Label-Specific Feature Augmentation for Long-Tailed Multi-Label Text Classification

Pengyu Xu, Lin Xiao, Bing Liu, Sijin Lu, Liping Jing*, Jian Yu

Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, Beijing, China
{pengyu, 17112079, 22120391, 22120406, lpjing, jianyu}@bjtu.edu.cn

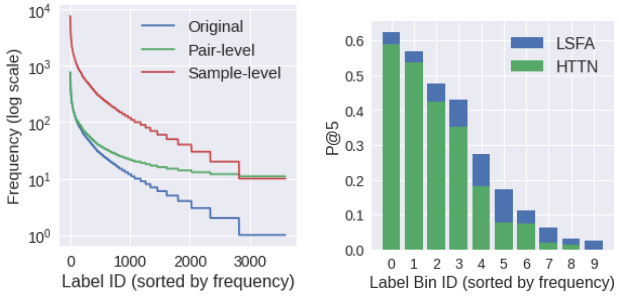
Abstract

Multi-label text classification (MLTC) involves tagging a document with its most relevant subset of labels from a label set. In real applications, labels usually follow a long-tailed distribution, where most labels (called as tail-label) only contain a small number of documents and limit the performance of MLTC. To facilitate this low-resource problem, researchers introduced a simple but effective strategy, data augmentation (DA). However, most existing DA approaches struggle in multi-label settings. The main reason is that the augmented documents for one label may inevitably influence the other co-occurring labels and further exaggerate the long-tailed problem. To mitigate this issue, we propose a new *pair-level* augmentation framework for MLTC, called **L**abel-**S**pecific **F**eature **A**ugmentation (LSFA), which *merely augments positive feature-label pairs for the tail-labels*. LSFA contains two main parts. The first is for label-specific document representation learning in the high-level latent space, the second is for augmenting tail-label features in latent space by transferring the documents second-order statistics (intra-class semantic variations) from head-labels to tail-labels. At last, we design a new loss function for adjusting classifiers based on augmented datasets. The whole learning procedure can be effectively trained. Comprehensive experiments on benchmark datasets have shown that the proposed LSFA outperforms the state-of-the-art counterparts.

Introduction

Multi-label text classification (MLTC) is a task of finding the most relevant labels for each text from a label set. It has a wide range of applications, such as topic recognition (Rubin et al. 2012), tag recommendation (Zhang et al. 2019), sentiment analysis (Yilmaz et al. 2021), profile identification (Gérardin et al. 2022) and so on.

Even though various techniques have been proposed for MLTC, it is still a challenging task due to the "long-tailed" label distribution (Wu et al. 2020; Guo and Wang 2021; Xiao et al. 2021). Figure 1 (a) illustrates the long-tailed label distribution in the EUR-Lex dataset (Loza Mencía and Fürnkranz 2008). Only 3% of the labels have more than 100 training instances (i.e., head-labels), while the remaining 97% are long-tail labels with much fewer training instances.



(a) Label distribution (b) Classification performance

Figure 1: (a) EUR-Lex shows a long-tailed distribution of labels (denoted as Original). Only 3% of the labels have more than 100 training instances. Sample-level denotes the distribution after augmenting 10 times for tail-labels in sample-level, and Pair-level denotes the distribution after augmenting 10 times for tail-labels in pair-level. (b) The classification performance of HTTN and LSFA (ours) on the EUR-Lex dataset. The bars show the macro-averaged P@5 scores of each algorithm over the label bins (with 400 labels per bin).

In this situation, training classification models for the tail-labels is much more difficult than that for head-labels, which suffers severely from the lack of sufficient training instances. Figure 1 (b) shows the performance of HTTN (Xiao et al. 2021), one of the state-of-the-art (SOTA) MLTC models on the EUR-Lex dataset. The vertical axis is the text classification performance measured in macro-averaged P@5 (higher the better) for binned labels (400 labels per bin). The green bars have the scores below to 0.2 for more than half of the total labels. In other words, even SOTA methods in MLTC perform poorly on the tail-labels.

One immediate approach to address the problem is data augmentation (DA) which can compensate the scarce data for tail-labels (Wang et al. 2019; Chu et al. 2020; Zhang et al. 2020, 2022a). Meanwhile, DA has shown its effectiveness in many low-resource data scenarios, such as low-resource NLP (Wei and Zou 2019; Wang et al. 2022; Wu et al. 2022), and zero/few-shot learning (Schwartz et al. 2018; Keshari, Singh, and Vatsa 2020; Xu and Le 2022).

*Corresponding author

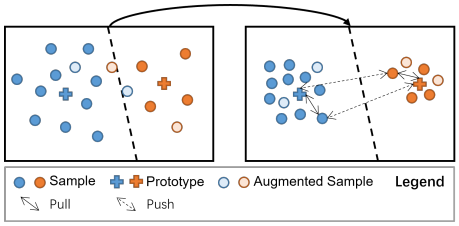


Figure 2: Illustration of cross-entropy (left) and prototypical supervised contrastive (right) loss based feature learning for MLTC. Cross-entropy loss learns skewed and loose features, which can result in biased classifier and augmentation. After adding PSC loss, each feature is pushed away from other prototypes and drawn toward the prototype of its own label.

However, because of the label co-occurrence, existing approaches of DA struggle in the multi-label scenario (Wu et al. 2020; Zhang et al. 2022a). A document usually contains several labels, making the selection for tail-labels no longer independent. For example, a document that contains tail-labels, e.g., "Neptune" and "Pluto", is likely to be also associated with the head-labels, e.g., "astronomy " and "physics". As shown in Figure 1 (a), the long-tailed problem is not necessarily eliminated and may even be exaggerated by directly augmenting in *sample-level*.

Instead of traditional *sample-level* augmentation, we propose a new *pair-level* augmentation framework for MLTC, called **L**abel-**S**pecific **F**eature **A**ugmentation (LSFA), which *merely augments positive feature-label pairs for the tail-labels* by introducing 1) decoupled representation learning and 2) head-to-tail feature augmentation.

