Hyper-Label-Graph: Modeling Branch-Level Dependencies of Labels for Hierarchical Multi-Label Text Classification

Wenmin Deng DENGWM@TJU.EDU.CN
Jing Zhang ZHANG_JING@TJU.EDU.CN
Peng Zhang* PZHANG@TJU.EDU.CN
Yitong Yao YITONGYAO@TJU.EDU.CN
Hui Gao HUI_GAO@TJU.EDU.CN
Yurui Zhang RACHELZ6164@TJU.EDU.CN
College of Intelligence and Computing, Tianjin University, Tianjin, China

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

In the task of Hierarchical Multi-label Text Classification (HTMC), there exist multiple multivariate relations between labels, particularly the semantic dependencies within label branches of the hierarchy. However, existing methods struggle to fully exploit these potential multivariate dependencies since they can only model binary relationships at best. In this paper, we address this limitation by focusing on leveraging semantic dependencies among labels within branches and propose a *Hyper-Label-Graph Model* (HLGM). Specifically, we first construct a label hypergraph based on the taxonomy hierarchy and utilize a hypergraph attention mechanism to learn branch-level multivariate dependencies among labels. Furthermore, the model employs a label-text fusion module to generate label-level text representations, facilitating the comprehensive integration of semantic features between text and labels. Additionally, we introduce a hierarchical triplet loss to enhance the ability to distinguish labels within the hyperedge structure. We validate the effectiveness of the proposed model on three benchmark datasets, and the experimental results demonstrate that HLGM outperforms competitive GNN-based baselines.

Keywords: Text Classification; Hierarchical Multi-label; Hypergraph Learning

1. Introduction

Hierarchical Multi-label Text Classification (HMTC) is a subtask of text classification where labels are organized in a structured hierarchy according to the multivariate semantic relations within labels. As shown in Figure 1, three related news labels *"sport"*, *"ball sport"* and *"soccer"* can be organized in a top-down branch, while multiple related labels can be organized in a tree-like taxonomy hierarchy with several interleaved branches. Therefore, how to adequately leverage these branch-level multivariate relations between labels to make more accurate predictions becomes a key challenge.

To address this challenge, many researchers have introduced various strategies (Kowsari et al., 2017; Mao et al., 2019), such as transfer learning (Banerjee et al., 2019; Linmei et al., 2019), capsule networks (Aly et al., 2019) and recursive regularization (Gopal and Yang, 2013). However, as the taxonomy hierarchy is exactly a tree-like structure, the above methods fail to capture the spatial feature of it. Some subsequent studies proposed

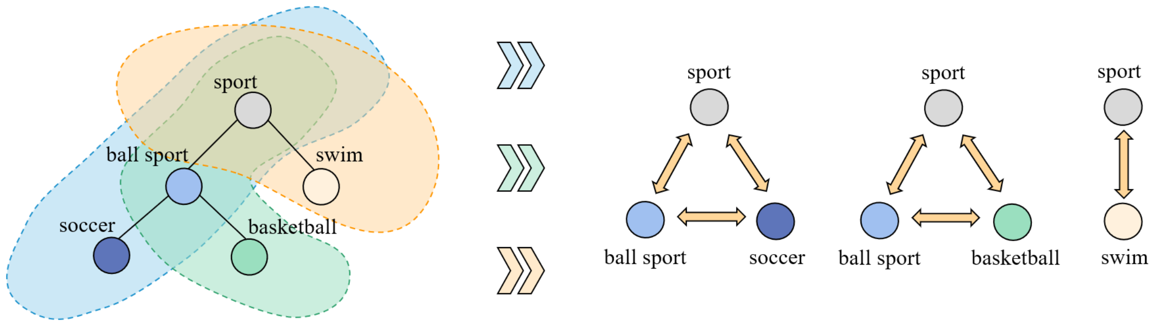

Figure 1: An example of taxonomy hierarchy with multiple spatial multivariate relations.

to formulate the hierarchy as a directed graph, and adopt variants of GNNs to leverage global structural label information relations (Zhou et al., 2020; Chen et al., 2021; Deng et al., 2021). For example, GCN and Tree-LSTM that integrate the label prior hierarchy knowledge are utilized to learn label representations (Zhou et al., 2020). These graph-based methods successfully achieve information propagation in a label-to-label way and process the hierarchical label structure in a global view, which prove to be more robust than previous top-down models.

However, in all of these methods, branch-level multivariate relations on taxonomy hierarchy are failed to model. As in the example of *"sport, ball sport and soccer"*, we know the label *"soccer"* has semantic dependency not only with its father label *"ball sport"*, but also with its grandfather label *"sport"*. Unfortunately, original graph-based methods have trouble modeling these relations well and may introduce noise when aggregating information from longer-distance labels (Feng et al., 2019; Yi and Park, 2020). These methods disassemble branch-level multivariate relations into multiple binary relations and mainly exploit the pairwise connections because of the lack of the connection between multi-hop-neighbor labels on a branch.

In this paper, we focus on capturing the multivariate label relations on branches of taxonomy hierarchy. Hypergraph is a type of graph structure where an edge can connect more than two nodes. Compared with simple graph, hypergraph has significant advantage on encoding non-pair-wise relations with its degree-free hyperedges (Bai et al., 2021), which makes it suitable to be introduced in HTMC. As a result, we propose a novel Hyper-Label-Graph Model (HLGM), where a label hypergraph has been constructed to model the label relations in branch level and a hierarchical triplet loss has been applied to further enhance label discriminative ability.

