Correlation-Guided Representation for Multi-Label Text Classification

Qian-Wen Zhang[1*], **Ximing Zhang**[2*†], **Zhao Yan**[1], **Ruifang Liu**[2], **Yunbo Cao**[1] and **Min-Ling Zhang**[3,4]

[1]Tencent Cloud Xiaowei, Beijing 100080, China

[2]Beijing University of Posts and Telecommunications, Beijing 100876, China

[3]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China

[4]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China

cowenzhang@tencent.com, ximingzhang@bupt.edu.cn, zhaoyan@tencent.com, lrf@bupt.edu.cn, yunbocao@tencent.com, zhangml@seu.edu.cn

Abstract

Multi-label text classification is an essential task in natural language processing. Existing multi-label classification models generally consider labels as categorical variables and ignore the exploitation of label semantics. In this paper, we view the task as a correlation-guided text representation problem: an attention-based two-step framework is proposed to integrate text information and label semantics by jointly learning words and labels in the same space. In this way, we aim to capture high-order label-label correlations as well as context-label correlations. Specifically, the proposed approach works by learning token-level representations of words and labels globally through a multi-layer Transformer and constructing an attention vector through word-label correlation matrix to generate the text representation. It ensures that relevant words receive higher weights than irrelevant words and thus directly optimizes the classification performance. Extensive experiments over benchmark multi-label datasets clearly validate the effectiveness of the proposed approach, and further analysis demonstrates that it is competitive in both predicting low-frequency labels and convergence speed.

1 Introduction

Multi-label text classification (MLTC) deals with real-world objects with rich semantics, where each text is simultaneously associated with multiple class labels that tend to be correlated. It is a fundamental task in natural language processing (NLP), which aims to learn a predictive model that assigns an appropriate set of labels to an unseen text. It is worth noting that to learn from multi-label text data, one needs to pay attention to two key factors: 1) How to generate more discriminative text representation; 2) How to effectively mine correlations to facilitate the learning procedure.

Text representation is critical in multi-label text classification. Transformer-based studies [Devlin *et al.*, 2019; Lan *et al.*, 2019] demonstrate the effectiveness of Transformer module for capturing the dependencies of all words in a sequence and provide a contextualized representation for classification tasks. Nevertheless, utilizing only contextual information to generate the text representation is suboptimal as it ignores the information conveyed by class labels and thus fails to take advantage of potential correlations among label-label and word-label. The fact that different labels may share the same subset of words is beneficial to help generate strong text representation. For example, academic literature containing keywords such as "neural network" is often tagged with "artificial intelligence" and "deep learning". Closely related labels tend to co-occurr. Therefore, it is rather desirable to fully exploit potential correlation information in text representation generation, which could be investigated in two-fold: 1) On one hand, label-label correlations can be exploited to extract latent inter-dependent features; 2) On the other hand, context-label correlations can be exploited to enhance discriminative abilities of extracted features. As far as we know, the simultaneous exploitation of both correlations has still not been well studied.

Generally, the learning process must be facilitated by exploiting correlations among labels in order to tackle the challenge of an exponential-sized output space for MLTC. Specifically, CNN-RNN [Chen *et al.*, 2017] presents an ensemble method of CNN and RNN to capture semantics and model label correlations. SGM [Yang *et al.*, 2018] captures high-order correlations between labels through the sequence generation model. We argue that correlations change dynamically in different contexts, so if we can learn words and labels jointly in the same space, we will get better label-label correlations as well as context-label correlations that fit a text. To further model the context-label correlations, several label embedding methods, including C2AE [Liu *et al.*, 2017], LEAM [Wang *et al.*, 2018], LSAN [Xiao *et al.*, 2019], X-Transformer [Chang *et al.*, 2020], etc., are proposed to take advantage of label information and construct label-specific text representation through the refinement of the word embedding. However, they fail to provide implicit information among label space,

*Equal contribution.

†Work done during an internship at Tencent.

which leads to the prediction bias in favor of the majority classes while ignoring the minority classes. Furthermore, such methods are limited to a certain extent when the labels do not carry semantic description information. As an example, "deep learning" is a label with description, but the symbol "DL" has no description. Providing labels with abbreviations or symbols in a dataset can lead to poor prediction performance or inapplicability.