During decoupled representation learning, we present a label-specific feature learning module for decoupling representation of each label. Furthermore, we explore a prototypical supervised contrastive (PSC) (Wang et al. 2021) learning strategy to learn better decoupled representations in order to boost classification and augmentation. Intuitively, as shown in Figure 2, the features learned by cross-entropy loss are skewed and loose, which can result in biased classifier and misguided augmentation. After adding PSC loss, each feature is pushed away from other prototypes and drawn toward the prototype of its own label.

In head-to-tail feature augmentation, augmented features of tail-labels are generated by a prototype-based variational autoencoder (PVAE) model (Kingma and Welling 2013; Xu and Le 2022). The PVAE learns to associate a distribution of features to a conditioned prototype, i.e., intra-class semantic variations (Wang et al. 2019). It is assumed that such association generalizes across the head and tail labels (Chu et al. 2020; Xiao et al. 2021). Therefore, the PVAE trained with ample data from the head-labels can generate tail-label features that align with the real unseen features.

We summarize our main contributions as follows:

- We propose a novel *pair-level* label-specific feature augmentation (LSFA) framework for MLTC, which generates positive feature-label pairs to alleviate the data sparsity on tail-labels.
- In order to acquire better decoupled representations for classification and augmentation, we introduce prototypical supervised contrastive learning strategy to label-specific feature learning process.
- A prototype-based VAE-style feature generation model is designed to capture the intra-class semantic variations from head-labels, which will be applied to augment features for tail-labels.
- Our experiments show that LSFA significantly and consistently outperforms state-of-the-art baselines on three benchmark datasets, especially on tail-labels.

Our code and hyper-parameter settings are publicly available at https://github.com/stxupengyu/LSFA.

Related Work

In this section, we review previous literature from two aspects, multi-label text classification and data augmentation.

Multi-label Text Classification

The most straightforward strategy for dealing with multi-label text classification (MLTC) is to utilize the identical representation of a document to induce classification models (Liu et al. 2017; Yang et al. 2018; Zhang et al. 2018). For instance, Liu et al. (2017) utilize a CNN-based model with dynamic max pooling scheme that captures high-level feature for MLTC. Such a strategy, however, disregards the fact that different labels may concentrate on various tokens. Consequently, label-specific feature learning (You et al. 2019; Xiao et al. 2019; Ma et al. 2021; Zhang et al. 2021), which focuses on each label's unique traits, is a promising method for facilitating the discrimination of each label (Zhang, Fang, and Wang 2021; Hang and Zhang 2022). You et al. (2019) propose a label-specific attention network to focus on different tokens when predicting each label. Furthermore, Ma et al. (2021) adopt a graph convolution network which incorporates category information and models adaptive interactions in a label-specific way. Recently, Zhang et al. (2021) exploit correlation-guided representation to capture high-order document-label correlations. Wang, Dai et al. (2022) use a k nearest neighbor mechanism along with a multi-label contrastive learning strategy for MLTC. Bai, Kong, and Gomes (2021) impose the VAE to learn and align two embedding spaces for labels and features respectively. Bai, Kong, and Gomes (2022) also use contrastive loss to strengthen the label embedding learning by introducing feature embedding as the anchor.

Even though previous techniques have achieved encouraging performance in MLTC, it is still a challenging task due to the long-tailed label distribution (Chang et al. 2020; Xiao et al. 2021; Zhang et al. 2022b). In this case, training classification models for the tail-labels is much more difficult than that for head-labels, which suffer severely from the lack of sufficient training instances. Some existing works (Yang et al. 2020a; Huang et al. 2021) tackle it by proposing imbalanced loss objectives instead of the vanilla cross-entropy loss. Xiao et al. (2021) propose a head-to-tail network which transfers the meta-knowledge from the head-labels to tail-labels. Our proposed method LSFA also adopts the knowl-

edge transfer strategy, but focuses on explicit feature augmentation for tail-labels to facilitate the long-tailed problem for MLTC.

Data Augmentation

Data augmentation (DA) has shown its effectiveness in many low-resource data scenario, such as text classification (Wei and Zou 2019; Kumar, Choudhary, and Cho 2020), few-shot learning (Zhou et al. 2022; Xu and Le 2022), natural language understanding (Wang et al. 2022) and so on. They are mainly divided into two categories: DA in input space and DA in feature space. The most commonly used DA method in input space is the word substitution, such as synonym replacement and random swap (Wei and Zou 2019). Back translation (Yang et al. 2020b) is also widely used. In addition, large pretrained models have been used for DA (Kumar, Choudhary, and Cho 2020; Zhang et al. 2020). They take advantage of the pretrained models, such as GPT-2 (Zhang et al. 2020), BERT (Kenton and Toutanova 2019), and BART (Lewis et al. 2020) to generate label-invariant perturbations of the input texts to augment the existing training data. Other DA techniques, such as autoencoder (AE) (Schwartz et al. 2018; Liu et al. 2020), which is offered for better data diversity based on the perturbation in the feature space. Furthermore, Xu and Le (2022) propose a feature generation method using a conditional variational autoencoder (VAE). Unfortunately, due to the label co-occurrence, it is challenging for these prior methods to handle MLTC (Wu et al. 2020; Zhang et al. 2022a). Instead of previous sample-level augmentation, we create a new pair-level augmentation strategy, which merely augments positive feature-label pairs for the tail-labels.

Method

As depicted in Figure 3, our method LSFA is composed of two major modules: decoupled representation learning and head-to-tail feature augmentation. Specifically, decoupled representation learning describes how to extract decoupled semantic components from the mixture of label information in each document; and the head-to-tail feature augmentation illustrates how to acquire the intra-class semantic variations from head-labels to the tail-labels.