Specifically, we first construct a hypergraph by connecting each group of labels that are on a top-down branch of taxonomy hierarchy together with hyperedges. Then, a Hypergraph Attention Network is employed to incorporate the attention mechanism into label information propagation. Secondly, we design a label-text fusion layer to generate a set of label-level text representations, which corresponds the most related local features of text for different labels and can be directly fed into the classifier for prediction. Moreover, inspired by triplet loss (Schroff et al., 2015), we propose a hierarchical triplet loss for label-level text representations under the guidance of hyperedge structure. The hierarchical triplet loss aims to pull the label-level text representations for same labels closer and push those with different labels away to varying degrees, therefore encouraging model to learn more discriminative label features.

The contributions of this paper are as follows:

- We construct a label hypergraph to model the multivariate semantic dependencies between hierarchical labels in branch level. As far as we know, this is the first work to introduce hypergraph structure into the HTMC task.
- A hierarchical triplet loss is proposed to enhance the discriminability of label feature based on hyperedges, thereby improving the model's ability of classification.
- We propose a novel end-to-end Hyper-Label-Graph Model (HLGM) which fuses text and label features with a label-text fusion layer. Extensive experiments show that HLGM achieves better performance than the compared GNN-based methods on three datasets.

2. Related Work

Existing methods for HMTC can be divided into local methods and global methods based on their ways of leveraging the label hierarchy. Local methods transform the entire classification problem into multiple local sub-problems and propagate information from top to down of label hierarchy (Koller and Sahami, 1997; Kowsari et al., 2017). Strategies such as transfer learning have been introduced to model dependencies between parent and child labels (Banerjee et al., 2019; Linmei et al., 2019). Global methods, however, coalesce the hierarchical information from a global perspective. Researchers have tried to employ methods such as Hierarchical-SVM (Cai and Hofmann, 2004), recursive regularization (Gopal and Yang, 2013), capsule networks (Aly et al., 2019), meta-learning (Wu et al., 2019) to utilize structural information of top-down branches.

Recently, some studies demonstrate that employing a structure encoder such as GCN and Tree-LSTM to encode the holistic label structure is an effective approach and achieves better performance (Zhou et al., 2020; Lu et al., 2020). HiAGM (Zhou et al., 2020) utilizes hierarchy-GCN and Tree-LSTM that integrate the label prior hierarchy knowledge to learn label representations. Ye et al. (2021) further incorporates meta-data information. However, in all of these GNN-based methods, multivariate semantic dependencies between all labels on a branch of label hierarchy are ignored. Subsequently, HiMatch (Chen et al., 2021) further exploits the correlation between labels on a branch. However, it focuses to capture text-label matching relationships, which is not convincing enough because it is hard to define the semantic similarity between text-level representation and each label in multi-label task.

3. Problem Definition

In practical text classification scenarios, labels are sometimes naturally hierarchical structured, i.e., labels can be organized at different levels of the hierarchy branch based on semantic subordinate relationships. Hierarchical Multi-label Text Classification (HMTC) aims to learn a mapping function $\mathcal{F} : x \to y$ from an input document x to a label space y, where y is a subset of hierarchical label set $\mathcal{Y}$ and the size of set $\mathcal{Y}$ is $|L|$.

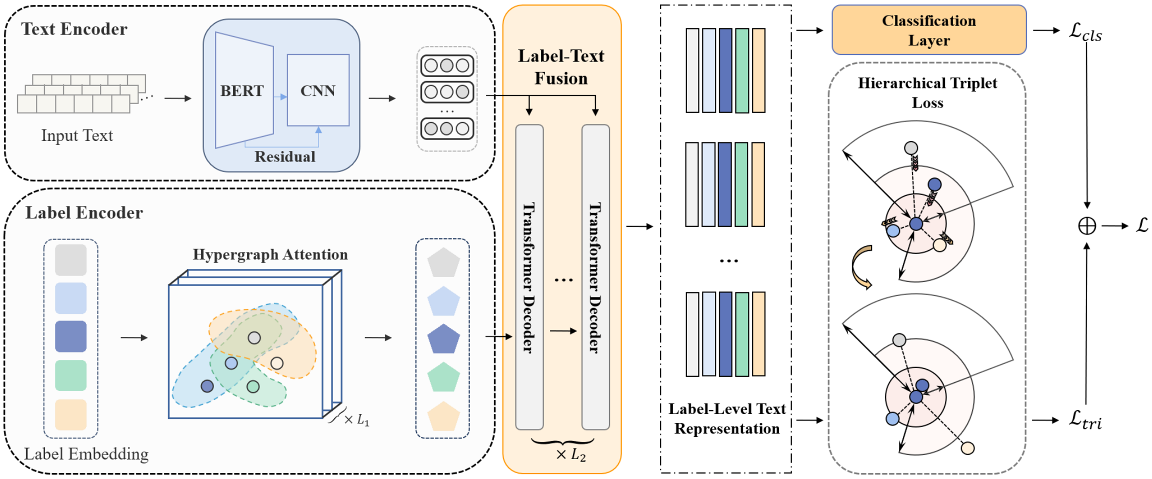

Figure 2: Illustration of HLGM Framework

.

4. Hyper-Label-Graph Model

In this paper, we propose a novel end-to-end Hyper-Label-Graph Model for HTMC. Next, we will introduce our proposed framework in detail, and the overall architecture of the model is shown in Figure 2.

4.1. Text Encoder

We adopt BERT (Devlin et al., 2018) and multiple CNN kernels as our text encoder to capture text contextual information.

