demszky et al. 2020)	43,227	5,423	5,421	1.2	1	5	28	15.8

Table 3: Statistics of four datasets.

In this paper, we combine these two loss functions to achieve a trade-off between uniformity and diversity using a hyper-parameter α:

$$\mathcal{L}_{\text{multi}} = \alpha\mathcal{L}_{\text{vinyal}} + (1-\alpha)\mathcal{L}_{\text{qin}}. \tag{8}$$

Label Smoothing. Label smoothing is a technique that redistributes the probability mass of one-hot distribution uniformly across the vocabulary (Müller, Kornblith, and Hinton 2019). In this paper, we incorporate label smoothing as a regularization term to reduce the model's over-confidence in specific label sequences. Notably, we do not apply label smoothing to special tokens, as their likelihood of being predicted as other tokens is minimal. The final loss for joint optimization is formulated as follows:

$$\mathcal{L} = (1-\beta)\mathcal{L}_{\text{multi}} + \frac{\beta}{|\mathcal{V}|}\mathcal{L}_{\text{smooth}}, \tag{9}$$

$$\mathcal{L}_{\text{smooth}} = -\sum_{l\in L_{\text{unordered}}}\sum_{t=1}^{|\boldsymbol{y}^{(l)}|} I^{y_t^{(l)}} \sum_{v\in\mathcal{V}} \log p_\theta(v|\boldsymbol{y}_{<t}^{(l)}, \boldsymbol{x}), \tag{10}$$

where β is a hyper-parameter, $\mathcal{V}$ denotes the vocabulary, and $\boldsymbol{y}_{<t} = y_1 y_2 \cdots y_{t-1}$. In addition, I^{y_t} is a binary value indicating whether y_t is not a special token.

$$I^{y_t} = \begin{cases} 0 & y_t \in \{\texttt{<sos>}, \texttt{<sep>}, \texttt{<tbc>}, \texttt{<eos>}\}; \\ 1 & \text{otherwise.} \end{cases}$$

Further Optimization on MLC. After obtaining the ordered label sequence $\boldsymbol{y}_{\text{ordered}}$ through Algorithm 1, we continue to train the SEQ2SEQ model on $(\boldsymbol{x}, \boldsymbol{y}_{\text{ordered}})$ for m iterations to ensure its convergence on the MLC task.

Decoding with EOS Penalty

During inference, we generate the label sequence $\hat{\boldsymbol{y}}$ through autoregressive decoding:

$$\hat{y}_t = \underset{v\in\mathcal{V}}{\text{argmax}}\, p_\theta(v|\hat{y}_1\hat{y}_2\cdots\hat{y}_{t-1}, \boldsymbol{x}). \tag{11}$$

We observe that the SEQ2SEQ model often terminates decoding before generating all the labels. To mitigate this issue, we introduce an eos penalty during decoding. Specifically, we modify the probability of <eos> by simply multiplying it with a weight γ that is less than 1:

$$p_\theta(\texttt{<eos>}|\hat{\boldsymbol{y}}, \boldsymbol{x}) = p_\theta(\texttt{<eos>}|\hat{\boldsymbol{y}}, \boldsymbol{x}) * \gamma. \tag{12}$$

Experiments

Experiment Settings

Following previous works Qin et al. (2019); Madaan et al. (2022), we evaluate our approach on four typical multi-label classification datasets, namely `Reuters-21578`, `RCV1-V2`, `Slashdot`, and `GoEmotions`. The statistical information of these datasets is displayed in Table 3. The evaluation metrics we use are macro precision, macro recall, and macro F1-score. We leverage BART-base (Lewis et al. 2020) as the backbone and implement our approach on top of it. To reduce randomness, we repeat our approach three times with different seeds and report the average results.

