Does the Order Matter? A Random Generative Way to Learn Label Hierarchy for Hierarchical Text Classification

Jingsong Yan, Piji Li, Haibin Chen, Junhao Zheng and Qianli Ma, *Member, IEEE*

***Abstract*—Hierarchical Text Classification (HTC) is an essential and challenging task due to the difficulty of modeling label hierarchy. Recent generative methods have achieved state-of-the-art performance by flattening the *local label hierarchy* into a label sequence with a specific order. However, the order between labels does not naturally exist and the generation of the current label should incorporate the information in all other target labels. Moreover, the generative methods usually suffer from the error accumulation problem. To this end, we propose a new framework named sequence-to-label (Seq2Label) with a random generative way to learn label hierarchy for hierarchical text classification. Instead of using only one specific order, we shuffle the label sequence by a Label Sequence Random Shuffling (LSRS) mechanism so that a text will be mapped to several different order label sequences during the training phase. To alleviate the error accumulation problem, we further propose a Hierarchy-aware Negative Sampling (HNS) strategy with a negative label-aware loss to better distinguish target labels and negative labels. In this way, our model can capture the hierarchical and co-occurrence information of the target labels of each text. The experimental results on three benchmark datasets show that Seq2Label achieves state-of-the-art results.**

***Index Terms*—Hierarchical text classification, label sequence random shuffling, error accumulation, hierarchy-aware negative sampling.**

I. Introduction

HIERARCHICAL text classification (HTC) is an important subtask of a multi-label text classification (MLC) [1], which is widely used in the news classification [2], advertising systems [3], information retrieval [4], fine-grained entity typing [5], etc. Different from MLC, HTC aims to assign each document to one or more node-paths from a taxonomic hierarchy structure. The taxonomic hierarchy structure is always represented as a tree or a directed acyclic graph [6], as depicted in Figure 1.

Modeling the label hierarchy is crucial for improving model performances in HTC [7]–[10]. The ideal label representation should incorporate the **hierarchical** information and the **co-occurrence** information of labels, which allows models to learn better label representations. Most of the existing work in HTC focuses on modeling *global label hierarchy* and utilizes encoders, such as Tree-LSTM/GCN [7], and Graphormer [10] to learn label representations. The *global label hierarchy* contains all labels in the dataset, which can be divided into target (ground-truth) and non-target (wrong) labels of a text. These methods [7]–[10] utilize the same label hierarchy information for each text, and cannot distinguish the target and non-target labels for a specific text. However, the non-target labels are irrelevant and noisy information [9], which may hurt model performance. Another strand of research utilizes the *local label hierarchy*, which refers to modeling the target labels of each text independently. In order to model the *local label hierarchy*, some researchers [11]–[15] formulated HTC as a sequence generation problem and applied the sequence-to-sequence (Seq2Seq) framework to predict label sequences, which have achieved great success.

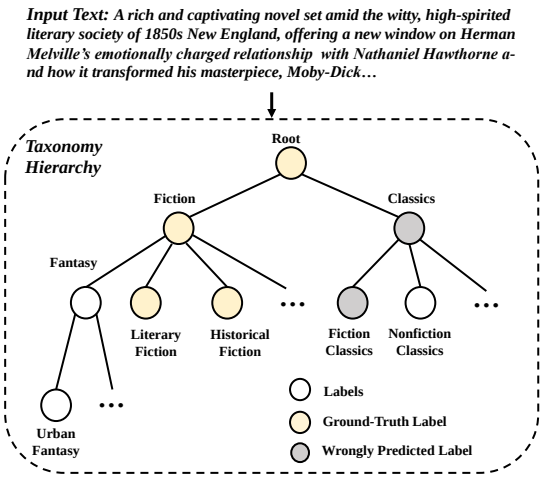


Fig. 1. A hierarchical text classification example. The figure shows a common problem in HTC: wrongly predicting a label node usually leads to its descendant nodes are also wrongly predicted.

In the Seq2Seq frameworks for HTC, the target labels of each text will be flattened to a linear label sequence with a specific order via sorting [11], Depth-First Search (DFS) [14] or Breadth-First Search (BFS) [15]. In each step of prediction phase, these methods generate the next label based on the text sequence and labels previously generated. Therefore, the model can easily capture the level and path dependency information among labels [15]. For example, in Figure 1, the target labels (yellow color) can be flattened to a label sequence [“Fiction” “Literary Fiction” “Historical Fiction”] by BFS

Jingsong Yan, Haibin Chen, Junhao Zheng and QianliMa are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510006, China (e-mail: yanjingsong12@gmail.com, haibin_chen@foxmail.com, junhaozheng47@outlook.com, qianlima@scut.edu.cn).

Piji Li is with College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, 211106, China (e-mail: pjli@nuaa.edu.cn).

Corresponding authors: Piji Li and Qianli Ma.

strategy. After the label "Fiction" is generated, the probability of predicting "Literary Fiction" in the next decoding step will be higher. The reason is that the generative model learns the specific order to generate label sequences during training. However, the order between labels does not naturally exist. Intuitively, solely learning label representations in a fixed order cannot fully explore the mutual dependencies among labels. On the one hand, the target labels at the same level (*e.g.,* "Literary Fiction" and "Historical Fiction") exhibit co-occurrence relationships and are equal in priority. On the other head, for enhanced label representation learning, a node should encompass hierarchical information from its ancestor and descendant nodes. However, in Figure 1, when predicting "Fiction", the model can not utilize the information and clues from the subsequent child labels ("Literary Fiction" and "Historical Fiction"). Although Kervy *et al.* [13] try to introduce the reverse order of the BFS sequence as an auxiliary synthetic task to conduct bi-directional dependency of labels, the performance improvement is not significant. Essentially and intuitively, we claim that **the generation of the current label needs to incorporate the hierarchical and co-occurrence information of all other target labels**, which is neglected by the existing generative methods.

Moreover, the generative methods usually suffer from the **error accumulation problem** [16], i.e., if the preceding labels are not correct, they will have negative impacts on the subsequent predictions. In other generation tasks such as machine translation and text summarization, the error accumulation always causes degenerate behaviors such as repetition, a lack of diversity, dullness, and incoherence [17]. In HTC, the error accumulation problem may cause the model to further predict the descendant nodes of a wrong label after the wrong label node is predicted. In other words, the wrongly predicted labels predicted by generative HTC models usually correspond to at least one subpath or node-path in the global label hierarchy. For instance, in Figure 1, the wrongly predicted labels "Classics" and "Fiction Classics" correspond to one node-path. As the non-target label "Classic" is predicted, the prediction of the following label will incorporate the information of the label "Classics" and will likely be predicted to be the descendant label "Fiction Classics" of "Classics".

