Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.054581

ARTICLE

Leveraging Uncertainty for Depth-Aware Hierarchical Text Classification

Zixuan Wu[1], Ye Wang[1,*], Lifeng Shen[2], Feng Hu[1] and Hong Yu[1,*]

[1]Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing, 400000, China

[2]Division of Emerging Interdisciplinary Areas, Hong Kong University of Science and Technology, Hong Kong, 999077, China

*Corresponding Authors: Ye Wang. Email: wangye@cqupt.edu.cn; Hong Yu. Email: yuhong@cqupt.edu.cn

Received: 01 June 2024 Accepted: 04 August 2024 Published: 12 September 2024

ABSTRACT

Hierarchical Text Classification (HTC) aims to match text to hierarchical labels. Existing methods overlook two critical issues: first, some texts cannot be fully matched to leaf node labels and need to be classified to the correct parent node instead of treating leaf nodes as the final classification target. Second, error propagation occurs when a misclassification at a parent node propagates down the hierarchy, ultimately leading to inaccurate predictions at the leaf nodes. To address these limitations, we propose an uncertainty-guided HTC depth-aware model called DepthMatch. Specifically, we design an early stopping strategy with uncertainty to identify incomplete matching between text and labels, classifying them into the corresponding parent node labels. This approach allows us to dynamically determine the classification depth by leveraging evidence to quantify and accumulate uncertainty. Experimental results show that the proposed DepthMatch outperforms recent strong baselines on four commonly used public datasets: WOS (Web of Science), RCV1-V2 (Reuters Corpus Volume I), AAPD (Arxiv Academic Paper Dataset), and BGC. Notably, on the BGC dataset, it improves Micro-F1 and Macro-F1 scores by at least 1.09% and 1.74%, respectively.

KEYWORDS

Hierarchical text classification; incomplete text-label matching; uncertainty; depth-aware; early stopping strategy

1 Introduction

Hierarchical Text Classification (HTC) is a classic multi-label text classification problem, where labels exhibit a hierarchical structure represented by a tree or directed acyclic graph [1]. Many related tasks represent hierarchy, such as international patent classification [2], product annotation [3], web page categorization [4], and news classification [5]. Accurate HTC helps the system to organize and retrieve information more effectively, provide personalized recommendation services, and improve user experience and operational efficiency. In the real world, one text sample may have multiple labels. As shown in Fig. 1, taking news classification as an example, the text corresponds to labels such as 'News', 'Business', 'Sports' and 'Stock', which usually contain hierarchical dependencies. However, in practical scenarios, some texts cannot be matched to appropriate leaf node labels. The text is matched with the hierarchical labels "News", "Business", "Sports" and "Stock". While, regarding the

leaf node labels “Football” and “Tennis”, the text cannot be assigned to appropriate labels, resulting in the classification stopping at the parent node label “Sports”. In practical applications, existing label hierarchies might not comprehensively cover all possible text topics. This means that some texts may not have corresponding leaf node labels and can only be matched to higher-level parent node labels. Therefore, for texts that cannot be assigned to appropriate leaf node labels, the model should only predict the correct parent node labels. Due to the incomplete match between text data and the hierarchical structure, directly predicting the leaf node labels becomes challenging [6].

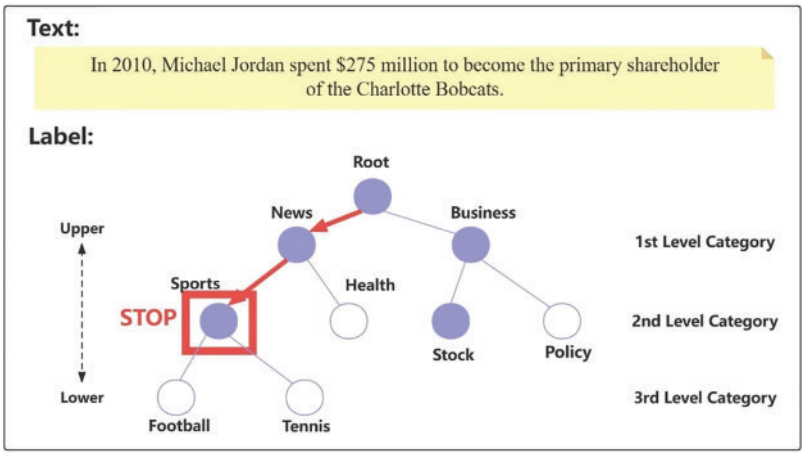

Figure 1: Samples that cannot be assigned to leaf nodes

Global approaches are the mainstream methods in current HTC, regarding the label structure as flattened, which leads to the issue of “incomplete text-label matching” in classification results [7]. The Seq2Seq-based approaches can solve this problem by generating label sequences from the root node to the leaf node along the hierarchical structure [8]. The current Seq2Seq procedures focus on classifying leaf node labels as the end goal, concentrating on improving classification performance [9] and reducing parameter size [10]. However, existing ways overlook certain issues where some texts cannot be perfectly matched to leaf node labels, and not all samples should be classified as leaf nodes. One solution is to introduce multiple nodes within the hierarchical structure, capturing semantic information of texts at different levels, balancing specificity and correctness [11]. Further, other methods leverage probability to provide better classification by normalizing the output of Softmax into probability distributions for each category [7]. Currently, these hierarchical text classification (HTC) methods focus on extracting label features to achieve better hierarchical representations. They flatten the hierarchical structure for label prediction, which may result in inconsistent predicted labels due to ignoring the hierarchy. Moreover, errors can affect subsequent label predictions because errors in parent node labels propagate step-by-step to leaf node labels. Accumulated errors can significantly impact the model’s performance.

To address the aforementioned issue, we propose a model (DepthMatch[1]) leveraging uncertainty for dynamic depth matching. Further, to prevent incomplete text-label matching, we propose to classify texts into appropriate parent node labels instead of leaf node labels. Inspired by the Dempster-Shafer evidence theory (DST), we leverage evidence to describe the uncertainty of the classifier’s prediction on label sequences. Meanwhile, we design an adaptive depth-aware method integrated with the hierarchical structure, dynamically determining whether to stop or proceed with classification.

[1] Our code is available at https://anonymous.4open.science/r/DepthMatch-157E/ (accessed on 1 August, 2024).

During the prediction process, we proactively stop classification to prevent error propagation and ensure credible and robust predictions. The DeepMatch model addresses the issue of improperly matched texts by introducing uncertainty measures and a dynamic adaptive classification approach. Additionally, DeepMatch employs a depth-aware strategy based on uncertainty to mitigate error propagation. The contributions of this paper are as follows:

1. We propose an uncertainty model based on a sequence-to-sequence structure, leveraging the Dempster-Shafer evidence theory to obtain uncertainty sharing among hierarchical sequences. Further, text and local label sequences are cross-fused, enhancing comprehensive representations with uncertainty from parent node labels to classify leaf nodes.

2. To address the problem of error propagation in classification, we propose a dynamic early stopping strategy. Specifically, for texts with incomplete text-label matching, it is necessary to adaptively determine the depth of classification to prevent error propagation.

3. We conduct comprehensive experiments on four public datasets to validate our proposed model, representing its superior performance over the current state-of-the-art (SOTA) models. Furthermore, we analyze the performance without leaf node labels in the ground truth, demonstrating the necessity of the proposed strategy.