Inspired by the potential of correlations, we import label semantics as auxiliary information by a global embedding strategy. The encoder learns word-word, label-label, and word-label correlations globally through Transformer module. Since not all text words contribute equally to the prediction, we construct an attention vector from the word-label correlation matrix to extract more discriminative words. The attention mechanism can improve performance with interpretability for text classification, which means that it helps relevant words to get higher attention than irrelevant words. To the best of our knowledge, we are the first to learn label-label and context-label correlations together with an attention-based two-step framework, and the main contributions of this paper include:

1. We propose a basic global embedding strategy that represents context and all class labels in the same latent space by Transformer to generate token-level representations, which captures correlations and reduces the dependence on label description information.
2. We propose an effective and novel method, called CORE, which exploits ***CO**rrelation-guided **RE**presentation* for multi-label text classification. CORE utilizes higher-order context-label correlations to guide attention processes and attempts to produce a semantic-aware representation.
3. Experimental results show that CORE achieves competitive performance against other state-of-the-art multi-label text classification approaches. We further provide a series of BERT-based methods and analyze the performance with macro-based and rank-based metrics. Results show that the utilization of label embedding and label correlations have a significant impact on the performance of our approach.

2 The CORE Approach

In this section, we first introduce the standard formal definition of multi-label text classification. Afterwards, the formulation of our method is illustrated. The technical details of CORE are detailed in three steps, including global embedding strategy, text representation learning, and predictive model induction.

2.1 Problem Formulation

Given a training set $S = \{(X_i, Y_i)\}_i^N$ of multi-label text classification data, where $X_i \in \mathcal{X}$ is the text sequence and $Y_i \subseteq \mathcal{Y}$ is its corresponding labels, the task of MLTC is to learn a predictive function f. More specifically, an input text sequence X of length m is composed of word tokens: $X = \{x_1, x_2, ..., x_m\}$, and $\mathcal{Y} = \{y_1, y_2, ..., y_l\}$ is the label space with l labels. Different from the single-label classification where only one label is associated with X_i, the multi-label classification function $f : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ assigns a set of possible class labels (Y, $0 \leq |Y| \leq l$) for the unseen text. Here, y_i is either regarded to be relevant ($y \in Y$) or irrelevant ($y \notin Y$) for instance $\mathcal{X}$. Note that we use $k_i \in \{0, 1\}$ to denote the categorical information of y_i.

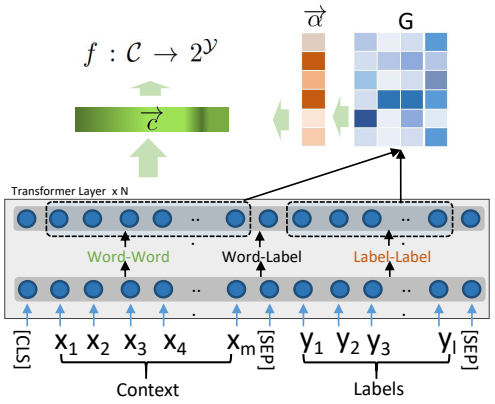


Figure 1: The framework of CORE. Specifically, word and label representations are first generated by a multi-layer Transformer encoder, which learns effectively about word-word, word-label, and label-label correlations through self-attention. After that, we focus on learning context-label correlation matrix by the output representations. Text representation vector is generated by the part of context output and attention vector, which is finally used to predict labels.

A typical text classification approach first preprocesses text data X for the model to obtain text representation $\mathcal{X}$. Then, the classifier annotates the text representation with a set of proper labels Y. Intuitively, the approaches utilize only the information from the input text sequence. Our method extends the input by adding label information. Therefore, the new input sequence of CORE is overlaid with both text and labels, which is composed of all tokens like: $\{X; \mathcal{Y}\} = \{x_1, x_2, ..., x_m; y_1, y_2, ..., y_l\}$, the number of labels is fixed to l in the data. The preprocessing is to obtain text representation $\mathcal{C}$ from context and labels. The aim of the predictive function $f : \mathcal{C} \rightarrow 2^{\mathcal{Y}}$ is to minimize a loss function which ensures that the model predicts relevant and irrelevant labels for each training instance with minimal misclassification.