To address the above-mentioned issues, we propose a new framework named sequence-to-label (Seq2Label) with a random generative way to learn label hierarchy for HTC. Instead of flattening target labels into a sequence with a specific and static order, we design a **Label Sequence Random Shuffling (LSRS)** mechanism to capture the hierarchical and co-occurrence information of target labels of each text. Specifically, for each epoch in training, we randomly shuffle the order of the label sequence, which means a text will be mapped to several different label sequences during the training phase. In this way, each label in the sequence has a certain probability of appearing at an arbitrary position in the sequence. For a label l in target labels, the subsequence before it is random in each epoch of training. The subsequence not only may contain labels that are hierarchically dependent on l, but it may contain labels that are unrelated to l. For example, in Figure 2, the local label hierarchy can be flattened into different label sequences. For label l_2, the subsequences before it can be "l_8" (the first row) or "$l_1 l_5 l_3 l_7$" (the second row), in which l_5 is hierarchically dependent on it and l_1, l_3, l_7, l_8 are unrelated to it. Therefore, all target labels beside l_2 contribute to the learning of the label representation of l_2. And the label representation of l_2 can incorporate both hierarchical and co-occurrence information with other target labels.

In addition, we propose a **Hierarchy-aware Negative Sampling (HNS)** strategy to alleviate the error accumulation problem. The principal idea is to give more punishments to the non-target labels with the same parent or the same level as target labels, and we call these non-target labels negative labels. Firstly, for a label sequence, we construct its negative label sequence. Specifically, for a label in the label sequence, we randomly sample a negative label from its non-target sibling label set or non-target label set with the same level. Then, we propose a negative label-aware loss to give more punishments to the negative label. In this way, our model can better distinguish the target and the negative label.

The main contributions of our work can be summarized as follows:

- We propose a sequence-to-label (Seq2Label) framework, in which the LSRS mechanism is a better way to model the hierarchical and co-occurrence information between the target labels in each text than the way with a specific order. This demonstrates the order does not matter.
- To alleviate the error accumulation problem, we propose an HNS strategy with a negative label-aware loss to better distinguish target labels and negative labels.
- We conduct experiments on three datasets and show that our method achieves state-of-the-art performances in HTC. Additionally, our visualizations of label representations show how the LSRS mechanism captures label hierarchical information.

II. Related Work

A. Hierarchical Text Classification

Hierarchical text classification (HTC) is a challenge task due to its large-scale, imbalanced, and structured label hierarchy [18]. Existing work of HTC can be divided into two categories: discriminative methods and generative methods.

1) Discriminative Methods: The discriminative methods also can be divided into local and global methods [7]. Most of the local methods [19]–[21] tend to build multiple local classifiers and integrate the results of classifiers to get the final classification results. Some works [22], [23] transfer the parent classifier to binary classifiers at lower levels and fine-tune it on the child category classification task. Different from the local methods, global methods construct only one classifier. Early global methods regard HTC as a flat multi-label classification problem [24] while neglecting the label hierarchy. Recent state-of-the-art approaches focus on incorporating text and the global labels representation. Zhou *et al.* [7] proposes an effective hierarchy-aware global model that extracts label-wise text features with hierarchy encoders based on prior hierarchy information. Chen *et al.* [8] considers the text-label semantics matching relationship and formulates HTC as

a semantic matching problem. Wang *et al.* [10] constructs apositive text sample with label hierarchy and proposes a Hierarchy-Guided Contrastive Learning to obtain hierarchy-aware text representation for HTC. Later on, considering the huge gap between pretrained masked language model BERT [25] and classification tasks with sophisticated label hierarchy, Wang *et al.* [26] designed a hierarchy-aware prompt tuning (HPT) method to sovle it.

2) Generative Methods: The generative methods consider modeling the *local label hierarchy* and utilize a sequence-to-sequence framework to generate label sequence with a specific order in an autoregressive manner [11]–[15]. Yang *et al.* [11] firstly views the multi-label classification task as a sequence generation problem, and flattens the local label hierarchy into a label sequence via sorting. However, they neglect the dependence of hierarchical labels. Later on, Yu *et al.* [14] and Huang *et al.* [15] flatten the local label hierarchy using DFS and BFS to capture the hierarchical information, respectively. Nonetheless, these generative methods ignore that the prediction of a label needs to incorporate the information of all target labels, and also suffer from the error accumulation problem. We propose a Label Sequence Random Shuffling mechanism and a Hierarchy-aware Negative Sampling strategy to solve the above-mentioned issues.

B. Sequence-to-Sequence Framework

The sequence-to-sequence (Seq2Seq) framework has been long studied in the natural language generation (NLG) field to tackle various tasks, such as machine translation [27], [28], speech recognition [29], text summarization [30], dialogue generation [31] etc. Some researchers also use Seq2Seq to conduct various natural language understanding (NLU) tasks, including aspect-based sentiment analysis [32], named entity recognition [33], multi-label text classification [11]. On the basis of the tremendous success of the Seq2Seq model in NLU, we transform HTC into a hierarchical label sequence generation problem and propose a sequence-to-label (Seq2Label) framework for HTC. In this paper, we use the pre-trained sequence-to-sequence model BART [34] as our backbone. The BART-Base model contains a 6-layer bidirectional encoder and a 6-layer autoregressive decoder. It is worth noting that other sequence-to-sequence pre-trained models such as T5 [35] can also be applied in our architecture.

III. Problem Definition

HTC tasks can be formulated as:

$$\mathcal{F}(\mathcal{X}, \mathcal{T}) \to \mathcal{L} \tag{1}$$

where $\mathcal{X} = \{X_1, \ldots, X_m\}$ is a text set, X_i is a text sequence, $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ is a taxonomic hierarchy predefined by dataset, $\mathcal{L} = \{L_1, \ldots, L_m\}$ is the aligned sequence of target label set of $\mathcal{X}$. The taxonomic hierarchy $\mathcal{T}$ contains a set of labels (nodes) $\mathcal{V}$ and parent-child relationships $\mathcal{E}$ between nodes, where the latter satisfies asymmetric, anti-reflexive, and transitive [6]. Given a text sequence $X = [x_1, \ldots, x_n]$, the goal is to learn the best model to predict the target label set $L = \{l_1, \ldots, l_k\}$ efficiently, where x_i is a word, n is the number of words and k is the number of target labels.

IV. Methodology

In this section, we will describe the details of our sequence-to-label (Seq2Label) model. Figure 2 shows the overall architecture of our proposed model. We ignores the start-of-token "<s>" and end-of-token "</s>" in our equations for simplicity.