Problem Definition

Let calligraphic letter (e.g., $\mathcal{A}$) indicates set, capital and lower-case letters (e.g., A, a) for scalars, lower-case bold letter (e.g., $\boldsymbol{a}$) for vector and capital bold letter (e.g., $\boldsymbol{A}$) for matrix. The input of the training stage includes N instances $\mathcal{P} = \{(doc^i, \boldsymbol{y}^i)\}_{i=1}^N$, each of which consists of a document doc and several labels $\boldsymbol{y} = (y_1, ..., y_j, ..., y_L)$ related to the document. Here $y_j \in \{0, 1\}$, where $y_j = 1$ indicates that the j-th label is associated with document doc, and L is the total number of candidate labels. Each document contains D tokens. In the testing stage, we aim to recommend the most relevant labels for a new document.

Decoupled Representation Learning

Label-Specific Encoder The label-specific encoder is proposed to focus on each label's unique traits for facilitating the discrimination of each label (You et al. 2019; Ma et al. 2021; Hang and Zhang 2022). Our model (Figure 3) is composed of a shared earlier layer $\boldsymbol{\phi}$ and L label-specific later layers $\{\boldsymbol{\psi}_1, \boldsymbol{\psi}_2, ..., \boldsymbol{\psi}_L\}$.

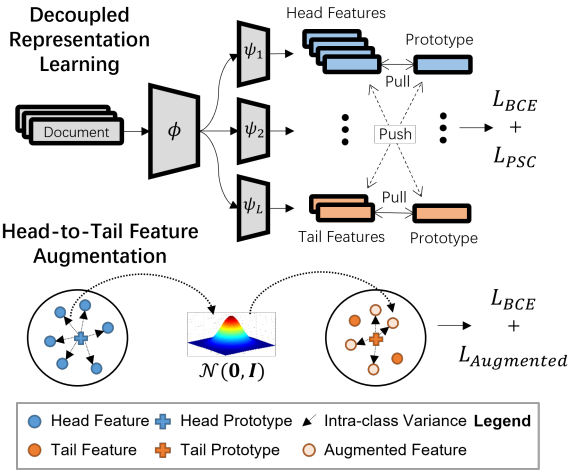


Figure 3: The architecture of the proposed LSFA. Head Feature denotes feature of head-label, and Head Prototype denotes prototype of head-label.

In order to capture the forward and backward sides contextual information, the document is extracted by a bidirectional LSTM (BiLSTM) $\boldsymbol{\phi}_{BiLSTM}$ to obtain the hidden representation of each token:

$$\boldsymbol{\phi}_{BiLSTM}(doc) = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_D\}. \quad (1)$$

The shared encoder can be replaced with any modern language models such as BERT (Kenton and Toutanova 2019), XLNet (Yang et al. 2019), BART (Lewis et al. 2020) and etc.

Similar to multi-label attention (You et al. 2019), our model is designed to let each label attentively select the key tokens from the document. Specifically, we treat labels as queries to retrieve the salient tokens in the document. Finally, for label l, the label-specific feature is:

$$\boldsymbol{v}_l = \boldsymbol{\psi}_l(\boldsymbol{\phi}_{BiLSTM}(doc)). \quad (2)$$

The label-specific layer $\boldsymbol{\psi}_l$ is:

$$\boldsymbol{\psi}_l(\boldsymbol{\phi}_{BiLSTM}(doc)) = \boldsymbol{\alpha}_l \cdot \boldsymbol{\phi}_{BiLSTM}(doc), \quad (3)$$

$$\boldsymbol{\alpha}_l = \text{softmax}(\boldsymbol{e}_l^T(\boldsymbol{\phi}_{BiLSTM}(doc))), \quad (4)$$

where $\boldsymbol{v}_l$ is the trainable parameter, $\boldsymbol{\alpha}_l$ is the normalized coefficient of $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_D\}$ and $\boldsymbol{e}_l$ is the label embedding.

After the label-specific representation $\boldsymbol{v}_l$ is obtained, a simple one-versus-all (OVA) approach realizes the scoring function as:

$$\hat{y}_l = \text{sigmoid}(\boldsymbol{w}_l^T \boldsymbol{v}_l), \quad (5)$$

where $\boldsymbol{w}_l$ is the trainable classifier parameter for the label l. Then, we use corresponding binary cross entropy as the loss function:

$$L_{BCE} = \sum_{l=1}^{L} y_l \log(\hat{y}_l) + (1 - y_l) \log(1 - \hat{y}_l). \quad (6)$$

Prototypical Supervised Contrastive Learning In this section, we explore a prototypical supervised contrastive (PSC) (Wang et al. 2021) learning strategy to learn better decoupled representations in order to boost classification and augmentation. Contrastive learning here is performed at the label-specific feature level. Since the features are already associated with only one label by label-specific encoder, we avoid two inputs share other labels in multi-label setting. First, after the label-specific learning, we only consider the positive feature-label pairs $\{\boldsymbol{v}^i, y^i\}_{i<N_p}$, where y^i denotes the label index, and N_p is the total number of the positive pairs in a mini-batch $\mathcal{Q}$. Instead of unsupervised contrastive learning (He et al. 2020), there are multiple positives per anchor under the supervised contrastive (SC) learning (Khosla et al. 2020). We define all the positive features of an anchor $\boldsymbol{v}^i$ as $\{\boldsymbol{v}^{i^+}\} = \{\boldsymbol{v}^j | y^j = y^i, i \neq j\}$, then the corresponding SC loss is:

$$L_{SC} = \sum_{i=1}^{N_p} L_{SC}^i(\boldsymbol{v}^i), \tag{7}$$

$$L_{SC}^i(\boldsymbol{v}^i) = \frac{-1}{|\{\boldsymbol{v}^{i^+}\}|} \sum_{\boldsymbol{v}^j \in \{\boldsymbol{v}^{i^+}\}} \log \frac{\exp(\boldsymbol{v}^i \cdot \boldsymbol{v}^j / \tau)}{\sum_{\boldsymbol{v}^k, k \neq i} \exp(\boldsymbol{v}^i \cdot \boldsymbol{v}^k / \tau)}, \tag{8}$$

where, $\tau \in \mathcal{R}^+$ is the scalar temperature parameter.