2.2 Global Embedding Strategy

We utilize BERT [Devlin *et al.*, 2019], which outperforms state-of-the-art models on a wide range of NLP tasks, as the base encoder in the CORE framework. The basic architecture of BERT is a multi-layer bidirectional self-attention Transformer. For classification tasks, a special token [CLS] is put to the beginning of the text and the output vector of the token [CLS] is designed to correspond to the final text representation. Different from this operation, we unite the input text with all class labels, which are packed into a single sequence and separated by a special token [SEP].

Let $\{[CLS], x_1, x_2, ..., x_m, [SEP], y_1, y_2, ..., y_l, [SEP]\}$ be the token sequence which input into Transformer module. Note that the input representation of each given token

is constructed by summing the corresponding token, segment, and position embeddings. We simply use the same pre-trained model parameters as the official public model to initialize our model, and finetune all the parameters end-to-end to obtain the contextualized token-level representations H, i.e., $\{h_{[CLS]}, h_{x1}, ..., h_{xm}, h_{[SEP]}, h_{y1}, ..., h_{yl}, h_{[SEP]}\}$. As shown in Figure 1, the utilization of global embedding strategy guarantees that we consider both label correlations and context-label correlations in the same space at the beginning. Note that when label descriptions are unavailable, we represent each label with a new word token (unused token) to learn the hidden representation.

2.3 Text Representation Learning

To characterize the underlying structure of the contextualized representations, CORE works by constructing an attention vector $(H_x, H_y) \rightarrow \overrightarrow{\alpha}$. H_x corresponds to the set of context sequence representations and H_y corresponds to the set of label sequence representations. Since the input text is often flexible, we fixed the length for ease of use, i.e., the excess is trimmed off and the deficient is padded.

Attention Discovery

A simple way to measure the deep context-label correlations is to multiply matrix H_x by matrix H_y:

$$G = H_x H_y^T \tag{1}$$

where $G \in \mathcal{R}^{m \times l}$, note that H_x and H_y have been normalized by L2 norm.

We consider a further generalization of Eq. (1), which aims to strengthen the relative spatial information among consecutive tokens. In particular, for a text fragment of length $2r+1$ centered at p, the local matrix block $G_{p-r;p+r}$ in G measures the correlation in the word-label fragment pairs. Since the matrix G can be viewed as an extension in the spatial orientation, we use $g(\cdot)$ to denote a convolution layer with ReLU activation to learn the higher-order correlation matrix:

$$M = g(G_{p-r;p+r} W_1 + b_1) \tag{2}$$

where $M \in \mathcal{R}^{l \times m}$, $p \in \{1, ..., m\}$. Here, $W_1 \in \mathcal{R}^{2r+1}$ and $b_1 \in \mathcal{R}^l$ represent weight matrix and bias vector respectively. We compress the matrix M into a vector of length m by selecting the maximum value and reduce the effect of the fluctuating value:

$$\overrightarrow{\alpha} = \Omega(M) \tag{3}$$

Here, the length of $\overrightarrow{\alpha}$ is m, softmax and hyperbolic tangent are executed sequentially in $\Omega(\cdot)$. By learning the context-label correlation matrix M from Eq.(2), the attention vector $\overrightarrow{\alpha}$ can be instantiated in a manageable range in Eq.(3).

Text Representation Generation

Given the attention vector $\overrightarrow{\alpha}$, the original contextualized representations of text sequence can be transformed into an enriched version. In CORE, the final text representation is generated by aggregation of word representations, weighted by attention vector:

$$\overrightarrow{c} = \overrightarrow{\alpha} \cdot H_x \tag{4}$$

Intuitively, the text representation uses higher-order context-label correlations to guide attention processes. The nonlinear interaction between context and labels has been adequately considered to improve the performance.

Dataset	$\lvert\mathcal{S}\rvert$	$L(\mathcal{S})$	$WCard(\mathcal{S})$	$LCard(\mathcal{S})$
AAPD	55,840	54	163.42	2.41
RCV1-V2	804,414	103	123.94	3.24

Table 1: Characteristics of datasets. Here, $|\mathcal{S}|$ and $L(\mathcal{S})$ denote the total number of samples and labels, respectively. $WCard(\mathcal{S})$ means *Label Cardinality*, which is the average number of words per sample. $LCard(\mathcal{S})$ means *Label Cardinality*, which is the average number of labels per sample.