A. Hierarchical Label Sequence Generation

The label sequence of HTC can be generated as follow:

$$P(Y \mid X) = \prod_{t=1}^{k} P\left(y_t \mid X, Y_{<t}\right) \tag{2}$$

where $X = [x_1, \ldots, x_n]$ is the input text sequence, $Y = [y_1, \ldots, y_k]$ is the target labels sequence. We use BART [34] model as the backbone of our framework, which consists of two components: Text Sequence Encoder and Label Sequence Decoder.

Text Sequence Encoder is applied to encode the input text sequence X into matrix H^e, which is represented as follows:

$$\mathbf{H}^e = \mathrm{Encoder}(X) \tag{3}$$

where $\mathbf{H}^e \in \mathbb{R}^{n \times d}$, n is the length of the input sequence and d is the dimension of the hidden state.

Label Sequence Decoder is an autoregressive decoder which is to get the label probability distribution $\mathbf{p}_t = P\left(y_t \mid X, Y_{<t}\right)$ for each step, and the last hidden state at step t can be calculated by:

$$\mathbf{h}_t^d = \mathrm{Decoder}(\mathbf{H}^e; \hat{Y}_{<t}) \tag{4}$$

where $\mathbf{h}_t^d \in \mathbb{R}^d$ and $\hat{Y}_{<t} = [\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{t-1}]$ is the label sub-sequence that has been predicted before step t. Then the last hidden state $\mathbf{h}_t^d$ is fed into a linear layer, and a softmax function is used for calculating the probability distribution.

$$\mathbf{z}_t = \mathbf{W}^d \mathbf{h}_t^d \tag{5}$$

$$\mathbf{p}_t = \mathrm{softmax}\left(\mathbf{z}_t\right) \tag{6}$$

where $\mathbf{W}^d \in \mathbb{R}^{m \times d}$ is linear layer learnable parameters, $\mathbf{z}_t \in \mathbb{R}^m$ is the output of the linear layer.

During the inference, we use an autoregressive manner to generate the target label sequence. Considering the efficiency of decoding, we choose greedy search as the decoding strategy of our model. Greedy search simply selects the label with the highest probability as its next label:

$$y_t = \arg\max_y P(y|X, \hat{Y}_{<t}) \tag{7}$$

The decoding procedure will end when the token "</s>" occurs. The phase of training will be discussed in Section IV-D2.

B. Label Flattening

In this section, we will flatten a *local label hierarchy* into a label sequence. It is very crucial to retain the hierarchical information of labels in the flattening process. Yu *et al.* [14] and Huang *et al.* [15] have proved that a label sequence with a specific order is beneficial for incorporating hierarchical information. To this end, we first sort the label sequence with

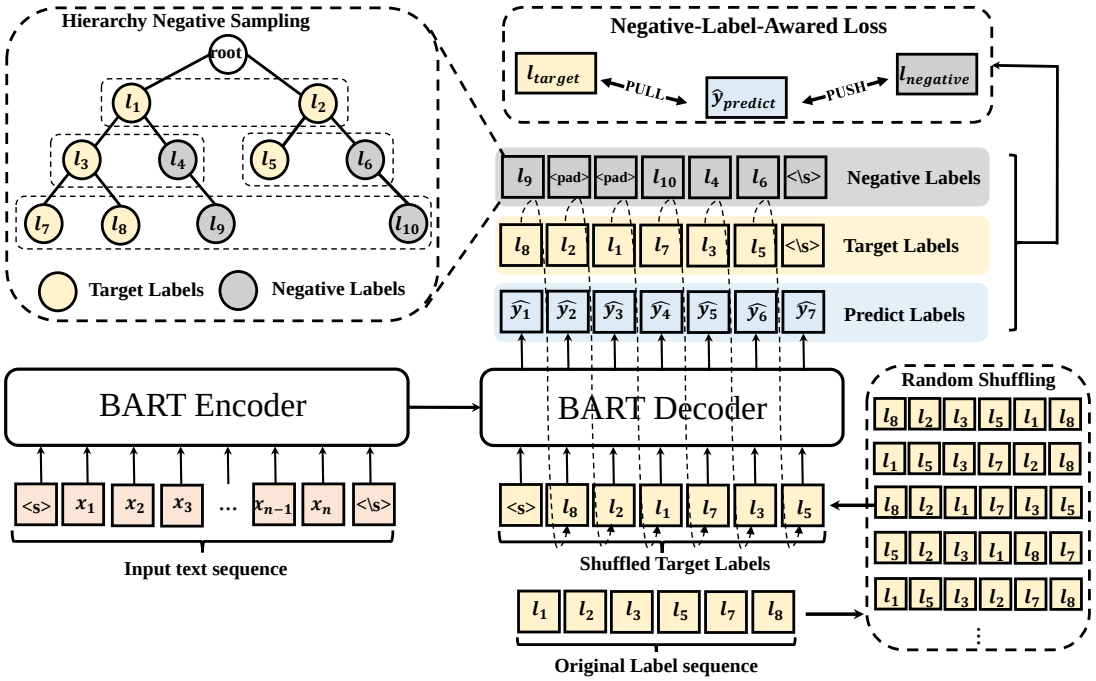


Fig. 2. The overview of the Seq2Label model. The encoder encodes text sequence, and the label decoder generates labels autoregressively. "<s>", "</s>" and "<pad>" are the predefined start-of-sentence, end-of-sentence and padding tokens in BART, respectively. The original label sequence is firstly converted to a shuffled label sequence by random shuffling and then input to the BART decoder. A negative label sequence is obtained by the Hierarchy-aware Negative Sampling strategy and is appended to a negative label-aware loss for the model optimization.

a specific order which can represent hierarchical information directly. A simple way is to utilize Breadth-First Search (BFS) [36] to convert *local label hierarchy* to a label sequence. However, BFS doesn't make use of the label relationships effectively. More specifically, when generating high-level labels, the information of low-level labels can not be utilized. We give an example in Figure 2 for demonstration. The *local label hierarchy* is flattened as a label sequence "$l_1l_2l_3l_5l_7l_8$" by BFS. When predicting label "l_3", the information of previously generated labels can be captured to facilitate the prediction, but the information of labels after l_3 is ignored due to the autoregressive generation manner. In this paper, we just use BFS for label sequence initialization. And we will use the backbone with BFS for later comparison.

