MULTI-LABEL TEXT CLASSIFICATION VIA DOCUMENT ENHANCEMENT AND LABEL CORRELATIONS LEARNING

Chuzhen Li[1,2]*, Mohd Juzaiddin Ab Aziz[3], Mohd Ridzwan Yaakub[4]

[1*]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia (UKM), Bangi 43600, Malaysia.

[2]Department of Information Technology, Guangdong Technology College, Zhaoqing 526100, China.

[3]Center for Software Technology and Management (SOFTAM), Faculty of Information Science and Technology, University Kebangsaan Malaysia (UKM), Bangi 43600, Selangor, Malaysia.

[4]Center for Artificial Intelligence Technology (CAIT), Faculty of Information Science and Technology, University Kebangsaan Malaysia (UKM), Bangi 43600, Selangor, Malaysia.

Emails: p111565@siswa.ukm.edu.my[1,2]* (Corresponding Author), juzaiddin@ukm.edu.my[3], idzwanyaakub@ukm.edu.my[4]

ABSTRACT

Multi-label text classification (MLTC) has become increasingly popular due to its broader applicability and closer alignment with real-world objects' inherent properties and rules. Numerous approaches have been suggested to capture the label correlations. Yet, most of them capture relationships between labels in an implicit manner and typically do not explicitly distinguish or define label similarity correlations and label pairing correlations, but rather treat them as a unified label correlation. To this end, in this paper, we propose an approach to distinguish and explicitly define label similarity correlations and pairing correlations. The approach begins by acquiring text and label representations simultaneously. Next, the document representations are enhanced by concatenating with the most similar document subsets. Finally, the label similarity correlations and pairing correlations are explicitly learned in the label correlations learning. This approach shows that the performance surpasses the previous competitive models, with micro-F1 scores of 75.3% and 89.6% on the AAPD and RCV1-V2 datasets, respectively.

Keywords: Multi-label text classification; Label correlation; Document representation.

1.0 INTRODUCTION

The process of multi-label text classification is to distribute one or more categories to documents simultaneously [1], [2]. It is widely used in real-world life, such as information retrieval [3], emotion classification [4], news topic classification [5], etc. There are studies that have shown that label correlations can boost classification accuracy, especially when modeling low-frequency labels [6], [7], [8]. Although various approaches to capture correlations have been proposed, such as reinforcement learning [9], adaptive sorting [10], deep forest [11], reasoning mechanisms [12], label embedding [13], multi-task learning [14], [15], graph convolutional networks (GCN) [2], [16], [17], [18], [19], and prompt learning [1], [20], [21], the majority of these approaches implicitly capture the relationships between labels, generally without explicitly differentiating or specifying label similarity correlations and pairing correlations. As a result, the captured label correlations lack interpretability, the model struggles to manage complex label structures correctly, and classification accuracy diminishes.

This paper discusses an approach that could explicitly predict the relationship between label similarity and label pairing based on enhanced documents. Specifically, the document representation enhancement module utilizes the concept of a document subset. Each document is enriched by concatenating with the most similar document subsets. Additionally, labels' similarity and pairing relationships are defined and learned explicitly. Label representations are put into the label correlation learning module, and then cosine similarity and co-occurrence are utilized to assess the similarity and pairing relationships between labels.

To validate that our model surpasses other baselines, we apply it to classify the two benchmark datasets. The outcome will show performance comparable to or exceeding previous models.
The contributions of our study can be outlined below:

1. A new approach for MLTC is proposed, focusing on explicitly predicting the label similarity correlations and the label pairing correlations. In our work, we distinguish and define the label similarity and pairing

correlations to understand the relationships between labels better. Furthermore, we enhance labels by integrating label similarity and pairing correlations using an attention mechanism. This approach enhances the model's interpretability and generalization ability, helping the model better understand and handle the relationships between long-tail labels, thereby improving classification performance.

2. Inspired by the idea that labels may share a subset of documents, we concatenate the original document representations with the most similar document subsets to enhance the raw document representations. Motivated by the research conducted by [8] and [22], a joint embedding mechanism is employed, wherein representations for both documents and labels are acquired using a transformer-based encoder.
3. We perform comparative and ablation experiments, and provide a thorough analysis and summary of the results.

2.0 DOCUMENT REPRESENTATION LEARNING

Document representation learning is an essential step in multi-label text classification as it significantly influences the accuracy and efficiency of the MLTC models [23], [24]. Until now, various methods have been suggested for acquiring distinctive text representations, including the vector space model (VSM)[25], topic model [23], and word embedding [26], [27], [28]. Among them, VSM is one of the oldest and most commonly used methods, where each text is represented as a vector consisting of the weights of feature terms. The topic model is one of the probabilistic generative models, where each document is viewed as a mixture of topics and is composed of a set of words. Word embedding is a method for mapping words into real domain vectors. There are some main word embedding algorithms, such as Word2Vec [27], Glove [26], Bert [28], and so on.

Despite the success of the above approaches, they are still limited because they do not consider that documents do not always contain rich semantic information. Scholars have explored some studies to enhance document representation. Sinoara et al. [29] introduced a technique for improving document embeddings by incorporating semantic knowledge. This method utilizes both word embeddings and word-sense embeddings to produce enriched document representations. The central concept involves merging word sense disambiguation tools with pre-trained embeddings for words and word senses, creating semantically enhanced, low-dimensional representations. Peng et al. [30] presented another method to improve document embeddings for legal documents by integrating explicit knowledge elements and a multi-task learning framework. Ennajari et al. [24] proposed the method of knowledge-enhanced spherical representation learning. They use knowledge graphs as external domain knowledge to obtain entities in documents and represent them as knowledge graph embedding to enhance document representation. Chalkidis et al. [31] achieved enhancement of the original document representation by searching the top K nearest neighbor documents among all documents through cross-attention and then integrating and concatenating them with the original document as the input of the document. To augment the semantic representation of texts, LeBERT [32] was applied in sentiment analysis, which recognizes word N-grams through a sentiment lexicon and transforms the selected parts into word vectors by BERT. Xiong et al. leveraged a pre-trained BERT model to encode the document, obtaining hidden vectors for each word, and averaging these vectors to represent the entire document. Simultaneously, they introduced a contrastive learning mechanism that enhances document representation by comparing different encoded results of the same document [33]. However, the effect of contrastive learning is highly dependent on the selection of negative samples and the tuning of temperature hyperparameters. In the EMGAN model, the authors enriched the feature representation of short texts by constructing a heterogeneous graph and incorporating various types of nodes and edges. While the model provides significant improvements, it relies on external knowledge sources.

Unlike their approaches, our objective is to concatenate the original document representations with the most similar document subsets to enhance the raw document representations.

3.0 LABEL CORRELATION LEARNING

As MLTC gained popularity, many researchers have been dedicated to studying the correlation of labels to improve classification performance. According to the degree of label correlation, the methods of MLTC can be categorized into first-order correlation, second-order correlation, and higher-order correlation [34]. The algorithm adaptation methods, ML-DT [35] and ML-KNN [36], are first-order correlation methods, which can process each label independently. The second-order correlation methods, such as Rank-SVM [37], CMLPC [38], and LPLC [39], only consider pairwise correlations between labels and cannot consider correlations among all labels. Currently, most methods are based on deep learning, which belongs to high-order correlation. They can capture the correlation between all labels, but still face some challenges.

Yang et al. [6] are the first to use sequence generation models to capture high-order correlations of labels by using Bi-LSTM for generating context vectors and LSTM for sequential label prediction. After that, an enhanced sequence generation model using CNN and a fully initialized connection was proposed by Liao et al.[40]. However, they are sensitive to the order of labels, and errors tend to accrue rapidly.

To circumvent the issues mentioned above, a variety of methods have been developed, including reinforcement learning [9], sorting adaptively [10], deep forest [11], reasoning mechanism [12], label embedding [13], multi-task learning [14], [15], graph convolutional network (GCN) [2], [16], [17], [18], [19], etc.

More recently, scholars have been exploring how to apply prompt learning to MLTC better. Wei et al. [21] regarded exercises-concepts as MLTC tasks. They designed prefix templates for each exercise and then adopted MLM to predict tokens with the help of a threshold mechanism. Zhu et al. [20] pioneered the use of prompt learning in the context of short text classification. The proposed method builds a suffix template, which takes text and template together as input and utilizes a masked language model to predict the masked token label. Then, it grabs the top N concepts related to entities in short texts through the knowledge graph and then calculates the distance between each selected concept and each label to determine the final expanded label set. Song et al. [1] designed a prompt template system and employed MLM to predict the masked original texts and label tokens to learn semantic correlations between labels and texts. Wang et al. [41] discussed two methods of how conceptual knowledge can help text classification to achieve good classification performance explicitly.

Although the above works mentioned show better performance, most of them do not explicitly capture label correlations and do not distinguish or define label correlations. Unlike them, we explicitly differentiate and define the label similarity and pairing correlations when capturing label correlations. Moreover, we enrich labels with label similarity correlations and pairing correlations by attention mechanism, thereby improving classification accuracy and making label correlations interpretable.

4.0 SUBMISSION NOTICE

Fig.1 presents the comprehensive structure of our proposed model, segmented into four distinct modules: document-label joint embedding, document representation enhancement, new label correlation learning, and multi-label text classification.