However, this operation is intractable in MLTC setting. Due to the large label size of MLTC, it demands an extremely large mini-batch $\mathcal{Q}$. Therefore, we utilize a tractable version of SC loss, i.e., PSC loss:

$$L_{PSC} = \alpha \cdot \sum_{i=1}^{N_p} \log \frac{\exp(\boldsymbol{v}^i \cdot \boldsymbol{p}_{y^i} / \tau)}{\sum_{j=1, j \neq y^i}^{L} \exp(\boldsymbol{v}^i \cdot \boldsymbol{p}_j / \tau)}, \tag{9}$$

where $\boldsymbol{p}_j$ is the prototype of label j (calculated by Eq. 10), and α is the coefficient of PSC loss. As shown in Figure 3, after adding PSC loss, each feature is pushed away from other prototypes and drawn toward the prototype of its own label, which could boost classification and augmentation afterward.

Head-to-Tail Feature Augmentation

Prototype-based Variational Autoencoder After the decoupled representation learning, we calculate the prototype of each label as the mean of every single dimension in the vector:

$$\boldsymbol{p}_l = \frac{1}{n^l} \sum_{j=1}^{n^l} \boldsymbol{v}_l^j, \tag{10}$$

where $\boldsymbol{v}_l^j$ is a feature vector from the label l and n^l is the total number of features of label l.

Then, we divide the labels to head-labels and tail-labels according to the hyper-parameter, head-to-tail threshold $N_t \in \mathcal{R}^+$. label l is a tail-label if $n^l < N_t$. After that, we obtain a head-label set $\mathcal{H}$ and a tail-label set $\mathcal{T}$.

In order to augment the features of tail-labels, we design a prototype-based variational autoencoder (PVAE) model to learn the intra-class semantic variations (Wang et al. 2019) of head-labels (see Figure 3). It is assumed that such association generalizes across the head and tail labels (Wang et al. 2019; Xiao et al. 2021; Xu and Le 2022).

The PVAE is composed of an encoder $E(\boldsymbol{v}_l^j, \boldsymbol{p}_l)$, which maps a vector $\boldsymbol{v}_l^j$ and its prototype $\boldsymbol{p}_l$ to a latent variation $\boldsymbol{z}$, and a decoder $G(\boldsymbol{z}, \boldsymbol{p}_l)$ which reconstructs $\boldsymbol{v}_l^j$ from $\boldsymbol{z}$ and its prototype $\boldsymbol{p}_l$. The variational parameters $\boldsymbol{\mu}_j^h$ and $\boldsymbol{\sigma}_j^h$ of the encoder are implemented by multilayer perceptrons (MLPs) $f_*(\cdot)$.

$$\boldsymbol{\mu}_l^j = f_\mu([\boldsymbol{v}_l^j; \boldsymbol{p}_l]) \tag{11}$$

$$\boldsymbol{\sigma}_l^j = f_\sigma([\boldsymbol{v}_l^j; \boldsymbol{p}_l]) \tag{12}$$

Here, $[;]$ denotes the concatenation operation. Then the decoding process is:

$$\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \tag{13}$$

$$\boldsymbol{z} = \boldsymbol{\mu}_l^j + \boldsymbol{\epsilon} \odot \boldsymbol{\sigma}_l^j \tag{14}$$

$$\hat{\boldsymbol{v}}_l^j = f_\phi([\boldsymbol{z}; \boldsymbol{p}_l]) \tag{15}$$

where $\odot$ denotes the Hadamard product. Eq. 13 and Eq. 14 are the implementation of reparameterization trick to smooth the gradients (Kingma and Welling 2013).

In the training stage, the objective function is defined based on the negative variational lower bound, which consists of two parts. The first part is the KL divergence that indicates discrepancy between prior distribution $p(\boldsymbol{z})$ and posterior distribution $q(\boldsymbol{z}|\boldsymbol{v}, \boldsymbol{p})$ about the latent variable $\boldsymbol{z}$, and the second part reflects the reconstruction loss:

$$L_l^j = D_{KL}(q(\boldsymbol{z}|\boldsymbol{v}_l^j, \boldsymbol{p}_l) || p(\boldsymbol{z})) - \log(p(\boldsymbol{v}_l^j | \boldsymbol{z}, \boldsymbol{p}_l)), \tag{16}$$

where $q(\boldsymbol{z}|\boldsymbol{v}_l^j, \boldsymbol{p}_l)$ and $p(\boldsymbol{v}_l^j|\boldsymbol{z}, \boldsymbol{p}_l)$ represent the encoder and decoder respectively.

The overall loss of PVAE is over all features from head-labels $\mathcal{H}$:

$$L_{PVAE} = \sum_{l \in \mathcal{H}} \sum_{i=1}^{n^l} L_l^j. \tag{17}$$

Augmenting tail-label features After the PVAE is trained on the head-labels, we generate a set of features for each tail-label l by inputting the respective prototype and a noise vector $\boldsymbol{z}$ from the Gaussian distribution:

$$\mathcal{D}^l = \{\hat{\boldsymbol{v}} | \hat{\boldsymbol{v}} = G(\boldsymbol{z}, \boldsymbol{p}_l), \boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})\}. \tag{18}$$

We augment N_a times for each tail-label.

Loss Function for Augmented Features It is worth noting that the augmented samples of LSFA are label-specific. In other words, the samples can be regarded as the positive feature-label pairs for tail-labels, which are used to adjust the classifiers. The loss function is:

$$L_{Augmented} = \gamma \cdot \sum_{l \in \mathcal{T}} \sum_{j \in \mathcal{D}^l} \log(\text{sigmoid}(\boldsymbol{w}_l^T \hat{\boldsymbol{v}}_j)), \tag{19}$$

where γ is the coefficient of augmented loss. Consequently, in the adjustment phrase, we obtain a more balanced loss for the classifiers by adding $L_{Augmented}$ to L_{BCE}.