C. Label Sequence Random Shuffling

To model *local label hierarchy* and capture the hierarchical and co-occurrence relationships between the corresponding labels of each text sample, we randomly shuffle the input label sequence. For example, in Figure 2, the input label sequence is shuffled as "$l_8l_2l_1l_7l_3l_5$". It is worth noting that the same input of label sequence may obtain shuffled lable sequence with different order after shuffling randomly, as depicted in Figure 2. The process of random shuffling can be formulated as:

$$Y = \text{Shuffle}(Y) \tag{8}$$

where Y is a label sequence. For a text sample, we define the length of its corresponding label sequence is N, and there are $N!$ different permutations, which means that we have $N!$ different <text, label sequence> pairs for training. Actually, the methods Kervy *et al.* [13] proposed can be viewed as a special case of our random shuffling strategy, which includes both BFS and its reversed permutations.

D. Hierarchy-aware Negative Sampling

To address the error accumulation problem, we expect to avoid predicting the non-target labels, which label information will lead to more prediction errors. For each label in the label sequence, to distinguish target labels and non-target labels, the straight way is to give all non-target labels more punishment. However, we empirically find that there is a parent-child or sibling relationship between the incorrectly predicted labels and the target labels. Therefore, it is more effective to penalize those non-target labels that have a parent-child or sibling relationship with the target labels.

1) Negative Label Sequence Constructing: Specifically, given a label sequence $Y = [y_1, \ldots, y_k]$, we construct a negative label sequence $Y^{neg} = [y_1^{neg}, \ldots, y_k^{neg}]$, where y_i^{neg} is a negative label which represents a non-target label with similar hierarchical semantics to target label y_i. For each label y in label sequence Y, it has a sibling label set $Y_y^{sibling}$ and a level label set Y_y^{level}. Each label in $Y_y^{sibling}$ has the same parent node label as y. Each label in Y_y^{level} has the same level as y in the taxonomic hierarchy. We randomly sample a negative label for y from its non-target sibling label set $Y_y^{neg\_sibling}$ or its non-target the same level label set $Y_y^{neg\_level}$, where these two sets are defined as:

$$Y_y^{neg\_sibling} = Y_y^{sibling} - Set(Y) \tag{9}$$

$$Y_y^{neg\_level} = Y_y^{level} - Set(Y) \tag{10}$$

Algorithm 1 Hierarchy-aware Negative Sampling Strategy

Input: Target label sequence $Y = [y_1, y_2, \ldots, y_k]$
Output: Negative sample label sequence $Y^{neg} = [y_1^{neg}, y_2^{neg}, \ldots, y_k^{neg}]$

Initialization: $i = 1$, Y^{neg} = []
while $i \leq k$ **do**
  Use Equation (9)(10) to compute $Y_i^{neg\_sibling}$, $Y_i^{neg\_level}$ respectively
  if $Y_i^{neg\_sibling} \neq \varnothing$ **then**
    $y_i^{neg} = \text{Sample}\left(Y_i^{neg\_sibling}\right)$
    Y^{neg}.append(y_i^{neg})
  else if $Y_i^{neg\_level} \neq \varnothing$ **then**
    $y_i^{neg} = \text{Sample}\left(Y_i^{neg\_level}\right)$
    Y^{neg}.append(y_i^{neg})
  else
    Y^{neg}.append(<pad>)
  end if
  $i = i + 1$
end while
return Y^{neg}

where $Set(Y)$ is target label set. We present the negative label sequence construction method in Algorithm 1. For example, in Figure 2, the $Y_{l_1}^{neg\_level} = \varnothing$, so the negative label of l_1 is "<pad>". The $Y_{l_7}^{neg\_sibling} = \varnothing$ and the $Y_{l_7}^{neg\_level} = \{l_9, l_{10}\}$, so the negative label of l_7 is l_{10} sampled from $\{l_9, l_{10}\}$.

2) Negative label-aware Loss: After constructing the negative label sequence, we will prevent the target label from being predicted as its negative label during the optimization process. During the training phase, most generation tasks feed the last hidden state into a linear layer, and a softmax function is used for calculating the probability distribution:

$$\mathbf{p}_{ti} = \text{softmax}\left(\mathbf{z}_t^i\right) = \frac{e^{\mathbf{z}_t^i}}{\sum_{j=1}^{n} e^{\mathbf{z}_t^j}} \tag{11}$$

where $\mathbf{z}_t^i$ is the value of the i-th dimension of $\mathbf{z}_t$ and $\mathbf{p}_{ti}$ is the probability of label i at step t. However, such function treats each label equally, which cannot represent the difference between negative labels and other non-target labels. To more penalize the negative label, we introduce a hyperparameter λ to reformulate the softmax function:

$$\mathbf{p}_t^{I_t} = \frac{e^{\mathbf{z}_t^{I_t}}}{\sum_{j=1}^{n} e^{\mathbf{z}_t^j} + \lambda e^{\mathbf{z}_t^{I_t^{neg}}}} \tag{12}$$

where I_t is the index of label y_t in label set $\mathcal{V}$. After obtaining the probability distribution, we use the teacher forcing to train our model and the negative log-likelihood to optimize the model.

$$\mathbf{loss} = -\frac{1}{MK} \sum_{i=1}^{M} \sum_{t=1}^{K} \log\left(\mathbf{p}_{it}^{I_t}\right) \tag{13}$$

where K is the length of the label sequence, and $\mathbf{p}_{it}^{I_t}$ is the probability of the target label at step t of the i-th text.

V. Experimental Settings

A. Dataset

To evaluate the classification effect of our model, we experiment on three widely used datasets for HTC, including Web Of Science (WOS) [20], RCV1-V2 [2], BlurbGenreCollection (BGC)[1]. The description of the three datasets is illustrated in the Supplementary Material.

B. Evaluation Metrics

To facilitate the comparison with the experimental results of other methods, we use standard evaluation metrics [37] Micro-F1 and Macro-F1 to measure the experimental results. The Micro-F1 computes a global average F1 score by counting the sums of the true positives (TP), false negatives (FN), and false positives (FP), while the Macro-F1 score is computed by taking the arithmetic mean of all the per-class F1 scores.

C. Comparison Methods

The experimental results are compared with other start-of-the-art models including discriminative methods: HAN [38], TextCNN [39], TextRCNN [40], TextRNN [41], HR-DGCNN-3 [42], HFT(M) [22], Htrans [23], HMCN [43], HiLAP-RL [18], HE-AGCRCNN [44], HiAGM [7], HiMatch [8], HTCInfoMax [9], HGCLR [10], HPT [26], and generative methods SGM [11], Seq2Tree [14], PAAM-HiA-T5 [15], BART [34]. The description of the main comparison models are listed in the Supplementary Material.