2 Related Work

2.1 Hierarchical Classification

In hierarchical multi-label classification, a text corresponds to multiple class labels, and these labels have natural hierarchical dependencies, such as parent-child relationships. Effectively utilizing the hierarchical structure is crucial for HTC. Various studies have focused on representing hierarchical information. The hierarchical-aware global model HiAGM [12] represents the hierarchy as a directed graph and then aggregates node information using the prior probabilities of label dependencies. Based on HiAGM, the HTCInfoMax model [13] was proposed, which is based on maximizing mutual information between text and labels. Additionally, some techniques utilize both local and global hierarchical information and unify them. The HARNN model [14] uses attention mechanisms in local classifiers to extract label features, while the global classifier concatenates features extracted from each level for prediction. The LA-HCN model [15] uses common factors to establish connections among sibling categories, propagating text representations from parent to child layers to determine the most compatible category in the child layer and giving it more attention. The hierarchical-guided contrastive learning HGCLR model [16] embeds the hierarchy into the text encoder rather than modeling it separately. Although these hierarchical classification methods focus on extracting label features to obtain better hierarchical representations, they often flatten the hierarchy to predict labels. This can lead to incomplete text-label matching due to the limited size of the label hierarchy.

2.2 Sequence-to-Sequence Learning

Sequence to Sequence (Seq2Seq) learning [17] is widely used in machine translation tasks and text generation tasks. Researchers use the Seq2Seq ways for multi-label classification, encoding each text into contextual representations. Then they integrate historical information into the attention mechanism to assist in label decoding [18]. The Seq2Image means [19] converts genome sequences into images and uses Convolutional Neural Networks (CNNs) for classification. In multi-label sentiment classification, the application of Seq2Seq performs better than other approaches by implicitly modeling emotion relevance [20].

In the application of Seq2Seq to hierarchical classification, the Seq2Tree [7] framework introduces a sequence-to-tree approach. It addresses the "incomplete text-label matching" problem in HTC, where each predicted leaf node within a path should not conflict with its parent node. They combine the tree structure with the Depth-First Search (DFS) algorithm, ensuring that nodes within the same path can be predicted in a top-down order. Initially, they use DFS to convert hierarchical labels into label sequences, and then map the text and label sequences in a Seq2Seq manner. Additionally, they design a Constrained Decoding (CD) strategy, which guides the generation process using label dependencies. The candidate labels generated by the CD strategy are constrained to the child nodes of the generated parent node. Specifically, after encoding the input sequence, the decoder predicts the DFS label sequence. The i-th token in the sequence and its decoder expression are:

$$\widehat{y}_i,\ h_i^d = Decoder\ \left(\left[H_i,\ h_{<i}^d\right],\ \widehat{y_{i-1}},\ T\right),\ \widehat{y}_i \in DV_{\widehat{y}_i}, \tag{1}$$

where $Decoder$ represents the decoder, $h_{<i}^d$ is the state from the previous time step, $\widehat{y_{i-1}}$ depicts the token from the last time step, and $DV_{\widehat{y}_i}$ describes the generated vocabulary.

In summary, the DepthMatch (Ours) model is compared with the Seq2Tree model as shown in Fig. 2. The Seq2Tree model adopts the DFS strategy and uses the Softmax output of category probability distribution as the classification basis. However, errors occurring at parent nodes during the decoding process can affect the classification of child nodes, leading to error accumulation over time. Our DepthMatch model quantifies the prediction uncertainty of each layer label in a top-down manner using evidence theory, combining uncertainty with parent-child dependencies during decoding. During decoding, we can quantify the model's confidence in classification. When uncertainty is high, we early stop classification at parent nodes to ensure robust predictions and avoid error propagation.

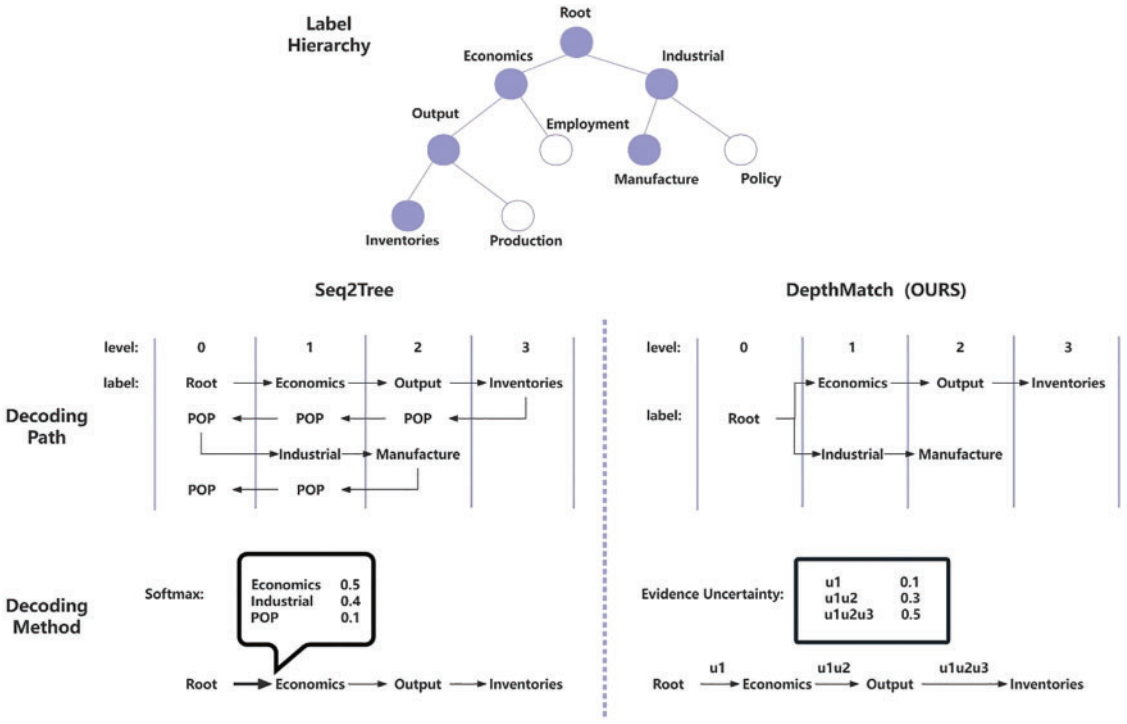

Figure 2: Seq2Tree *vs.* the proposed DepthMatch model

3 Methodology

The overall framework of the model is illustrated in Fig. 3. The training process consists of three parts: constructing local label sequences, measuring the evidence uncertainty of hierarchy, and performing label encoding-decoding under a depth-aware strategy. We first flatten the labels in each layer and construct label sequences based on dependencies. Then, we measure the uncertainty of the label sequences using the Dempster-Shafer evidence theory (DST). Finally, we share text and label embeddings to generate text representations with local label information, and then decode label sequences guided by uncertainty.