Baselines

We select multi-label classification models from recent years as our baselines, which can be classified into two groups:

Discriminative methods generally utilize BERT as the language encoder and then predict labels through multiple binary classifiers. To learn label semantics or model label correlations, ML-REASONER (Wang et al. 2021) applies a novel iterative reasoning mechanism; CORE (Zhang et al. 2021a) joint learns samples and labels in the same space; BOX (Onoe et al. 2021) and HIDDEN (Chatterjee et al. 2021) investigate label embeddings in high-dimensional space; LACO (Zhang et al. 2021b) introduces a novel multi-task learning approach; NPCRF (Jiang et al. 2022) formulates MLC problems under pairwise conditional random field.

Generative methods usually concatenate multiple labels together into a sequence and then use SEQ2SEQ models for modeling. The challenge within generative methods lies in determining the order of labels. One straightforward approach is to use the order in which labels occur in the dataset. SGM (Yang et al. 2018) sorts labels according to their frequencies in the training set and places high-frequency labels at the beginning of the sequence. SETAUG (Madaan et al. 2022) exploits label co-occurrences to establish an order. Other methods opt to minimize the influence of order. SETRNN (Qin et al. 2019) samples multiple orders for each sample and optimizes the model on these orders simultaneously. SEQ2SET employs an order-agnostic reinforcement learning framework to guide the training of the SEQ2SEQ model. OTSEQ2SET (Cao and Zhang 2022) further integrates bipartite matching and optimal transport distance to enhance SEQ2SET.

Some methods have not published their performance on specific datasets. For these cases, we run their official implementations on these datasets. In addition, some methods

Methods	Reuters-21578			RCV1-V2			Slashdot			GoEmotions			Avg-F1
	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	Prec	Rec	F1	
Discriminative Methods													
BERT (Devlin et al. 2019)	69.8	57.3	61.7	77.3	61.9	66.7	41.3	30.9	34.3	54.2	43.9	47.5	52.6
BERT+ML-REASONER (Wang et al. 2021)	71.9	58.7	63.2	77.3	69.1	70.7	44.5	32.4	35.6	57.3	45.1	49.0	54.6
BERT+CORE (Zhang et al. 2021a)	71.2	59.9	63.6	75.9	68.4	70.3	44.1	31.2	35.7	57.1	45.5	49.7	54.8
BERT+BOX (Onoe et al. 2021)	72.5	59.3	63.6	75.9	67.3	69.2	46.7	33.8	36.9	58.7	46.4	50.1	55.0
BERT+HIDDEN (Chatterjee et al. 2021)	71.6	62.8	64.9	73.7	69.3	69.5	45.3	35.1	37.2	59.2	42.9	48.5	55.0
BERT+NPCRF (Jiang et al. 2022)	70.3	64.6	65.2	78.0	69.4	71.2	40.1	38.9	36.9	55.0	52.5	51.8	56.3
BERT+LACO (Zhang et al. 2021b)	71.8	63.4	65.3	77.6	71.5	73.1	42.3	35.7	36.1	56.6	48.3	51.5	56.5
Generative Methods													
SEQ2SET (Yang et al. 2019)	66.3	55.8	58.3	77.1	66.5	68.8	37.4	31.5	31.2	49.0	45.6	45.5	51.0
OTSEQ2SET (Cao and Zhang 2022)	67.0	57.1	59.2	75.9	68.7	69.7	38.9	31.8	32.7	50.7	46.2	46.8	52.1
BART (Lewis et al. 2020)	71.0	60.6	63.6	79.4	70.4	72.9	42.5	34.7	36.2	56.9	47.9	50.9	55.9
BART+SGM (Yang et al. 2018)	71.9	61.5	64.8	81.0	70.6	73.1	43.2	35.4	36.8	57.3	48.4	51.2	56.5
BART+SETRNN (Qin et al. 2019)	72.3	61.7	65.0	80.3	70.9	73.4	43.8	36.0	37.4	57.8	48.9	51.7	56.9
BART+SETAUG (Madaan et al. 2022)	72.8	62.9	65.9	81.8	71.1	73.9	43.0	36.1	37.1	54.6	52.4	51.7	57.2
Our Approach													
Dynamic Label-Order Learning (DLOL)	74.9	66.3	**68.8**	81.3	72.7	**75.0**	38.7	42.0	**39.0**	56.3	53.9	**53.8**	**59.2**
w/ vinyal loss	75.3	64.5	67.5	80.8	72.6	74.7	39.2	40.1	37.8	57.5	51.5	52.8	-1.0
w/ qin loss	76.0	64.0	67.7	81.4	72.4	74.9	40.3	39.8	38.4	53.3	54.9	52.9	-0.7
w/o label smoothing	75.4	65.1	67.9	81.3	71.9	74.6	40.8	39.7	38.5	57.4	51.1	53.2	-0.6
w/o eos penalty	76.5	64.5	68.3	81.4	72.2	74.7	40.6	38.5	37.9	60.9	49.7	53.3	-0.6