D. Implement Details

The backbone pre-trained model we adopt is BART-base [34]. The maximum length of the token inputs of the encoder is set as 800, and the maximum length of the label sequence of the decoder in WOS, RCV1-V2, and BGC are 6, 20, and 16, respectively. For each dataset, we create a label vocab to replace the vocab of the BART decoder, where the label vocab contains all labels in the taxonomic hierarchy, sequence start token "<s>", sequence end token "</s>" and padding "<pad>". The batch size is set to 16, and the epoch of training is set to 100. The hyperparameters λ in WOS, RCV1-V2, and BGC are set to 9, 5, and 9, respectively. We optimize the model with AdamW [45] with a learning rate of 2e-5. For training, we train the model with a training set and evaluate it with the validation set in each epoch. We update the model if the validation set achieves better Micro-F1 or Macro-F1 scores. For validation and inference, we use greedy search to generate label sequences. Our experiments are all conducted on a RTX 3090 GPU.

VI. Results and Discussions

A. Main Results

The main experimental results on WOS, RCV1-V2 and BGC compared to other state-of-the-art models are shown in Table I, Table II and Table III, respectively. Compared

[1]The dataset is obtained from https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/blurb-genre-collection.html

TABLE I
THE EXPERIMENTAL RESULTS ON WOS COMPARED TO OTHER STATE-OF-THE-ART MODELS

Model	Micro-F1	Macro-F1
Discriminative Model		
TextRNN	77.94	69.65
TextCNN	82.00	76.18
TextRCNN	83.55	76.99
HiAGM	85.82	80.28
HiMatch	86.20	80.53
HTCInfoMax	85.58	80.05
BERT (Vanilla Fine Tuning)	86.26	80.58
BERT+HiAGM	86.04	80.19
BERT+HTCInfoMax	86.30	79.97
BERT+HiMatch	86.70	81.06
HGCLR	87.11	81.20
HPT	87.16	81.93
Generative Model		
SGM-T5	85.83	80.79
Seq2Tree	87.20	**82.50**
PAAM-HiA-T5	**90.36**	81.64
BART+BFS	86.70	81.23
Seq2Label (T5-based)	87.15	81.89
Seq2Label (Ours)	87.31	81.86

TABLE II
THE EXPERIMENTAL RESULTS ON RCV1-V2 COMPARED TO OTHER STATE-OF-THE-ART MODELS

Model	Micro-F1	Macro-F1
Discriminative Model		
TextCNN	76.60	43.00
TextRCNN	81.57	59.25
HR-DGCNN-3	76.18	43.34
HFT(M)	80.29	51.40
Htrans	80.51	58.49
HMCN	80.80	54.60
HAN	75.30	40.60
HiLAP-RL	83.30	60.10
HiAGM	83.96	63.35
HTCInfoMax	85.58	80.05
HiMatch	84.73	64.11
BERT (Vanilla Fine Tuning)	85.65	67.02
BERT+HiAGM	85.58	67.93
BERT+HTCInfoMax	85.53	67.09
BERT+HiMatch	86.33	68.66
HGCLR	86.49	68.31
HPT	87.26	69.53
Generative Model		
SGM	77.30	47.49
SGM-T5	84.39	65.09
Seq2Tree	86.88	70.01
PAAM-HiA-T5	87.22	70.02
BART+BFS	86.43	68.81
Seq2Label (T5-based)	87.03	70.17
Seq2Label (Ours)	**87.35**	**70.60**

with BART, a strong baseline that builds the level dependency by BFS, Seq2Label outperforms it on three datasets with a significant improvement, which validates that the improvement of our method is mainly brought by our design on the framework rather than BART.

Compared with all discriminative models with or without pretrained model, Seq2Label achieves new state-of-the-art results on all three datasets. This is because Seq2Label learns only the information between the target labels and ignores the noisy information of the non-target labels. Table V shows that BERT-based models such as BERT+HiAGM, BERT+HiMatch have slightly more parameters than Seq2Label, which means the comparison between BERT-based methods and Seq2Label is fair.

TABLE III
THE EXPERIMENTAL RESULTS ON BGC COMPARED TO OTHER STATE-OF-THE-ART MODELS

Model	Micro-F1	Macro-F1
Discriminative Model		
HMC-Capsule	74.37	-
HiAGM	77.22	57.91
HiMatch	76.57	58.34
BERT+HiMatch	78.89	63.19
Generative Model		
SGM-T5	77.84	60.91
Seq2Tree	79.72	63.96
BART+BFS	79.59	65.12
Seq2Label (T5-based)	**80.61**	66.44
Seq2Label (Ours)	80.54	**66.76**

Generative methods have been proven effective for HTC due to their powerful ability to model the local label hierarchy. On the simple dataset WOS, Seq2Label achieves competitive results. Although our model does not outperform T5-based models Seq2Tree and PAAM-HiA-T5, its Micro-F1 score and Macro-F1 score are better than Seq2Tree and PAAM-HiA-T5, respectively. Specially, for each text in the WOS dataset, the length of its label sequence is 2 so that the permutations of target labels are only 2!. Therefore, compared with two more complex datasets, our proposed strategies do not achieve significant improvement on this dataset. Moreover, the parameters of Seq2Label are 37% less than T5-based models, which means that there is an advantage in the speed of inference of our model. On more complex datasets RCV1-V2 and BGC, Seq2Label achieves state-of-the-art results. Specifically, the Macro-F1 score on BGC improves by 4.4%. This demonstrates our method is more effective at modeling complex local label hierarchy. For each text sample on RCV1-V2 or BGC, our LSRS strategy provides more different <text, label sequence> pairs for training due to the number of target labels permutations is always greater than 3!, in which permutations include the label sequence obtained by BFS, DFS and other orders. Therefore, Seq2Label captures not only the hierarchical information between labels, but also the co-occurrence information between labels.

B. Ablation Analysis

The ablation study results on three datasets are shown in Table IV, the details are as follows:

"w/o LSRS & HNS": Baseline BART, the local label hierarchy is flattened by BFS. It's also called "BART+BFS".

"w/o HNS": Not using the hierarchy-aware negative sampling strategy. It's also called "BART+LSRS".

TABLE IV
ABLATION STUDY

Ablation Models	WOS		RCV1-V2		BGC	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
w/o LSRS & HNS	86.70	81.23	86.43	68.81	79.59	65.12
w/o HNS	87.21	81.57	87.19	69.62	80.34	66.17
w/o LSRS	86.92	81.51	86.85	69.50	80.14	66.39
r.p. Random Sample	87.06	81.72	87.33	70.30	80.37	66.49
Full Model	**87.31**	**81.86**	**87.35**	**70.60**	**80.54**	**66.76**

TABLE V
NUMBER OF PARAMETERS OF COMPARABLE MODELS

Model	Parameters
BERT+HiAGM	143M
BERT+HiMatch	153M
T5-based model	220M
Seq2Label	**139M**

"w/o LSRS": Not using the label sequence random shuffling mechanism.