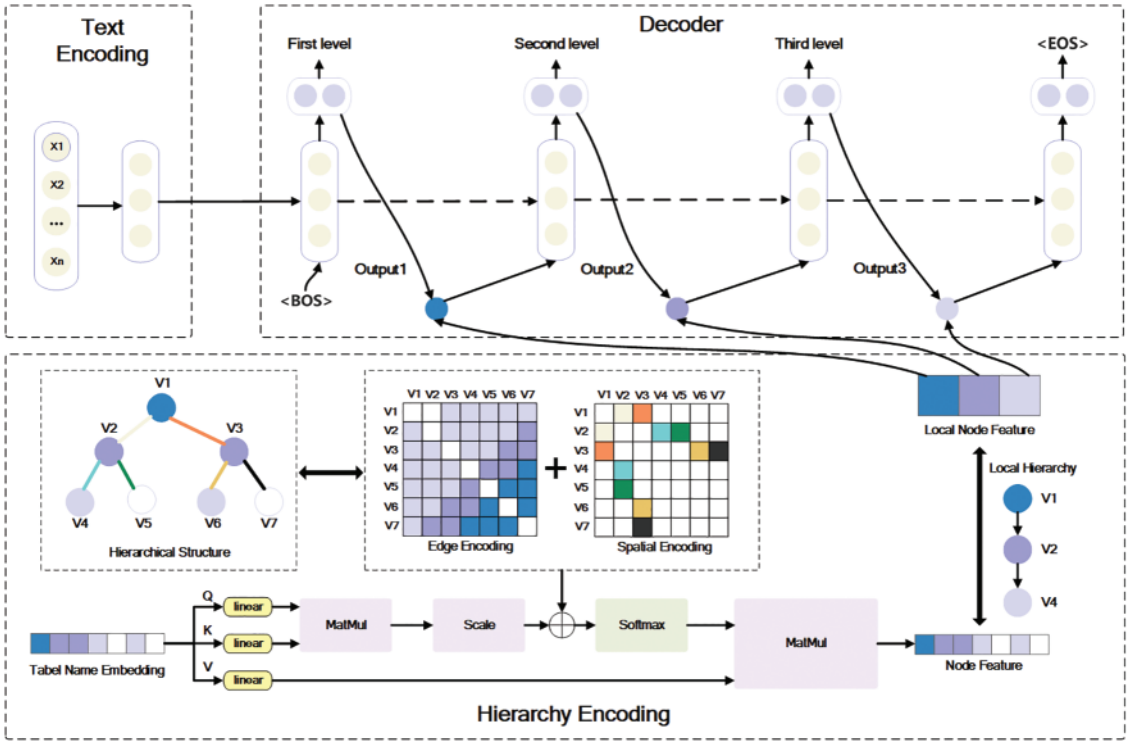

Figure 3: The overall framework of the proposed DepthMatch, an uncertainty-guided depth-aware model for hierarchical text classification

3.1 Problem Definition

Hierarchical Text Classification (HTC) aims to predict corresponding labels from input text. Given input text $T = \{c_1, c_2, \cdots, c_n\}$, where n is the sequence length, the task is to predict a subset of the label set $K = \{k_1, k_2, \cdots, k_m\}$, where m is the length of the label set. Each label corresponds to a unique node in the hierarchical structure, typically a tree-like structure. The predicted label subset includes leaf nodes and their parent nodes. Current approaches target classifying leaf nodes, but there exist cases where the ground-truth labels for some texts do not include leaf node labels. Therefore, our task is to ensure that the model can classify text T to both parent node labels and the correct leaf node labels.

3.2 Local Label Sequence Construction

We represent a local label hierarchy as a subgraph of the global label structure, flattening a local label hierarchy into a label sequence.

$$\mathbf{S}_h = \sum_{j}^{j \in l^h} \mathbf{y}_j, \quad \mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_z\}. \tag{2}$$

Here, $\mathbf{S}_h \in R^d$ represents the h-th layer in the hierarchy, l_j denotes the true labels for the h-th layer, $\mathbf{y}_j$ defines the label feature containing hierarchical information, z indicates the maximum number of layers, and $\mathbf{S}$ is the local hierarchical sequence representation.

To combine local hierarchical sequences with textual features, we propose a top-down sequence-to-sequence approach:

$$p(\mathbf{S}|\mathbf{x}) = \prod_{h=1}^{z} p\ (\mathbf{S}_h|\mathbf{S}_{<h},\ \mathbf{x}), \tag{3}$$

$\mathbf{S}_{<h}$ represents the hierarchical sequence representation of layers lower than the h-th layer, and $\mathbf{x}$ is the hidden layer representation corresponding to each token in the text. $\mathbf{S}_h$ corresponds to the hierarchical representation of labels for the h-th layer, which ensures a top-down directionality in sequence-to-sequence modeling.

3.3 Hierarchical Evidence Uncertainty Quantification

The Dempster-Shafer evidence theory (DST) is a way for uncertain reasoning, where different textual features are utilized to obtain classification evidence. It quantifies the uncertainty of assigning a text to labels, thus providing uncertainty estimates for each label in the label sequence. For a classification problem, the set of all possible labels is represented by the identification framework Θ. Any label corresponding to a text belongs to a subset of Θ, denoted by the set 2^{Θ}. Evidence from the text can provide support for labels at different levels in the sequence, and this support can be obtained through a basic trust allocation function.

The basic trust assignment function $m(C)$ is a mapping from the set 2^{Θ} to [0, 1], where C represents any subset of the identification framework Θ, denoted as $C \in 2^{\Theta}$. The basic trust assignment function $m(C)$ satisfies properties $0 \leq m(C) \leq 1$, $m(\varnothing) = 0$, and $\sum_{A \in 2^{\Theta}} m(C) = 1$, where $\varnothing$ is the empty set. Any subset C in the identification framework that satisfies $m(C) > 0$ is called a focal element. The basic trust assignment function $m(C)$ is distributed over the Dirichlet distribution:

$$D(p|\alpha) = \begin{cases} \dfrac{1}{B(\alpha)} \prod_{k=1}^{K} p_k^{\alpha_k - 1} & \text{for } p \in S_K \\ 0 & \text{otherwise} \end{cases}, \tag{4}$$

where α_k represents the distribution parameter, $B(\cdot)$ describes the beta function, and $S_K = \left\{p | \sum_{k=1}^{K} p_k = 1 \text{ and } 0 \leq p_k \leq 1, \forall k\right\}$ is the k-dimensional unit simplex. α_k represents the Dirichlet parameters for the k-th class, while $S = \sum_{k=1}^{K} \alpha_k$ denotes the Dirichlet strength.

We adopted the text encoder component of the pre-trained BERT model, which consists of a stack of 12 layers of Transformer Encoder structure. We transform the given input text T into word embeddings $\mathbf{x}^0 = \mathbf{W}^0 T$, $\mathbf{x}^0 = \{x_1,\ x_2,\ \cdots,\ x_n\}$, where $\mathbf{W}^0$ represents the weights of the word embedding

layer. The hidden state of the given $\mathbf{x}^0$ text encoder can be represented as:

$$\mathbf{x} = \texttt{TextEncoder}\left(\mathbf{x}^0\right). \tag{5}$$

Here, $\mathbf{x} \in R^{n\times d_h}$ represents the hidden layer representations for each token, and d_h is the size of the hidden layer. The hidden layer features $\mathbf{x} \in R^{n\times d_h}$ are input to a linear layer, resulting in logits, representing the evidence $e = [e_1, e_2, \cdots, e_K]$ for each class. The parameters of the Dirichlet distribution can be computed using $\alpha_k = e_k + 1$, and the total quantity of uncertainty and belief mass is a constant, denoted as $u + \sum_{k=1}^{K} m(C_k) = 1$. Both the uncertainty and belief mass are determined by the parameters: $u = \dfrac{K}{S}$ and $m(C_k) = \dfrac{\alpha_k - 1}{S}$.

