HCL-MTC: Hierarchical Contrastive Learning for Multi-label Text Classification

Anonymous EMNLP submission

Abstract

Multi-label text classification is a big challenging subtask in text classification, where labels generally form a tree structure. Existing solutions learn the label tree structure in a shallow manner and ignore the distinctive information between labels. To address this problem, we propose a Hierarchical Contrastive Learning for Multi-label Text Classification (HCL-MTC), which constructs the graph based on the contrastive knowledge between labels. Specifically, we formulate the MTC as a multi-task learning by introducing a sampling hierarchical contrastive loss, which learns both the correlative and distinctive label information and is beneficial in learning deep label hierarchy. The experimental results show that the proposed model can achieve considerable improvements on both public datasets (i.e., RCV1-v2 and WoS).

1 Introduction

Text classification is a fundamental task in natural language processing, which has attracted increasing attention recently. Text classification has been widely used in many applications such as sentiment analysis (Pang and Lee, 2008; Li et al., 2020; Ding et al., 2020), document classification (Yang et al., 2016), medical codes prediction (Mullenbach et al., 2018), law study (Chalkidis et al., 2019), patent categorization (Tang et al., 2020), and financial study (Maia et al., 2021). Multi-label text classification (MTC) is one of the most challenging subtasks, where the classification result contains more than one label where label set generally forms a tree structure, i.e., there exists relationships between each label and one label can be inferred based on the information of another.

Existing solutions for MTC task can be divided into two groups: 1) predicting labels simply from text information and 2) predicting labels from hybrid information of both labels and texts. The first group predicts text labels by utilizing the local and global information extracted from text encoders. Previous works (Shimura et al., 2018; Yang et al., 2020a) proposed CNN-based models to overcome the data imbalance problem caused by lack of child label samples. Some other works (e.g. Lin et al., 2018) tend to utilize the semantic information from text. These methods only focus on text information but ignore the information between labels. The second group tends to combine text information with label information, such as weight initialization (Baker and Korhonen, 2017), label hierarchy learning (Huang et al., 2019), and capsule network (Chen et al., 2020). While these approaches increased the efficiency of multi-label text classification by including label information, they learned the label hierarchy in a shallow manner. The GCN-based model proposed by Zhou et al. (2020) is able to learn deep label hierarchy. However, they do not take full advantage of the label information as they only learned label correlative information but ignore the label distinctive information.

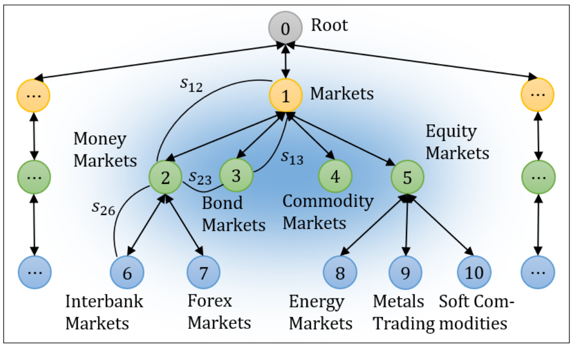

Figure 1: Sample of label tree structure from RCV1-v2 dataset where grey, yellow, green, blue denotes root, first-level, second-level, third-level labels respectively. The variable s_{ij} indicates the similarity between label i and label j .

Learning both correlative and distinctive information is beneficial in learning deep label hierarchy and thus improves classification effect for MTC. For instance, in Figure 1, the similarity s_{23}

between node 2 and node 3 denotes the distinctive information which is assumed to be as large as possible since there is no edge connects them and the similarity s_{26} denotes the correlative information which is assumed to be as small as possible. In this paper, we propose a Hierarchical Contrastive Learning for Multi-label Text Classification (HCL-MTC). In order to demonstrate the efficiency of our contrastive learning method in modelling the label hierarchy, we adopt the state-of-art GCN framework and compare the results in Zhou et al. (2020). The HCL-MTC explicitly models the hierarchical label structure as a directed graph and formulates the graph edge as the contrastive knowledge between labels. To further increase the performance of the label contrastive learning, we introduce a sampling hierarchical contrastive loss function. The goal of the contrastive loss is to maximize the distinction between parent labels and minimize the similarity between parent and child labels.

Specifically, given train texts, the model first generates text features based on the local and global information extracted from the text encoder. A single linear transformer then transforms the text feature to the label-wise feature. Finally, the contrastive learner aggregates the information of each label from its correlated labels based on their contrastive knowledge.