"r.p. Random Sample": Replace the hierarchy-aware negative sampling with random negative sampling the non target labels.

1) Ablation Study and Analysis on Label Sequence Random Shuffling: It is evident that "w/o HNS" greatly outperforms the "w/o LSRS & HNS" both in Micro-F1 and Macro-F1 on three datasets. Compared with "w/o LSRS & HNS" on BGC, "w/o HNS" boosts Micro-F1 by 0.9% and achieves a significant 1.6% improvement in Macro-F1. In addition, In the case of joining HNS, Seq2Label also outperforms "w/o LSRS" in two metrics. These results suggest that LSRS is powerful in modeling local label hierarchy. More detail on the effect of LSRS will be discussed in Section VI-C and Section VI-D.

2) Ablation Study and Analysis on Hierarchy-aware Negative Sampling: "w/o LSRS" greatly increases Micro-F1 and Macro-F1 especially in Macro-F1 compared with "w/o LSRS & HNS". In the case of joining LSRS, Seq2Label also outperforms "w/o HNS" in two metrics. With the introduction of HNS, the Macro-F1 scores improve significantly on the RCV1-V2 and BGC datasets, while only slightly on the WOS dataset. The reason is that HNS alleviates the error accumulation problem leading to accuracy degradation of low-level sparse labels. The effect of HNS was more pronounced on samples with more complex local hierarchy. Due to Macro-F1 equally weighting all labels and being more sensitive to lower-level sparse labels, the improvement of the Macro-F1 score is more significant than the Micro-F1 scores on the RCV1-V2 and BGC datasets, respectively. More detail on the effect of HNS will be discussed in Section VI-E and Section VI-F.

C. Effect of Label Sequence Random Shuffling

Table VI shows the comparison of experimental results of baseline BART with different orders on RCV1-V2. The "Random", "BFS" and "DFS" are strategies for label sequence initialization, and label sequences obtained by these strategies remain in the training phase. The hierarchical initialization strategies "BFS" and "DFS" outperform "Random" by a huge

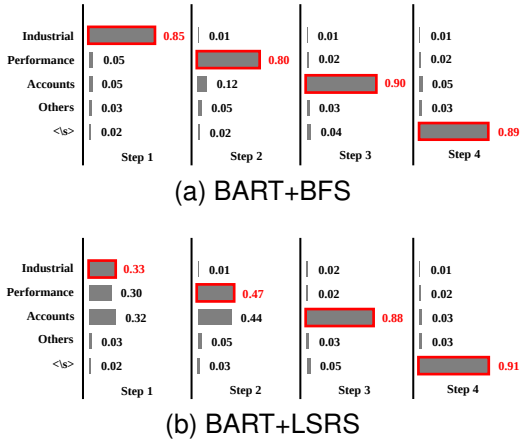


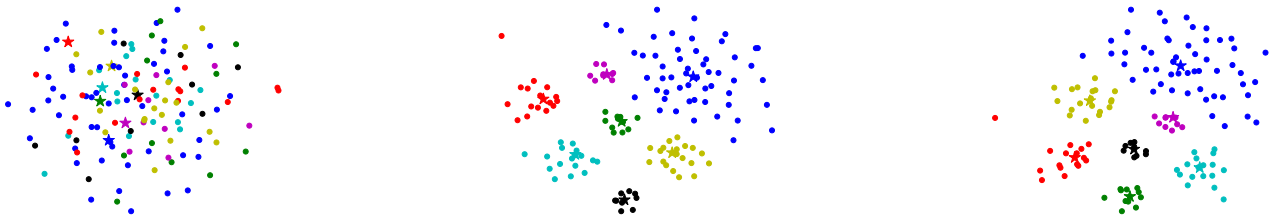
Fig. 3. Comparison between the outputs probability distribution of the (a) "BART+BFS" and the (b) "BART+LSRS".

TABLE VI
COMPARISON OF EXPERIMENTAL RESULTS OF BASELINE WITH DIFFERENT ORDER ON RCV1-V2

Model	Micro-F1	Macro-F1
BART+Random	85.54	67.00
BART+BFS	86.43	68.81
BART+DFS	86.69	68.78
BART+LSRS	**87.19**	**69.62**

margin in Micro-F1 and Macro-F1 due to the success of building the level dependency and capturing the hierarchical information [14], [15]. Compared with models with specific order "BART+BFS" and "BART+DFS", "BART+LSRS" achieves significant improvement in two metrics, which experimentally demonstrates the order does not matter. The counter-intuitive view that "the order of the labels does not matter" is based on the intuitive idea that "the generation of the current label needs to incorporate the hierarchical and co-occurrence information of all other target labels", and the improvement proves that this intuitive idea is effective. As illustrated in Section IV-C, a text sample with N target labels can potentially generate $N!$ different <text, label sequence> pairs for training, including but not limited to "BFS" and "DFS" pairs. Therefore, compared with "Random", LSRS helps the model capture not only the hierarchical information between labels, but also the co-occurrence information between labels. Based on the above analysis, LSRS could be viewed as a special form of data augmentation. The specificity is discussed in Supplementary Material.

To further analyze the effect of LSRS, we show the probabil-



(a) BART+BFS (b) BART+LSRS (c) Full Model

Fig. 4. T-SNE visualization of the label representation on the WOS dataset. "⋆" means the most top level labels in the taxonomy hierarchy. "•" means all labels except the top-level labels. Dots have the same color as a star means the star is the primogenitor of these dots.

ity distributions of "BART+BFS" and "BART+LSRS" in each decoding step in Figure 3. When using BFS, the probability of the label predicted at each decoding step is significantly higher than others. When introducing LSRS, however, the probabilities of the target labels yet to be predicted at each decoding step are comparable and much higher than other labels. For example, in decoding step 2 of Figure 3b, the probabilities of the target labels "Performance" and "Accounts" to be predicted are relatively close to each other and much higher than the probabilities of "Industrial", "Others" and "</s>". Note that "Others" denotes other labels in the hierarchy which do not appear in Figure 3. The main reason is that shuffling the sequence order encourages each label to appear at each position in the sequence. After multiple training epochs, LSRS helps the model learn the co-occurrence relationships between target labels.

D. T-SNE Visualization of Label Sequence Random Shuffling

The LSRS mechanism seems counter-intuitive in raising the performance of a hierarchy, as it does not seem to make use of the hierarchical information of the target labels explicitly. We claim that LSRS incorporates hierarchical information implicitly. We view weight matrix W_d as label representations and plot their T-SNE projections in different versions of our model in Figure 4. Since a label and its father label should be classified simultaneously, the representation of a label and its father should be similar [10], which means that labels with the same father should be clustered towards the father label. Comparing with Figure 4a and Figure 4b, the label representation of "BART+BFS" is scattered, whereas "BART+LSRS" are clustered, which demonstrates that our LSRS can learn a hierarchy-aware representation implicitly. The comparison of Figure 4b and Figure 4c shows that HNS does not have a negative effect on LSRS.