The advantage of evidence uncertainty lies in its modeling based on the Dirichlet distribution, which directly parameterizes the belief mass from the neural network outputs. Through the Dirichlet distribution, it is possible to flexibly control the allocation of belief mass, hence better reflecting the uncertainty of the model at different hierarchical sequences during decoding.

3.4 Uncertainty-Based Depth-Aware Hierarchical Classification

By leveraging DST, we can better handle classification uncertainty and flexibly perform inference and decision-making under label sequence decoding. In DST, we can derive more accurate conclusions by combining evidence from multiple different sequences of preceding layers. The uncertainty from multiple sequences is combined incrementally, formalizing the pairwise Dempster combination rule:

$$u^1 \oplus u^2 = \frac{1}{1-E} u^1 u^2, \tag{6}$$

where $E = \sum_{C_1 \cap C_2 = \varnothing} m(C_1)\, m(C_2)$ is discordant factors. In DST evidence theory, when the earlier labels in the sequence reach a consensus on the majority of the belief mass, it indicates that they can be confidently combined despite the presence of discordant factors.

Utilizing the combination rule, labels are sequentially combined from the front to the back of the sequence:

$$u = u^1 \oplus u^2 \oplus \cdots \oplus u^N = \frac{\prod_{n=1}^{N} u^n}{\prod_{n=1}^{N} (1 - E^n)}, \tag{7}$$

where $E = \sum_{C_1 \cap C_2 \cap \cdots \cap C_n = \varnothing} \left(\prod_{1 \le i \le n} m(C_i)\right)$ represents the discordant factors between labels across multiple consecutive layers. Uncertainty combination takes into account the independent uncertainty of each layer's label as well as the consistency between different layers. By applying combination rules, uncertainty is propagated between the layers of the label sequence.

Next is evidence combination, defining the accumulation of uncertainty for each layer as:

$$\begin{cases} w^1 = 1; \\ w^2 = u^1; \\ w^{n+1} = w^n \oplus u^n = \dfrac{1}{1-E^n} w^n u^n, \\ \text{for n} = 2, 3, \cdots, N-1 \end{cases}. \tag{8}$$

The uncertainty accumulation w^n is a measure of the overall uncertainty from the current layer up to the nth layer. This dynamic weight allocation mechanism allows the model to flexibly choose the

stopping position for decoding hierarchical labels based on different samples. For easily classifiable samples, decoding can proceed to leaf nodes as much as possible, while for complex samples, robust predictions can be made based on uncertainty. Through the adjustment of uncertainty accumulation, the model can intelligently determine the stopping position for early termination, effectively reducing error propagation during the sequence-to-sequence prediction process. Specific depth-aware objectives are presented in Section 3.5.2.

3.5 Training Process

3.5.1 Encoding of Hierarchical Label Structure

To obtain the comprehensive feature representation, we utilize Graphormer to model the hierarchical structure of labels. Then, we map category indices to corresponding embedding vectors $\mathbf{L} = [l_1,\ l_2,\ \cdots, l_k]$, with an output size of $R^{k\times d_h}$ vectors. For each node's embedding vector $\mathbf{L}$ from previous iterations, we enhance the vectors using self-attention layers to fuse hierarchical relationships. Within each graph layer's iteration, node features are passed as input to the hidden layers, and modifications are made to the Query-Key product matrix $\mathbf{A}^H$ of the self-attention layer using spatial encoding and edge encoding:

$$\mathbf{A}_{ij}^H = \frac{\left(l_i\mathbf{W}_Q^H\right)\left(l_j\mathbf{W}_K^H\right)}{\sqrt{d_h}} + c_{ij} + b_{\phi\left(y_i,y_j\right)}, \tag{9}$$

In the given context, $c_{ij} = \frac{1}{D}\sum_{n=1}^{D} w_{e_n}$ represents edge encoding, indicating the edge information between two nodes, where $w_{e_i} \in R^1$ denotes the learnable weight of each edge. $D = \phi\left(y_i,\ y_j\right)$ describes the distance between two nodes, y_i and y_j. $\mathbf{A}_{ij}^H$ depicts attention weights, where query and key are projected onto $\mathbf{W}_Q^H \in R^{d_h\times d_h}$ and $\mathbf{W}_K^H \in R^{d_h\times d_h}$. $b_{\phi\left(y_i,y_j\right)}$ explains spatial encoding, measuring the connectivity between two nodes.

To achieve effective training and accelerate convergence, we apply softmax to the attention weights $\mathbf{A}^H$, followed by element-wise multiplication with the value matrix $\mathbf{V}$. Additionally, to enhance model training efficiency and generalization capability, we utilize residual connections and layer normalization operations. The self-attention is computed as follows:

$$\mathbf{Y} = \mathrm{LayerNorm}\left(\mathrm{Softmax}\left(\mathbf{A}^H\right)\mathbf{V} + \mathbf{L}\right), \tag{10}$$

We obtain the label feature representation $\mathbf{Y} = [y_1,\ y_2,\ \cdots,\ y_k]$, completing the encoding of the hierarchical structure.

3.5.2 Decoding of Label Sequences

During the training phase, our DepthMatch model employs local hierarchical sequences to classify each layer's label in a top-down manner. To achieve shared text and label embedding weights, we extend and perform element-wise multiplication between the attention mask of the input text and the label attention mask, resulting in a cross attention mask:

$$\mathbf{A}_{mn} = \frac{q_m k_n^T}{\sqrt{d_h}}. \tag{11}$$

This mask is used to control the multiplication matrix between Query and Key in the cross-attention mechanism. Here, $q_m = x_m\mathbf{W}_Q$ and $k_n = y_n\mathbf{W}_K$ correspond to the text embeddings and label

features, while $\mathbf{W}_Q \in R^{d_h \times d_h}$ and $\mathbf{W}_K \in R^{d_h \times d_h}$ are weight matrices. By computing the dot-product attention between query and key, the correlation between the current text and labels can be measured. In this way, during the decoding phase, different parts of the input sequence are weighted based on the current position of the decoder and the generated parts, leading to a better understanding of the dependency relationship within the hierarchical sequence. We input text embeddings, local hierarchical sequence representation, and cross-attention mask into the encoder part of BERT to obtain the hidden layer representation of the h-th layer.

$$h_h = \text{BERTEncoder}\left(\left[\mathbf{x}^0, \mathbf{S}_{<h}\right], \mathbf{A}_{mn}\right), \tag{12}$$

where $\mathbf{x}^0$ is the word embedding of the text, $\mathbf{S}_{<h}$ represents the hierarchical sequence representation lower than the h-th layer. The hidden features are input into a linear layer, and probabilities are calculated using the sigmoid function to output the probability of appearing on the j-th sub-label of the current label v_i, thus obtaining the model's prediction result. The calculation formula is as follows:

$$p_{ij} = sigmoid\ \left(\mathbf{W}\mathbf{h}_h + b\right)_j, \tag{13}$$

where $\mathbf{W}$ is the weight coefficient, b is the bias term, $\mathbf{h}_i$ is the feature vector, and p_{ij} is the predicted probability.