Our main contributions can be summarized as follows:

- We propose a Hierarchical Contrastive Learning for Multi-label Text Classification (HCL-MTC). The HCL-MTC models the label tree structure as a directed graph and constructs the graph based on the contrastive knowledge between labels.

- To further utilize the label contrastive knowledge, we propose a sampling hierarchical contrastive loss which can increase the performance for MTC.

- Experimental results on two public datasets demonstrate the effectiveness of HCL-MTC.

2 Related Work

Multi-label Text Classification aims to assign labels with hierarchical structure to the given text. Existing solution for MTC can be categorized into text information based approach and hybrid information based approach.

Text Information based Approaches: Since a text contains rich information from both word level and sentence level, previous studies (e.g. Yang et al., 2016) have developed various methods to take advantage of this information to predict hierarchical labels. Convolutional Neural Network (CNN) (Kim, 2014) based methods have been widely used in MTC task due to its local performance. To name a few, Lin et al. (2018) proposed a Seq2Seq model which utilizes dilated convolution and hybrid attention method to capture the semantic unit from texts. Shimura et al. (2018) proposed a fine-tuning technique in CNN which attempts to contribute upper level information to lower levels. Yang et al. (2020a) integrated two single CNNs using siamese approach for tail categories. However, the above mentioned models only used information extracted from texts and ignored the relationship between labels.

Hybrid Information based Approaches: In order to incorporate label information, various approaches have been proposed. For instance, Baker and Korhonen (2017) initialized the final hidden layer of a CNN model such that it can leverage the label co-occurrence relations. Chen et al. (2020) proposed a capsule network which incorporates the label probabilities. Some existing methods incorporate label embedding vectors to the model and learn the label structure from upper levels to lower levels (Huang et al., 2019; Yang et al., 2018). However, these methods learn the label hierarchy in a shallow manner. Since labels in MTC task can be formulated as tree structure or directed acyclic graph (DAG) structure. Recently, GCN-based models (Peng et al., 2018; Zhou et al., 2020) have obtained promising performance on the MTC task. These models formulate the edge feature based on word co-occurrence or label dependencies which are over-reliance on the prior probability.

Edge Feature Formulation in GCN-based Model: Traditional GCN-based models (Marcheggiani and Titov, 2017; Lu et al., 2020) formulate the adjacency matrix by random initialization. Some works (Yao et al., 2018; Henaff et al., 2015; Peng et al., 2018) define weight of edge by word information such as word co-occurrence, word similarity and point-wise mutual information. In contrast, we formulates the edge feature based on contrastive knowledge between labels.

3 Model

We propose a Hierarchical Contrastive Learning for Multi-label Text Classification (HCL-MTC) in which the contrastive learning methods are represented in two aspects: 1) the transition matrix parameter of GCN, and 2) the sampling hierarchical contrastive loss. We first introduce the problem formulation, then describe our proposed Hierarchical Contrastive Learning for Multi-label Text Classification (HCL-MTC).

3.1 Problem Formulation

In the MTC task, there are m predefined labels $L = \{l_1, l_2, ..., l_m\}$. Given a training set $\{(T_1, Y_1), (T_2, Y_2), ..., (T_N, Y_N)\}$, where $T_i = \{x_1, x_2, ..., x_n\}$ indicates the i^{th} text, n indicates the text length, x_i indicates the i^{th} word and Y_i is the subset of L assigned to T_i. The goal of the MTC task is to predict $\hat{y}_i$ for each test text. Note that: i) every text has one or more labels; ii) labels generally form a tree structure, which indicates that there exists both correlative and distinctive information between labels; iii) The sample size of the child node is much lower than that of its parent node.

3.2 Hierarchical Contrastive Learning for MTC

As demonstrated in Figure 2, our proposed model contains four parts, a text encoder, a feature extractor, a linear transformer and a hierarchical contrastive learner. Given a sentence, the text encoder and the feature extractor extract local and global information as text feature. The linear transformer transforms the text feature to the label-wise feature, which directly changes the text feature dimension to the label feature dimension. The hierarchical contrastive learner learns the contrastive knowledge between labels and considers it as the transition probability. The overall model structure is depicted in Figure 2.

Input: Before transferring to the text encoder, the original text is embedded by the pre-trained embedding matrix. Given a text $T = \{x_1, x_2, ..., x_n\}$, each of word x_i will be converted to the vector ω_i which constructs the input matrix $I = \{\omega_1, \omega_2, ..., \omega_n\}$.