Table 4: Comparison results on four multi-label classification datasets (%). The best results are highlighted in bold, while the second-best results are underlined.

(such as SGM and SETRNN) do not use BART-base as the backbone. For fairness of comparison, we reproduce these methods using BART-base as the backbone.

Main Results

Table 4 presents the comparison results between our approach and existing methods. As can be observed, our approach consistently outperforms previous methods across all four datasets, achieving an average improvement of 2% in F_1-score. This outcome demonstrates the effectiveness of our approach. It is especially noteworthy that our approach exhibits a significantly higher recall, aligning with our claim that dynamic label-ordering methods can reduce the likelihood of missing labels.

Furthermore, inspired by the recent success of Large Language Models (LLMs) (Ouyang et al. 2022), we conduct additional experiments on two prominent LLMs, namely `GPT-3.5-Turbo` and `Text-Davinci-003`. Unlike traditional deep learning models, LLMs gain task-specific capabilities through natural language instruction and demonstration examples. Many works (Liu et al. 2022; Dai et al. 2023) have observed that demonstration examples significantly influence model performance. Meanwhile, despite their robust capabilities, LLMs still adopt an autoregressive decoding approach for text generation. Thus, in tasks like MLC, the decoding order of labels can be just as important for LLMs as for smaller models. To verify this, we extend the application of our approach to LLMs. Since the perplexity calculation is infeasible on these two LLMs, we instead input the sample and its gold labels into LLMs and prompt them to sort the labels based on difficulty. According to the results in Table 5, LLMs exhibit a high sensitivity to label order, and our principle attains the best performance.

Method	Prec	Rec	F1
`GPT-3.5-Turbo`			
Alphabetical	41.0	36.0	35.7
Random(ind)	46.5	37.9	37.6
Difficulty-prioritized	**53.1**	**40.9**	**42.1**
`Text-Davinci-003`			
Alphabetical	40.4	41.5	36.5
Random(ind)	49.2	36.8	37.4
Difficulty-prioritized	**48.5**	**43.7**	**40.4**

Table 5: Experimental results on LLMs (%). The dataset used for evaluation is `GoEmotions`, chosen due to its relatively short samples, which permit the inclusion of sufficient demonstration examples. We randomly select ten samples from the training set that have three or more labels to serve as demonstration examples.

Method	Reuters	RCV1	Slashdot	GoEmo
Alphabetical	65.2	73.4	36.0	52.0
Freq-first	65.1	72.9	36.2	51.7
Rare-first	66.3	73.8	37.6	52.2
Difficulty-prioritized(sta)	67.1	74.3	38.2	51.9
Ease-prioritized	66.0	72.7	36.9	51.5
Difficulty-prioritized(dyn)	**68.8**	**75.0**	**39.0**	**53.8**

Table 6: Comparison results of different label-ordering principles (F_1-score, %). In the *difficulty-prioritized(sta)* principle, we pre-estimate the difficulty of labels based on their performance on the development set. In the *difficulty-prioritized(dyn)* principle, we estimate the difficulty of labels based on perplexity.