2.4 Predictive Model Induction

According to the objective function $f : \mathcal{C} \rightarrow 2^{\mathcal{Y}}$, the correlation-guided representation replaces the original text representation for multi-label prediction. We choose standard neural networks to annotate the correlation-guided representation with a set of relevant labels:

$$p = Sigmoid(W_2 \overrightarrow{c}^T + b_2) \tag{5}$$

where $W_2 \in \mathcal{R}^{l \times |\overrightarrow{c}|}$ and $b_2 \in \mathcal{R}^l$ are parameters to be learned. Notice that, the sigmoid function allows to deal with non-exclusive labels, while the softmax function only deals with exclusive classes.

In CORE, binary cross-entropy losses are used to measure probability errors in multi-label classification tasks where each class is independent, rather than mutually exclusive:

$$loss_i = -[k_i \ln p_i + (1 - k_i) \ln(1 - p_i)] \tag{6}$$

In order to minimize the loss function, we train the model end-to-end with all above parameters.

3 Experimental Setup

In this section, the datasets, comparing algorithms, evaluation metrics and parameter settings are introduced.

3.1 Datasets

We use two datasets for MLTC: AAPD [Yang *et al.*, 2018] and RCV1-V2 [Lewis *et al.*, 2004]. Table 1 summarizes the detailed characteristics of the two datasets. Each dataset is divided into a training set, a validation set, and a test set, which are used as basic divisions in the performance experiments of each algorithm [Yang *et al.*, 2018].

3.2 Comparing Algorithms

The performance of CORE is compared against the following multi-label algorithms:

• **SGM** [Yang *et al.*, 2018] proposes the sequence-to-sequence model with an attention mechanism to capture label correlations. Although label correlations are exploited, it ignores the use of label semantics to construct text representations.

• **Seq2Set** [Yang *et al.*, 2019] utilizes deep reinforcement learning to improve the performance of seq2seq model, which reduces the dependency of the label order. Similar to SGM, it lacks the use of label information.

• **LSAN** [Xiao *et al.*, 2019] makes use of content and label text to learn the label-specific text representation with the help of self-attention and label-attention mechanisms.

Algorithm	methods LE	methods LC	AADP dataset HL↓	AADP dataset Micro-P↑	AADP dataset Micro-R↑	AADP dataset Micro-F1↑	RCV1-V2 dataset HL↓	RCV1-V2 dataset Micro-P↑	RCV1-V2 dataset Micro-R↑	RCV1-V2 dataset Micro-F1↑
BR	no	no	0.0316	0.644	0.648	0.646	0.0086	0.904	0.816	0.858
CNN	no	no	0.0256	**0.849**	0.545	0.664	0.0089	0.922	0.798	0.855
BERT	no	no	0.0224	0.786	0.687	0.734	0.0073	**0.927**	0.832	0.877
CC	no	yes	0.0306	0.657	0.651	0.654	0.0087	0.887	0.828	0.857
LP	no	yes	0.0312	0.662	0.608	0.634	0.0087	0.896	0.824	0.858
CNN-RNN	no	yes	0.0278	0.718	0.618	0.664	0.0085	0.889	0.825	0.856
SGM	no	yes	0.0251	0.746	0.659	0.699	0.0081	0.887	0.850	0.869
Seq2Set	no	yes	0.0247	0.739	0.674	0.705	0.0073	0.900	0.858	0.879
LSAN	yes	no	0.0242	0.777	0.646	0.706	0.0076	0.913	0.841	0.875
LEAM	yes	no	0.0261	0.765	0.596	0.670	0.0090	0.871	0.841	0.856
LEAM$_{w/BERT}$	yes	no	0.0237	0.753	0.700	0.726	0.0077	0.893	0.857	0.875
BERT$_{onelab}$	yes	no	0.0239	0.775	0.659	0.712	0.0077	0.909	0.835	0.871
BERT$_{labseq}$	yes	yes	0.0236	0.742	**0.727**	0.734	0.0074	0.897	0.862	0.879
Ours (CORE)	yes	yes	**0.0210**	0.803	0.704	**0.750**	**0.0069**	0.911	0.864	**0.887**

Table 2: Predictive performance of each comparing algorithm on two datasets. **BERT**$_{onelab}$ and **BERT**$_{labseq}$ are comparable baselines that we proposed. Note that LE and LC indicate whether the algorithm considers **label embedding** and **label correlations**, respectively. HL, Micro-P, Micro-R denote hamming loss, micro-precision, and micro-recall, respectively. The best performance is highlighted in bold.