Text Encoder: A variety of text encoders have been used to extract global information within texts, for instance, RNN (Werbos, 1990) and its variants (e.g. LSTM (Hochreiter and Schmidhuber, 1997), GRU (Cho et al., 2014)). Recently, pretraining model with fine-tune procedure (e.g BERT (Devlin et al., 2019), XLNet (Yang et al., 2020b))has shown great performance in many NLP tasks and can also be utilised as the text encoder. For the purpose of experimental comparison, we adopt the same text encoder (i.e. Bi-GRU) proposed in Zhou et al. (2020). The input of the Bi-GRU encoder layer is a matrix $I = \{\omega_1, \omega_2, ..., \omega_n\}$, and the hidden vector of a Bi-GRU is calculated as follows:

$$\begin{aligned}\overrightarrow{h}_t &= GRU(\overrightarrow{h}_{t-1}, \omega_t),\\ \overleftarrow{h}_t &= GRU(\overleftarrow{h}_{t+1}, \omega_t),\\ h_t &= [\overrightarrow{h}_t, \overleftarrow{h}_t],\end{aligned} \tag{1}$$

where $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ are the forward hidden vector and backward hidden vector at time step t. The output $h_t \in \mathbb{R}^{2\mathbb{u}}$ of the Bi-GRU is the concatenation of $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ where $\mathbb{u}$ indicates the number of hidden units of each unidirectional GRU. The resulting global feature maps are $H = \{h_1, h_2, ..., h_n\}$.

Feature Extractor: We apply the CNN model to extract n-gram feature from global feature maps H get from the text encoder. Let $F \in \mathbb{R}^{g \times 2\mathbb{u}}$ denotes a convolutional kernel and $H_{i:i+g-1}$ denotes a global feature map region of g words. The local feature can be formulated as follows:

$$c_i = F \odot H_{i:i+g-1} + b, \tag{2}$$

where $\odot$ denotes the component-wise multiplication, and $b \in \mathbb{R}$ denotes a bias term. The feature maps of f filters at i^{th} channel can be denoted as $C_i = \{c_i^1, c_i^2, ..., c_i^f\}$. Next, we apply the k-max pooling method to filter the top k most informative word combinations which can be formulated as follows:

$$\begin{aligned}P =& flatten(\\ & max(k, [C_1, C_2, ..., C_{n-g+1}]))\end{aligned} \tag{3}$$

Suppose K convolutional kernels are used and the final text feature is the concatenation of the output P denoted as $O = [P^1, P^2, ..., P^K]$.

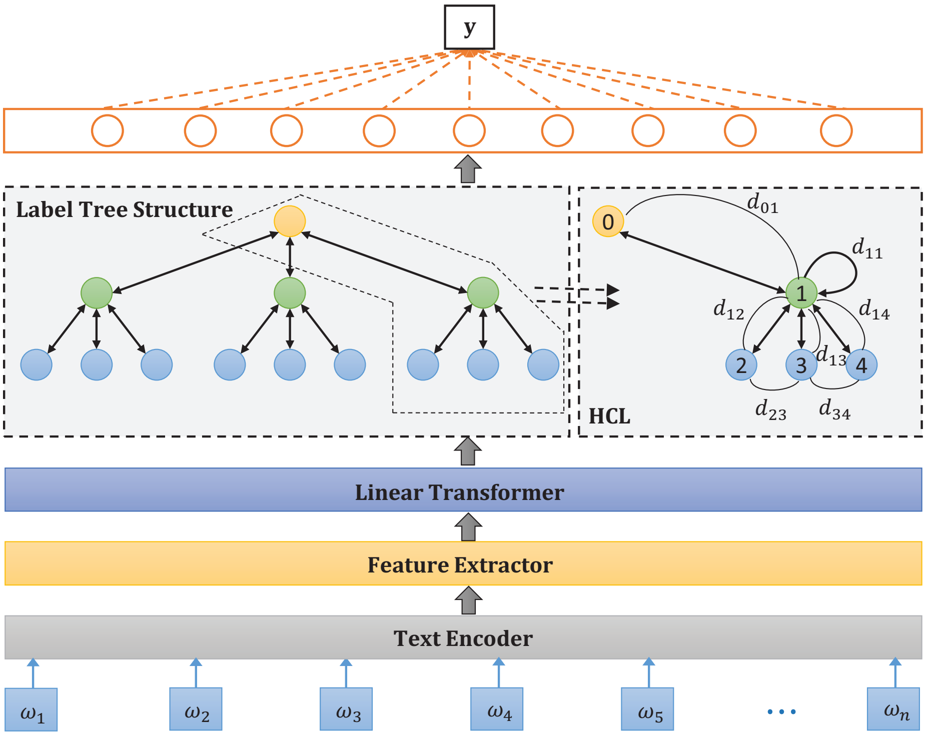

Figure 2: The overall structure of our HCL-MTC model.