Method	Reuters	RCV1	Slashdot	GoEmo
Joint Optimization	**68.8**	**75.0**	**39.0**	**53.8**
Separate Optimization	67.4	74.3	37.7	53.1
No Order Optimization	66.9	73.9	38.2	52.7

Table 7: Ablation results on optimization of label-order learning and MLC learning (F_1-score, %).

This affirms the strong universality of our approach.

Ablation Studies

We conduct ablation experiments to analyze the impact of each component in our approach. As illustrated in Table 4, we make the following observations: (1) In multi-reference training, the exclusive use of either Vinyals loss or Qin loss would lead to a decline in model performance; (2) The removal of label smoothing or eos penalty also results in a drop in model performance. Notably, the absence of the eos penalty leads to a more notable decrease in recall. These observations highlight the necessity of the aforementioned components in our approach.

Comparison of Label-Ordering Principles. Our approach utilizes the difficulty-prioritized principle to construct the label sequence. To examine its effectiveness, we replace it with several other principles while keeping the rest of our approach unchanged. As depicted in Table 6, other principles fall short in performance compared to the difficulty-prioritized principle, which confirms its effectiveness. Moreover, it is noticeable that dynamically estimating label difficulty based on perplexity yields better results than estimating it based on the development set performance.

Joint Optimization. In our approach, we use the same model for both label-order learning and MLC learning and optimize them using a unified loss function. First, we attempt to separate these two learning processes and optimize them individually. As shown in Table 7, separate optimization significantly decreases model performance, suggesting that shared knowledge exists between label-order learning and MLC learning and joint optimization is beneficial. Second, we try optimizing only the MLC learning process, which entails estimating label difficulty based on the original weights of BART-base. The results in Table 7 indicate a further decline in model performance. In summary, experimental results emphasize the importance of joint optimization in our approach.

Label Pair		Label1-first	Label2-first
Label1	Label2		
corn	wheat	60.0	40.0
soybean	oilseed	63.6	36.4
corn	oilseed	64.1	35.9
soybean	corn	65.6	34.4
yen	dlr	60.0	40.0

Table 8: Statistics of the relative order of label pairs in Reuters-21578 (%).

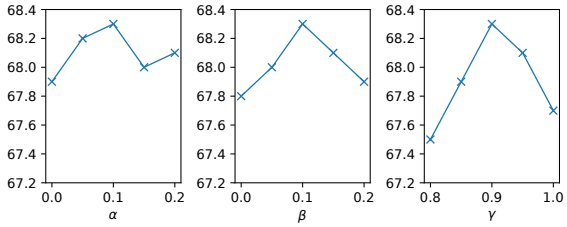

Figure 3: Impact of three hyper-parameters on the development set of Reuters-21578 (F_1-score, %).

Further Analysis

Label Order Variation Across Samples. In this paper, we hypothesize that the optimal label order for a specific sample is inherent in its unique semantics. To substantiate this assertion, we undertake an analysis focused on label order variation across different samples. For this analysis, we presume that the label order learned by our approach is the "optimal order". Subsequently, we select the five pairs of labels most frequently co-occurring in the training set and statistically examine their relative positioning. The corresponding results are displayed in Table 8. We can observe that an absolute order is not exhibited by these label pairs, with the highest proportion of a label appearing first amounting to only 65.6%. Such observations underscore the variability in the optimal label order among different samples, further illustrating that it is unreasonable to impose a unified label order on all samples.