In multi-label classification, for the h-th layer, we use a binary cross-entropy loss function for the label v_i, as shown below:

$$L_h = -\frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J} y_{ij}\log\left(p_{ij}\right) + \left(1 - y_{ij}\right)\log\left(1 - p_{ij}\right), \tag{14}$$

where J represents the number of sub-labels, y_{ij} is the j-th sub-label of the current label v_i, and p_{ij} is the corresponding probability.

The final loss function integrated with the depth-aware strategy is:

$$L = \lambda\left(w^h\right)\sum_{h=1}^{H} L_h, \quad s.t\lambda\left(w^h\right) = \begin{cases} 0, \text{ if } w^h > \delta \\ 1, \text{ otherwise} \end{cases} \tag{15}$$

where w^h is the measure of uncertainty from the previous $h-1$ layers to the current h-th layer, and $\lambda(\cdot)$ represents considering whether to stop classification based on the magnitude of uncertainty. In this approach, we recognize the importance of not classifying to leaf nodes when uncertainty is high, to obtain more robust prediction results. If the accumulated uncertainty of a sample in the preceding layers of the label sequence exceeds a threshold δ, then the loss on that sample in subsequent sequences will be eliminated. Consequently, the entire sequence decoding process exhibits a decreasing trend, allowing the model to have more confidence in decoding each layer's label.

4 Experiments

4.1 Dataset

We conduct our experiments on four datasets: WOS (Web of Science) [21], RCV1-V2 (Reuters Corpus Volume I) [22], AAPD (Arxiv Academic Paper Dataset) [23], and BGC. The BGC dataset is available at: www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/blurb-genre-collection.html (accessed on 1 August, 2024). We evaluate our experimental results using multiple metrics. WOS covers abstracts of academic papers published in the Web of Science database, AAPD collects abstracts of Arxiv academic papers along with corresponding subject category information, RCV1-V2 is a news

classification corpus, and BGC dataset consists of blurbs (short texts) of books and metadata such as authors, publication dates, and page numbers. WOS is used for single-path HTC, while RCV1-V2, AAPD, and BGC contain multi-path classification labels. The labels in all four datasets form hierarchical tree-like structures. Detailed statistics are shown in Table 1.

Table 1: The statistical details of datasets

Dataset	|Y|	Avg (y_i)	Depth	#Train	#Dev	#Test
WOS	141	2.0	2	30,070	7518	9397
RCV1-V2	103	3.24	4	20,833	2316	781,265
AAPD	54	2.41	2	43,872	10,968	1000
BGC	146	3.01	4	58,800	14,700	18,394

4.2 Baseline Models and Evaluation Metrics

We selected several baseline methods for comparison, including FastText [24], TextVDCNN [25], HTCInfoMax [13], TextRCNN [26], HiAGM [12], HBGL [27], HiMatch [28], HGCLR [16], HiTIN [29] and Seq2Tree [7] models.

We follow the evaluation metrics of baseline models such as [12,13,27,28], using Micro-F1 and Macro-F1. The specific formulas are as follows:

$$Micro - F1 = 2 \cdot \frac{P_{\text{micro}} \cdot R_{\text{micro}}}{P_{\text{micro}} + R_{\text{micro}}} \tag{16}$$

$$Macro - F1 = \frac{1}{N} \sum_{i=1}^{N} \frac{2 \cdot \mathrm{P}_i \cdot \mathrm{R}_i}{\mathrm{P}_i + \mathrm{R}_i} \tag{17}$$

where $P_{\text{micro}} = \sum_{i=1}^{N} \mathrm{TP}_i / \sum_{i=1}^{N} (\mathrm{TP}_i + \mathrm{FP}_i)$, $R_{\text{micro}} == \sum_{i=1}^{N} \mathrm{TP}_i / \sum_{i=1}^{N} (\mathrm{TP}_i + \mathrm{FN}_i)$, $\mathrm{P}_i = \mathrm{TP}_i / \mathrm{TP}_i + \mathrm{FP}_i$, $\mathrm{R}_i = \mathrm{TP}_i / \mathrm{TP}_i + \mathrm{FN}_i$. Micro-F1 is evaluated by first calculating the global True Positives (TP), False Positives (FP), and False Negatives (FN) for all categories. Macro-F1 is evaluated by separately calculating the Precision and Recall for each category, and then averaging these values. By using these two metrics, we can comprehensively understand the model's performance from different perspectives.

4.3 Experimental Settings

For the text encoder, we use the BERT model, with the transformer's bert-base-uncased as the base architecture. For Graphormer, we set the adaptive graph attention heads to 8 and feature size to 768. The model is trained on the training set, and after each epoch, evaluation is performed on the validation set. The detailed hyperparameter information is shown in Table 2.

4.4 Experimental Results

The proposed model is experimentally compared with baseline models on the WOS, RCV1-V2, AAPD, and BGC datasets. The specific experimental results are shown in Table 3. On the four public datasets, compared to previous mainstream models, our proposed model achieves better performance. The model's Micro-F1 values on the WOS, RCV1-V2, AAPD, and BGC datasets are 87.59%, 86.90%, 78.81%, and 80.46%, respectively, all reaching state-of-the-art (SOTA) performance. The Macro-F1

values are 81.54%, 69.32%, 63.37%, and 66.55%, respectively, achieving optimal results on WOS and BGC as well. The performance is significantly improved.

Table 2: The list of hyperparameters along with their explanations and optimal settings

Hyperparameter	Explanation	Default
lr	Learning rate	3e-5
batch	Batch size	16
early_stop	Epoch before early stop	10
update	Gradient accumulate steps	1
warmup	Warmup steps	2000
thre	Threshold for keeping tokens	0.02
layer	Label layer	2 (WOS, AAPD), 4 (RCV1-V2, BGC)
δ	If prefix weight $\leq\delta$, the loss of expert m on the sample will be eliminated	0 (WOS), 0.5 (RCV1-V2, AAPD, BGC)

Table 3: The comparison of different models on WOS, RCV1-V2, AAPD and BGC

Dataset	WOS		RCV1-V2		AAPD		BGC	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
FastText	74.21	69.13	72.25	29.13	66.80	56.96	59.43	44.69
TextVDCNN	75.71	59.77	63.25	28.66	65.89	51.20	64.96	52.35
TextRCNN	83.55	76.99	81.57	59.25	66.58	55.16	72.37	54.18
HiAGM	85.82	80.28	83.96	63.35	75.96	58.26	77.19	58.01
HTCInfoMax	85.28	79.76	83.95	61.13	77.84	57.99	75.14	56.97
HiMatch	86.10	80.44	84.73	64.11	76.47	59.97	76.88	57.96
HGCLR	87.03	81.18	86.49	68.33	78.44	63.20	78.63	64.59
HiTIN	87.08	81.42	86.65	**69.45**	78.69	**63.47**	79.59	65.41
Seq2Tree	86.70	81.22	86.41	69.03	78.57	63.18	79.05	65.24
DepthMatch (Ours)	**87.59**	**81.54**	**86.90**	69.32	**78.81**	63.37	**80.46**	**66.55**