Given a document x, we first feed it into BERT as a form of token sequence $\boldsymbol{x} = \{[CLS], x_1, x_2, \ldots, x_{k-2}, [SEP]\}$, where $[CLS]$ is the classification token and $[SEP]$ is the separator token which denotes the end of the sequence here:

$$\boldsymbol{H} = \Phi_{\text{BERT}}(\boldsymbol{x}) \tag{1}$$

where $\Phi_{\text{BERT}}(\cdot)$ denotes the BERT model. The obtained $\boldsymbol{H} = \{\boldsymbol{h}_{[CLS]}, \boldsymbol{h}_1, \ldots, \boldsymbol{h}_{k-2}, \boldsymbol{h}_{[SEP]}\} \in \mathbb{R}^{k \times d}$ contains hidden representations for each token and d is the hidden dimension. Next, we utilize CNN kernels to generate n-gram features and feed the concatenation of these features into a linear layer for feature fusion:

$$\tilde{\boldsymbol{H}} = \text{Linear}(\text{Concat}(\Phi_{\text{CNN}}(\boldsymbol{H}))) \tag{2}$$

where $\Phi_{\text{CNN}}(\cdot)$ denote a CNN layer with multiple CNN kernels. Finally, we add $\boldsymbol{h}_{[CLS]}$ to $\tilde{\boldsymbol{H}}$ to achieve a “shortcut connection” and obtain the text representation $\boldsymbol{S} = \{\tilde{\boldsymbol{h}}_1 + \boldsymbol{h}_{[CLS]}, \tilde{\boldsymbol{h}}_2 + \boldsymbol{h}_{[CLS]}, \ldots, \tilde{\boldsymbol{h}}_k + \boldsymbol{h}_{[CLS]}\} \in \mathbb{R}^{k \times d}$.

4.2. Label Encoder

Taxonomic hierarchy significantly describes the subordinate dependencies between labels in a branch. Therefore, we convert the taxonomic hierarchy into a hypergraph, and adopt attention mechanism to aggregate label information to learn hierarchical-aware label representations for better classification.

4.2.1. Hierarchical-Aware Label Hypergraph

Hypergraph is a type of graph where an edge can connect two or more nodes, and the edges within are defined as hyperedges. With the advantage of hypergraph in modeling high-order correlations among data, we introduce it to model the essential branch-level label dependencies in taxonomy hierarchy. Specifically, we build hyperedges to connect labels on top-down branches of taxonomy hierarchy. In this way, the connectivity of labels on a branch is achieved and information transfer between labels comes more direct, therefore model is able to learn label feature incorporating hierarchical label association.

Formally, we denote the label hypergraph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, consisting of a node set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and a hyperedge set $\mathcal{E} = \{e_1, e_2, \ldots, e_m\}$. The structure of the label hypergraph can be represented by an incidence matrix $\mathbf{A} \in \{0,1\}^{n \times m}$ where each entry $\mathbf{A}_{ij}$ indicates whether the node v_i is in the hyperedge e_j (or whether the label v_i is on the branch e_j). Since each node in the hypergraph correlates a label to be classified and each hyperedge correlates a branch of hierarchy, we denote both a label and a node as v while both a branch and a hyperedge as e:

$$\mathbf{A}_{ij} = \begin{cases} 1, v_i \in e_j \\ 0, v_i \notin e_j \end{cases} \tag{3}$$

For instance, as shown in Figure 3(a), an example taxonomy hierarchy with five labels and three branches is converted into a hierarchical-aware label hypergraph with the shown incidence matrix.

4.2.2. Hypergraph Attention Network

With the constructed label hypergraph, we introduce a module named Hypergraph Attention Network (HGAT), which generalizes attention mechanism on branches to propagate message between branch-related labels. The HGAT allows to learn label representations considering the correlations among labels defined by different branches.

Specifically, we initialize the label features $\boldsymbol{C}^0 \in \mathbb{R}^{|L| \times d}$ with the average of BERT token embedding of corresponding label text, d indicates the dimension of label embedding and is equal to that of BERT output. Due to the specificity of the hypergraph structure, instead of directly propagating information node by node, HGAT performs two aggregation operations separately: first forms branches feature by gathering information from labels on the branches, and then updates labels feature from related branches. Both in these two operations, HGAT learns a dynamic transition matrix to better reveal the relationships between labels and branches, as in Figure 3(b).

We stack L_1 HGAT sub-layers to fully capture multi-hop high-order label relationships. The output of $(l-1)^{th}$ HGAT sub-layer is the input for the l^{th} layer. We will introduce the aggregation operations on the l^{th} as an example to describe Hypergraph Attention Network in detail as follows:

Label To Branch. Given a branch e_i, we generate its representation $\boldsymbol{f}_i$ on l^{th} sub-layer by aggregating information from labels on it. As each label has different correlation