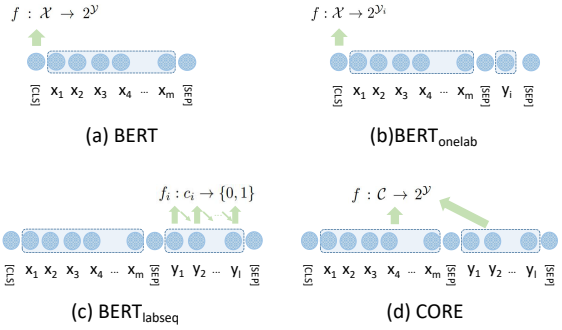


Figure 2: Illustration of different BERT-based methods. Subfigure(a) corresponds to the classical **BERT** of the classification task. Subfigure(b) represents a label embedding method that does not consider label correlations, corresponding to **BERT**$_{onelab}$. Subfigure(c) uses the label-related parts of the context-label representations, corresponding to **BERT**$_{labseq}$. Subfigure(d) is a simplified diagram of Figure 1, corresponding to **CORE**.

• **LEAM** [Wang *et al.*, 2018] applies label embedding in text classification, which obtains each label's embedding by its corresponding text descriptions. Here, we provide a comparable baseline method **LEAM**$_{w/BERT}$, which uses BERT to provide text encoding, but its labels are encoded independently.

• **BERT** [Devlin *et al.*, 2019] is a recently proposed language representation model that generates contextualized word vectors. For multi-label text classification, it only uses text as input, no label information. As is shown in Figure 2, we propose two comparable baseline methods. **BERT**$_{onelab}$ is input only one label at a time, and divides l labels into l times to perform the binary classification task. While incorporating label embedding, no label correlations can be learned. **BERT**$_{labseq}$ uses the output of the part of label sequence H_y as multiple contextual representations for multiple binary classification. It considers our global embedding strategy and solves the problem as a sequence annotation task with an additional BiLSTM-CRF layer.

More information about other baselines can be found in Binary Relevance (BR) [Boutell *et al.*, 2004], CNN [Kim, 2014], Classifier Chains (CC) [Read *et al.*, 2011], Label Powerset (LP) [Tsoumakas and Katakis, 2007] and CNN-RNN [Chen *et al.*, 2017].

3.3 Evaluation Metrics

Following the previous work [Yang *et al.*, 2018; Zhang and Zhou, 2014], we adopt **Hamming Loss** and **Micro-F1** as our main metrics, the micro-precision and micro-recall are also reported for reference. We further used **Macro-F1** assuming equal label weights as the key metric, which provides a different analytical perspective. The macro-precision, macro-recall are also reported for reference. When the intermediate real-valued function is available, macro-AUC, ranking loss, and coverage are provided as rank-based metrics, which expect related labels to score higher than the unrelated labels. For each evaluation metric, "↓" indicates "the smaller the better", while "↑" indicates "the larger the better".

3.4 Experimental Setting

We implement our experiments in Tensorflow on NVIDIA Tesla P40. In the experiments, we fine-tuned models on the base-uncased versions of BERT for English texts. The batch size is 32, the learning rate is $5e^{-5}$, and the window size of additional layer is 10. Based on $WCard(\mathcal{S})$ and $L(\mathcal{S})$ in Table 1, the maximum total input sequence length is 320. In addition, learning rate decay is added to the BERT training part, which starts with a large learning rate and then decays multiple times [Clark *et al.*, 2019]. Note that all BERT-based models in this paper use learning rate decay technique to improve performance.