E. Effect of Hierarchy-aware Negative Sampling Strategy

To further illustrate the effect of the hierarchy-aware negative sampling strategy, we replace the negative sampling strategy in HNS with random negative sampling. As shown in Table IV, the results of "r.p. Random Sample" and the full model on two metrics are better than "w/o HNS", which demonstrates the negative sampling strategy is an effective strategy for HTC. Compared with "r.p. Random Sample", the full model gets better scores of two metrics on three datasets, which demonstrates hierarchy-aware negative sampling is a better negative sampling strategy. The reason is that HNS gives more punishment to those non-target labels that have a parent-child or sibling relationship with the target labels, as Section IV-D mentioned.

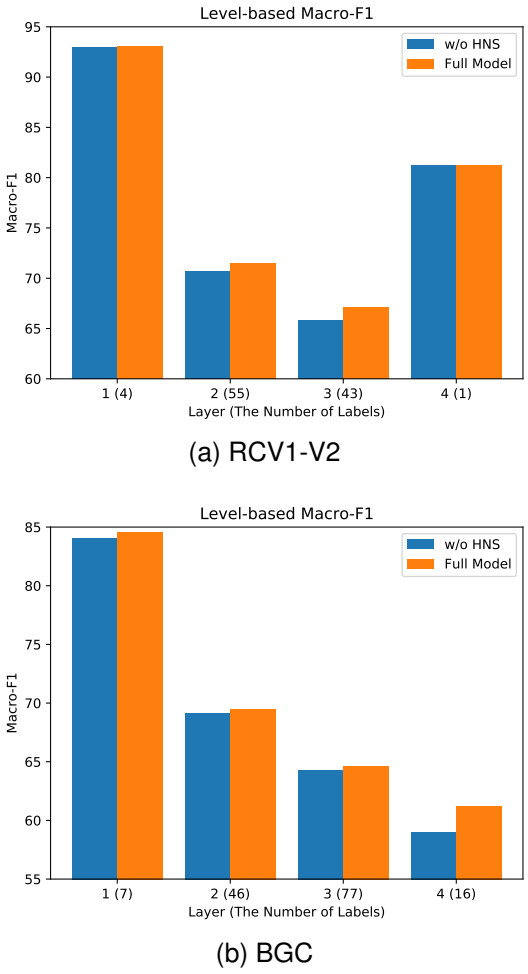


Fig. 5. Macro-F1 scores of label clusters grouped by depth in the hierarchy on (a) RCV1-V2 and (b) BGC.

To illustrate how HNS alleviates the error accumulation problem, we analyze performance on different label clusters grouped by depth in the hierarchy for "w/o HNS" and the

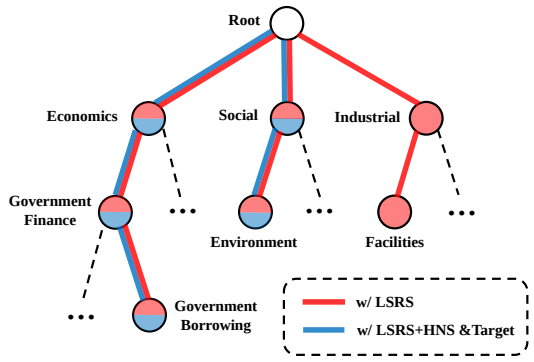


Fig. 6. An RCV1-V2 example generated in two ways. Labels (nodes) with the same color are generated by the same way.

full model on RCV1-V1 and BGC, as shown in Figure 5. For BGC in Figure 5b, compared with "w/o HNS", the full model achieves performance improvement on all levels, especially on the lower level (level 4). However, for RCV1-V2 in Figure 5a, the Macro-F1 score of level 4 has almost no improvement, which is counter-intuitive. Note that the number of labels in level 4 is only one and the label is called "C1511". We find that the number of training samples containing the "C1511" label is 1.7% (371/20833) of the total number of training samples, which is sufficient for the model to learn the label representation of "C1511". In level 2 and level 3, the full model improves by 1.1% (71.49/70.68) and 2.0% (67.14/65.85) Macro-F1 scores, respectively, which means fewer or shorter error branches are generated during testing. This demonstrates that HNS can alleviate the error accumulation problem.

F. Case Analysis

In Figure 6, we select a case to further illustrate the effect of HNS. The text is entered into the input of the model w/ and w/o HNS, and two sets of label generation results are obtained. However, without HNS, the model predicts "Industrial" with similar hierarchical semantics to "Economics" or "Social", leading to the prediction of "Industrial"'s descendant labels "Facilities". The model with HNS achieves the same results as target labels. The reason is that when generating labels with the same level as the not-target label "Industrial", HNS gives "Industrial" more punishment during the training phase, thus avoiding the generation of the descendant labels "Facilities" of "Industrial".

VII. Conclusion

In this paper, we propose Seq2Label to learn label hierarchy for HTC. Seq2Label shows a new counter-intuitive view that the order of the labels does not matter. Instead of using only one specific order, we propose a LSRS mechanism, a better way to model the hierarchical and co-occurrence mutual dependency relationships between the target labels in each text. Moreover, we propose an HNS strategy, which effectively alleviates the error accumulation problem. Compared with existing methods, Seq2Label achieves significant improvements on three datasets. In future work, we plan to extend our model to the few-shot or zero-shot learning scenario.

Acknowledgments

We thank the anonymous reviewers for their helpful feedbacks. The work described in this paper was partially funded by the National Natural Science Foundation of China (Grant Nos. 62272173, 61872148), the Natural Science Foundation of Guangdong Province (Grant Nos. 2022A1515010179, 2019A1515010768), the Science and Technology Planning Project of Guangdong Province (Grant No. 2023A0505050106).

References

[1] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He, "A survey on text classification: From shallow to deep learning," *arXiv preprint arXiv:2008.00364*, 2020.

[2] D. D. Lewis, Y. Yang, T. Russell-Rose, and F. Li, "Rcv1: A new benchmark collection for text categorization research," *Journal of machine learning research*, vol. 5, no. Apr, pp. 361–397, 2004.

[3] R. Agrawal, A. Gupta, Y. Prabhu, and M. Varma, "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*, 2013, pp. 13–24.