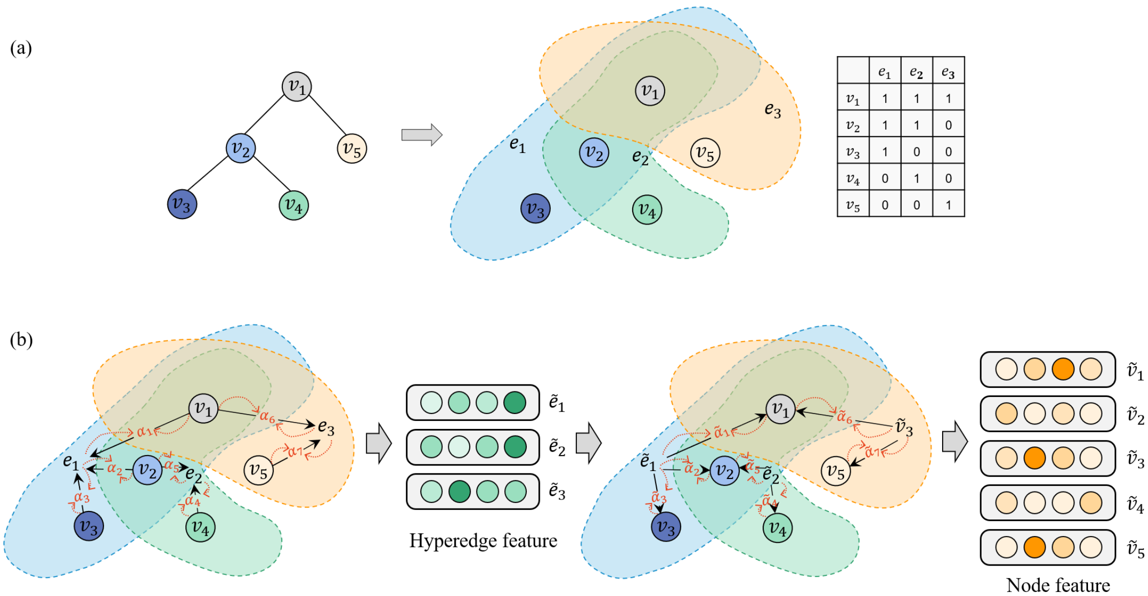

Figure 3: (a) An example of the construction of a label hypergraph with three hyperedges and five nodes. (b) Illustration of the Hypergraph Attention Network.

with the branch, we pay varying attention on the information from labels while aggregating them together and the importance of afferent information flow is calculated with attention mechanism:

$$\boldsymbol{f}_i^l = \sigma\Big(\sum_{v_j \in e_i} \alpha_{ij} \mathbf{W}_1^l \boldsymbol{c}_j^{l-1}\Big) \tag{4}$$

$$\alpha_{ij} = \frac{\exp(\boldsymbol{a}^l \mathbf{W}_2^l \boldsymbol{c}_j^{l-1})}{\sum_{v_k \in e_i} \exp(\boldsymbol{a}^l \mathbf{W}_2^l \boldsymbol{c}_k^{l-1})} \tag{5}$$

where α_{ij} denotes the attention score of label v_j for branch e_i and σ is a nonlinear activation function. $\mathbf{W}_1^l$ and $\mathbf{W}_2^l$ are both trainable weight matrices and $\boldsymbol{a}^l$ is a trainable weight vector in l^{th} layer. $\boldsymbol{c}_j^{l-1}$ here refers to feature of label v_j learned from previous layer.

Branch To Label. The procedure of propagating branches information to labels is similar. Given a label v_i and a branch set $\xi_i = \{e_j | v_i \in e_j\}$, we apply attention mechanism to highlight the informative hyperedges for label v_i and update representation of it:

$$\boldsymbol{c}_i^l = \sigma\Big(\sum_{e_j \in \xi_i} \tilde{\alpha}_{ij} \mathbf{W}_3^l \boldsymbol{f}_j^l\Big) \tag{6}$$

$$\tilde{\alpha}_{ij} = \frac{\exp(\mathbf{W}_4^l \boldsymbol{f}_j^l \mathbf{W}_5^l \boldsymbol{c}_i^{l-1})}{\sum_{e_k \in \xi_i} \exp(\mathbf{W}_4^l \boldsymbol{f}_k^l \mathbf{W}_5^l \boldsymbol{c}_i^{l-1})} \tag{7}$$

where $\mathbf{W}_3^l$, $\mathbf{W}_4^l$ and $\mathbf{W}_5^l$ are trainable weight matrices, and $\tilde{\alpha}_{ij}$ denotes the attention score of branch e_j for label v_i.

Finally, the outputs $\boldsymbol{C} = \boldsymbol{C}^{L_1} \in \mathbb{R}^{|L| \times d}$ of L_1^{th} HGAT sub-layer are the updated label representations incorporating high-order multi-hop label relationships.

4.3. Label-Text Fusion Module

Next, to model the interaction between text semantic features and label semantic features, we propose a label-text fusion module to generate label-level text representations.

As the Transformer Decoder which has built-in attention mechanism is exactly a complete and robust module to capture local discriminative features, we follow the structure to design our fusion module. Specifically, we stack L_2 Transformer Decoder layer where each Decoder is made up of two Multi-Head Attention layers (a self-attention layer and a cross-attention layer) and a Feed-Forward network (FFN). The output of last layer is the input for the next layer.

Since we do not perform auto-regressive generation, we do not use attention masks for Multi-Head Attention. We feed label representations as query, key and value for self-attention layer, while we treat label representations $\boldsymbol{C} \in \mathbb{R}^{|L| \times d}$ as query and text token representations $\boldsymbol{S}$ as key and value for cross-attention layer. The fusion procesure in l^{th} Decoder layer can be formulated by:

$$\boldsymbol{Q}^{l} = \mathrm{Decoder}(\boldsymbol{Q}^{l-1}, \boldsymbol{S}, \boldsymbol{S}) \tag{8}$$

where $\boldsymbol{Q}^0 = \boldsymbol{C}$ in the first layer. As a result, the model capture label-related information from input text via attention mechanism layer by layer. Since the final outputs of label-text fusion module can be regarded as sub components of text for corresponding labels, we name these outputs as label-level text representations and denote them as $\boldsymbol{Q} = \boldsymbol{Q}^{L_2} \in \mathbb{R}^{|L| \times d}$.

4.4. Hierarchical Triplet Loss

Triplet loss aims to pull samples with the same label as close as possible, and push samples with different labels apart from each other. Inspired by this, we propose a hierarchical triplet loss that regards label-level text representations as samples. Our principle idea is: As each label-level representation can be regarded as one aspect of the corresponding text, the representations of different texts but for same label should be similar, while representations for different labels should be different.