4 Experimental Results

We report the detailed experimental results of all comparing algorithms on two datasets in Table 2. The following obser-

Algorithm	methods LE	LC	Macro-P↑	Macro-R↑	Macro-F1↑	Macro-AUC↑	RL↓	coverage↓
			AADP dataset					
BERT	no	no	0.687	0.521	0.572	0.8843	0.0866	0.2018
LEAM	yes	no	0.547	0.386	0.453	0.9372	0.0451	0.1048
LEAM$_{w/BERT}$	yes	no	0.627	0.555	0.577	0.8446	0.1123	0.2448
BERT$_{onelab}$	yes	no	0.666	0.483	0.534	0.9200	0.0480	0.1195
BERT$_{labseq}$	yes	yes	0.610	**0.585**	0.586	**0.9463**	**0.0362**	**0.0962**
Ours (CORE)	yes	yes	**0.704**	0.546	**0.595**	0.8866	0.0614	0.1545
			RCV1-V2 dataset					
BERT	no	no	**0.773**	0.619	0.667	0.9460	0.0310	0.1140
SGM	no	yes	0.713	0.680	0.681	-	-	-
LEAM	yes	no	0.741	0.649	0.692	0.9881	0.0073	0.0440
LEAM$_{w/BERT}$	yes	no	0.743	0.676	0.684	0.9543	0.0357	0.1176
BERT$_{onelab}$	yes	no	0.752	0.616	0.659	0.9866	0.0075	0.0436
BERT$_{labseq}$	yes	yes	0.735	0.678	0.686	**0.9949**	**0.0037**	**0.0308**
Ours (CORE)	yes	yes	0.759	**0.684**	**0.703**	0.9909	0.0064	0.0414

Table 3: The performance of different models on macro-based and rank-based metrics. Note that Macro-P, Macro-R, RL denote macro-precision, macro-recall, and ranking loss, respectively.

vations can be made according to the results:

1) Our proposed CORE presents the best performance in terms of hamming loss and micro-F1. We perform a significant test among the comparing algorithms suggesting that performance is statistically significant ($p < 0.05$). On AAPD dataset, compared to the traditional deep learning model CNN which only considers text content, CORE decreases by 17.97% on hamming loss and improves by 12.95% on micro-F1. As for BERT, CORE continues to perform well, with a relative reduction of 6.25% on hamming loss and an improvement of 2.18% on micro-F1. On RCV1-V2 dataset, compared with Seq2Set which only uses label semantics for corrected predictions, CORE achieves a reduction of 5.48% on hamming loss and an improvement of 0.91% on micro-F1. It shows that modeling correlations with label semantics can lead to performance gains.

2) With our global embedding strategy, BERT$_{labseq}$ has made a significant improvement in micro-recall compared to BERT. We argue that the potential correlations among label-label and word-label can help capture more meaningful features. BERT$_{onelab}$ predicts labels one by one, but compared to BERT, it achieves a reduction of 3.00% micro-F1 score on AAPD dataset. LEAM$_{w/BERT}$ improves LEAM by Transformer, but its performance is rather slightly lower than BERT because the labels are encoded independently. It indicates that the lack of label correlations may lead to performance degradation.

3) Algorithms like CNN, BERT, LSAN, etc., are biased to predict positive examples as negative examples, resulting in fewer matches than the actual number of samples in each class. CORE ensures good micro-precision while improving micro-recall. We attribute this phenomenon to the effect of the attention mechanism. According to the above observations, it is noteworthy that no algorithm significantly outperforms CORE across all evaluation metrics.

To summarize, comparing the proposed CORE against the recent state-of-the-art models, our method significantly improved previous state-of-the-art results in the main metrics.

5 Further Analysis

In this section, several studies are used to argue intuitively that CORE has good capability to learn high-order corrections and generate correlation-guided representation with competitive convergence speed. Moreover, CORE can classify each class well, even if that class is low-frequency.

We provide the macro-based and rank-based metrics in Table 3 to quantify the prediction performance from different analytical perspectives. Our proposed CORE shows the best performance on macro-F1, which proves that our method effectively improves the performance of all classes. In addition, BERT$_{labseq}$, which we proposed to validate the global embedding strategy, has the best performance on rank-based metrics. The direct use of label sequence representations has a more primitive preservation of label-label correlation, which favors relevant labels to rank higher than irrelevant ones. However, this method has weak word-label correlation and is prone to misclassification.

Alternatively, we divide labels on AAPD into three groups according to their occurring frequency. Nearly 56% of labels appearing more than 60 times are high-frequency labels and form Group1. Labels appearing 15-60 times form Group2 (34%), and the remaining 10% of labels form Group3. Figure 3 shows that all algorithms perform better on high-frequency labels (Group1) than on low-frequency labels (Group3), which is reasonable since there are more samples of high-frequency labels. More significantly, CORE improves macro-F1 on Group2 and Group3 compared to other methods, and it is more robust to classify mid/low-frequency labels. These results demonstrate the superiority of our proposed models in predicting low-frequency labels.