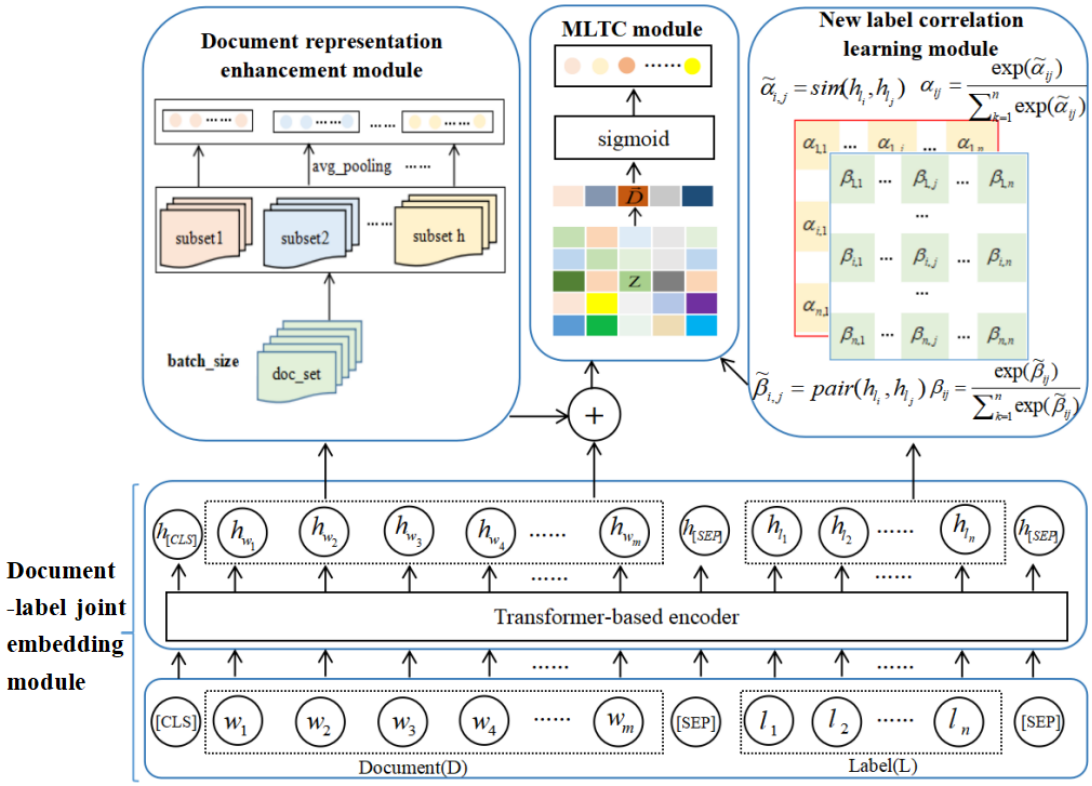


Fig.1: The proposed model

4.1 Problem formulation

Multi-label text classification aims to determine which categories each document belongs to. Suppose there is a training dataset, $T = \{(D_i, L_i)|1 \leq i \leq k\}$, which has k training documents. The labels of these documents form a label set $\ell = \{l_1, l_2, \ldots l_n\}$, where n is label size. Each document sequence $D_i = \{\omega_1, \omega_2, \ldots \omega_m\}$ is composed of m word tokens, and $L_i \subseteq \ell$ for each document D_i. MLTC develops a classification function $f(\cdot): D \rightarrow 2^{\ell}$ from the training dataset T, and then uses the classification function $f(\cdot)$ to predict the most relevant label set $L^* \subseteq \ell$ for unseen documents in the testing dataset.

4.2 Document-label joint embedding

Motivated by the research conducted by Zhang et al.[8] and Xiong et al.[22]. We also adopt a joint embedding mechanism that utilizes a transformer-based encoder to acquire representations for both documents and labels concurrently. With this design, the model can capture the direct interaction between labels and text. Among the numerous models based on transformers, BERT [28] stands out as a highly influential and extensively utilized model because of its ability to acquire word embedding that effectively captures the contextual meaning of words. Hence, we leverage BERT as an encoder to capture the joint embedding of documents and labels.

Specifically, an original document and labels are concatenated with a [SEP] token, and the start of each document is appended with a [CLS] token. They constitute an input sequence that is subsequently fed into the encoder. The hidden representation of documents and labels is ultimately derived through the Transformer-based encoder, taking BERT as an example. Formally, a document D is denoted as $D = \{\omega_1, \omega_2, \dots, \omega_m\}$ and its corresponding labels are presented as $L = \{l_1, l_2, \dots, l_n\}$, where m represents the length of the document and n signifies the number of labels. Consequently, the concatenated input sequence is denoted as $\{[CLS], \omega_1, \omega_2, \dots, \omega_m, [SEP], l_1, l_2, \dots, l_n, [SEP]\}$. Through the Bert model, the output contextualized document representation is expressed as $H_D = \{h_{\omega_1}, h_{\omega_2}, \dots, h_{\omega_m}\}$ and the label representation is represented as $H_L = \{h_{l_1}, h_{l_2}, \dots, h_{l_n}\}$. Accordingly, the entire document-label representation is shown as $H = \{h_{[CLS]}, h_{\omega_1}, h_{\omega_2}, \dots, h_{\omega_m}, h_{[SEP]}, h_{l_1}, h_{l_2}, \dots, h_{l_n}, h_{[SEP]}\}$.

4.3 Document-label joint embedding

Not all documents contain rich semantic features, especially when doing short text classification and few-shot classification [24]. For this reason, when performing MLTC, the classification performance tends not to be optimal if the original documents are used solely for classification. To improve accuracy and performance, inspired by the idea that labels may share a subset of documents, we fuse the context-hidden states of the most similar document subsets for document enhancement.

Here, we give more detailed explanations in two scenarios, one where the dataset has explicit categories, topics, or keywords, and the other where the dataset does not. Assume numerous documents constitute a document set. For the first case, i.e., the documents contain information on corresponding categories, topics, or keywords, we can directly divide the document set into multiple document subsets based on this information and then splice the original document representation with the document subset vectors exhibiting the highest similarity with the document representation. To accelerate computation speed, prevent overfitting, and retain key features, this paper opts for average pooling to reduce the dimensionality of the document subset vectors. Hence, the concatenated process is shown in Equations (1) ~ (2). It is assumed here that the document subset Sub_S is the most similar to the document D.

$$H_{Sub\_S} = avg\_pooling(H_{D_1}, H_{D_2}, \dots, H_{D_k}) \quad (1)$$

$$H_D^{'} = concat(H_{Sub\_S}, H_D) \quad (2)$$

where H_{D_k} represents the context representation of the $k-th$ document in the document subset Sub_S, H_{Sub_S} denotes the context representation of Sub_S after avg_pooling, H_D and $H_D^{'}$ denote the original and enhanced context representation of document D, respectively.

For the second scenario, we achieve the goal of document representation enhancement by multiplying the document matrix $S \in R^{k \times m}$ with a projection matrix $W_1 \in R^{m \times h}$ which is a trained parameter matrix. Specifically, it performs a linear transformation on document vectors in the document matrix S, resulting in a new document matrix $C \in R^{k \times h}$. This new matrix C maps each document vector to all category labels. The definition of the document matrix S is shown in Equation (3).

$$S = \{H_{D_1}, H_{D_2}, \dots, H_{D_k}\}^T = \begin{bmatrix} H_{\omega_{11}} & H_{\omega_{12}} & \cdots & H_{\omega_{1m}} \\ H_{\omega_{21}} & H_{\omega_{22}} & \cdots & H_{\omega_{2m}} \\ \vdots & \vdots & \vdots & \vdots \\ H_{\omega_{k1}} & H_{\omega_{k2}} & \cdots & H_{\omega_{km}} \end{bmatrix} \quad (3)$$

where H_{D_k} represents the $k-th$ contextualized document representation, k is the number of documents in the batch_size, $H_{\omega_{km}}$ represents the contextual representation of the $m-th$ word in the $k-th$ document, m represents the number of words per document.

A new matrix C is obtained as shown in Equation (4).

$$C = SW_1 = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1h} \\ p_{21} & p_{22} & \cdots & p_{2h} \\ \vdots & \vdots & \vdots & \vdots \\ p_{k1} & p_{k2} & \cdots & p_{kh} \end{bmatrix} \quad (4)$$

where h is the number of document subsets, p_{kh} denotes the probability value of the $k-th$ document assigning to the $h-th$ subset, $W_1 \in R^{m \times h}$ is a randomly initialized projection matrix.

To convert the values in the matrix C into probability distributions, we choose the SoftMax function for normalization. Equation (5) illustrates the calculation process.

$$O = soft\,max(C) \quad (5)$$

After normalization, the function $argmax()$ can retrieve the index associated with the highest probability for each document. The calculation process is shown in Equation (6).

$$Y = argmax(O) \quad (6)$$

Here $Y \in R^{k \times 1}$ denotes a $k-$ dimensional column vector and each dimension represents the index of the highest probability of the category corresponding to each document.

Then, we divide documents with the same index into the same document subset, forming multiple document subsets. Ultimately, similar to the first scenario, we concatenate the original document representation with the document subset vectors exhibiting the highest similarity with the original document as in Equation (1) to (2). In conclusion, the process of document representation enhancement can be represented in Algorithm 1.

Algorithm 1: The process of document representation enhancement

Input: Document matrix S in batch_size
Output: Enhanced document representation $H_{D_j}^{'}$

1. Sub_S := []
2. if S has no explicit categories do:
3. For H_{D_i} in S do:
4. $C_i \leftarrow H_{D_i} \times W_1$
5. $O_i = soft\,max(C_i)$
6. $Y_i = argmax(O_i)$
7. $Sub\_S \leftarrow H_{D_i}$ if $Y_i = Y_j$
8. else:
9. For H_{D_i} in S do:
10. $Sub\_S \leftarrow H_{D_i}$ if $Y_i = Y_j$
11. For H_{D_j} in Sub_S do:
12. $H_{Sub\_S} \leftarrow avg\_pooling(H_{D_1} + H_{D_2} + \cdots + H_{D_k})$
13. $H_{D_j}^{'} \leftarrow concat(H_{Sub\_S}, H_{D_j})$
14. end

4.4 New label correlations learning

As mentioned earlier, most studies do not explicitly capture label correlations and do not distinguish or define label correlations. Therefore, in our work, the correlation of labels is refined into similarity and pairing correlations. Our paper focuses on measuring these two correlations and making them interpretable. Here, we leverage cosine similarity to measure the similarity of two label representations. It refers to the degree to which two or more labels exhibit high similarity in the semantic or feature space. The detailed calculation process can be found in Equations (7) through (8).

$$\tilde{\alpha}_{ij} = sim(h_{l_i}, h_{l_j}) = cos(h_{l_i}, h_{l_j}) = \frac{h_{l_i} \times {h_{l_j}}^T}{|h_{l_i}| \times |h_{l_j}|} = \frac{\sum_{k=1}^{n}(h_{l_{ik}} \times h_{l_{jk}})}{\sqrt{\sum_{k=1}^{n}(h_{l_{ik}})^2} \times \sqrt{\sum_{k=1}^{n}(h_{l_{jk}})^2}} \quad (7)$$

$$\alpha_{ij} = \frac{exp(\tilde{\alpha}_{ij})}{\sum_{k=1}^{n} exp(\tilde{\alpha}_{ik})} \quad (8)$$

where h_{l_i} and h_{l_j} represent the $i-th$ and $j-th$ label representation of documents, respectively.
In addition, there is a situation where the similarity between two labels is minimal, but they often appear simultaneously. That is, one label often appears together with the other. We call this correlation a label pairing correlation. It can be expressed by Equations (9) ~ (11).

$$\tilde{\beta}_{ij} = pair(h_{l_i}, h_{l_j}) = \frac{co-occurrence(h_{l_i}, h_{l_j})}{sim(h_{l_i}, h_{l_j})} \quad (9)$$

$$\beta_{ij} = \frac{exp(\tilde{\beta}_{ij})}{\sum_{k=1}^{n} exp(\tilde{\beta}_{ik})} \quad (10)$$

$$\text{here } co-occurrence(h_{l_i}, h_{l_j}) = \frac{f(h_{l_i}, h_{l_j})}{max(f(h_{l_i}), f(h_{l_j}))} \quad (11)$$

The function $f(.)$ is used to calculate the number of occurrences of a label.
Finally, the enhancement result of the final label representation is shown in Equations (12) ~ (13).

$$H_{l_i} = \sum_{j=1}^{n} \alpha_{ij} h_{l_j} + \sum_{j=1}^{n} \beta_{ij} h_{l_j} \quad (12)$$

$$H_L^{'} = (H_{l_1}, H_{l_2}, \dots, H_{l_n}) \quad (13)$$

4.5 Multi-label text classification

4.5.1 Document-label attention

The labels convey specific details and establish semantic connections with corresponding words in the original text. Hence, we employ the dot product to gauge the semantic similarity between the enhanced words and labels. The calculation process can be found in Equation (14).