Datasets	N_{trn}	N_{tst}	D_{vocab}	L	L_{avg}	N_{avg}	W_{trn}	W_{tst}
AAPD	54,840	1,000	69,399	54	2.41	2444.04	163.42	171.65
RCV1	23,149	781,265	47,236	103	3.18	729.67	259.47	269.23
EUR-Lex	11,585	3,865	171,120	3,956	5.32	15.59	1225.20	1248.07

Table 1: Data statistics. N_{trn}, N_{tst} refer to the number of documents in the training and test sets, respectively. D_{vocab} is the vocabulary size of documents. L is the number of labels. L_{avg} is the average number of labels per documents. N_{avg} is the average number of documents per label. W_{trn}, W_{tst} refer to the average number of words per document in the training and test sets, respectively. The three benchmark datasets are the same as LSAN (Xiao et al. 2019) and LDGN (Ma et al. 2021) for fair comparison.

Experiments

Experimental Setup

Datasets There are several MLTC datasets. We evaluate the proposed model on three benchmark datasets of them, which are AAPD (Yang et al. 2018), RCV1 (Lewis et al. 2004) and EUR-Lex (Loza Mencía and Fürnkranz 2008). Table 1 contains the statistics of these three benchmark datasets.

Evaluation Metric Following the settings of previous works (You et al. 2019; Xiao et al. 2019; Ma et al. 2021; Xiao et al. 2021), We chose precision at k (P@k) and normalized discounted cumulative gain at k (N@k) as our evaluation metrics for performance comparison. We also examined the performance on tail-labels by propensity scored precision at k (PSP@k) and macro-averaged P@k.

Implementation Details For all three datasets, we used the most frequent words that appeared in the training set as a limited-size vocabulary (below 500,000). We truncated each text after 500 words for efficiency. These are the conventional setups for MLTC methods (Xiao et al. 2019; You et al. 2019). Our model was trained by Adam (Kingma and Ba 2014) with the learning rate of 1e-3. We also used stochastic weight averaging (You et al. 2019) with a constant learning rate to enhance the performance. We empirically set the $\alpha = 0.1, \gamma = 1$ for balancing the loss . As for the key hyper-parameters of our proposed method: head-to-tail threshold N_t and times of augmentation N_a, we set $N_t = 1000, N_a = 500$ for AAPD. For RCV1 and EUR-Lex, we set $N_t = 500, N_a = 200$ and $N_t = 50, N_a = 10$ respectively.

Baselines We compare our proposed LSFA method to the most representative and state-of-the-art (SOTA) MLTC methods:

- XML-CNN (Liu et al. 2017): a CNN-based model using dynamic max pooling scheme to capture high-level feature.
- SGM (Yang et al. 2018): a sequence generation model which models the correlations between labels.
- DXML (Zhang et al. 2018): a deep embedding method which models the feature and label space simultaneously.
- AttentionXML (You et al. 2019): a deep learning model which uses a multi-label attention to extract information for each label.

Models	P@1	P@3	P@5	N@3	N@5
XML-CNN	74.38	53.84	37.79	71.12	75.93
SGM	75.67	56.75	35.65	72.36	75.30
DXML	80.54	56.30	39.16	77.23	80.99
AttentionXML	83.02	58.72	40.56	78.01	82.31
EXAM	83.26	59.77	40.66	79.10	82.79
LSAN	85.28	61.12	41.84	80.84	84.78
HTTN	83.84	59.92	40.79	79.27	82.67
LDGN	86.24	61.95	42.29	83.32	86.85
LSFA	**86.95**	**62.88**	**43.43**	**83.96**	**87.53**

Table 2: Comparisons with SOTA methods on the AAPD dataset. The experimental results of all baseline models are directly cited from [(Ma et al. 2021), Table 2] and [(Xiao et al. 2021), Table 2]. The best results are highlighted in bold.

- EXAM (Du et al. 2019): a framework that employs the interaction mechanism to compute the word-level interaction signals.
- LSAN (Xiao et al. 2019): a label-specific attention model based on self-attention and label-attention mechanisms.
- HTTN (Xiao et al. 2021): a head-to-tail network which transfers the meta-knowledge from the head-labels to tail-labels.
- LDGN (Ma et al. 2021): a graph convolution network which incorporates category information and models adaptive interactions of labels.

Most results of all these baseline methods are obtained from [(Ma et al. 2021), Table 2 and Table 3] and [(Xiao et al. 2021), Table 2 and Table 3].

Performance Comparison

From Table 2 to Table 4, we show the main comparison results on three datasets. After analyzing the results, we have the following observations.

Label-specific feature learning is crucial for MLTC. XML-CNN performs relatively poorly as it utilizes the identical representation of a document to induce classification model. AttentionXML, LDGN and LSAN outperform XML-CNN by a large margin on each dataset. This is not surprising since these methods introduce the label-specific encoder, which focuses on each label's unique traits in the representation learning phrase. Meanwhile, LDGN achieves

Models	P@1	P@3	P@5	N@3	N@5
XML-CNN	95.75	78.63	54.94	89.89	90.77
SGM	95.37	81.36	53.06	91.76	90.69
DXML	94.04	78.65	54.38	89.83	90.21
AttentionXML	96.41	80.91	56.38	91.88	92.70
EXAM	93.67	75.80	52.73	86.85	87.71
LSAN	96.81	81.89	56.92	92.83	93.43
HTTN	95.86	78.92	55.27	89.61	90.86
LDGN	97.12	82.26	57.29	93.80	95.03
LSFA	**97.21**	**82.52**	**57.52**	**94.20**	**95.42**

Table 3: Comparisons with SOTA methods on the RCV1 dataset. The experimental results of all baseline models are directly cited from [(Ma et al. 2021), Table 3] and [(Xiao et al. 2021), Table 3]. The best results are highlighted in bold.