Linear Transformer: The linear transformer transforms the text feature to the label feature:

$$V = Reshape(MO), \tag{4}$$

where $O \in \mathbb{R}^{d_c}$ is the text feature, $M \in \mathbb{R}^{d_w \times d_c}$ is the trainable weight matrix, $V \in \mathbb{R}^{m \times d_n}$ denotes the label feature, the reshape operation change the size from d_w to $(m \times d_n)$, d_c, m, and d_n denotes the text feature length, number of labels and label feature dimension respectively.

Hierarchical Contrastive Learner: GCN (Kipf and Welling, 2017) is a graph representations of structural information between nodes (e.g classification labels). The graph edge in a graph represents the relationship between each node. Traditional GCNs (Marcheggiani and Titov, 2017; Lu et al., 2020) randomly initialize the transition matrix and learn the relationship among nodes by error back propagation method, which ignores the node correlation information. Zhou et al. (2020) overcomes the issue by formulating the edge feature by the prior probability of label dependencies. While they learn the label correlative information, they ignore the label distinctive information. In contrast to Zhou et al. (2020), we propose the hierarchical contrastive learner which connects graph nodes by label contrastive knowledge.

Our proposed hierarchical contrastive learner adopts the framework of Hierarchy-GCN proposed in Zhou et al. (2020). The label tree constructs a directed graph where the current node can aggregate the information transferred from parent nodes, child nodes and itself. This operation is realized by a weighted adjacent matrix learned from the contrastive information between label nodes.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph where $\mathcal{V} \in \mathbb{R}^{m \times d_n}$ indicates set of nodes, $\mathcal{E}$ indicates set of edges. We define $v_k \in \mathbb{R}^{d_n}$ as the feature of node k and $N(k) = \{n_k, child(k), parent(k)\}$ as its connected neighbourhood. The hidden state of node k are computed as follows:

$$\begin{aligned} a_{j,k} &= \left| \frac{v_j \cdot v_k}{||v_j|| \cdot ||v_k||} \right|, \\ \mu_{j,k} &= a_{j,k} v_j + b_l^k, \\ g_{j,k} &= \sigma(W_g^{d(j,k)} v_j + b_g^k), \\ h_k &= ReLU(\sum_{j \in N(k)} g_{j,k} \cdot \mu_{j,k}), \end{aligned} \tag{5}$$

where $W_g^{d(j,k)} \in \mathbb{R}^n$ indicates the gate weight in the direction from node j to node k, b_l^k and b_g^k indicate the transferring bias and gate bias of node k respectively. $\mu_{j,k} \in \mathbb{R}^n$ denotes the information of node j transfers to node k and $g_{j,k} \in \mathbb{R}^n$ controls the flow of information. Edge $d(j,k)$ contains three directions, including top-down, bottom-up and self-loop. $a_{j,k}$ is the contrastive knowledge computed by cosine similarity, which indicates the transition probability from node j to node k. With the decrease of the similarity, one node is more likely to transfer to another. Thus, the model can learn deep along hierarchical label structure. Note that $a_{j,k}$ from top-down flow and $a_{k,j}$ from bottom-up flow is equal and for self-loop, $a_{k,k} = 1$. Finally, the output hidden state h_k of node k denotes the aggregation of information transferred from its neighborhood in $N(k)$ activated by ReLU activation function.