$$Z = H_D^{'T} H_L^{'} \quad (14)$$

where $H_D' = \{H_{w_1}, H_{w_2}, \dots, H_{w_m}\}$ is the enhanced contextual representation of document, $H_L' = (H_{l_1}, H_{l_2}, \dots, H_{l_n})$ is the enhanced contextual representation of label and $Z \in R^{m \times n}$. Taking into account the semantic relationships among consecutive words, we refer to [8] to extend M using a nonlinear network. First of all, the correlation of the label-phrase pairs is measured by the local matrix block $Z_{i-r;i+r}$ which is centered at i. Then, in hidden layers, the CNN with ReLU is used to enhance the efficiency of sparse regularization. Subsequently, we apply max-pooling and tanh sequentially within the function Ω. Hence, the final document representation $\vec{D}$ can be represented as (15).

$$\vec{D} = \Omega(Z_{i-r;i+r}) H_D^{'} \quad (15)$$

4.5.2 Label prediction

After we obtain the final document representation, a fully connected layer is employed to construct a classifier that integrates category-discriminative information. Subsequently, a sigmoid function is applied for non-linear transformation to estimate the probability of each document being associated with each label. The calculation process is shown in Equation (16).

$$\vec{p} = sigmoid(W_2 \vec{D}^T + b_2) \quad (16)$$

Finally, a threshold mechanism is used to identify labels associated with documents, formulated as (17):

$$\vec{p}(D) = \{l_i | \vec{p} > t, l_i \in L\} \quad (17)$$

where t is a threshold. For convenience, we set t to 0.5.

4.5.3 Loss function

Research has established the efficacy of utilizing binary cross-entropy loss (BCE) with sigmoid activation for addressing multi-label tasks, demonstrating its superior performance over cross-entropy loss [7]. Consequently, within the context of our paper, BCE is employed as the designated loss function for parameter learning in the MLTC task, represented by formulation (18):

$$Loss = -\sum_{i=1}^{n}[y_i\, log(\, p_i) + (1 - y_i)\, log(1 - p_i)] \qquad (18)$$

where n indicates label size, $y_i \in \{0,1\}$ and $p_i \in \{0,1\}$ represent the true outcome and predicted outcome of the document belonging to the label l_i, respectively.

5.0 EXPERIMENTS

Within this section, we employ two multi-label datasets to appraise our approach. Initially, we furnish details of datasets, baselines, evaluation metrics, and experimental configurations. Subsequently, a comparative analysis is conducted, pitting the experimental outcomes of our approach against those of the baseline methods. Ultimately, we perform ablation studies on our model.

5.1 Datasets

To validate the effectiveness of our proposed model, similar to [8], [12], [40], [42], we use two golden public datasets, AAPD and RCV1-V2, for experiments.

1) **Arxiv Academic Paper Dataset (AAPD):** It was acquired from the Arxiv website and covers the abstract and associated subjects of 55,840 computer science publications with a total of 54 labels [6].
2) **Reuters Corpus Volume I (RCV1-V2):** It comprises 804,414 manually classified news reports grouped into 103 labels. It is offered for research purposes by Reuters Ltd [43]. Each news report can have multiple themes associated with it.

The specific information of the datasets is shown in Table 1. This work divides these three datasets according to [6]. Each dataset is segmented into training, validation, and testing sets.

Table 1: Overview of the experimental datasets

Datasets	Number of samples	Number of labels	Average number of words in the sample	Average number of labels in the sample
AAPD	55, 840	54	163.43	2.41
RCV1-V2	804, 414	103	123.94	3. 24

5.2 Baselines

To thoroughly assess the behavior of our model, we conduct comparative experiments with various baseline algorithms, encompassing classical machine-learning-based methods (i.e., BR [44], CC [45]), traditional word embedding-based methods (i.e., SGM [6], Seq2Set [9], ML-Reasoner [12]), and pre-trained-based methods (i.e., BERT [28], MAGNET [2], LACO [8], LP-MTC [1], CFTC [46]).

- **Binary Relevance (BR):** By treating each label as a separate binary classification, the BR algorithm transforms the MLTC issue into many independent binary classification problems.
- **Classifier Chains (CC):** It reformulates MLTC as a sequence of binary classification processes. Each classifier in the chain takes as input the complete set of labels generated before it, considering the correlation between them.
- **SGM:** The model transforms the problem of MLTC into a problem of generating sequences to capture label correlations [6].
- **Seq2Set:** Based on SGM, a set decoder is incorporated to minimize the effects of disordered labels on MLTC by utilizing the disorder of the set. Seq2Set presents deep reinforcement learning to capture label correlations.
- **ML-Reasoner:** It uses a binary classifier for simultaneous prediction of all labels and employs an innovative iterative reasoning approach to acquire relationships between labels.
- **BERT:** A language representation model designed to jointly consider both the left and right context across all layers to pre-train deep bidirectional representations from documents. It can output the probability of each label independently.
- **MAGNET:** It leverages two matrices, one is feature vectors and the other is a correlation matrix, to get label correlation automatically on the basis of a graph neural network.
- **LACO:** It employs a joint embedding mechanism to acquire representations for text and labels and uses self-attention to learn label correlations.

- **LP-MTC:** Its idea is to transform the MLTC task into a prompt learning task. A prompt template system that integrates labels and documents as input is designed, and masked language models are utilized for label prediction.
- **CFTC:** It aims to acquire label correlations while eliminating correlation bias. It utilizes a predict-and-adjust framework to obtain label information ingrained within label dependencies. Following this, it employs the counterfactual de-bias technique to impede correlation shortcuts.

5.3 Evaluation metrics

The same as the earlier research [6], [8], [9], our paper applies micro/macro-F1 score ($F1$) and hamming Loss (HL) as the critical metrics for evaluation, while also including micro/macro-P (P) and micro/macro-R (R) to support the analysis.

Suppose that n is the label size, j denotes the $j-th$ label ($1 \le j \le n$), k presents document size in the testing dataset, $\hat{Y}_j$ and Y_j indicates the predicted and actual dataset of the $j-th$ label, respectively. The calculation formulas are shown as follows:

$$micro - P = \frac{\sum_{j=1}^{n} |\hat{Y}_j \cap Y_j|}{\sum_{j=1}^{n} |\hat{Y}_j|} \quad (19)$$

$$micro - R = \frac{\sum_{j=1}^{n} |\hat{Y}_j \cap Y_j|}{\sum_{j=1}^{n} |Y_j|} \quad (20)$$

$$micro - F1 = \frac{2micro\_P \times micro\_R}{micro\_P + micro\_R} \quad (21)$$

$$macro - P = \frac{1}{n}\sum_{j=1}^{n} \frac{|\hat{Y}_j \cap Y_j|}{|\hat{Y}_j|} \quad (22)$$

$$macro - R = \frac{1}{n}\sum_{j=1}^{n} \frac{|\hat{Y}_j \cap Y_j|}{|Y_j|} \quad (23)$$

$$macro - F1 = \frac{2micro\_P \times micro\_R}{micro\_P + micro\_R} \quad (24)$$

$$HL = \frac{1}{k \times n}\sum_{i=1}^{k}\sum_{j=1}^{n} xor(\hat{Y}_{ij}, Y_{ij}) \quad (25)$$

where xor denotes the XOR operation, $\hat{Y}_{ij}$ and Y_{ij} respectively mean the predicted outcome and true outcome of the $j-th$ label in the $i-th$ sample.

5.4 Experimental settings

Our experiments were run on an autoDL platform, which was built with TensorFlow 2.9.0, Python 3.8, GPU with V100 (32GB) *2, and Memory 120GB. We adopt a 12-layer BERT-base case as a pre-trained language model with a vector dimension of 768 and a total of 110 M parameters. Each batch is configured to 64 documents with an epoch of 200. The maximum input sequence lengths for the two datasets are 120 and 512, respectively. We train and fine-tune models while tracking micro-F1 on the validation set. In addition, to accelerate network training, we also choose AdamW with a learning rate of 1e-4 as an optimizer. Meanwhile, the learning rate is adjusted using lr_scheduler, with the stepLR function parameter step_size set to 1000 and gamma set to 0.99.

5.5 Main results

Tables 2 and 3 report the outcomes of our model and all baseline models on the AAPD and RCV1-V2 datasets, respectively. Results show our model significantly surpasses most of the baseline models in the primary evaluation metrics on both the AAPD and RCV1-V2 datasets.

From Table 2, we can see that compared with traditional machine learning methods, our model achieves apparent improvements in AAPD. It gains an improvement of 15.14% in micro-F1 and a decline of 28.1% in hamming loss relative to CC. This advantage stems from machine learning methods' inherent challenges in accurately extracting semantic information from text.

In addition, our model also outperforms word embedding-based baselines across the primary evaluation metrics. For instance, the proposed model demonstrates a 12.35% reduction in hamming loss and a 7.73% increase in micro-F1 score compared to SGM. Compared to Seq2Set, it achieves a 10.93% decrease in hamming loss and a 6.81% enhancement in micro-F1. This indicates that our model, which is based on BERT, is overall superior to that of word embedding-based models. We attribute this superiority to pre-trained language models, such as BERT, which can capture more nuanced semantic information than traditional deep learning algorithms.

Finally, our model has a comparable or exceeding outcome compared to other algorithms utilizing a pre-trained language model. For instance, our model improves the micro-F1 by 2.59% and decreases hamming loss by 1% than CFTC. Therefore, the comparison results imply that the concatenation of document representations with the most similar document subsets can help capture richer semantic information, and by explicitly modeling label similarity and pairing correlations, the model can better understand the relationships between labels.

Table 2: Performance on AAPD

Model	micro (+)			macro (+)			HL (-)
	P	R	F1	P	R	F1	
BR[44]	0.644	0.648	0.646	-	-	-	0.0316
CC[45]	0.657	0.651	0.654	-	-	-	0.0306
SGM[6]	0.746	0.659	0.699	-	-	-	0.0251
Seq2Set[9]	0.739	0.674	0.705	-	-	-	0.0247
ML-Reasoner[12]	0.726	**0.718**	0.722	-	-	-	0.0248
BERT[28]	0.786	0.687	0.734	0.687	0.521	0.572	0.0224
MAGNET[2]	-	-	0.696	-	-	-	0.0252
LACO[8]	**0.802**	0.696	0.745	0.704	0.540	0.591	0.0213
LP-MTC[1]	0.774	0.711	0.741	-	-	-	0.0221
CFTC[46]	0.793	0.684	0.734	-	-	-	0.0222
Our model	0.797	0.714	**0.753**	**0.707**	**0.546**	**0.595**	**0.0220**

Table 3: Performance on RCV1-V2

Model	micro (+)			macro (+)			HL (-)
	P	R	F1	P	R	F1	
BR[44]	0.904	0.816	0.858	-	-	-	0.0086
CC[45]	0.887	0.828	0.857	-	-	-	0.0087
SGM [6]	0.887	0.850	0.869	-	-	-	0.0081
Seq2Set [9]	0.900	0.858	0.879	-	-	-	0.0073
ML-Reasoner[12]	0.890	0.852	0.871	-	-	-	0.0079
BERT[28]	**0.927**	0.832	0.877	**0.773**	0.619	0.667	0.0073
MAGNET[2]	-	-	0.885	-	-	-	0.0079
LACO[8]	0.908	0.856	0.881	0.759	0.666	0.692	0.0072
CFTC [38]	0.905	**0.874**	0.889	-	-	-	0.0068
Our model	0.921	0.872	**0.896**	0.771	**0.675**	**0.720**	**0.0061**

The first group of methods is based on machine learning, the second group relies on traditional word embedding, and the third group employs pre-trained models. The '-' signifies that better performance is associated with a lower score, whereas '+' indicates that a higher score is preferred. The bold denotes the results are the best.