Models	P@1	P@3	P@5	N@3	N@5
XML-CNN	70.40	54.98	44.86	58.62	53.10
SGM	70.45	60.37	43.88	60.72	55.24
DXML	75.63	60.13	48.65	63.96	53.60
AttentionXML*	79.66	64.88	52.99	68.66	62.33
EXAM	74.40	61.93	50.98	65.12	59.43
LSAN	79.17	64.99	53.67	68.32	62.47
HTTN*	80.45	65.57	55.68	69.01	63.76
LDGN	81.03	67.79	56.36	71.81	66.09
LSFA	**83.75**	**70.74**	**58.95**	**74.13**	**68.25**

Table 4: Comparisons with SOTA methods on the EUR-Lex dataset. Results with a trailing reference are reproduced by ourselves. Other results are taken from [(Ma et al. 2021), Table 2]. The best results are highlighted in bold.

the second-best results, since its label-specific encoder is enhanced by the deeper correlations between categories.

Head-to-tail transfer learning effectively alleviates the long-tailed problem. An interesting point is that, transfer learning based HTTN is worse than LSAN on AAPD and RCV1 datasets, while HTTN is superior to LSAN on the EUR-Lex. The reason is that, there are more tail-labels on the EUR-Lex. As a consequence, HTTN could introduce the meta-knowledge from the data-rich head-labels to data-poor tail-labels, while LSAN ignores the data sparsity on tail-labels.

The results demonstrate the superiority of the proposed LSFA on all metrics for MLTC. Especially, on the EUR-Lex dataset, the relative improvements of LSFA are 3.36%, 4.35% and 4.60% compared with its best competitors on the P@1, P@3 and P@5, respectively. For some label-specific methods (LSAN and LDGN), although they designed different strategies to enhance the representation learning, they suffer severely from the high data scarcity on tail-labels. LSFA avoids this problem by augmenting features for long-tailed labels, and decoupling the representation learning to further improve the robustness of tail-labels augmentation and classification.

Models	P@5	PSP@1	PSP@3	PSP@5
LSAN	53.67	36.41	41.27	43.42
HTTN	55.68	38.96	43.28	45.74
LSFA	**58.95**	**42.50**	**48.03**	**50.69**
Improvement	*5.87%*	*9.09%*	*10.98%*	*10.82%*

Table 5: Performance on tail-labels on the EUR-Lex dataset. Improvement denotes the relative improvement of LSFA over the best baseline. Best results highlighted in Bold.

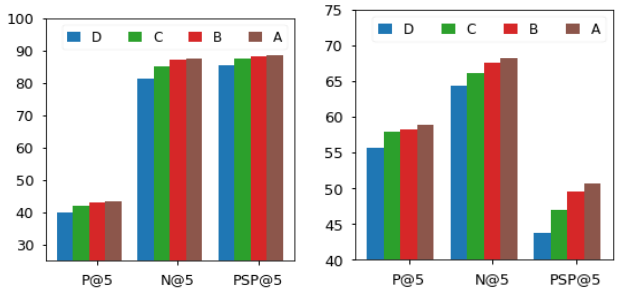
Ablation Test

We analyze the impacts of key components in LSFA via ablation test. The complete LSFA (denoted as A) is compared with the following variants: LSFA removes the PSC loss (denoted as B), LSFA removes the PSC loss and feature augmentation (denoted as C), LSFA removes the PSC loss, feature augmentation and label-specific encoder (denoted as D). Figure 4 shows the results evaluated on AAPD and EUR-Lex datasets in terms of P@5, N@5 and PSP@5. There are several interesting observations:

It is always preferable to use the PSC strategy, as shown by the superior performance of A. The reason is that it decoupled the representation of each label in the feature space, which is crucial for classification and augmentation. The result of B is always better than C, because the augmented features effectively alleviate the data sparsity on tail-labels, improving the performance. C is better than D, it demonstrates the effectiveness of incorporating the label-specific learning for MLTC. Such a label-specific learning module allows our model to capture various intensive parts of the document for each label, thus enabling better classification.

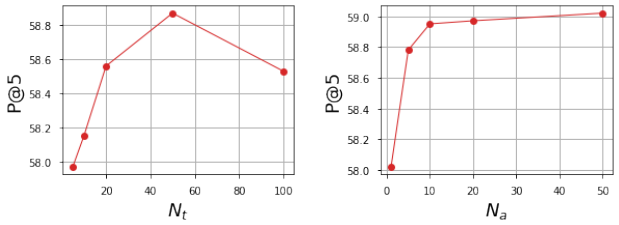
Performance Analysis on Tail-Labels

To further verify the effectiveness of the proposed LSFA in alleviating the long-tailed problem, we compare the performance of LSFA with SOTA baselines by PSP@k (Jain, Prabhu, and Varma 2016; You et al. 2019; Ma et al. 2021).



(a) AAPD (b) EUR-Lex

Figure 4: Ablation test of LSFA on two datasets. A denotes the complete LSFA, B denotes LSFA without the PSC loss, C denotes LSFA without the PSC loss and feature augmentation, D denotes LSFA without the PSC loss, feature augmentation and label-specific encoder.



(a) Head-to-tail threshold (b) Times of augmentation

Figure 5: Effect of parameters on LSFA for EUR-Lex dataset.

Table 5 shows LSFA has remarkable improvement compared to the baselines in tail-labels classification. It is reasonable since LSFA addresses the data sparsity on tail-labels by augmenting the tail-label samples in the feature space with the intra-class semantic variations, which are learned from the head-labels with ample samples.