4.5 Performance Analysis

4.5.1 Performance Analysis of Text with Incomplete Text-Label Matching

We first extract all texts from the ground-truth labels in the RCV1-V2 dataset that have parent node labels but no corresponding leaf node labels. In Table 4, we present statistics on the number of texts corresponding to each parent node label and their proportion in the dataset. By summarizing, we observe that texts with incomplete matching leaf node labels account for 23.47% of the entire test set, indicating that approximately one-fourth of the texts should not be assigned corresponding leaf node labels. Therefore, we test the performance of the Seq2SESeq and HGCLR models on these texts,

as shown in Fig. 4. The horizontal axis represents the performance of different parent node labels corresponding to the texts, all of which should terminate classification at the current parent node label. We use lines to indicate the Micro-F1 and Macro-F1 scores of the two models on these classes, and use bar charts to reflect the performance gain compared to the SOTA model. It can be seen that we outperform the SOTA model in the majority of classes, with a 33% improvement in Micro-F1 score for class E41 and over 120% increase in Macro-F1 score for Class C18.

Table 4: Statistics of the parent node labels for the subset of texts with incomplete matching

Parent node	CCAT	C15	C151	C17	C18	C31	C33	C41	ECAT	E12	E13
Amount	2064	225	56710	4609	29	28402	13710	1057	606	24333	126
Proportion (%)	0.26	0.03	7.26	0.59	0.004	3.64	1.75	0.14	0.08	3.11	0.02
Parent node	14	E21	E31	E41	E51	GCAT	G15	MCAT	M13	M14	Total
Amount	416	920	571	14490	3915	23811	1492	878	440	4584	183388
Proportion (%)	0.05	0.12	0.07	1.85	0.50	3.05	0.19	0.11	0.06	0.59	23.47

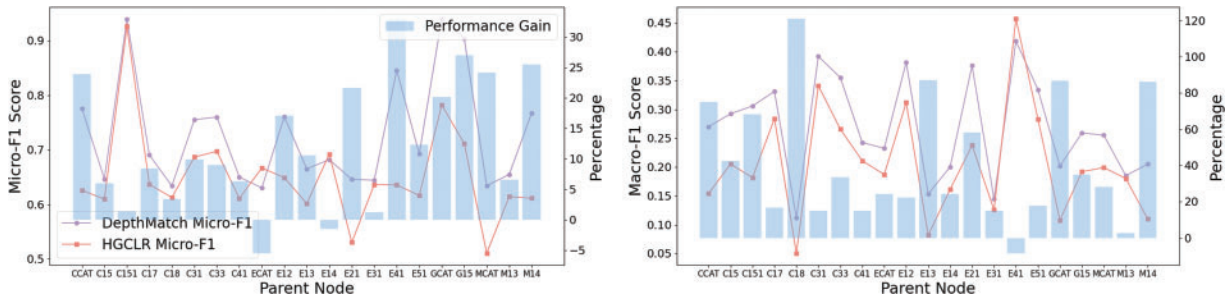

Figure 4: The Micro-F1 and Macro-F1 scores and their performance gain between DepthMatch and HGCLR under incomplete text-label matching

4.5.2 Mean and Standard Deviation

We analyze two types of F1 scores and their standard deviations for the model under different random seeds during the training process, as shown in Fig. 5. Two dashed lines represent two types of F1 scores, and the shaded area indicates the standard deviation across multiple experiments. It is commonly observed across the four datasets that during the early stages of training, the model's performance varies significantly, and there is also a considerable difference across multiple experiments. As the number of epochs increases, the model gradually stabilizes.

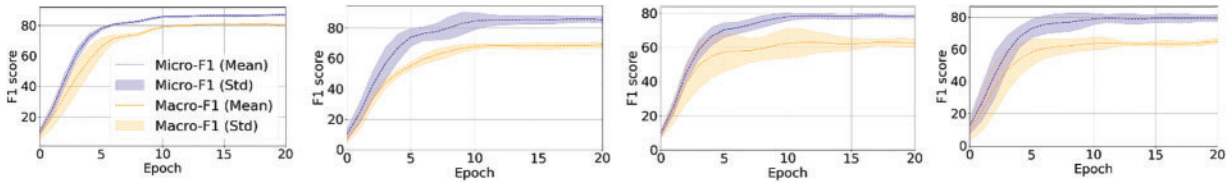

Figure 5: Mean and standard deviation of F1 score

4.5.3 Model Parameter Analysis

We analyze the parameter count of our model and compare it with other mainstream models, as shown in Fig. 6. Our model's parameter count is slightly smaller than HGCLR but significantly smaller than other types of models. One important reason is that, apart from our text encoder and local hierarchical structure encoder, no additional parameters are required. In contrast, models like HiAGM and HiMatch consume additional space to project text feature parameters onto labels, while HTCInforMax introduces additional auxiliary neural networks.

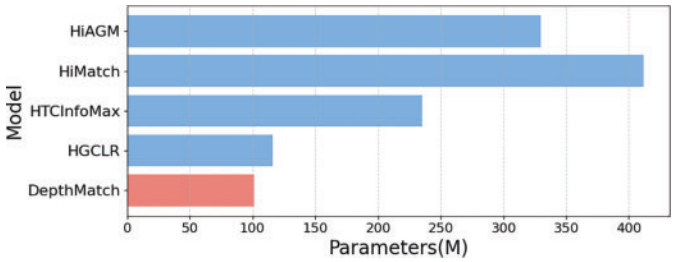

Figure 6: Comparison of model parameters between DepthMatch and current mainstream models

4.5.4 Analysis of Uncertainty and Accuracy

We visualize the data uncertainty of difficult samples with low accuracy and the model uncertainty one layer above the leaf nodes, as shown in Fig. 6. We utilize the semantic vectors extracted directly from the extensively pre-trained BERT model to measure the evidence uncertainty of the samples, indicating the difficulty of the samples themselves. As seen from Fig. 7a, this uncertainty typically ranges between 0.6 and 0.9. Directly using such texts for classification poses significant risks. The model uncertainty one layer above the leaf nodes accumulates based on the uncertainty of predictions from higher-level models, typically exceeding 0.75. Such high uncertainty indicates challenges in robustly classifying leaf nodes. Fig. 7b demonstrates the impact of using an early stopping strategy on difficult samples. It can be observed that the error rate in classification significantly decreases for multiple difficult classes, with a reduction of 15.1% for Class E31 and an average reduction of around 10%. The bar chart results confirm the effectiveness of the depth-aware strategy, enabling more accurate classification results for samples with high uncertainty.