As for the results on RCV1-V2, our method and the baselines perform better overall compared to AAPD, as shown in Table 3. We believe this is due to RCV1-V2 having more samples and labels than AAPD during model training. More samples can help the model learn additional features, providing richer information about label correlations.

Furthermore, consistent with the results observed on the AAPD, our model outperforms most baseline models on the primary evaluation metrics for RCV1-V2. From the comparison result that our model improves micro-F1 by 2.17% and decreases hamming loss by 16.44% compared to Bert, we can conclude that Bert focuses more on global information in documents. In contrast, our proposed model also considers information between labels. In addition, compared with LACO, our model achieves a 15.28% reduction in hamming loss and a 1.7% improvement in micro-F1 score. This further demonstrates our model has a more remarkable ability to predict labels with higher accuracy by incorporating document subsets and explicitly modeling label correlations.

5.6 Ablation experiment

In this subsection, we analyze how different parameters affect our proposed method, focusing on input sequence lengths (N), batch size, and subsets within the document representation enhancement module. During the experiments, one parameter is varied while the others remain constant. The input sequence lengths, batch size, and the number of document subsets are selected from the sets {30, 60, 80, 120, 320, 640, 1024}, {2, 4, 8, 16, 32, 64, 128}, {1,2,3,4,5,6,7,8,9,10}, respectively.

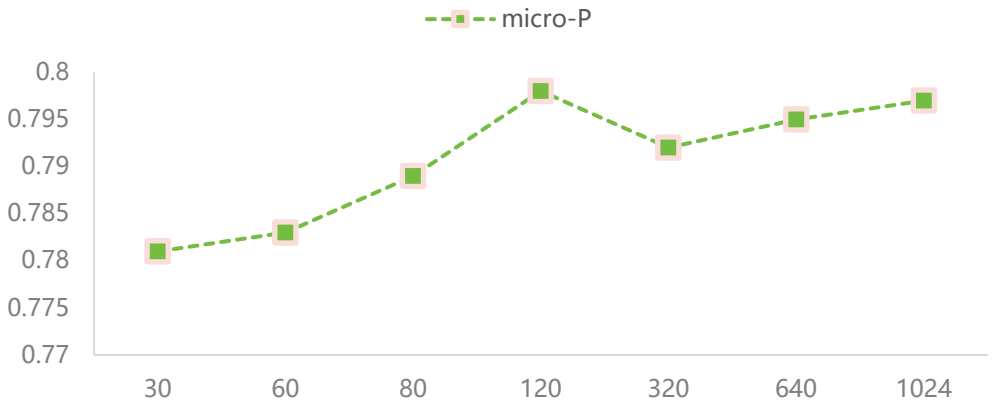
5.6.1 Impact of input sequence lengths

Based on Table 4 and Fig.2, it's evident that the length of the input text influences different evaluation metrics in classification performance. This occurs primarily because the amount of contextual information captured by the Transformer-based encoder depends on the text length. Specifically, shorter input lengths are inadequate in capturing sufficient contextual information, resulting in lower performance metrics. As the input length increases, the model gains the ability to capture more relevant context, which enhances classification performance. With an input length of 120, the model captures the optimal amount of contextual information without being overwhelmed by excessive data, achieving the highest micro-F1 score of 75.3% and the lowest Hamming loss of 2.2%. Beyond an input length of 320, the performance plateaus since additional context does not notably improve classification, likely due to the model's capacity constraints or diminishing returns from the extra information.

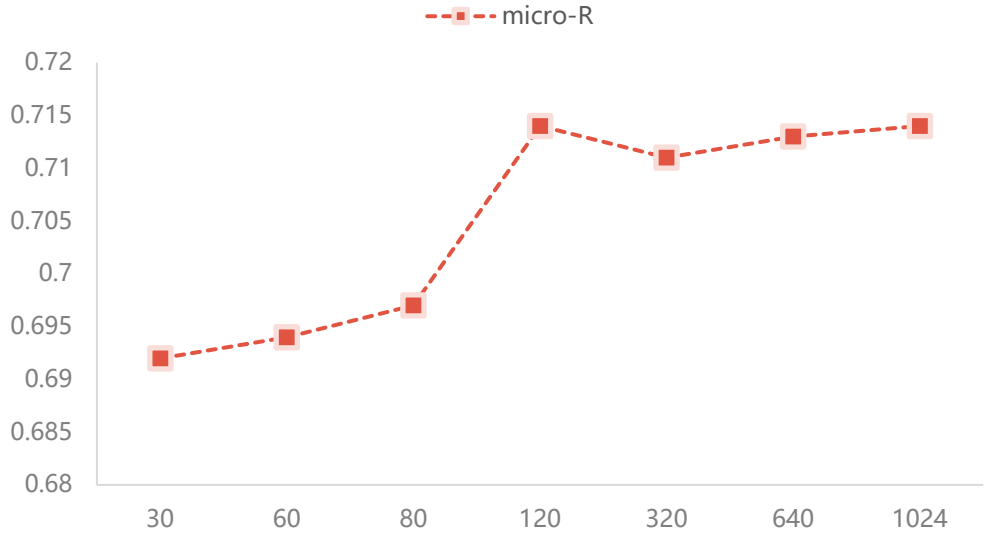
Table 4: Impact of input sequence lengths on AAPD

N	micro-P	micro-R	micro-F1	macro-P	macro-R	macro-F1	HL
30	0.781	0.692	0.733	0.694	0.536	0.582	0.0254
60	0.783	0.694	0.735	0.695	0.537	0.582	0.0247
80	0.789	0.697	0.740	0.698	0.539	0.586	0.0235
120	**0.798**	**0.714**	**0.753**	0.706	0.544	0.592	**0.0220**
320	0.792	0.711	0.749	0.705	**0.546**	0.593	0.0222
640	0.795	0.713	0.751	**0.707**	0.545	**0.595**	0.0221
1024	0.797	**0.714**	**0.753**	**0.707**	**0.546**	**0.595**	**0.0220**