3.3 Sampling Hierarchical Contrastive Loss

Let $s(v_{p_i}, v_{p_j})$ denote the similarity between parent node i and parent node j; $s(v_{p_i}, v_{c_k})$ denote the similarity between parent node i and its child node k. In a label tree, a parent-child label pair is able to transfer information in both directions but a parent can not transfer information to other parents. This indicates that there exists correlative and distinctive information between labels which is the purpose of the hierarchical contrastive loss. Thus, the goal of the hierarchical contrastive loss is to maximize the distinctive information $s(v_{p_i}, v_{p_j})$ and minimize the correlative information $s(v_{p_i}, v_{c_k})$. The hierarchical contrastive loss can be formulated as follows:

$$
\begin{aligned}
s(v_{p_i}, v_{p_j}) &= \left| \frac{v_{p_i} \cdot v_{p_j}}{||v_{p_i}|| \cdot ||v_{p_j}||} \right|, \\
s(v_{p_i}, v_{c_k}) &= \left| \frac{v_{p_i} \cdot v_{c_k}}{||v_{p_i}|| \cdot ||v_{c_k}||} \right|, \\
L_d &= \sum_{p_i \in \mathcal{V}} \sum_{p_j \in \mathcal{V}} \sum_{c_k \in child(i)} \\
&\quad exp(s(v_{p_i}, v_{p_j}) - s(v_{p_i}, v_{c_k}))
\end{aligned}
\tag{6}
$$

However, to enumerate all node pairs can be time-consuming. Instead, we apply the sampling mechanism. For each level, only two randomly selected parent nodes and one randomly selected child node will participate in the calculation of the hierarchical contrastive loss.

3.4 Classification

The final node features are fed into a fully connected layer and the probability of node k can be formulated as follows:

$$
p_k = \sigma(W_k h_k + b^k), \tag{7}
$$

where $W_k \in \mathbb{R}^n$ and $b^k \in \mathbb{R}^n$. The model will assign labels with probability greater than the preset threshold θ to a test text.

3.5 Loss Function

The HCL-MTC applies three losses, including binary cross-entropy loss, recursive regularization loss(Gopal and Yang, 2013) and sampling hierarchical distance loss. The total loss can be formulated as:

$$
\begin{aligned}
L_c = &- \sum_{i=1}^{m} [y_i log(y'_i) \\
&+ (1 - y_i) log(1 - y'_i)], \\
L_r = &\sum_{i \in \mathcal{V}} \sum_{j \in child(i)} \frac{1}{2} ||\omega_i - \omega_j||^2, \\
L = &L_c + \lambda_1 L_r + \lambda_2 L_d,
\end{aligned}
\tag{8}
$$

where L_r utilizes the parameters of the final fully connected layer, y_i indicates the ground truth and y'_i denotes the predicted probability of label i.

4 Experiments

4.1 Dataset Description

We evaluate the effectiveness of our proposed model on two published dataset, including RCV1-v2 and Web-of-Science (WoS).

Reuters Corpus Volume I (RCV1-v2): This dataset is a correction version of the original data RCV1-v1 provided by (Lewis et al., 2004) for research purposes. It includes a total of 804,414 manually categorized newswire stories and 103 topics where each newswire stories can be assigned multiple topics.

Web of Science (WoS): The WoS dataset is a collection of meta-data on 46985 published papers provided by (Kowsari et al., 2017) which consists of abstract, domain and keywords. The abstract is regarded as the input for text classification and the domain is the label with hierarchy. The keywords are descriptions of the next label level. There are 141 domains in total.

Dataset	# Total	# Train	# Valid	# Test	$|\mathcal{C}|$	# Depth	Avg Words	Avg $|\mathcal{C}|$
RCV1-v2	804414	22917	232	781265	103	4	136.54	3.24
WoS	46985	30070	7518	9397	141	2	131.19	2.0

Table 1: Statistics of the datasets where $|\mathcal{C}|$ indicates total label numbers and Avg $|\mathcal{C}|$ indicates average label numbers in each text.

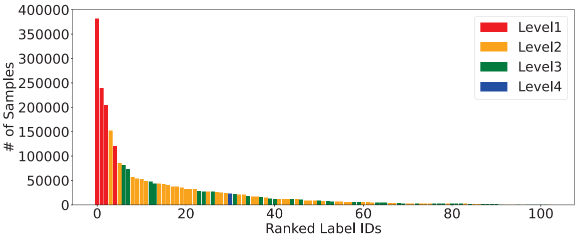

Figure 3: Text number distribution in label hierarchy of RCV1-v2 dataset.

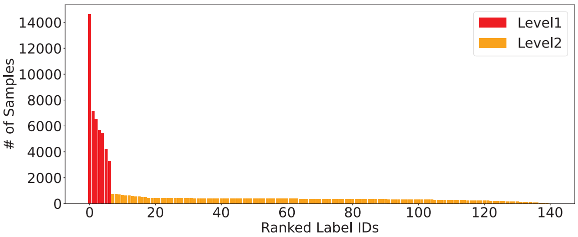

Figure 4: Text number distribution in label hierarchy of WoS dataset.