[4] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-y. Wang, "Representation learning using multi-task deep neural networks for semantic classification and information retrieval," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 912–921. [Online]. Available: https://aclanthology.org/N15-1092

[5] P. Xu and D. Barbosa, "Neural fine-grained entity type classification with hierarchy-aware loss," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, Jun. 2018, pp. 16–25. [Online]. Available: https://aclanthology.org/N18-1002

[6] C. N. Silla and A. A. Freitas, "A survey of hierarchical classification across different application domains," *Data Mining and Knowledge Discovery*, vol. 22, no. 1, pp. 31–72, 2011.

[7] J. Zhou, C. Ma, D. Long, G. Xu, N. Ding, H. Zhang, P. Xie, and G. Liu, "Hierarchy-aware global model for hierarchical text classification," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 1106–1117. [Online]. Available: https://aclanthology.org/2020.acl-main.104

[8] H. Chen, Q. Ma, Z. Lin, and J. Yan, "Hierarchy-aware label semantics matching network for hierarchical text classification," 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)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4370–4379. [Online]. Available: https://aclanthology.org/2021.acl-long.337

[9] Z. Deng, H. Peng, D. He, J. Li, and P. Yu, "HTCInfoMax: A global model for hierarchical text classification via information maximization," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Online: Association for Computational Linguistics, Jun. 2021, pp. 3259–3265. [Online]. Available: https://aclanthology.org/2021.naacl-main.260

[10] Z. Wang, P. Wang, L. Huang, X. Sun, and H. Wang, "Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 7109–7119. [Online]. Available: https://aclanthology.org/2022.acl-long.491

[11] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, "SGM: sequence generation model for multi-label classification," in *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, 2018, pp. 3915–3926.

[12] P. Yang, F. Luo, S. Ma, J. Lin, and X. Sun, "A deep reinforced sequence-to-set model for multi-label classification," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5252–5258.

[13] K. Rivas Rojas, G. Bustamante, A. Oncevay, and M. A. Sobrevilla Cabezudo, "Efficient strategies for hierarchical text classification: External knowledge and auxiliary tasks," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 2252–2257. [Online]. Available: https://aclanthology.org/2020.acl-main.205

[14] C. Yu, Y. Shen, and Y. Mao, "Constrained sequence-to-tree generation for hierarchical text classification," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 1865–1869.

[15] W. Huang, C. Liu, B. Xiao, Y. Zhao, Z. Pan, Z. Zhang, X. Yang, and G. Liu, "Exploring label hierarchy in a generative way for hierarchical text classification," in *Proceedings of the 29th International Conference on Computational Linguistics*. Gyeongju, Republic of Korea: International Committee on Computational Linguistics, Oct. 2022, pp. 1116–1127. [Online]. Available: https://aclanthology.org/2022.coling-1.95

[16] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," *arXiv preprint arXiv:1511.06732*, 2015.

[17] K. Arora, L. El Asri, H. Bahuleyan, and J. C. K. Cheung, "Why exposure bias matters: An imitation learning perspective of error accumulation in language generation," in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022, pp. 700–710.

[18] Y. Mao, J. Tian, J. Han, and X. Ren, "Hierarchical text classification with reinforced label assignment," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 445–455. [Online]. Available: https://aclanthology.org/D19-1042

[19] R. Cerri, R. C. Barros, and A. C. De Carvalho, "Hierarchical multi-label classification using local neural networks," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 39–56, 2014.

[20] K. Kowsari, D. E. Brown, M. Heidarysafa, K. Jafari Meimandi, , M. S. Gerber, and L. E. Barnes, "Hdltex: Hierarchical deep learning for text classification," in *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*. IEEE, 2017.

[21] Y. Meng, J. Shen, C. Zhang, and J. Han, "Weakly-supervised hierarchical text classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 6826–6833.

[22] K. Shimura, J. Li, and F. Fukumoto, "Hft-cnn: Learning hierarchical category structure for multi-label short text categorization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 811–816.

[23] S. Banerjee, C. Akkaya, F. Perez-Sorrosal, and K. Tsioutsiouliklis, "Hierarchical transfer learning for multi-label text classification," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 6295–6300. [Online]. Available: https://aclanthology.org/P19-1633

[24] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Denver, Colorado: Association for Computational Linguistics, May–Jun. 2015, pp. 103–112. [Online]. Available: https://aclanthology.org/N15-1011

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "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)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: https://aclanthology.org/N19-1423

[26] Z. Wang, P. Wang, T. Liu, B. Lin, Y. Cao, Z. Sui, and H. Wang, "Hpt: Hierarchy-aware prompt tuning for hierarchical text classification," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 2022.

[27] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations*, 2015.

[28] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, Sep. 2015, pp. 1412–1421. [Online]. Available: https://aclanthology.org/D15-1166

[29] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2016, pp. 4960–4964.

[30] P. Li, W. Lam, L. Bing, and Z. Wang, "Deep recurrent generative decoder for abstractive text summarization," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017, pp. 2091–2100.

[31] J. Li, W. Monroe, A. Ritter, D. Jurafsky, M. Galley, and J. Gao, "Deep reinforcement learning for dialogue generation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 1192–1202. [Online]. Available: https://aclanthology.org/D16-1127

[32] H. Yan, J. Dai, T. Ji, X. Qiu, and Z. Zhang, "A unified generative framework for aspect-based sentiment analysis," 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)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 2416–2429. [Online]. Available: https://aclanthology.org/2021.acl-long.188

[33] H. Yan, T. Gui, J. Dai, Q. Guo, Z. Zhang, and X. Qiu, "A unified generative framework for various NER subtasks," 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)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 5808–5822. [Online]. Available: https://aclanthology.org/2021.acl-long.451

[34] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "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*, 2020, pp. 7871–7880.

[35] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020.

[36] A. Bundy and L. Wallen, "Breadth-first search," in *Catalogue of artificial intelligence tools*. Springer, 1984, pp. 13–13.

[37] S. Gopal and Y. Yang, "Recursive regularization for large-scale classification with hierarchical and graphical dependencies," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 257–265.

[38] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.

[39] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[40] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.

[41] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," *arXiv preprint arXiv:1605.05101*, 2016.

[42] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, and Q. Yang, "Large-scale hierarchical text classification with recursively regularized deep graph-cnn," in *Proceedings of the 2018 world wide web conference*, 2018, pp. 1063–1072.

[43] J. Wehrmann, R. Cerri, and R. Barros, "Hierarchical multi-label classification networks," in *International conference on machine learning*. PMLR, 2018, pp. 5075–5084.

[44] H. Peng, J. Li, S. Wang, L. Wang, Q. Gong, R. Yang, B. Li, S. Y. Philip, and L. He, "Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 6, pp. 2505–2519, 2019.

[45] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2018.