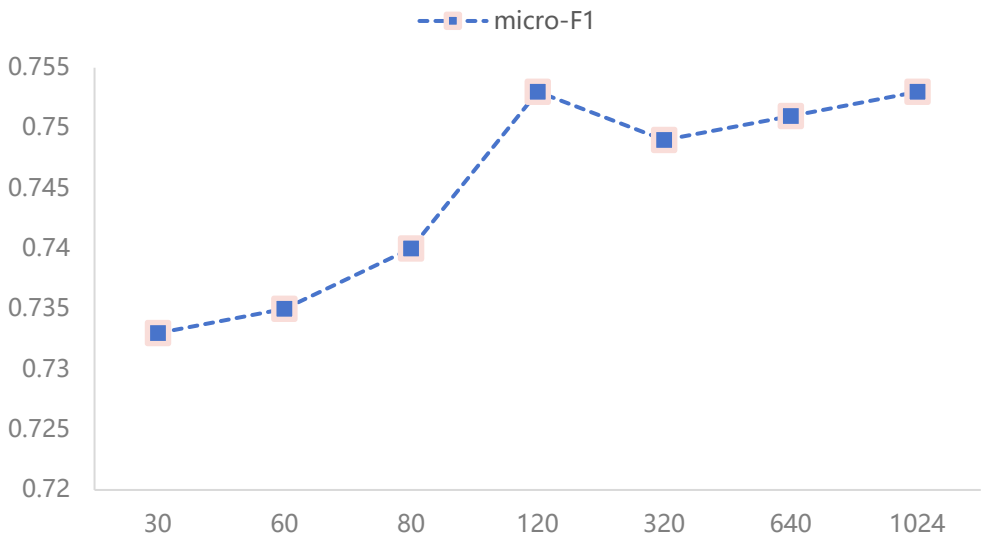
The bold denotes the optimal results



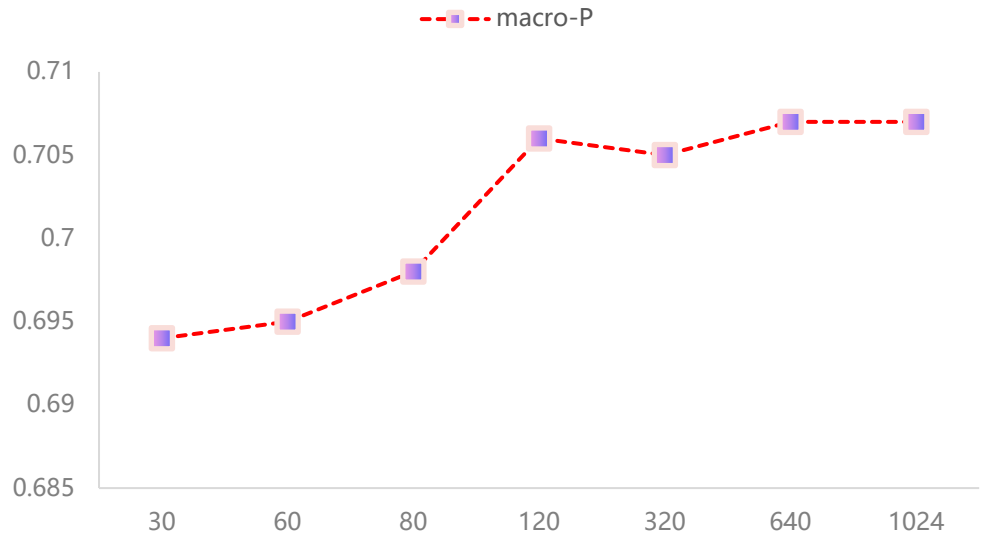
(a)micro-P



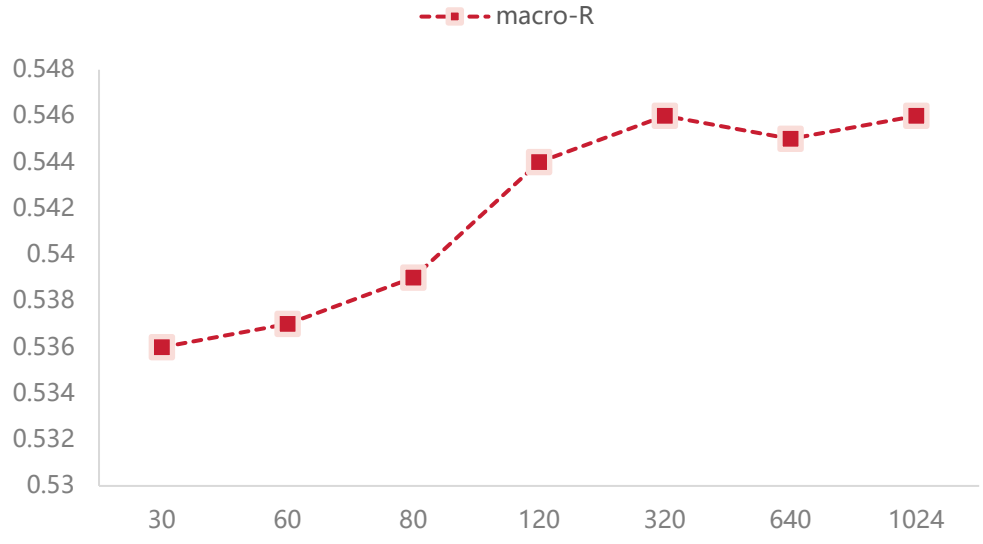
(b)micro-R



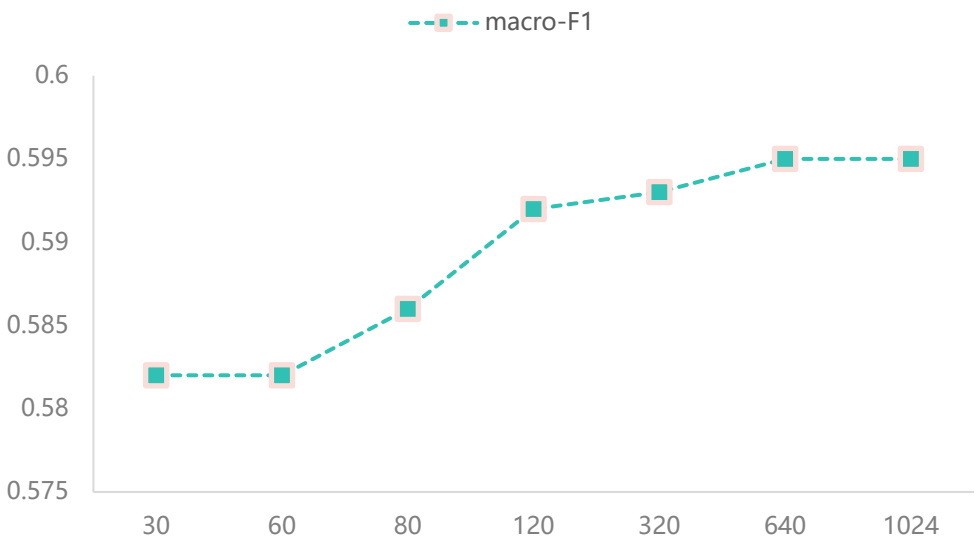
(c)micro-F1



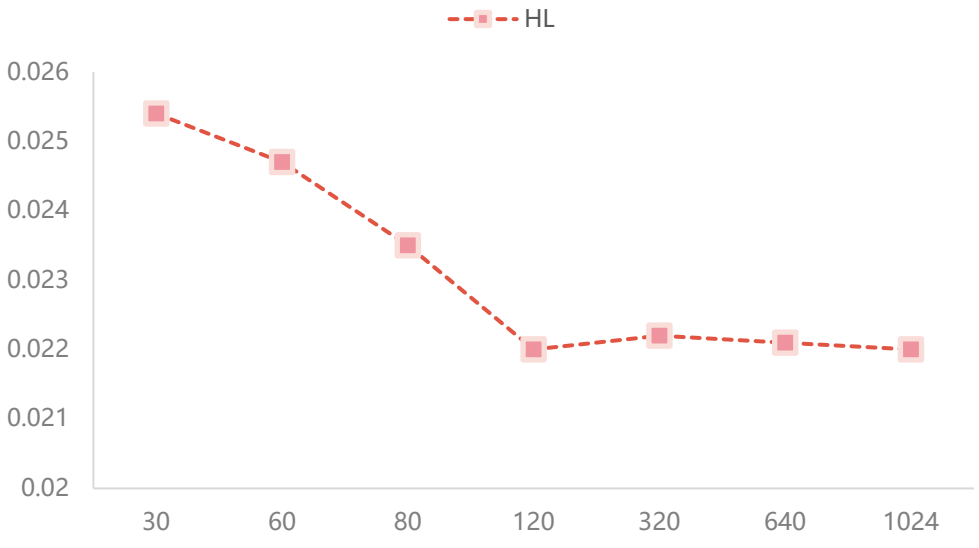
(d)macro-P



(e)macro-R



(f)macro-F1



(g)HL

Fig. 2: The metrics with different input sequence lengths on the AAPD dataset

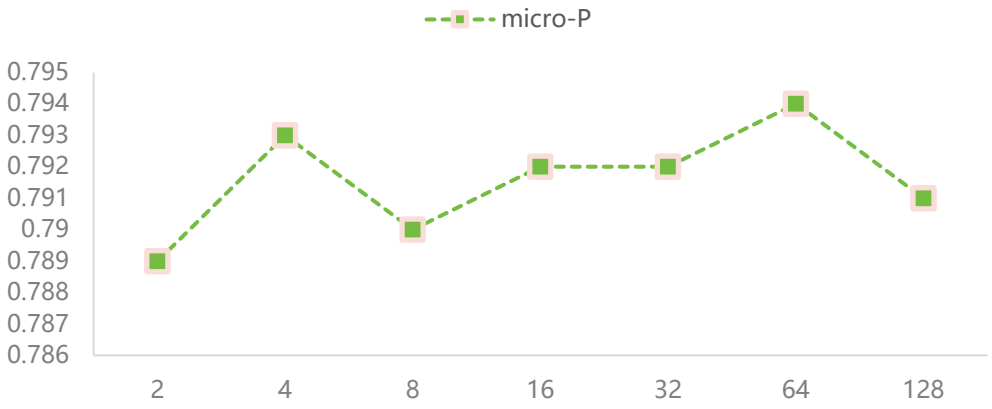
5.6.2 Impact of batch size

The size of the batch affects the model's computational efficiency, gradient estimation stability, convergence speed, and generalization ability. To identify the most suitable batch size for our model, we conduct experiments with different batch sizes. As shown in Table 5, smaller batches introduce more noise in gradient estimation, leading to less stable training and higher variance in performance. Medium batch provides a better balance between noise and stability, resulting in improved performance. The model achieves a balance between stability and computational efficiency with a batch size of 64, leading to the best performance in terms of micro-F1 score and Hamming loss. Consequently, we can infer that this optimal batch size enables the model to more effectively capture the comprehensive features of the document set, along with the inter-document relationships and contextual information. In addition, a larger batch causes the document features captured in the document representation enhancement module to be too smooth and unable to capture the subtle differences between documents. Fig.3 is included to more effectively illustrate the variations in each performance metric across different batch sizes.

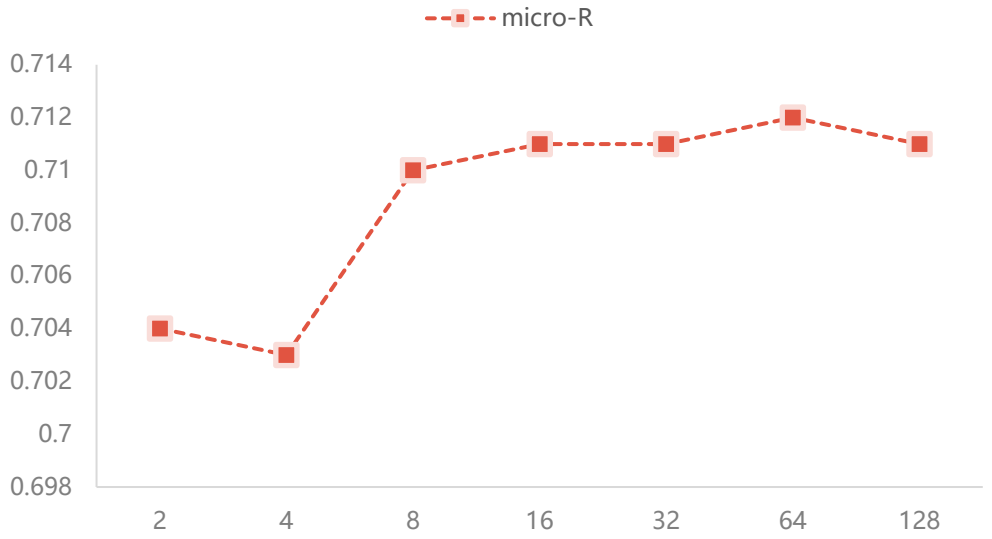
Table 5: Impact of batch size on AAPD

batch size	micro-P	micro-R	micro-F1	macro-P	macro-R	macro-F1	HL
2	0.789	0.704	0.744	0.701	0.538	0.588	0.0236
4	0.793	0.703	0.745	0.703	0.539	0.59	0.0232
8	0.790	0.71	0.748	0.701	0.539	0.589	0.0229
16	0.792	0.711	0.749	0.703	0.544	0.591	0.0240
32	0.792	0.711	0.749	0.705	**0.546**	**0.593**	0.0222
64	**0.794**	**0.712**	**0.751**	0.704	**0.546**	**0.593**	**0.0221**
128	0.791	0.711	0.749	**0.706**	0.545	**0.593**	0.0222