Furthermore, Figure 1 (b) shows their macro-averaged P@5 on label bins on the EUR-Lex. We can see that the macro P@5 of LSFA on tail-label bins [2-9] are much higher than those from the HTTN. It is noteworthy that, the performance of bins [3-5] achieves the most outstanding improvement when comparing with bins [6-9]. The reason is that each label of bins [6-9] has only 1-5 documents, which is insufficient to learn a better prototype as of bins [3-5].

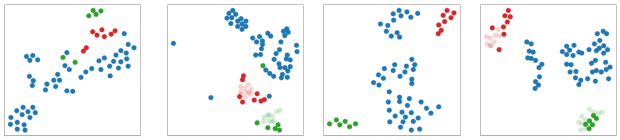
In summary, LSFA do help to build effective tail-label predictor by augmenting for tail-labels.

Parameter Sensitivity

There are two major hyper-parameters we proposed in LSFA, including head-to-tail threshold N_t and times of augmentation N_a that controlling the intensity of the augmentation. For investigating the impacts of the N_t and N_a, we vary them and show their influences on P@5 in Figure 5. The performance improves at first and then decreases as the N_t increases. As N_t increases, less head-labels are available for head-to-tail transfer learning, which is not conducive to LSFA. As N_t decreases, the number of tail-labels becomes smaller, reducing the effectiveness of the feature augmentation. Finally, we find a trade-off between them. Also, increasing N_a from 1 to 10 can greatly help LSFA to gain strong improvement. That's to say, augmenting more features for tail-labels can effectively strengthen the generalizability of the classifiers.

Case Study

To further illustrate the effectiveness of our decoupled learning and feature augmentation, we show the t-SNE (Van der Maaten and Hinton 2008) representation of head and tail labels in figure 6. From left to right, we visualize the distribution of the features learned by LSFA without PSC strategy (a), the features augmented by LSFA without PSC strategy (b), the features learned by LSFA (c), and the features augmented by LSFA (d). Figure 6 (a) and (c) visualize the effect of our prototypical contrastive learning strategy, while



(a) (b) (c) (d)

Figure 6: Feature visualization. We show the t-SNE visualization of the original features (marked as dark points) and augmented features (marked as transparent points) on the EUR-Lex dataset. The features are from two tail-labels and a head-label. Different colors represent different labels. From left to right, we show the features learned by LSFA without PSC strategy (a), the augmented features generated by LSFA without PSC strategy (b), the features learned by LSFA (c), and the augmented features generated by LSFA (d).

each feature is pulled towards the prototype of its class and pushed away from prototypes of other classes. Figure 6 (d) visualizes features generated from LSFA within PSC strategy, that lie closer to the real features, showing the effectiveness of contrastive learning enhanced decoupled learning and feature augmentation.

Conclusions and Future Work

In this paper, we propose a novel *pair-level* label-specific feature augmentation (LSFA) framework for MLTC, which augments positive feature-label pairs for the tail-labels in the feature space. Firstly, we use a prototypical supervised contrastive learning strategy to learn better decoupled representations for the document. After that, a prototype-based VAE-style transfer learner is designed to capture the intra-class semantic variations from head-labels to tail-labels. Finally, we design a new loss function to adjust the classifiers based on the original and augmented features. Extensive experiments demonstrate that LSFA outperforms the state-of-the-art approaches, especially on tail-labels.

In the future, we would like to explore how to leverage LSFA in the scenarios with larger label space. In addition, we are also interested in boosting input-level augmentation for MLTC.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China under Grant 62176020; the National Key Research and Development Program (2020AAA0106800); the Beijing Natural Science Foundation under Grant (Z180006, L211016); CAAI-Huawei MindSpore Open Fund; and Chinese Academy of Sciences(OEIP-O-202004). Discussions with Xin Liu and Mingyang Song are gratefully acknowledged. We are also grateful for the anonymous reviewers and the editor for their helpful comments.

References

Bai, J.; Kong, S.; and Gomes, C. 2021. Disentangled variational autoencoder based multi-label classification with

covariance-aware multivariate probit model. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, 4313–4321.

Bai, J.; Kong, S.; and Gomes, C. P. 2022. Gaussian mixture variational autoencoder with contrastive learning for multi-label classification. In *International Conference on Machine Learning*, 1383–1398. PMLR.

Chang, W.-C.; Yu, H.-F.; Zhong, K.; Yang, Y.; and Dhillon, I. S. 2020. Taming pretrained transformers for extreme multi-label text classification. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 3163–3171.

Chu, P.; Bian, X.; Liu, S.; and Ling, H. 2020. Feature space augmentation for long-tailed data. In *European Conference on Computer Vision*, 694–710. Springer.

Du, C.; Chen, Z.; Feng, F.; Zhu, L.; Gan, T.; and Nie, L. 2019. Explicit interaction model towards text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 6359–6366.

Gérardin, C.; Wajsbürt, P.; Vaillant, P.; Bellamine, A.; Carrat, F.; and Tannier, X. 2022. Multilabel classification of medical concepts for patient clinical profile identification. *Artificial Intelligence in Medicine*, 128: 102311.

Guo, H.; and Wang, S. 2021. Long-tailed multi-label visual recognition by collaborative training on uniform and re-balanced samplings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15089–15098.

Hang, J.-Y.; and Zhang, M.-L. 2022. Dual Perspective of Label-Specific Feature Learning for Multi-Label Classification. In *International Conference on Machine Learning*, 8375–8386. PMLR.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9729–9738.

Huang, Y.; Giledereli, B.; Köksal, A.; Özgür, A.; and Ozkirimli, E. 2021. Balancing Methods for Multi-label Text Classification with Long-Tailed Class Distribution. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 8153–8161.

Jain, H.; Prabhu, Y.; and Varma, M. 2016. Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 935–944.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Keshari, R.; Singh, R.; and Vatsa, M. 2020. Generalized zero-shot learning via over-complete distribution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13300–13308.

Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; and Krishnan, D. 2020. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33: 18661–18673.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kumar, V.; Choudhary, A.; and Cho, E. 2020. Data Augmentation using Pre-trained Transformer Models. In *Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems*, 18–26.