Figure 3 and Figure 4 show the text number distribution in label hierarchy of RCV1-v2 dataset and WoS dataset respectively. From two distributions, we can observe that most texts are in high levels and with the label hierarchy goes deeper, the number of texts decreases, which lead to the data imbalance problem. The explanation of this phenomenon is that labels form a tree structure and the lower leaves are constructed based on the upper leaves. Therefore, it is important to learn the label hierarchy and contribute high-level label information to the low-level labels.

We remove the English stop words for the two datasets and divide each each dataset into training, validation and test follows Zhou et al. (2020). The only difference is that we choose 1% training set as the validation set. The statistics of the datasets is shown in Table 1.

4.2 Experimental Setup

Evaluation Metrics: We use the standard evaluation metrics of Micro-F1 and Macro-F1 (Gopal and Yang, 2013) to measure our experimental results.

- **Micro-F1** considers the overall performance of the model which is calculated by overall precision and recall of all the labels.
- **Macro-F1** considers the local performance of the model which gives equal weight to all labels.

To be specific, the computation of Micro-F1 score and Macro-F1 score are illustrated below:

$$\begin{aligned} microF1 &= \frac{2\sum_{l\in L} TP_l}{\sum_{l\in L} 2TP_l + FP_l + FN_l}, \\ macroF1 &= \frac{1}{|L|}\sum_{l\in L}\frac{2TP_l}{2TP_l + FP_l + FN_l}, \end{aligned} \tag{9}$$

where TP_t, FP_t, FN_t indicates the true-positives, false-positives and false-negatives for the label $l \in L$.

Baselines: We compare our proposed model with multiple traditional MLP baselines for all datasets and the state-of-art models on RCV1-v2 dataset. The baselines and their performance are reported in Zhou et al. (2020).

- Traditional MLP baselines: CNN (Kim, 2014) is a local method which uses multiple convolution kernels to extract text information and MLP to predict labels. RNN is a global method which employs a variational Bi-GRU network (Cho et al., 2014) to learn the word dependencies in a long distance. RCNN is a combination of above two methods which first extracts global text features and then feeds to the CNN architecture to extract the local information.
- State-of-art models: HR-DGCNN (Peng et al., 2018) employs deep CNN to extract the local text information from graph word embedding of documents for HMTC and add the recursive regularization to the final MLP. HE-AGCRCNN (Peng et al., 2019) is similar to HR-DGCNN which proposes an attentional capsule RCNN netwrok for HMTC. HiLAP (Mao et al., 2019) is a deep reinforcement

Description	Values	Description	Values	Description	Values
GRU depth	1	Learning rate	0.0001	Train batch size	64
GRU hidden units	64	Prediction threshold	0.5	Test batch size	512
CNN depth	3	Dropout	0.5	Momentum β_1	0.9
CNN filter region size	{2,3,4}	GRU dropout	0.1	Momentun β_2	0.999
Token length	256	Node dropout	0.05	Momentum ϵ	1×10^{-6}

Table 2: Implementation details: Dropout shows the dropout rate in the embedding layer and MLP layer, GRU dropout shows the dropout rate in the Bi-GRU layer and Node dropout shows the dropout rate in the node transformation layer.

learning based model which aims to learn the label assignment policy for HMTC. HMCN (Wehrmann et al., 2018) is a deep neural network for HMTC which aggregates the information of local and global data flow in the label hierarchy. HFT(M) (Shimura et al., 2018) is a CNN-based model with fine-tune mechanism which forces the low-level inference utilize the high-level information. Similar to HFT, HTrans (Banerjee et al., 2019) learns the label hierarchy by utilizing the parameter of parent category classifiers to fine-tune the child category classifiers. SGM (Yang et al., 2018) models HMTC as a Seq2Seq task which predicts the current label based on the previous predicted label. HiAGM (Zhou et al., 2020) employs RCNN to extract the text information and structure encoder to learn the label hierarchy.