Hyper-Parameter Analysis. Our approach has three hyper-parameters for multi-reference training, label smoothing, and eos penalty. We conduct experiments to analyze the impact of these hyper-parameters. As shown in Figure 3, we observe that for Reuters-21578, the optimal values for these hyper-parameters are $\alpha = 0.1$, $\beta = 0.1$, and $\gamma = 0.9$, respectively. Moreover, our approach appears to be relatively sensitive to these hyper-parameters, suggesting a potential for further performance boost through comprehensive grid search. Meanwhile, such sensitivity to hyper-parameters constitutes an inherent challenge to our

Sample	Ground-truth	Alphabetical	Our Approach
Gotta love these kids! When they do or say the cutest things like that, it really makes u melt..... Enjoy!!!	joy, love	love	love, joy
Oh absolutely. No way he should run in 2020. I don't know what alternatives there are though.	confusion, disapproval	disapproval	disapproval, confusion
I'm glad you have your kids, but have you seen Brooklyn 99 like the op stated? It's worth living for too.	admiration, confusion, joy	joy	joy, admiration

Table 9: Case Study.

Method	Reuters		RCV1		Slashdot		GoEmo	
	Time	F1	Time	F1	Time	F1	Time	F1
BART	67	63.6	231	72.9	196	36.2	26	50.9
Ours	92	68.8	453	75.0	186	39.0	31	53.8

Table 10: Comparison results on training time (minutes) and performance (F_1-score, %). Training time refers to the time it takes to train the model on the training set to convergence.

approach, and we look forward to future work addressing this issue.

Time Complexity Analysis. For a sample, we assume that its length is l, the size of its label set is n, and the number of training epochs is m. The time complexity of using a SEQ2SEQ model for MLC learning would then be $O(ml + mn)$.[1] The process of using a SEQ2SEQ model for label-order learning requires n iterations, with each of which has a complexity approximating $O(l + n^2)$. This culminates in an overall complexity of $O(nl+n^3)$. This implies that the time complexity of our label-order learning process is cubically related to n. This property could be seen as disadvantageous, but it is mitigated by two factors: (1) By caching the activations computed in previous steps, we can further reduce the time complexity to $O(l + n^2)$; (2) In real-world datasets, n typically isn't large, as corroborated by Table 3. Furthermore, we record the time cost of training our approach, as shown in Table 10. It can be observed that the additional cost of label-order learning is limited and acceptable when considering the enhanced performance it enables.

Case Study

We further illustrate our approach through several representative cases, as presented in Table 9. In the first case, the sample includes the phrase "*gotta love these kids*", which strongly suggests the presence of the label love. Both the alphabetical order method and our approach predict love first. However, in alphabetical order, joy precedes love, causing the alphabetical order method to overlook joy once love has been predicted. Similar observations can be made for the second and third cases.

[1]When the sequence is not too long, we can ignore the consumption of attention, that the time complexity of a SEQ2SEQ model is linearly related to the sequence length.

Related Works

Multi-Label Classification (MLC) aims to assign multiple labels to a single sample. The most naive approach is to formulate MLC as multiple binary classification problems (Boutell et al. 2004). To further model label correlations, Read et al. (2011) propose classifier chains, which sequentially predicts each label and augments the classifiers with all prior predictions in the chain. However, this method incurs a high computational cost. In recent years, the SEQ2SEQ model has been widely used in MLC due to its ability to model high-order label correlations and generate a varying number of labels (Nam et al. 2017; Yang et al. 2018; Qin et al. 2019; Madaan et al. 2022). Using SEQ2SEQ for MLC requires concatenating multiple labels into a label sequence. How to determine the order of labels is an important issue, as many works have shown that it has a significant impact on label generation (Vinyals, Bengio, and Kudlur 2016; Jin and Nakayama 2016; Chen et al. 2018). One line of work investigates how to determine an appropriate label order. Yang et al. (2018) exploit label frequencies, and Madaan et al. (2022) exploit label co-occurrences. We believe the issue with this line of work is that it assigns a uniform label order to all samples, neglecting the unique semantics of each sample. Another line of work investigates the use of order-agnostic methods to train SEQ2SEQ or to reduce the impact of order. Yang et al. (2019); Cao and Zhang (2022); Yazici et al. (2020) employ reinforcement learning or introduce bipartite graph matching methods to train SEQ2SEQ models. Qin et al. (2019) mitigate the impact of order by considering multiple orders simultaneously. The issue with this line of work is that they either weaken the ability of modeling label correlations or come with a high computational cost.