Specifically, we create positive sample pairs with label-level representations of different texts but for same label, while negative sample pair are those representations for different labels in a minibatch: Given a batch of N_{batch} texts with a label set $\mathcal{Y}_{batch} = \{y_{ij} \in \{0,1\} | i \in \{1, \ldots, N_{batch}\}, j \in \{1, \ldots, |L|\}$, we have label-level text representations $\boldsymbol{Q}_{batch} = \{\boldsymbol{q}_{ij} \in \mathbb{R}^d | i \in \{1, \ldots, N_{batch}\}, j \in \{1, \ldots, |L|\}$ from label-text fusion module. Firstly, we employ a project network $\mathrm{Proj}(\cdot)$ to map $\boldsymbol{Q}_{batch}$ into the embedding space where hierarchical triplet loss is applied and get new representations:

$$\mathcal{Z} = \{\boldsymbol{z}_{ij} = \mathrm{Proj}(\boldsymbol{q}_{ij}) \in \mathbb{R}^d | \boldsymbol{q}_{ij} \in \boldsymbol{Q}_{batch}\} \tag{9}$$

Next, we define an activate embedding set $\mathcal{A} = \{\boldsymbol{z}_{ij} \in \mathcal{Z} | y_{ij} = 1\}$ that contains label-level representations with active ground-truth labels. With above notations, for a given activate embedding $\boldsymbol{z}_{ij} \in \mathcal{A}$, we can form a set of triplets $\mathcal{T}_{ij} = \{\tau_{ij}^{kpq} = (\boldsymbol{z}_{ij}, \boldsymbol{z}_{kj}, \boldsymbol{z}_{pq}) | \boldsymbol{z}_{ij}, \boldsymbol{z}_{kj}, \boldsymbol{z}_{pq} \in$

$\mathcal{A}, q \neq j\}$ that regards $\boldsymbol{z}_{ij}$ as anchor sample. Therefore, we can define the hierarchical triplet loss $\mathcal{L}_{tri}(\mathcal{T}_{ij})$ for $\boldsymbol{z}_{ij}$ as:

$$\mathcal{L}_{tri}(\mathcal{T}_{ij}) = \sum_{\tau_{ij}^{kpq} \in \mathcal{T}_{ij}} l_{tri}(\tau_{ij}^{kpq}) \tag{10}$$

$$l_{tri}(\tau_{ij}^{kpq}) = \Big[Dis(\boldsymbol{z}_{ij}, \boldsymbol{z}_{kj}) - Dis(\boldsymbol{z}_{ij}, \boldsymbol{z}_{pq}) + \gamma_{ij}^{kpq}\Big]_{+} \tag{11}$$

where $[\cdot] = max(0, \cdot)$ and distance $Dis(\cdot, \cdot)$ here is calculated using the cosine distance. Specially, we consider the multivariate semantic dependencies between labels in branch level when designing the loss. The margin γ_{ij}^{kpq} for triplet τ_{ij}^{kpq} is set according to the dependencies between label v_j and v_q:

$$\gamma_{ij}^{kpq} = \begin{cases} \frac{\exp(\frac{|l_j - l_q|}{|D|})}{|D|}, v_j \in e_a, v_q \in e_a \\ 1, v_j \in e_a, v_q \in e_b, a \neq b \end{cases} \tag{12}$$

where l_j and l_q represent the hierarchical level number of v_j and v_q on the taxonomy hierarchy, and $|D|$ represents the depth of the hierarchy. Thus, for a minibatch in training procedure, the hierarchical triplet loss is calculated as:

$$\mathcal{L}_{tri} = \sum_{z_{ij} \in A} \mathcal{L}_{tri}(\mathcal{T}_{ij}) \tag{13}$$

In this way, each activate embedding $\boldsymbol{z}_{ij} \in \mathcal{A}$ is pulled closer to embeddings under label v_j of different texts, and is pushed away from embeddings under different labels in varying degree. The proposed hierarchical triplet loss further enhance the consistency of label-level representation with same labels, while increasing the dependency of that with relevant labels and strengthening the distinctiveness of that with irrelevant labels.

4.5. Classification and Objective Function

We feed the label-related component of each input document learned by label-text fusion module into a linear layer for classification, and the predicted probability of j^{th} label for i^{th} document can be computed as:

$$\tilde{y_{ij}} = \text{sigmoid}(\mathbf{W}\boldsymbol{c}_{ij}) \tag{14}$$

where $\mathbf{W}$ is the trainable weights of the classification layer. We adopt multi-label cross-entropy loss (BCE loss) as classification loss function and it can be formulated by:

$$\mathcal{L}_{cls} = -\sum_{i=1}^{N} \sum_{j=1}^{|L|} [(y_{ij}\text{log}(\tilde{y_{ij}})) + (1 - y_{ij})\text{log}(1 - \tilde{y_{ij}})] \tag{15}$$

where N is the number of training samples, $y_{ij} \in \{0, 1\}$ is the ground truth for whether i^{th} document belongs to j^{th} label.

Conbining classification loss $\mathcal{L}_{cls}$ and proposed hierarchical triplet loss $\mathcal{L}_{tri}$, we have our final loss function:

$$\mathcal{L} = \mathcal{L}_{cls} + \beta\mathcal{L}_{tri} \tag{16}$$

where β is a trade-off hyperparameter controlling the hierarchical triplet loss weight.

Dataset	$\lvert L \rvert$	Depth	Avg($\lvert L_i \rvert$)	Train	Val	Test
RCV1-v2	103	4	3.42	20,833	2,316	781,265
BGC	146	4	3.01	58,715	14,785	18,394
NYTimes	166	8	7.6	23,345	5,834	7,292

Table 1: Dataset statistics. $|L|$ is the size of label set. Depth is the maximum level of hierarchy. Avg($|L_i|$) is average number of labels for per sample. Train/Test/Val are sizes of train/validation/test set.