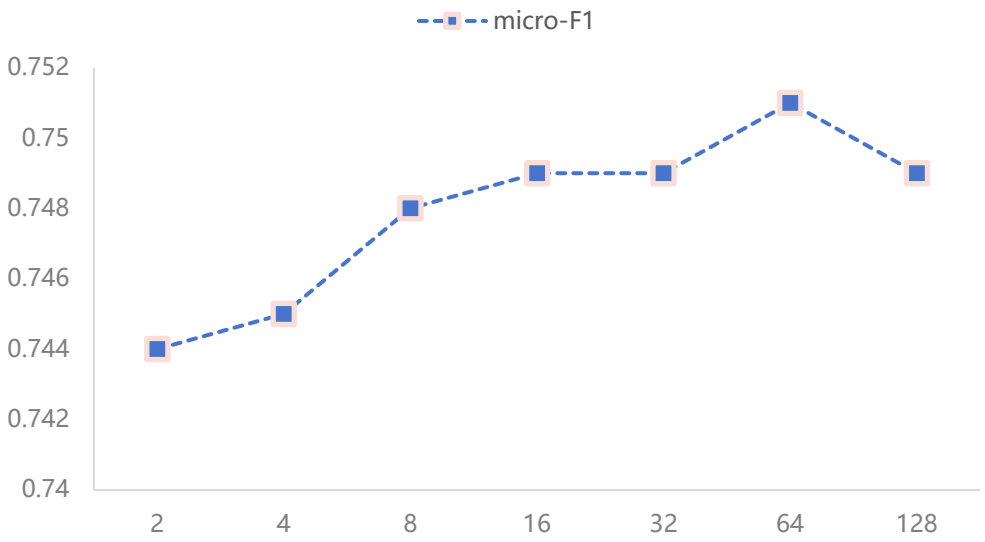
The bold denotes the optimal results



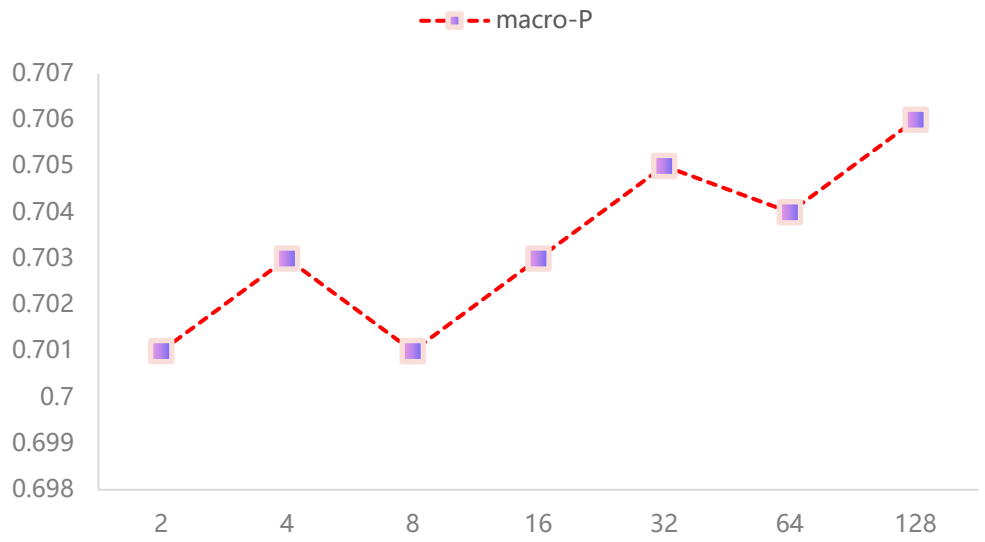
(a)micro-P



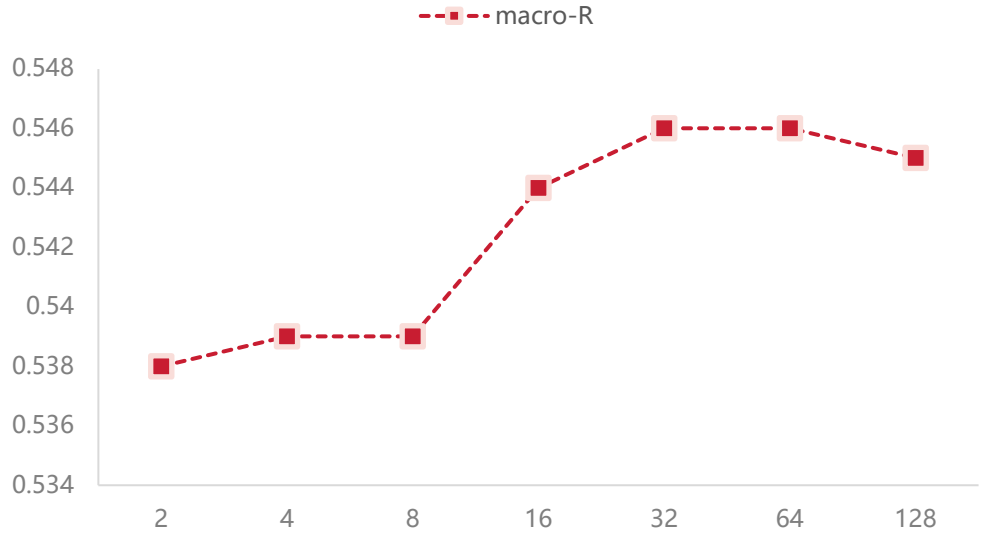
(b)micro-R



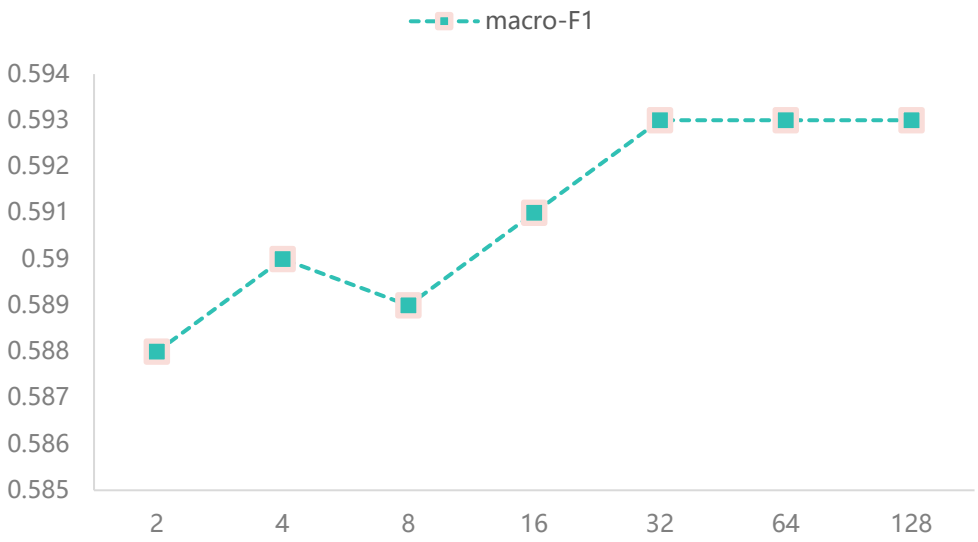
(c)micro-F1



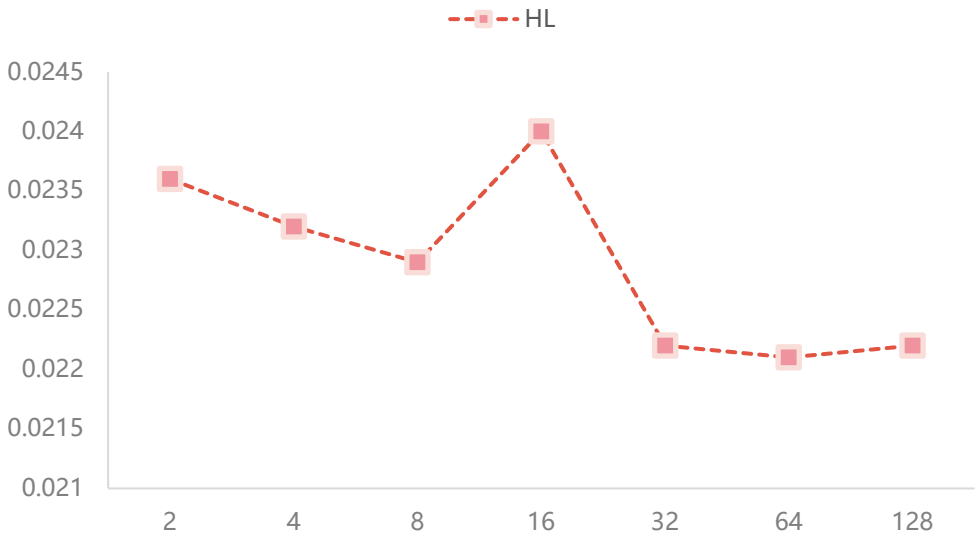
(d)macro-P



(e)macro-R



(f)macro-F1



(g)HL

Fig.3: The metrics with different batch sizes on the AAPD dataset

5.6.3 Impact of subset

Given that the number of document subsets can influence the classification outcomes to some degree during the enhancement of document representations, we conduct experiments to investigate this. Table 6 summarizes the results, with bold indicating the best results. Fig.4 is provided to more clearly demonstrate the changes in each performance metric for different numbers of document subsets.

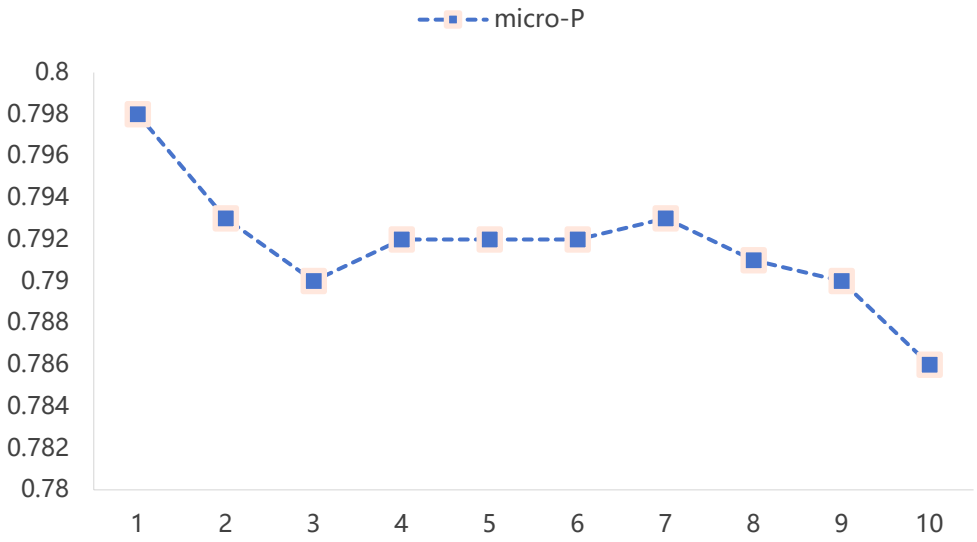
Fig.4 (a) demonstrates that the micro-P value reaches its maximum when there is only one subset and its minimum when there are 10 subsets, and obtains a stable value of 0.792 when the number of subsets is 4 to 6. From Fig.4 (b) and (c), we can see that within the range of 1 to 4 subsets, the value of micro-R and micro_F1 gradually increases, and then performance remains until the number of subsets reaches 6. Subsequently, the performance gradually weakens. According to Fig.4 (d), (e), and (f), the variation patterns of macro-P, macro-R, and macro-F1 are somewhat similar. They increase overall until the number of subsets reaches 6, followed by a gradual decrease. As shown in Fig. 4 (g), the Hamming loss reaches its lowest value when the subset quantity is 6.