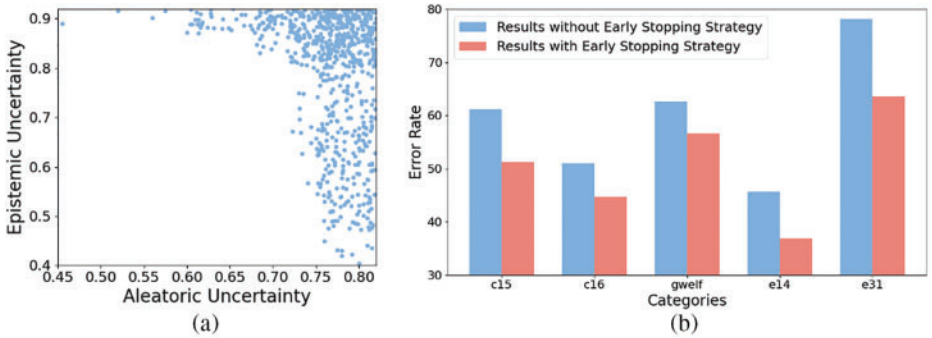

Figure 7: The uncertainty of difficult samples and the comparison of classification error rates on challenging samples with and without depth matching. (a) Epistemic and aleatoric uncertainty. (b) Error rates for classification

4.5.5 Ablation Experiments

We replace the components in DepthMatch with a standard self-attention model and a standard attention masking model, as shown in Table 5. After the replacement, the model performance declined. Firstly, compared to the standard self-attention model, we introduced edge encoding and spatial encoding. These components learn the information between two edges and the connectivity between two nodes, respectively, enhancing the representation capability of the label hierarchy. Secondly, compared to the standard attention masking model, we extended and performed a product operation on the label attention mask. Then, we generated a cross-attention mask to measure the relevance between the current text and the labels. Our model improves performance through the representation of the hierarchy and the interaction between the text and labels.

Table 5: The results of replacing components with the standard self-attention model and standard attention masking model on the WOS and RCV1-V2 datasets. "r.p." stands for replace

Ablation study	WOS		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
DepthMatch	**87.59**	**81.54**	**86.90**	**69.32**
r.p. standard self-attention model	86.65	80.37	86.21	68.67
r.p. standard attention masking model	86.33	79.98	86.01	68.48

We conduct experiments on the WOS and RCV1-V2 dataset, and the results are shown in Table 6, the model's classification performance is superior to previous results, with improvements of 1.48% and 1.39% in Micro-F1 and Macro-F1, respectively. A good hierarchical representation is crucial for HTC tasks. Similarly, under the depth-aware strategy, the model also shows significant improvement, with increases of 1.17% and 1.2% in Micro-F1 and Macro-F1, respectively. The model can produce more reliable classification results.

Table 6: The ablation study of different modules on WOS and RCV1-V2

Ablation study	WOS		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
DepthMatch	**87.59**	**81.54**	**86.90**	**69.32**
w/o $\lambda\left(w^h\right)$	86.42	80.34	86.53	68.65
w/o $\lambda\left(w^h\right)$ \&*Graph Encoder*	86.11	80.05	86.08	68.29
BERT	85.75	79.36	85.62	67.41

4.5.6 Statistical Analysis

In Table 7, we repeated the experiments for the proposed method and the baseline Seq2Tree five times (using different random seeds). We conducted experiments on the WOS and RCV1-V2 datasets. It can be seen that these improvements are statistically significant based on the paired t-test at the 95% significance level.

Table 7: The standard deviation and *p*-value of different models on WOS and RCV1-V2

Model	WOS		RCV1-V2	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
DepthMatch	**87.39** ± **0.2**	**81.16** ± **0.4**	**86.50** ± **0.4**	**68.81** ± **0.6**
Seq2Tree	86.12 ± **0.6**	80.45 ± **0.8**	85.69 ± **0.7**	68.33 ± **1.1**
p-value	4.96×10^{-3}	6.82×10^{-2}	9.60×10^{-2}	3.59×10^{-1}

5 Conclusions

Hierarchical Text Classification (HTC) aims to match text with labels in a structured manner. To address the issues of incomplete text-label matching and error propagation, this paper proposes an uncertainty-guided HTC deep awareness model called DepthMatch. The model employs Dempster-Shafer Evidence Theory to enable uncertainty sharing between hierarchical sequences. Additionally, a dynamic stopping strategy is introduced, using uncertainty to determine the depth of text classification and prevent error propagation. In real-world datasets, most labels have relatively few data, especially at the leaf nodes of the hierarchy. This situation poses challenges for model learning and can lead to unpredictable problems. Therefore, addressing the long-tail problem in hierarchical multi-label text classification is an important research direction for the future.

Acknowledgement: This work was partly supported by the National Natural Science Foundation of China (62136002 and 62306056), the National Natural Science Foundation of Chongqing (cstc2022ycjh-bgzxm0004), and the Science and Technology Commission of Chongqing Municipality (CSTB2023NSCQ-LZX0006), respectively.

Funding Statement: This work was sponsored by the National Key Research and Development Program of China (No. 2021YFF0704100), the National Natural Science Foundation of China (No. 62136002), and the Chongqing Natural Science Foundation (No. cstc2022ycjh-bgzxm0004).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Zixuan Wu, Ye Wang; data collection: Zixuan Wu; analysis and interpretation of results: Zixuan Wu, Ye Wang, Lifeng Shen; draft manuscript preparation: Zixuan Wu, Ye Wang, Lifeng Shen, Feng Hu, Hong Yu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Publicly available datasets were analyzed in this study. The Web of Science Dataset is available at https://data.mendeley.com/datasets/9rw3vkcfy4/6 (accessed on 1 August 2024), the Reuters Corpus Volume I Dataset is available at https://trec.nist.gov/data/reuters/reuters.html (accessed on 1 August 2024), the Arxiv Academic Paper Dataset is available at https://drive.google.com/file/d/1QoqcJkZBHsDporttTxaYWOM_ExSn7-Dz/view (accessed on 1 August 2024), the BGC dataset is available at https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/blurb-genre-collection.html. (accessed on 1 August 2024).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Y. K. Cao, Z. Y. Wei, Y. J. Tang, and C. K. Jin, "Hierarchical label text classification method with deep-level label-assisted classification," in *2023 IEEE 12th Data Driven Control Learn. Syst. Conf. (DDCLS)*, Xiangtan, China, IEEE, 2023, pp. 1467–1474. doi: 10.1109/DDCLS58216.2023.10166293.

[2] J. C. Gomez and M. Moens, "A survey of automated hierarchical classification of patents," in *Professional Search Modern World*, 2014, pp. 215–249. doi: 10.1007/978-3-319-12511-4_11.

[3] R. Aly, S. Remus, and C. Biemann, "Hierarchical multi-label classification of text with capsule networks," in *Proc. 57th Annu. Meeting Assoc. Comput. Linguistics: Student Res. Workshop*, Florence, Italy, 2019, pp. 323–330. doi: 10.18653/v1/P19-2045.

[4] X. Song, Y. Zhu, X. M. Zeng, and X. S. Chen, "Hierarchical contaminated web page classification based on meta tag denoising disposal," *Secur. Commun. Netw.*, vol. 2021, pp. 1–11, 2021. doi: 10.1155/2021/2470897.

[5] J. H. Wang and L. Zhang, "News text classification based on deep learning and TRBert model," in *2023 IEEE 3rd Int. Conf. Electron. Technol. Commun. Inf. (ICETCI)*, Changchun, China, IEEE, 2023, pp. 1244–1248. doi: 10.1109/ICETCI57876.2023.10176604.