5. Experiment

5.1. Experiment Setup

Datasets and Evaluation Metrics. We evaluate the performance of our proposed model on three hierarchical multi-label text classification datasets: RCV1-v2 (Lewis et al., 2004), BlurbGenreCollection-EN (BGC) [1] (Aly et al., 2019) and NYTimes (Sandhaus, 2008). For fair comparison, we apply the same data preprocessing procedure and dataset split for RCV1-v2 and NYTimes as Zhou et al. (2020), and keep the original division ratio of BGC dataset. The statistics of these datasets are illustrated in Table 1. Experimental results are measured with two benchmark metrics Micro-F1 and Macro-F1 following previous work.

Implementation Details. We implement our proposed network with PyTorch. In text encoder module, we adopt *bert-base-uncased* from Transformers (Wolf et al., 2020) as our base architecture. We apply Adam (Kingma and Ba, 2014) with the initial learning rate of $1e-5$ as our optimizer to minimize loss and the learning rate will gradually decrease during the training procedure. The training batch size is set to 16. We set HGAT layer number L_1 to 2 for all datesets, and set transformer decoder layer number L_2 to 1 for RCV1-v2 and 2 for NYTimes and BGC. The loss weight β is set to 0.1. Notebly, we divide our training procedure into two phases based on the loss function model uses: pre-training phase and fine-tuning phase. We train the model with only BCE loss in the pre-training phase, so that in this phase label embeddings are learned freely. Then after embeddings are learned, we start to fine-tune them with combination of BCE loss and hierarchical triplet loss, which encourages model to learn more discriminative label features.

5.2. Baselines

We select four representative graph-based methods as our baselines: (1) HiAGM (Zhou et al., 2020) adopts a bidirectional Tree-LSTM and a hierarchy-GCN as their graph encoder. (2) HiMatch (Chen et al., 2021) formulates the text-label semantics relationship as a semantic matching problem and adopts GCN as graph encoder following HiAGM. (3) HTCInfoMax (Deng et al., 2021) improves HiAGM by regularizing the label representation with a prior distribution. (4) HGCLR (Wang et al., 2022) introduces contrastive learning for the hierarchy-aware representation and uses Graphormer as label graph encoder.

1. BGC dataset is available at `https://www.inf.uni-hamburg.de/en/inst/ab/lt/resources/data/blurb-genre-collection.html`.

Model	RCV1-v2		BGC		NYTimes	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1	Micro-F1	Macro-F1
TextRCNN(Zhou et al., 2020)	81.57	59.25	-	-	70.83	56.18
HiAGM (Zhou et al., 2020)	83.96	63.35	75.42	56.82	74.97	60.83
HTCInfoMax (Deng et al., 2021)	83.51	62.71	76.12	58.56	72.22	60.05
HiMatch (Chen et al., 2021)	84.73	64.11	75.69	55.09	74.53	58.90
BERT (Our implement)	85.83	67.20	78.69	60.86	78.29	65.75
BERT+HiAGM (Our implement)	86.26	67.24	79.27	61.56	78.12	66.20
BERT+HTCInfoMax (Our implement)	85.99	67.65	78.91	61.25	78.32	66.06
BERT+HiMatch (Chen et al., 2021)	86.33	68.66	78.03	62.32	78.56	67.20
HGCLR(Wang et al., 2022)	86.49	68.31	79.13	61.03	78.86	**67.96**
HLGM(Ours)	**86.93**	**69.41**	**80.19**	**63.05**	**78.88**	66.77

Table 2: The performance of HLGM compared with baseline models.

5.3. Results and Analysis

In the comparison experiments, our baseline models can be divided into three categories: models using TextRCNN as encoder, GNN-based models and Graph Transformer-based models using BERT as encoder. Results of BERT, BERT + HiAGM and BERT + HTCInfoMax and results on BGC dataset of all baselines are implemented upon the released projects[2]. As shown in Table 2, the performance of graph-based models are significantly better than models without graph encoder (TextRCNN and BERT). Therefore, we mainly focus on analyzing the results of comparative experiments between graph-based models and HLGM.

Models based on TextRCNN include HiAGM, HTCInfoMax and HiMatch. Our model achieves better performance than these models because we have the strong BERT model as our text encoder and capture fuller semantic information. Our proposed label hypergraph learning and hierarchical triplet loss further improve the ability of model to classify.

Compared with GNN-based models (BERT + HiAGM, BERT + HTCInfoMax and BERT + HiMatch) with BERT encoder, our model performs the best on both RCV1-v1 and BGC. Experimental results are a proof of HLGM's superiority of modeling label relations with hypergraph in branch level, and the effectiveness of hierarchical triplet loss in enhancing the label features.

The HGCLR baseline uses the Graphormer model in encoding labels. Although the Transformer-based graph network also models only label-level dependencies, our model underperforms HGCLR on the NYTimes on Macro-F1 metric due to the strong power of Transformer in HGCLR in feature learning. However, our model outperforms HGCLR on both the NYTimes Micro-F1 metric and other datasets.

Thus, the above analysis shows that the proposed HLGM model generally outperforms strong baseline models in terms of hierarchical multi-label classification ability.