Table 6: The outcome on the AAPD with varying subset quantities

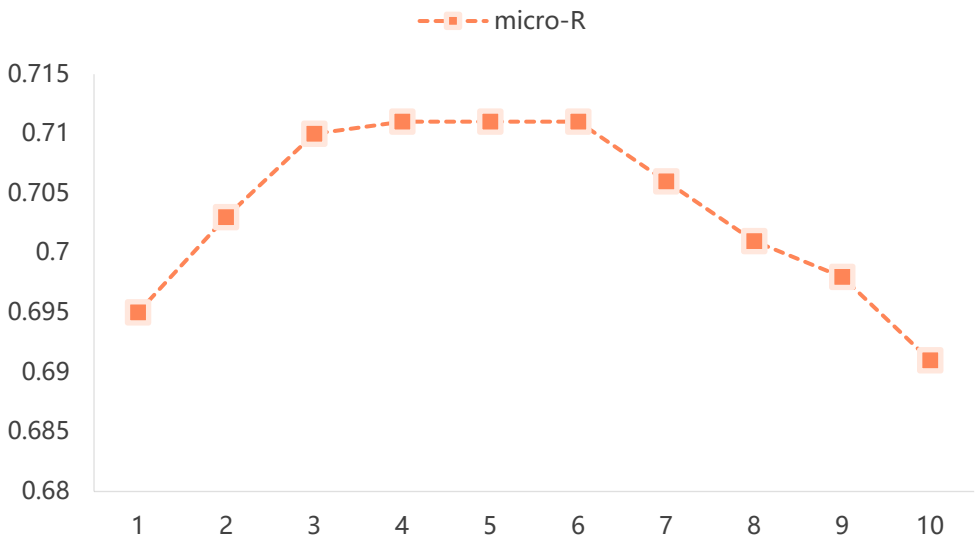
subset quantities	micro-P	micro-R	micro-F1	macro-P	macro-R	macro-F1	HL
1	**0.798**	0.695	0.743	0.697	0.539	0.586	0.0236
2	0.793	0.703	0.745	0.686	0.541	0.583	0.0243
3	0.790	0.710	0.748	0.701	0.544	0.590	0.0233
4	0.792	**0.711**	**0.749**	0.703	0.543	0.591	0.0230
5	0.792	**0.711**	**0.749**	0.704	0.544	0.592	0.0224
6	0.792	**0.711**	**0.749**	**0.705**	**0.546**	**0.593**	**0.0222**
7	0.793	0.706	0.747	0.702	0.540	0.588	0.0232
8	0.791	0.701	0.743	0.692	0.536	0.582	0.0239
9	0.790	0.698	0.741	0.687	0.528	0.575	0.0248
10	0.786	0.691	0.735	0.672	0.521	0.565	0.0254

The optimal outcome is achieved with a subset quantity of 6. With an increase in subset count from 1 to 5 or 6, most metrics display an upward trend, peaking at 5 or 6. However, beyond this point, the performance begins to decline. We attribute this phenomenon to the following factors:

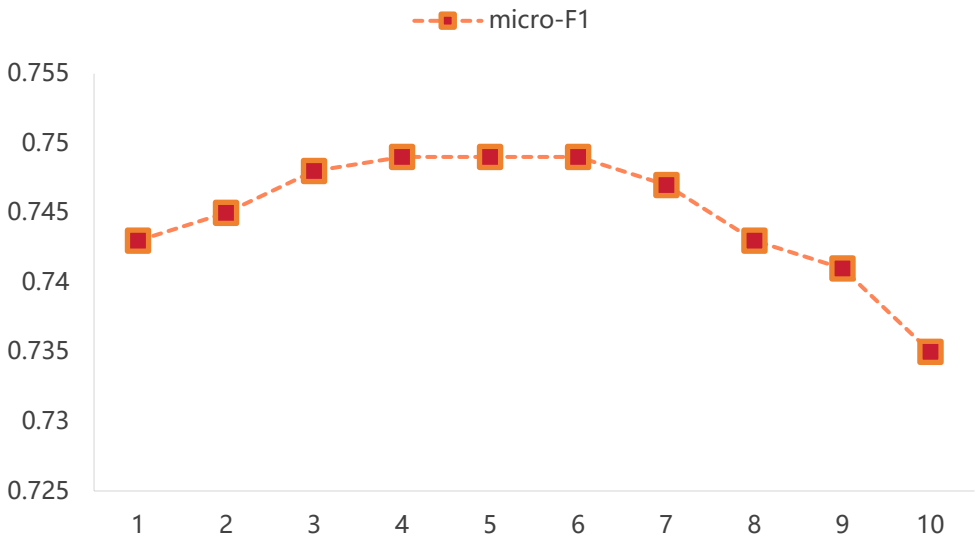
- Generally speaking, by enhancing original documents, the documents can contain more semantic information, thereby improving generalization ability and classification outcome. With more subsets, the model is allowed to capture more detailed and relevant features and similarities within each subset, thereby enhancing document representations. However, once the number of subsets exceeds the limit, the number of documents within each subset decreases, and the avg_pooling results may no longer be sufficiently representative. The feature information becomes sparse, making it difficult to effectively capture the overall characteristics of the documents.
- A greater number of subsets means that during avg_pooling, documents can utilize more contextual information, thereby better capturing the relationships and details between documents. This enables the classification model to make judgments based on more comprehensive features, thus improving classification accuracy. However, too many subsets result in reduced contextual information within each subset, thereby decreasing the quality of the avg_pooling results and affecting the overall performance of the model.



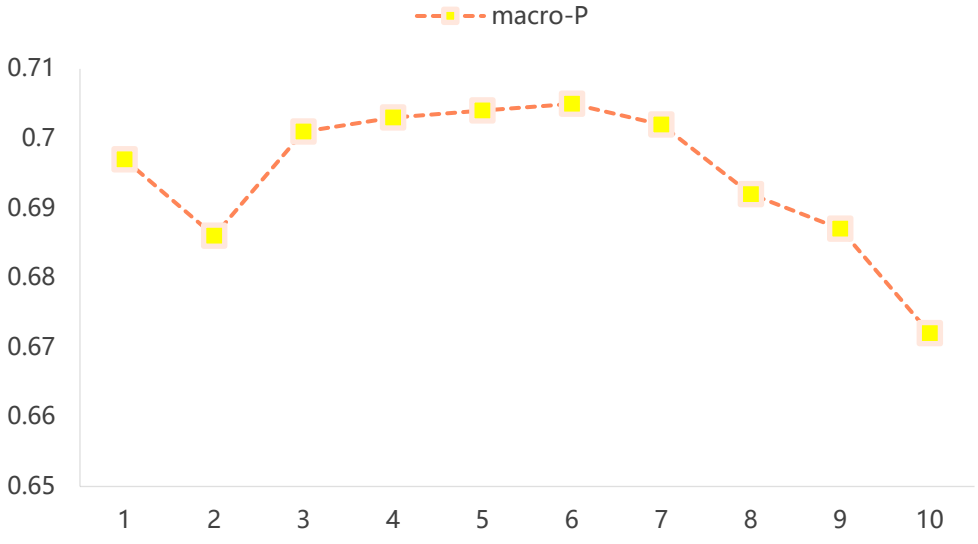
(a)micro-P



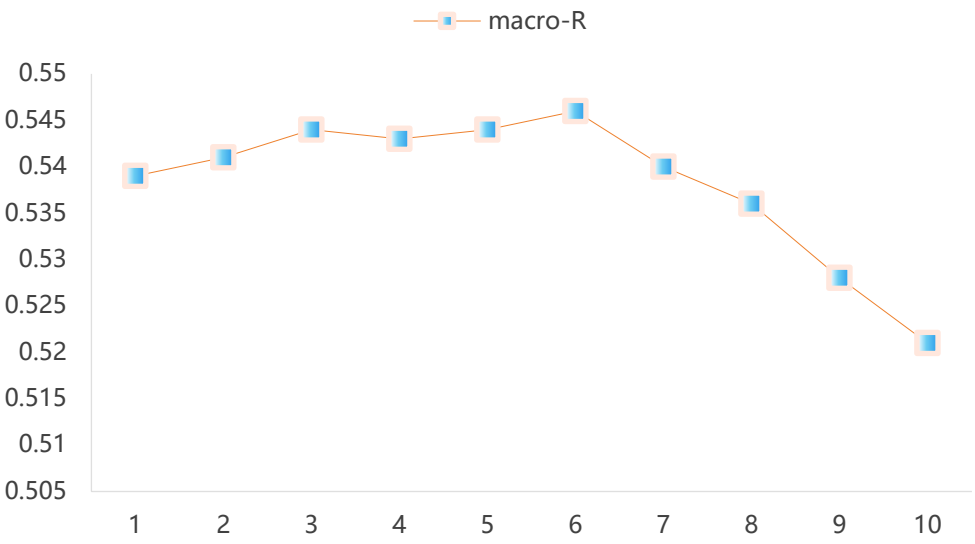
(b)micro-R



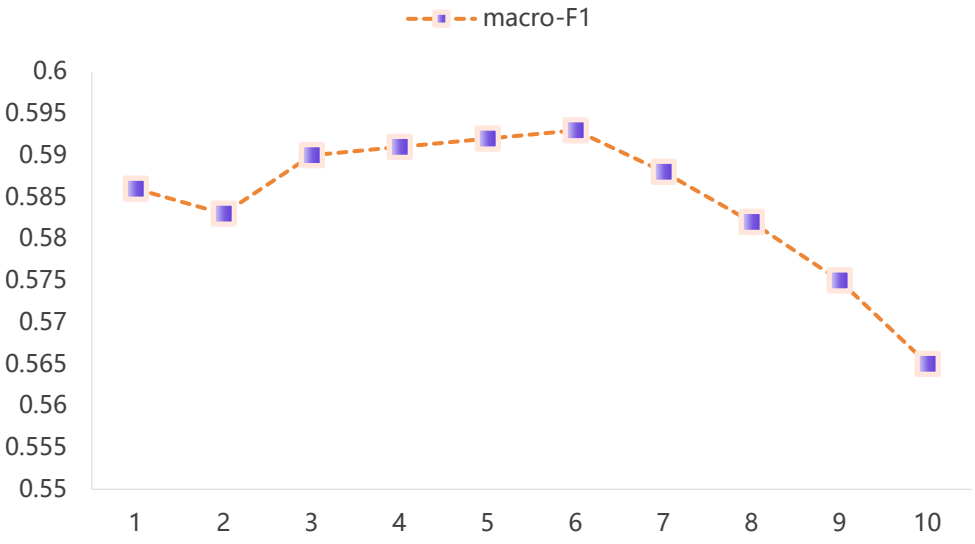
(c)micro-F1



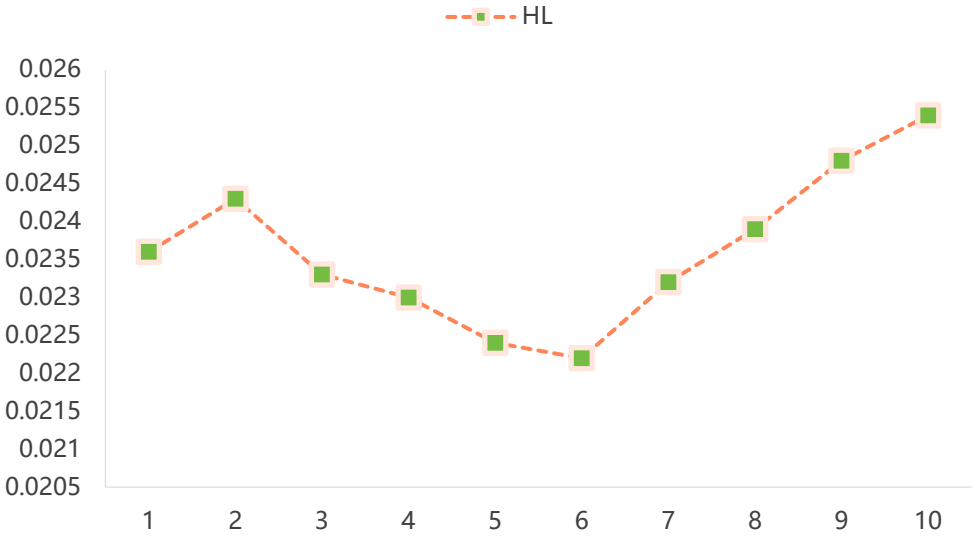
(d)macro-P



(e)macro-R



(f)macro-F1



(g)HL

Fig.4: The metrics with different subsets on the AAPD dataset

6.0 CONCLUSION AND FUTURE WORK

In this study, we present an innovative model that emphasizes improving document representation and learning label correlations in MLTC. We explicitly distinguish and define label similarity correlations and label pairing correlations. To enhance document representations, each document is concatenated with the context-hidden vector of the most similar document subsets. Our model shows performance that is on par with or surpasses the previous competitive models. Thorough analyses reveal the success of our model in integrating the most similar document subsets, which aids in obtaining a more discriminative text representation. Additionally, explicitly capturing the similarity and pairing relationship between labels can significantly improve classification performance. In the future, we will investigate applying this method in diverse domains to validate its generalizability and robustness. Furthermore, we will explore integrating multi-modal data (such as images, audio, etc.) into multi-label classification models to improve document representation and label correlation learning.