Conclusion

This paper investigates the label order in multi-label classification. We first uncover the significant impact of label order on label generation and highlight the limitations of previous label-ordering methods. Based on these insights, we propose a dynamic label-order learning approach that adaptively learns a label order for each sample. Empirical evaluations on public datasets demonstrate the superiority of our approach over previous methods. Moreover, experiments on large language models also indicate the strong universality of our approach. Subsequent ablation studies confirm the effectiveness of each component in our approach.

Acknowledgments

We thank the anonymous reviewers for their valuable suggestions to improve the quality of this work. This work was partially supported by the National Natural Science Foundation of China 62176076, Natural Science Foundation of Guangdong 2023A1515012922, Shenzhen Foundational Research Funding JCYJ20220818102415032, Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies 2022B1212010005, The Major Key Project of PCL PCL2023A09.

References

Agrawal, R.; Gupta, A.; Prabhu, Y.; and Varma, M. 2013. Multi-Label Learning with Millions of Labels: Recommending Advertiser Bid Phrases for Web Pages. In *Proceedings of the 22nd International Conference on World Wide Web*, WWW '13, 13–24. New York, NY, USA: Association for Computing Machinery. ISBN 9781450320351.

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern Recognition*, 37(9): 1757–1771.

Cao, J.; and Zhang, Y. 2022. OTSeq2Set: An Optimal Transport Enhanced Sequence-to-Set Model for Extreme Multi-label Text Classification. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 5588–5597. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Chatterjee, S.; Maheshwari, A.; Ramakrishnan, G.; and Jagarlapudi, S. N. 2021. Joint Learning of Hyperbolic Label Embeddings for Hierarchical Multi-label Classification. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2829–2841. Online: Association for Computational Linguistics.

Chen, S.-F.; Chen, Y.-C.; Yeh, C.-K.; and Wang, Y.-C. 2018. Order-Free RNN With Visual Attention for Multi-Label Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Dai, D.; Sun, Y.; Dong, L.; Hao, Y.; Ma, S.; Sui, Z.; and Wei, F. 2023. Why Can GPT Learn In-Context? Language Models Secretly Perform Gradient Descent as Meta-Optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, 4005–4019. Toronto, Canada: Association for Computational Linguistics.

Demszky, D.; Movshovitz-Attias, D.; Ko, J.; Cowen, A.; Nemade, G.; and Ravi, S. 2020. GoEmotions: A Dataset of Fine-Grained Emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4040–4054. Online: Association for Computational Linguistics.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Geirhos, R.; Jacobsen, J.-H.; Michaelis, C.; Zemel, R.; Brendel, W.; Bethge, M.; and Wichmann, F. A. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11): 665–673.

Hayes, P. J.; and Weinstein, S. P. 1990. CONSTRUE/TIS: A System for Content-Based Indexing of a Database of News Stories. In *Proceedings of the The Second Conference on Innovative Applications of Artificial Intelligence*, IAAI '90, 49–64. AAAI Press. ISBN 0262680688.

Huang, C.; Trabelsi, A.; Qin, X.; Farruque, N.; Mou, L.; and Zaiane, O. R. 2021. Seq2Emo: A sequence to multi-label emotion classification model. In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, 4717–4724.

Jiang, C.; Jiang, Y.; Wu, W.; Xie, P.; and Tu, K. 2022. Modeling Label Correlations for Ultra-Fine Entity Typing with Neural Pairwise Conditional Random Field. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 6836–6847. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Jin, J.; and Nakayama, H. 2016. Annotation order matters: Recurrent Image Annotator for arbitrary length image tagging. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2452–2457.