2. Codes are available at HiAGM (`https://github.com/Alibaba-NLP/HiAGM`), HiMatch (`https://github.com/RuiBai1999/HiMatch`), HTCInfoMax (`https://github.com/RingBDStack/HTCInfoMax`), HGCLR (`https://github.com/wzh9969/contrastive-htc`)

Ablation Model	RCV1-v2		BGC	
	Micro-F1	**Macro-F1**	**Micro-F1**	**Macro-F1**
-r.p. HGCN	86.52	68.85	80.17	62.54
-r.p. GAT	85.72	67.50	80.02	62.57
-r.p. GCN	85.92	68.24	79.95	62.19
-r.m. hierarchical triplet loss	86.51	68.91	79.87	62.50
HLGM	**86.60**	**69.26**	**80.19**	**63.05**

Table 3: Ablation studies for different parts in HLGM, where *-r.m.* refers to removing the module and *-r.p.* refers to replacing.

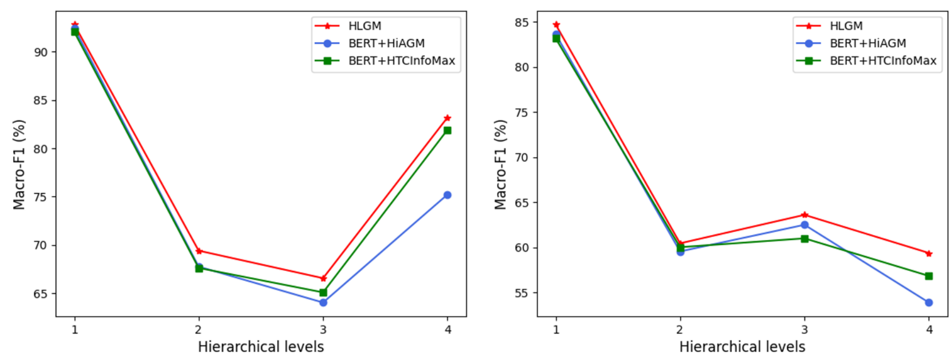

Figure 4: Level-based Macro-F1 on RCV-v2 (left) and BGC (right).

5.4. Ablation Study

To investigate the contribution of each module in HLGM, we conduct a series of ablation experiments in this section, and the results are reported in Table 3.

Firstly, we replace Hypergraph Attention Network (HGAT) with Hypergraph Convolution Network (HGCN) (*r.p.* HGCN), which is another hypergraph learning method aggregating node information without attention mechanism. The result shows both Micro-F1 and Macro-F1 will decrease when employing HGCN as graph encoder instead of HGAT, which proves that introducing an attention learning module to learn a dynamic incidence matrix helps better describe label relationships. Besides, we also remove the construction of label hypergraph but adopt GCN and GAT directly on label tree to update label representations (*r.p.* GCN and *r.p.* GAT). The results show both HGAT and HGCN outperforms GAT and GCN, even if GAT also involves the attention mechanism, which further proves the validity of hypergraph.

Apart from that, we remove the hierarchical triplet loss from HLGM and only adopt BCE loss as the objective function for training (*r.m. hierarchical triplet loss*). The results show that both two metrics decrease especially Macro-F1, which shows the effectiveness of hierarchical triplet loss to improve the ability to distinguish labels. Notebly, our report results are from average of the results of the two experiments.

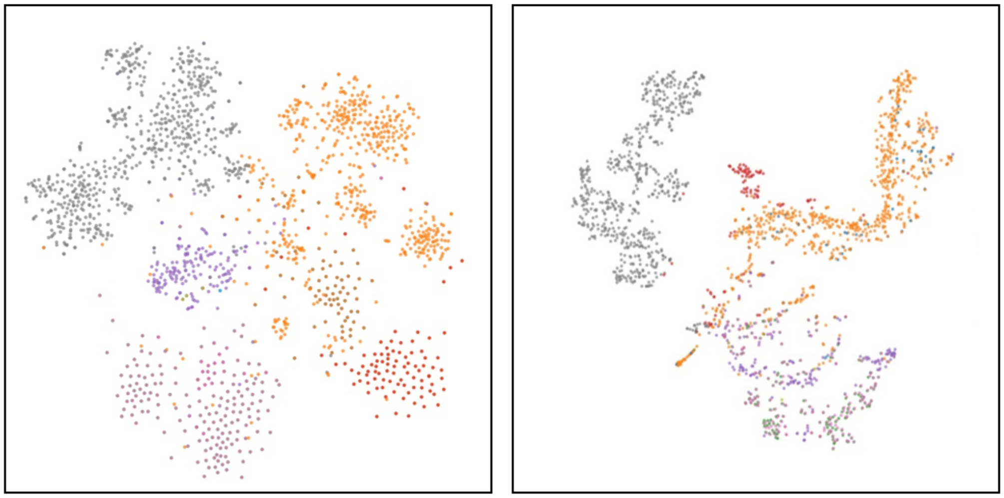

Figure 5: T-SNE visualization for label-level text representations training with only BCE loss (left), and with the combination of BCE loss and hierarchical triplet loss (right). Each dot represents one label-level text representations and different colors correspond to different labels.

5.5. Hypergraph Effect

Table 2 has shown the superiority of our method compared with baseline models on overall label set. In addition, performances on different hierarchical levels of label set are also of analytical value. We compute the level-based Macro-F1 of our model, BERT + HiAGM and BERT + HTCInfoMax and the results are shown in Figure 4. From the line charts we can observe that our model achieves better performance on all levels, especially on deeper levels.

In the taxonomy hierarchy, labels on deeper levels are appear less frequently and more fine-grained, which results in the insufficient training and makes it more difficult to learn their semantic features. Different from the compared models that apply GNNs as graph encoder, our method propose to augment the connectivity of labels on a branch with hyperedges and utilize the essential branch-level label dependencies to use the knowledge of upper-level labels in better learning representations of lower-level labels. In this way, the superiority of hypergraph becomes more apparent as level gets deeper.