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

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

Liu, D.; Gong, Y.; Fu, J.; Yan, Y.; Chen, J.; Lv, J.; Duan, N.; and Zhou, M. 2020. Tell Me How to Ask Again: Question Data Augmentation with Controllable Rewriting in Continuous Space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 5798–5810.

Liu, J.; Chang, W.-C.; Wu, Y.; and Yang, Y. 2017. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval*, 115–124.

Loza Mencía, E.; and Fürnkranz, J. 2008. Efficient pairwise multilabel classification for large-scale problems in the legal domain. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 50–65. Springer.

Ma, Q.; Yuan, C.; Zhou, W.; and Hu, S. 2021. Label-specific dual graph neural network for multi-label 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)*, 3855–3864.

Rubin, T. N.; Chambers, A.; Smyth, P.; and Steyvers, M. 2012. Statistical topic models for multi-label document classification. *Machine learning*, 88(1): 157–208.

Schwartz, E.; Karlinsky, L.; Shtok, J.; Harary, S.; Marder, M.; Kumar, A.; Feris, R.; Giryes, R.; and Bronstein, A. 2018. Delta-encoder: an effective sample synthesis method for few-shot object recognition. *Advances in neural information processing systems*, 31.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wang, P.; Han, K.; Wei, X.-S.; Zhang, L.; and Wang, L. 2021. Contrastive learning based hybrid networks for long-tailed image classification. In *Proceedings of the IEEE/CVF*

conference on computer vision and pattern recognition, 943–952.

Wang, R.; Dai, X.; et al. 2022. Contrastive Learning-Enhanced Nearest Neighbor Mechanism for Multi-Label Text Classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 672–679.

Wang, Y.; Pan, X.; Song, S.; Zhang, H.; Huang, G.; and Wu, C. 2019. Implicit semantic data augmentation for deep networks. *Advances in Neural Information Processing Systems*, 32.

Wang, Y.; Xu, C.; Sun, Q.; Hu, H.; Tao, C.; Geng, X.; and Jiang, D. 2022. PromDA: Prompt-based Data Augmentation for Low-Resource NLU Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4242–4255.

Wei, J.; and Zou, K. 2019. EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. 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)*, 6382–6388.

Wu, T.; Huang, Q.; Liu, Z.; Wang, Y.; and Lin, D. 2020. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *European Conference on Computer Vision*, 162–178. Springer.

Wu, X.; Gao, C.; Lin, M.; Zang, L.; and Hu, S. 2022. Text Smoothing: Enhance Various Data Augmentation Methods on Text Classification Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 871–875.

Xiao, L.; Huang, X.; Chen, B.; and Jing, L. 2019. Label-specific document representation for multi-label 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)*, 466–475.

Xiao, L.; Zhang, X.; Jing, L.; Huang, C.; and Song, M. 2021. Does Head Label Help for Long-Tailed Multi-Label Text Classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 14103–14111.

Xu, J.; and Le, H. 2022. Generating Representative Samples for Few-Shot Classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9003–9013.

Yang, P.; Sun, X.; Li, W.; Ma, S.; Wu, W.; and Wang, H. 2018. SGM: sequence generation model for multi-label classification. *arXiv preprint arXiv:1806.04822*.

Yang, W.; Li, J.; Fukumoto, F.; and Ye, Y. 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)*, 6716–6722.

Yang, Y.; Malaviya, C.; Fernandez, J.; Swayamdipta, S.; Le Bras, R.; Wang, J.-P.; Bhagavatula, C.; Choi, Y.; and Downey, D. 2020b. Generative Data Augmentation for Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1008–1025.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Yilmaz, S. F.; Kaynak, E. B.; Koç, A.; Dibeklioğlu, H.; and Kozat, S. S. 2021. Multi-label sentiment analysis on 100 languages with dynamic weighting for label imbalance. *IEEE Transactions on Neural Networks and Learning Systems*.

You, R.; Zhang, Z.; Wang, Z.; Dai, S.; Mamitsuka, H.; and Zhu, S. 2019. AttentionXML: Label Tree-based Attention-Aware Deep Model for High-Performance Extreme Multi-Label Text Classification. *Advances in Neural Information Processing Systems*, 32: 5820–5830.

Zhang, D.; Li, T.; Zhang, H.; and Yin, B. 2020. On data augmentation for extreme multi-label classification. *arXiv preprint arXiv:2009.10778*.

Zhang, J.; Liu, J.; Chen, S.; Lin, S.; Wang, B.; and Wang, S. 2022a. ADAM: An Attentional Data Augmentation Method for Extreme Multi-label Text Classification. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 131–142. Springer.

Zhang, M.-L.; Fang, J.-P.; and Wang, Y.-B. 2021. Bilabel-specific features for multi-label classification. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(1): 1–23.

Zhang, Q.-W.; Zhang, X.; Yan, Z.; Liu, R.; Cao, Y.; and Zhang, M.-L. 2021. Correlation-Guided Representation for Multi-Label Text Classification. In *IJCAI*, 3363–3369.

Zhang, R.; Wang, Y.-S.; Yang, Y.; Yu, D.; Vu, T.; and Lei, L. 2022b. Long-tailed Extreme Multi-label Text Classification with Generated Pseudo Label Descriptions. *arXiv preprint arXiv:2204.00958*.

Zhang, S.; Yao, Y.; Xu, F.; Tong, H.; Yan, X.; and Lu, J. 2019. Hashtag recommendation for photo sharing services. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 5805–5812.

Zhang, W.; Yan, J.; Wang, X.; and Zha, H. 2018. Deep extreme multi-label learning. In *Proceedings of the 2018 ACM on international conference on multimedia retrieval*, 100–107.

Zhou, J.; Zheng, Y.; Tang, J.; Jian, L.; and Yang, Z. 2022. FlipDA: Effective and Robust Data Augmentation for Few-Shot Learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 8646–8665.