Lewis, D. D.; Yang, Y.; Rose, T. G.; and Li, F. 2004. RCV1: A New Benchmark Collection for Text Categorization Research. *J. Mach. Learn. Res.*, 5: 361–397.

Lewis, M.; Liu, Y.; Goyal, N.; Ghazvininejad, M.; Mohamed, A.; Levy, O.; Stoyanov, V.; and Zettlemoyer, L. 2020. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. Online: Association for Computational Linguistics.

Liu, J.; Shen, D.; Zhang, Y.; Dolan, B.; Carin, L.; and Chen, W. 2022. What Makes Good In-Context Examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 100–114. Dublin, Ireland and Online: Association for Computational Linguistics.

Madaan, A.; Rajagopal, D.; Tandon, N.; Yang, Y.; and Bosselut, A. 2022. Conditional set generation using Seq2seq models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 4874–4896. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Müller, R.; Kornblith, S.; and Hinton, G. E. 2019. When does label smoothing help? In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Nam, J.; Loza Mencía, E.; Kim, H. J.; and Fürnkranz, J. 2017. Maximizing Subset Accuracy with Recurrent Neural Networks in Multi-label Classification. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Onoe, Y.; Boratko, M.; McCallum, A.; and Durrett, G. 2021. Modeling Fine-Grained Entity Types with Box Embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2051–2064. Online: Association for Computational Linguistics.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P. F.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 27730–27744. Curran Associates, Inc.

Prabhu, Y.; Kag, A.; Harsola, S.; Agrawal, R.; and Varma, M. 2018. Parabel: Partitioned Label Trees for Extreme Classification with Application to Dynamic Search Advertising. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, 993–1002. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee. ISBN 9781450356398.

Qin, K.; Li, C.; Pavlu, V.; and Aslam, J. 2019. Adapting RNN Sequence Prediction Model to Multi-label Set Prediction. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 3181–3190. Minneapolis, Minnesota: Association for Computational Linguistics.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2011. Classifier Chains for Multi-Label Classification. *Mach. Learn.*, 85(3): 333–359.

Tsai, C.-P.; and Lee, H.-Y. 2020. Order-Free Learning Alleviating Exposure Bias in Multi-Label Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04): 6038–6045.

Vinyals, O.; Bengio, S.; and Kudlur, M. 2016. Order Matters: Sequence to sequence for sets. In Bengio, Y.; and LeCun, Y., eds., *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Wang, R.; Ridley, R.; Su, X.; Qu, W.; and Dai, X. 2021. A novel reasoning mechanism for multi-label text classification. *Information Processing & Management*, 58(2): 102441.

Yang, P.; Luo, F.; Ma, S.; Lin, J.; and Sun, X. 2019. A Deep Reinforced Sequence-to-Set Model for Multi-Label Classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5252–5258. Florence, Italy: Association for Computational Linguistics.

Yang, P.; Sun, X.; Li, W.; Ma, S.; Wu, W.; and Wang, H. 2018. SGM: Sequence Generation Model for Multi-label Classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, 3915–3926. Santa Fe, New Mexico, USA: Association for Computational Linguistics.

Yazici, V. O.; Gonzalez-Garcia, A.; Ramisa, A.; Twardowski, B.; and Weijer, J. v. d. 2020. Orderless Recurrent Models for Multi-Label Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, M.-L.; and Zhou, Z.-H. 2014. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8): 1819–1837.

Zhang, Q.-W.; Zhang, X.; Yan, Z.; Liu, R.; Cao, Y.; and Zhang, M.-L. 2021a. Correlation-Guided Representation for Multi-Label Text Classification. In Zhou, Z.-H., ed., *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, 3363–3369. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Zhang, X.; Zhang, Q.-W.; Yan, Z.; Liu, R.; and Cao, Y. 2021b. Enhancing Label Correlation Feedback in Multi-Label Text Classification via Multi-Task Learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 1190–1200. Online: Association for Computational Linguistics.