Inplementation Details: All experiments are implemented in PyTorch (Paszke et al., 2017). To be comparable with (Zhou et al., 2020), we take the similar implementation parameters. The word embedding vector is initialized by 300-dimentional word embedding pretrained by GloVe (Pennington et al., 2014). We use a maximum size of 60000 most frequent words as vocabulary and remove words under the minimum count of 2. We use Adam (Kingma and Ba, 2017) optimizer to minimize the total loss. We set the penalty coefficient of recursive regularization to 1×10^{-6} and the penalty coefficient of sampling hierarchical distance loss to 1×10^{-5}. The maximum number of epochs is set to 400 and the model is stopped when there is no improvement in 50 epochs. Other implementation details is shown in Table 2.

4.3 Experimental Results

We evaluate our proposed model on two public datasets and compare it with 12 MLP baselines and state-of-art models in terms of micro-F1 and macro-F1. The results of our proposed model is evaluated on the test subset with the best model on the validation subset. The experimental results is shown in Table 3.

According to the experimental results, the following conclusions can be drawn. First, our proposed model outperforms all existing models in both RCV1-v2 and WoS datasets. Second, for RCV1-v2 dataset, HCL-MTC achieves an improvement of 0.02% micro-F1 score and 0.38% macro-F1 score compared with HiAGM-TP$_{GCN}$ model. For WoS dataset, HCL-MTC also achieves a considerable improvement by 0.23% and 0.35% in terms of micro-F1 and macro-F1.

Our proposed model is an improvement of HiAGM-TP$_{GCN}$ where we add the contrastive learning method to the basic framework of HiAGM-TP$_{GCN}$. Specifically, we utilize the similarity between label pairs as the transition parameters in the GCN network instead of using prior probability of label dependencies. We also add a sampling hierarchical contrastive loss to the total loss. The results show that HCL-MTC improves the ability of learning label hierarchy. Moreover, HCL-MTC mainly improves the macro-F1 score in both datasets which indicates that HCL-MTC can get access to deeper label hierarchy and has a strong ability to tackle the data sparsity problem.

4.4 Ablation Test

We conduct an ablation test to analyze the impact of the similarity transition matrix and sampling hierarchical contrastive loss to the proposed model. The results of the ablation study is shown in Table 4.

From the ablation study, we can observe that HCL-MTC without sampling hierarchical contrastive loss outperforms HiAGM-TP$_{GCN}$ on two datasets in terms of both Micro-F1 and Macro-F1.

Model	RCV1-v2		WoS	
	Micro-F1	Macro-F1	Micro-F1	Macro-F1
CNN	79.37	55.45	82.00	76.18
RNN	81.10	51.09	77.94	69.65
RCNN	81.57	59.25	83.55	76.99
HR-DGCNN	76.18	43.34	-	-
HE-AGCRCNN	77.80	51.30	-	-
HiLAP	83.30	60.10	-	-
HMCN	80.80	54.60	-	-
HFT(M)	80.29	51.40	-	-
HTrans	80.51	58.49	-	-
SGM	77.30	47.49	-	-
HiAGM-LA$_{GCN}$	82.21	61.65	64.61	79.37
HiAGM-TP$_{GCN}$	83.96	63.35	85.82	80.28
HCL-MTC	**83.98**	**63.73**	**86.05**	**80.63**

Table 3: Experimental results of MLP baselines, state-of-art models and our proposed model.

Model	**RCV1-v2**	
	Micro-F1	**Macro-F1**
HCL-MTC	83.98	**63.73**
w/o similarity	**84.09**	63.17
w/o contrastive loss	84.03	63.39
Model	**WoS**	
	Micro-F1	**Macro-F1**
HCL-MTC	**86.05**	**80.63**
w/o similarity	86.00	80.26
w/o contrastive loss	85.93	80.42

Table 4: Ablation study of the HCL-MTC with varying different components on RCV1-v2 and WoS datasets. *w/o similarity* denotes the HCL-MTC without similarity transition matrix and *w/o contrastive loss* denotes the HCL-MTC without sampling hierarchical contrastive loss.

It shows that the similarity transition matrix is undoubtedly beneficial to the HCL-MTC. The single contrastive loss does not help much for the HCL-MTC according to the results shown in *w/o similarity*. However, combining these two contrastive learning methods, the HCL-MTC can achieve better performance than only using single contrastive learning method. It shows that the contrastive loss increase the performance of similarity transition matrix. The reason is: 1) We want the child node aggregates more information from its parent or the parent node aggregates more information from its child nodes, so that the model can learn deep label hierarchy along the correct path. We use the similarity transition matrix to perform this process where the closer label pairs have higher similarity and, thus have higher transition probability. 2) The sampling hierarchical contrastive loss helps the model minimize the similarity information from parent node to its child nodes and maximize the distinction information between parent nodes. Therefore, the sampling hierarchical contrastive loss can help the model find a better solution during the training process.