5.6. Hierarchical Triplet Loss Effect

The goal of our proposed hierarchical triplet loss is to push label-level representations for same labels closer, and push representations for another labels away to different degrees according to label semantic relationships. To demonstrate its effectiveness in a clearer view, we use T-SNE to visualize the learned label-level text representations training with and without hierarchical triplet loss on BGC dataset for comparison.

More than 90 percent of innovation initiatives fail to achieve their return-on-investment targets. Poor management decisions and lack of marketplace savvy often undermine even huge research efforts. Can America continue to be a formidable global competitor with this kind of failure rate?

Target Label: *Business*

An original Disney Princess Little Golden Book starring Jasmine as a super-fun babysitter!A Disney Princess makes the best babysitter! Children ages 2 to 5 will love this Little Golden Book, which tells a fun story about Princess Jasmine spending the day taking care of a young princess from a nearby kingdom. Disney's Dream Big, Princess campaign focuses on the amazing stories and leadership qualities of each Disney Princess and shows how they relate to today's girl.

Target Label: *Children's Books*

A dazzling and haunting vision of the end of the world, Ice is a masterpiece of literary science fiction now in a new 50th anniversary edition with a foreword by Jonathan Lethem. One might become convinced that Kavan had seen the future . . .

Target Label: *Fiction Classics*

Figure 6: Visualization of label-text attention weights. The attention weights of "Target Label" are shaded in different colors. Note that darker color represents higher weight score.

As shown in Figure 5, compared with adopting BCE loss only, the clusters of label-level text representations that learned with hierarchical triplet loss have clearer boundaries, which visually demonstrates the ability of hierarchical triplet loss to enhance the discriminativeness of labels. As the distinction between labels becomes more apparent, the classifier is then able to achieve better classification.

5.7. Visualization of Label-Text Fusion

To gain a clearer view of the effectiveness of the label-text fusion module in modeling text and label semantic features, we present some concrete cases and visualize the attention weights between texts and labels from BGC dataset.

As shown in Figure 6, in the first case, label *"Business"* have higher attention scores with words like "formidable", "innovation", "achieve", "return" and etc, which are correlated with business. In the second case, label *"Children's Books"* pays more attention to words like "Dream","amazing", "princess" and "Little" which are all common words in fairy tales. In the third case, *"Fiction Classics"* is more related to "edition","masterpiece" and "literary".

6. Conclusion

In this paper, we proposed a novel end-to-end Hyper-Label-Graph Model (HLGM). We converted taxonomy hierarchy to a label hypergraph and learn the branch-level multivariate dependencies of the hierarchy by hypergraph attention mechanism. Moreover, based on constructed hypergraph, we proposed a hierarchical triplet loss to encourage model to learn more discriminative label features, thereby achieving better classification accuracy. Finally, experiments show that our proposed model outperforms other compared methods on three datasets, and the effectiveness of all components in our model are verified.

References

Rami Aly, Steffen Remus, and Chris Biemann. Hierarchical multi-label classification of text with capsule networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 323–330, 2019.

Song Bai, Feihu Zhang, and Philip HS Torr. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110:107637, 2021.

Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsioutsiouliklis. Hierarchical transfer learning for multi-label text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6295–6300, 2019.

Lijuan Cai and Thomas Hofmann. Hierarchical document categorization with support vector machines. In *Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 78–87, 2004.

Haibin Chen, Qianli Ma, Zhenxi Lin, and Jiangyue 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)*, pages 4370–4379, 2021.

Zhongfen Deng, Hao Peng, Dongxiao He, Jianxin Li, and Philip S Yu. Htcinfomax: A global model for hierarchical text classification via information maximization. *arXiv preprint arXiv:2104.05220*, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. Hypergraph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3558–3565, 2019.

Siddharth Gopal and Yiming 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*, pages 257–265, 2013.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Daphne Koller and Mehran Sahami. Hierarchically classifying documents using very few words. Technical report, Stanford InfoLab, 1997.

Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber, and Laura E Barnes. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, pages 364–371. IEEE, 2017.

David D Lewis, Yiming Yang, Tony Russell-Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr): 361–397, 2004.

Hu Linmei, Tianchi Yang, Chuan Shi, Houye Ji, and Xiaoli Li. Heterogeneous graph attention networks for semi-supervised short text classification. 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)*, pages 4821–4830, 2019.

Jueqing Lu, Lan Du, Ming Liu, and Joanna Dipnall. Multi-label few/zero-shot learning with knowledge aggregated from multiple label graphs. *arXiv preprint arXiv:2010.07459*, 2020.

Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. Hierarchical text classification with reinforced label assignment. *arXiv preprint arXiv:1908.10419*, 2019.

Evan Sandhaus. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752, 2008.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

Zihan Wang, Peiyi Wang, Lianzhe Huang, Xin Sun, and Houfeng Wang. Incorporating hierarchy into text encoder: a contrastive learning approach for hierarchical text classification. *arXiv preprint arXiv:2203.03825*, 2022.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

Jiawei Wu, Wenhan Xiong, and William Yang Wang. Learning to learn and predict: A meta-learning approach for multi-label classification. 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)*, pages 4354–4364, 2019.

Chenchen Ye, Linhai Zhang, Yulan He, Deyu Zhou, and Jie Wu. Beyond text: Incorporating metadata and label structure for multi-label document classification using heterogeneous graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3162–3171, 2021.

Jaehyuk Yi and Jinkyoo Park. Hypergraph convolutional recurrent neural network. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3366–3376, 2020.

Jie Zhou, Chunping Ma, Dingkun Long, Guangwei Xu, Ning Ding, Haoyu Zhang, Pengjun Xie, and Gongshen Liu. Hierarchy-aware global model for hierarchical text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1106–1117, 2020.