The convergence speed of three BERT-based models are shown in Figure 4. Both CORE and BERT$_{labseq}$ outperform BERT in terms of convergence speed. CORE converges significantly faster than BERT, which means that the performance of our proposed CORE can approach the optimal solution more efficiently by global embedding strategy and text representation learning.

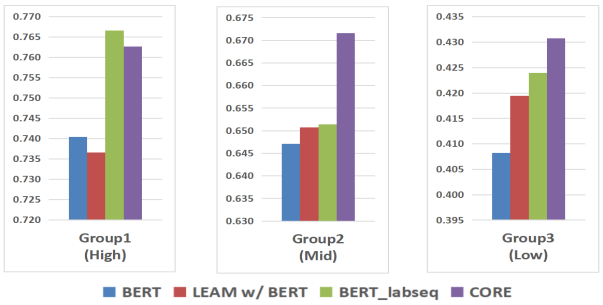
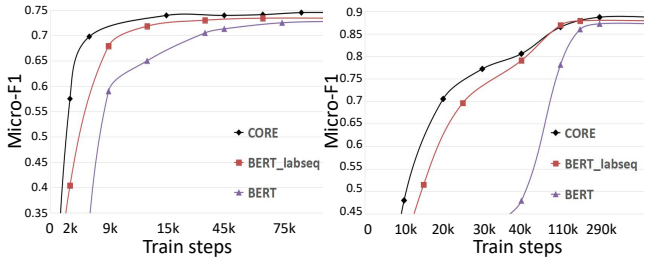


Figure 3: Macro-F1 for three groups on AAPD. All classes are sorted in descending order of their frequency of occurrence. Group1, Group2, and Group3 denote high-frequency labels, mid-frequency labels, and low-frequency labels, respectively.



(a) Covergence speed of AAPD (b) Covergence speed of RCV1-V2

Figure 4: The convergence speed of three BERT-based methods. The x-axis refers to the training steps, and the y-axis refers to the micro-F1 score performance.

In Figure 5, we visualize one example on the AAPD dataset. The dark orange means more important words. For the label "CV" and "CL", the selected informative words are `videos`, `image`, `movie`, `descriptions`, etc. For the label "CV" and "LG", the selected informative words are `images`, `machine`, etc. Benefiting from the interpretability of the attention mechanism, the text representation learning can correctly detect the key words with proper scores.

6 Related Work

Text representation plays a significant role in model performance. It is crucial to extract essential hand-crafted features for early models [Joachims, 1998], but features can be extracted automatically by DNNs. During the past decade, DNNs have been employed progressively in classification tasks by learning a set of nonlinear transformations that serve to map text directly to outputs, such as CNN [Kim, 2014]. Each word in the text is represented by a specific vector obtained through the word embedding technique [Joulin *et al.*, 2017]. Recently, Transformer, which is proposed by [Vaswani *et al.*, 2017], relies entirely on an attention mechanism to draw global dependencies between input and output. [Devlin *et al.*, 2019] is an important turning point in the development of text classification task and it works by generating contextualized word vectors using Transformer. [Yang *et al.*, 2018; Yang *et al.*, 2019] use the sequence-to-sequence model that consists of an encoder and a decoder connected through an attention mechanism. [Pang *et al.*, 2021] performs the few-shot learning for text classification. To further manipulate the

generating descriptions for videos has many applications including assisting blind people and human robot interaction the recent advances in image captioning.
- **reference labels**: "CV" and "CL"

compared to machines , humans are extremely good at classifying images into categories , especially when they possess prior knowledge of the categories at hand if this prior information is not available. we propose an interactive machine teaching algorithm that enables a computer to teach challenging visual concepts to a human our adaptive algorithm chooses
- **reference labels**: "CV" and "LG"

Figure 5: Visualization of attention mechanism on the AAPD dataset. The reference labels are displayed after abstracts. Note that "CV", "CL", and "LG" denote computer vision, computational language, and machine learning, respectively.

feature space, [Sun and Zhang, 2021] exploits distance metric to generate discriminative meta-level features.