5 Conclusions

We present a Hierarchical Contrastive Learning for Multi-label Text Classification (HCL-MTC). The HCL-MTC implements two contrastive learning methods based on the state-of-art framework HiAGM-TP$_{GCN}$ where we apply the similarity transition matrix to the GCN. Furthermore, a complementary sampling hierarchical contrastive loss is introduced to learn both the correlative and distinctive knowledge between labels and increase the performance of the similarity transition matrix. Extensive experiments are carried out on two public datasets, including RCV1-v2 and WoS datasets. The experimental results show that our proposed model outperforms all the existing model, especially in terms of Macro-F1. It indicates that our model has a strong ability to get access to the deep label hierarchy and is better to tackle with the data sparsity problem. For RCV1-v2 dataset, our best model obtains a Micro-F1 of 83.98% and a Macro-F1 of 63.73%. Our best model also achieves a Micro-F1 score of 86.05% and a Macro-F1 score of 80.63% for WoS dataset.

References

Simon Baker and Anna Korhonen. 2017. Initializing neural networks for hierarchical multi-label text classification. In *BioNLP 2017*, pages 307–315, Vancouver, Canada,. Association for Computational Linguistics.

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

Ilias Chalkidis, Manos Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-scale multi-label text classification on eu legislation.

Boli Chen, Xin Huang, Lin Xiao, and Liping Jing. 2020. Hyperbolic capsule networks for multi-label classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3115–3124, Online. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020. End-to-end emotion-cause pair extraction based on sliding window multi-label learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3574–3583, Online. Association for Computational Linguistics.

Siddharth Gopal and Yiming Yang. 2013. Recursive regularization for large-scale classification with hierarchical and graphical dependencies. KDD '13, page 257–265, New York, NY, USA. Association for Computing Machinery.

Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. 2019. Hierarchical multi-label text classification: An attention-based recurrent network approach. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 1051–1060, New York, NY, USA. Association for Computing Machinery.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks.

Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*.

David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397.

Yuncong Li, Cunxiang Yin, Sheng hua Zhong, and Xu Pan. 2020. Multi-instance multi-label learning networks for aspect-category sentiment analysis.

Junyang Lin, Qi Su, Pengcheng Yang, Shuming Ma, and Xu Sun. 2018. Semantic-unit-based dilated convolution for multi-label text classification.

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

Macedo Maia, Juliano Efson Sales, André Freitas, Siegfried Handschuh, and Markus Endres. 2021. A comparative study of deep neural network models on multi-label text classification in finance. In *2021 IEEE 15th International Conference on Semantic Computing (ICSC)*, pages 183–190.

Yuning Mao, Jingjing Tian, Jiawei Han, and Xiang Ren. 2019. 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)*, pages 445–455, Hong Kong, China. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text.

Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Found. Trends Inf. Retr.*, 2(1–2):1–135.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in pytorch.

Hao Peng, Jianxin Li, Qiran Gong, Senzhang Wang, Lifang He, Bo Li, Lihong Wang, and Philip S. Yu. 2019. Hierarchical taxonomy-aware and attentional graph capsule rcnns for large-scale multi-label text classification.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. 2018. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1063–1072, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Kazuya Shimura, Jiyi Li, and Fumiyo Fukumoto. 2018. 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*, pages 811–816, Brussels, Belgium. Association for Computational Linguistics.

Pingjie Tang, Meng Jiang, Bryan (Ning) Xia, Jed W. Pitera, Jeffrey Welser, and Nitesh V. Chawla. 2020. Multi-label patent categorization with non-local attention-based graph convolutional network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9024–9031.

Jonatas Wehrmann, Ricardo Cerri, and Rodrigo Barros. 2018. Hierarchical multi-label classification networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5075–5084. PMLR.

P.J. Werbos. 1990. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. Sgm: Sequence generation model for multi-label classification.

Wenshuo Yang, Jiyi Li, Fumiyo Fukumoto, and Yanming Ye. 2020a. HSCNN: A hybrid-Siamese convolutional neural network for extremely imbalanced multi-label text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6716–6722, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2020b. Xlnet: Generalized autoregressive pretraining for language understanding.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. 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*, pages 1480–1489, San Diego, California. Association for Computational Linguistics.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2018. Graph convolutional networks for text classification.

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