[6] Y. Wang, Q. H. Hu, H. Chen, and Y. H. Qian, "Uncertainty instructed multi-granularity decision for large-scale hierarchical classification," *Inf. Sci.*, vol. 586, pp. 644–661, 2022. doi: 10.1016/j.ins.2021.12.009.

[7] C. Yu, Y. Shen, and Y. Mao, "Constrained sequence-to-tree generation for hierarchical text classification," in *Proc. 45th Int. ACM SIGIR Conf. Res. Dev. Inf. Retrieval*, Madrid, Spain, 2022, pp. 1865–1869. doi: 10.48550/arXiv.2204.00811.

[8] J. H. Wang, Y. Cheng, J. T. Chen, T. T. Chen, D. Chen and J. Wu, "Ord2Seq: Regarding ordinal regression as label sequence prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Paris, France, 2023, pp. 5865–5875. doi: 10.48550/arXiv.2307.09004.

[9] B. L. Wang, I. Titov, J. Andreas, and Y. Kim, "Hierarchical phrase-based sequence-to-sequence learning," in *Proc. 2022 Conf. Empirical Methods Nat. Lang. Process.*, Abu Dhabi, United Arab Emirates, 2022, pp. 8211–8229. doi: 10.48550/arXiv.2211.07906.

[10] S. H. Im, G. B. Kim, H. Oh, S. Jo, and D. H. Kim, "Hierarchical text classification as sub-hierarchy sequence generation," in *Proc. AAAI Conf. Artificial Intell.*, Washington, DC, USA, 2023, vol. 37, pp. 12933–12941. doi: 10.48550/arXiv.2111.11104

[11] J. Valmadre, "Hierarchical classification at multiple operating points," in *Proc. 36th Int. Conf. Neural Inf. Process. Syst.*, New Orleans, LA, USA, 2022, pp. 18034–18045. doi: 10.48550/arXiv.2210.10929.

[12] J. Zhou *et al.*, "Hierarchy-aware global model for hierarchical text classification," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguist.*, Washington, DC, USA, 2020, pp. 1106–1117. doi: 10.18653/v1/2020.acl-main.104.

[13] Z. F. Deng, H. Peng, D. X. He, J. X. Li, and S. Y. Philip, "HTCInfoMax: A global model for hierarchical text classification via information maximization," in *Proc. 2021 Conf. North Am. Chapter Assoc. Comput. Linguist.: Hum. Lang. Technol.*, 2021, pp. 3259–3265. doi: 10.18653/v1/2021.naacl-main.260.

[14] W. Huang *et al.*, "Hierarchical multi-label text classification: An attention-based recurrent network approach," in *Proc. 28th ACM Int. Conf. Inf. Knowl. Manage.*, New York, NY, USA, 2019, pp. 1051–1060. doi: 10.1145/3357384.3357885.

[15] X. Y. Zhang, J. H. Xu, C. Soh, and L. H. Chen, "LA-HCN: Label-based attention for hierarchical multi-label text classification neural network," *Expert Syst. Appl.*, vol. 187, 2022, Art. no. 115922. doi: 10.1016/j.eswa.2021.115922.

[16] Z. H. Wang, P. Y. Wang, L. Z. Huang, X. Sun, and H. F. Wang, "Incorporating hierarchy into text encoder: A contrastive learning approach for hierarchical text classification," in *Proc. 60th Annu. Meeting Assoc. Computat. Linguist.*, Dublin, Ireland, 2022, pp. 7109–7119. doi: 10.48550/arXiv.2203.03825.

[17] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2014, vol. 2, pp. 3104–3112. doi: 10.48550/arXiv.1409.3215.

[18] Y. Q. Xiao, Y. Li, J. Yuan, S. R. Guo, Y. Xiao and Z. Y. Li, "History-based attention in seq2seq model for multi-label text classification," *Knowl.-Based Syst.*, vol. 224, 2021, Art. no. 107094. doi: 10.1016/j.knosys.2021.107094.

[19] N. Tavakoli, "Seq2Image: Sequence analysis using visualization and deep convolutional neural network," in *2020 IEEE 44th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Madrid, Spain, IEEE, 2020, pp. 1332–1337. doi: 10.1109/COMPSAC48688.2020.00-71.

[20] Q. Q. Guo, Z. F. Zhu, Q. Lu, D. Y. Zhang, and W. Q. Wu, "A dynamic emotional session generation model based on seq2seq and a dictionary-based attention mechanism," *Appl. Sci.*, vol. 10, no. 6, 2020, Art. no. 1967. doi: 10.3390/app10061967.

[21] K. Kowsari, D. E. Brown, M. Heidarysafa, K. J. Meimandi, M. S. Gerber and L. E. Barnes, "HDLTex: Hierarchical deep learning for text classification," in *2017 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Cancun, Mexico, IEEE, 2017, pp. 364–371. doi: 10.48550/arXiv.1709.08267.

[22] D. D. Lewis, Y. Yang, T. Russell-Rose, and F. Li, "RCV1: A new benchmark collection for text categorization research," *J. Mach. Learn. Res.*, vol. 5, pp. 361–397, 2004.

[23] P. C. Yang, X. Sun, W. Li, S. M. Ma, W. Wu and H. F. Wang, "SGM: Sequence generation model for multi-label classification," in *Proc. 27th Int. Conf. Comput. Linguist.*, Santa Fe, NM, USA, 2018, pp. 3915–3926. doi: 10.48550/arXiv.1806.04822.

[24] A. Joulin, É. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," in *Proc. 15th Conf. Eur. Chapter Assoc. Comput. Linguist.*, Valencia, Spain, 2017, pp. 427–431. doi: 10.48550/arXiv.1607.01759.

[25] A. Conneau, H. Schwenk, Y. L. Cun, and L. Barrault, "Very deep convolutional networks for text classification," in *15th Conf. Eur. Chapter Assoc. Comput. Linguist.*, Valencia, Spain, 2017, pp. 1107–1116. doi: 10.48550/arXiv.1606.01781.

[26] S. W. Lai, L. H. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. Twenty-Ninth AAAI Conf. Artificial Intell.*, Austin, TX, USA, 2015, pp. 2267–2273. doi: 10.1609/aaai.v29i1.9513.

[27] T. Jiang, D. Q. Wang, L. L. Sun, Z. Z. Chen, F. Z. Zhuang and Q. H. Yang, "Exploiting global and local hierarchies for hierarchical text classification," in *Proc. 2022 Conf. Empirical Methods Nat. Lang. Process.*, Abu Dhabi, United Arab Emirates, 2022, pp. 4030–4039. doi: 10.48550/arXiv.2205.02613.

[28] H. B. Chen, Q. L. Ma, Z. X. Lin, and J. Y. Yan, "Hierarchy-aware label semantics matching network for hierarchical text classification," in *Proc. 59th Annu. Meeting Assoc. Comput. Linguist. 11th Int. Joint Conf. Nat. Lang. Process.*, 2021, pp. 4370–4379. doi: 10.18653/v1/2021.acl-long.337.

[29] H. Zhu, C. Zhang, J. J. Huang, J. R. Wu, and K. Xu, "HiTIN: Hierarchy-aware tree isomorphism network for hierarchical text classification," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguist.*, 2023, pp. 7809–7821. doi: 10.48550/arXiv.2305.15182.