Label embedding techniques are gaining attention and application in large-scale heterogeneous networks [Tang *et al.*, 2015], multi-class learning [Joshi *et al.*, 2017], and label distribution learning [Peng *et al.*, 2018]. [Pappas and Henderson, 2019] propose a non-linear transformation to capture the relationships across labels, which performs significantly better on unseen labels. [Liu *et al.*, 2017] is the first DNN-based multi-label embedding method that seeks a deep latent space to jointly embed the instances and labels. [Chang *et al.*, 2020] considers the extreme multi-label text classification (XML) problem and performs label embedding via label text or positive instances.

7 Conclusions and Future Work

In this paper, we present the CORE approach, which exploits correlation-guided representation for multi-label text classification. We first introduce the global embedding strategy which learns high-order corrections between context and all class labels in the same space. Then, the attention mechanism is used to highlight the most informative words in the text sequence. Extensive comparative studies clearly validate the superiority of our proposed CORE against state-of-the-art multi-label classification algorithms.

Our method treats all class labels as a label sequence, which means that our default labels are ordered. However, it can also be treated as an unordered set. On the other hand, one label is virtualized as one single token. If the label has descriptive text, there could be multiple tokens for semantic learning, which might be useful for XML problems. The above issues should be further explored in the future.

References

[Boutell *et al.*, 2004] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[Chang *et al.*, 2020] Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit S Dhillon. Taming pretrained transformers for extreme multi-label text classifi-

cation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3163–3171, 2020.

[Chen *et al.*, 2017] Guibin Chen, Deheng Ye, Zhenchang Xing, Jieshan Chen, and Erik Cambria. Ensemble application of convolutional and recurrent neural networks for multi-label text categorization. In *Proceedings of the International Joint Conference on Neural Networks*, pages 2377–2383, 2017.

[Clark *et al.*, 2019] Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. Electra: Pre-training text encoders as discriminators rather than generators. In *Proceedings of the International Conference on Learning Representations*, 2019.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[Joachims, 1998] Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, pages 137–142. Springer, 1998.

[Joshi *et al.*, 2017] Bikash Joshi, Massih R Amini, Ioannis Partalas, Franck Iutzeler, and Yury Maximov. Aggressive sampling for multi-class to binary reduction with applications to text classification. In *Advances in Neural Information Processing Systems 30*, pages 4159–4168, 2017.

[Joulin *et al.*, 2017] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, 2017.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv:1408.5882*, 2014.

[Lan *et al.*, 2019] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *Proceedings of the International Conference on Learning Representations*, 2019.

[Lewis *et al.*, 2004] David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5(Apr):361–397, 2004.

[Liu *et al.*, 2017] Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124, 2017.

[Pang *et al.*, 2021] Ning Pang, Xiang Zhao, Wei Wang, Weidong Xiao, and Deke Guo. Few-shot text classification by leveraging bi-directional attention and cross-class knowledge. *SCIENCE CHINA Information Sciences*, 2021.

[Pappas and Henderson, 2019] Nikolaos Pappas and James Henderson. Gile: A generalized input-label embedding for text classification. *Transactions of the Association for Computational Linguistics*, 7:139–155, 2019.

[Peng *et al.*, 2018] Cheng-Lun Peng, An Tao, and Xin Geng. Label embedding based on multi-scale locality preservation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2623–2629, 2018.

[Read *et al.*, 2011] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333, 2011.

[Sun and Zhang, 2021] Yan-Ping Sun and Min-Ling Zhang. Compositional metric learning for multi-label classification. *Frontiers of Computer Science*, 15(5):1–12, 2021.

[Tang *et al.*, 2015] Jian Tang, Meng Qu, and Qiaozhu Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1165–1174, 2015.

[Tsoumakas and Katakis, 2007] Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13, 2007.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, 2017.

[Wang *et al.*, 2018] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2321–2331, 2018.

[Xiao *et al.*, 2019] Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 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*, pages 466–475, 2019.

[Yang *et al.*, 2018] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. Sgm: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926, 2018.

[Yang *et al.*, 2019] Pengcheng Yang, Fuli Luo, Shuming Ma, Junyang Lin, and Xu Sun. A deep reinforced sequence-to-set model for multi-label classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5252–5258, 2019.

[Zhang and Zhou, 2014] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1819–1837, 2014.