REFERENCES

[1] R. Song *et al.*, "Label prompt for multi-label text classification," *Applied Intelligence*, vol. 53, no. 8, pp. 8761–8775, Apr. 2023, doi: 10.1007/s10489-022-03896-4.

[2] A. Pal, M. Selvakumar, and M. Sankarasubbu, "Magnet: Multi-label text classification using attention-based graph neural network," *arXiv preprint arXiv:2003.11644*, 2020, doi: 10.5220/0008940304940505.

[3] A. Sharma and S. Kumar, "Machine learning and ontology-based novel semantic document indexing for information retrieval," *Comput Ind Eng*, vol. 176, p. 108940, Feb. 2023, doi: 10.1016/j.cie.2022.108940.

[4] I. Ameer, N. Bölücü, M. H. F. Siddiqui, B. Can, G. Sidorov, and A. Gelbukh, "Multi-label emotion classification in texts using transfer learning," *Expert Syst Appl*, vol. 213, p. 118534, Mar. 2023, doi: 10.1016/j.eswa.2022.118534.

[5] Y. Jiang and Y. Wang, "Topic-aware hierarchical multi-attention network for text classification," *International Journal of Machine Learning and Cybernetics*, vol. 14, no. 5, pp. 1863–1875, May 2023, doi: 10.1007/s13042-022-01734-0.

[6] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, "SGM: Sequence Generation Model for Multi-label Classification," *arXiv preprint arXiv:1806.04822*, Jun. 2018. http://arxiv.org/abs/1806.04822

[7] W. Liu, H. Wang, X. Shen, and I. W. Tsang, "The Emerging Trends of Multi-Label Learning," *IEEE Trans Pattern Anal Mach Intell*, vol. 44, no. 11, pp. 7955–7974, Nov. 2021, doi: 10.1109/TPAMI.2021.3119334.

[8] X. Zhang, Q.-W. Zhang, Z. Yan, R. Liu, and Y. Cao, "Enhancing Label Correlation Feedback in Multi-Label Text Classification via Multi-Task Learning," *arXiv preprint arXiv:2106.03103*, Jun. 2021. http://arxiv.org/abs/2106.03103

[9] P. Yang, F. Luo, S. Ma, J. Lin, and X. 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*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 5252–5258, doi: 10.18653/v1/P19-1518.

[10] K. Qin, C. Li, V. Pavlu, and J. A. Aslam, "Adapting RNN Sequence Prediction Model to Multi-label Set Prediction," *arXiv preprint arXiv:1904.05829*, Apr. 2019, http://arxiv.org/abs/1904.05829

[11] L. Yang, X.-Z. Wu, Y. Jiang, and Z.-H. Zhou, "Multi-Label Learning with Deep Forest," in *ECAI 2020*, Nov. 2020, pp. 1634–1641. http://arxiv.org/abs/1911.06557

[12] R. Wang, R. Ridley, X. Su, W. Qu, and X. Dai, "A novel reasoning mechanism for multi-label text classification," *Inf Process Manag*, vol. 58, no. 2, Mar. 2021, doi: 10.1016/j.ipm.2020.102441.

[13] H. Liu, G. Chen, P. Li, P. Zhao, and X. Wu, "Multi-label text classification via joint learning from label embedding and label correlation," *Neurocomputing*, vol. 460, pp. 385–398, Oct. 2021. doi: 10.1016/j.neucom.2021.07.031.

[14] C.-P. Tsai and H.-Y. Lee, "Order-free Learning Alleviating Exposure Bias in Multi-label Classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Sep. 2020, pp. 6038–6045. http://arxiv.org/abs/1909.03434

[15] W. Zhao, H. Gao, S. Chen, and N. Wang, "Generative Multi-Task Learning for Text Classification," *IEEE Access*, vol. 8, pp. 86380–86387, 2020, doi: 10.1109/ACCESS.2020.2991337.

[16] D. Xinkai, H. Quanjie, S. Yalin, L. Chao, and S. Maosong, "Label Dependencies-aware Set Prediction Networks for Multi-label Text Classification," *arXiv preprint arXiv:2304.07022*, Apr. 2023. http://arxiv.org/abs/2304.07022

[17] Q. Ma, C. Yuan, W. Zhou, and S. Hu, "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)*, pp. 3855–3864.

[18] W. Peng, Y. Hu, L. Xing, Y. Xie, Y. Sun, and Y. Li, "Control Globally, Understand Locally: A Global-to-Local Hierarchical Graph Network for Emotional Support Conversation," *arXiv preprint arXiv*, no. 2204, Apr. 2022. http://arxiv.org/abs/2204.12749

[19] L. Wang, Y. Feng, and Y. Hong, "An aspect-centralized graph convolutional network for aspect-based sentiment classification," N. 2021 Natural Language Processing and Chinese Computing: 10th CCF International Conference, Ed., Qingdao, China, Oct. 2021, pp. 260–271. http://www.springer.com/series/1244

[20] Y. Zhu, X. Zhou, J. Qiang, Y. Li, Y. Yuan, and X. Wu, "Prompt-Learning for Short Text Classification," *IEEE Trans Knowl Data Eng*, Feb. 2023, http://arxiv.org/abs/2202.11345

[21] L. Wei, Y. Li, Y. Zhu, B. Li, and L. Zhang, "Prompt Tuning for Multi-Label Text Classification: How to Link Exercises to Knowledge Concepts?," *Applied Sciences (Switzerland)*, vol. 12, no. 20, Oct. 2022, 10363. doi: 10.3390/app122010363.

[22] Y. Xiong, Y. Feng, H. Wu, H. Kamigaito, and M. Okumura, "Fusing Label Embedding into BERT: An Efficient Improvement for Text Classification," in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 1743–1750, doi: 10.18653/v1/2021.findings-acl.152.

[23] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[24] H. Ennajari, N. Bouguila, and J. Bentahar, "Knowledge-enhanced Spherical Representation Learning for Text Classification," in *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM)*, 2022, pp. 639–647.

[25] G. Salton, A. Wong, and C. S. Yang, "A Vector Space Model for Automatic Indexing," *Commun ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[26] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv preprint arXiv:1301.3781*, Jan. 2013. http://arxiv.org/abs/1301.3781

[28] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, Oct. 2018. http://arxiv.org/abs/1810.04805

[29] R. A. Sinoara, J. Camacho-Collados, R. G. Rossi, R. Navigli, and S. O. Rezende, "Knowledge-enhanced document embeddings for text classification," *Knowl Based Syst*, vol. 163, pp. 955–971, Jan. 2019, doi: 10.1016/j.knosys.2018.10.026.

[30] D. Peng, J. Yang, and J. Lu, "Similar case matching with explicit knowledge-enhanced text representation," *Applied Soft Computing Journal*, vol. 95, Oct. 2020, doi: 10.1016/j.asoc.2020.106514.

[31] I. Chalkidis and Y. Kementchedjhieva, "Retrieval-augmented Multi-label Text Classification," *arXiv preprint arXiv:2305.13058*, May 2023. http://arxiv.org/abs/2305.13058

[32] J. Mutinda, W. Mwangi, and G. Okeyo, "Sentiment Analysis of Text Reviews Using Lexicon-Enhanced Bert Embedding (LeBERT) Model with Convolutional Neural Network," *Applied Sciences (Switzerland)*, vol. 13, no. 3, Feb. 2023, doi: 10.3390/app13031445.

[33] J. Xiong, L. Yu, X. Niu, and Y. Leng, "XRR: Extreme multi-label text classification with candidate retrieving and deep ranking," *Inf Sci (N Y)*, vol. 622, pp. 115–132, Apr. 2023, doi: 10.1016/j.ins.2022.11.158.

[34] S. Nazmi, X. Yan, A. Homaifar, and E. Doucette, "Evolving Multi-label Classification Rules by Exploiting High-order Label Correlation," *Neurocomputing*, vol. 417, pp. 176–186, 2020. https://www.elsevier.com/open-access/userlicense/1.0/

[35] A. Clare and R. D. King, "Knowledge Discovery in Multi-label Phenotype Data," in *European conference on principles of data mining and knowledge discovery*, 2001, pp. 42–53.

[36] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[37] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," *Adv Neural Inf Process Syst*, vol. 14, 2001.

[38] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Mach Learn*, vol. 73, no. 2, pp. 133–153, 2008, doi: 10.1007/s10994-008-5064-8.

[39] J. Huang, G. Li, S. Wang, Z. Xue, and Q. Huang, "Multi-label classification by exploiting local positive and negative pairwise label correlation," *Neurocomputing*, vol. 257, pp. 164–174, Sep. 2017, doi: 10.1016/j.neucom.2016.12.073.

[40] W. Liao, Y. Wang, Y. Yin, X. Zhang, and P. Ma, "Improved sequence generation model for multi-label classification via CNN and initialized fully connection," *Neurocomputing*, vol. 382, pp. 188–195, Mar. 2020, doi: 10.1016/j.neucom.2019.11.074.

[41] L. Wang, R. Chen, and L. Li, "Knowledge-Guided Prompt Learning for Few-Shot Text Classification," *Electronics (Switzerland)*, vol. 12, no. 6, Mar. 2023, doi: 10.3390/electronics12061486.

[42] H. T. Vu, M. T. Nguyen, V. C. Nguyen, M. H. Pham, V. Q. Nguyen, and V. H. Nguyen, "Label-representative graph convolutional network for multi-label text classification," *Applied Intelligence*, vol. 53, no. 12, pp. 14759–14774, Jun. 2023, doi: 10.1007/s10489-022-04106-x.
[43] D. D. Lewis, Y. Yang, T. G. Rose, F. Li, and F. Li LEWIS, "RCV1: A New Benchmark Collection for Text Categorization Research," *Journal of Machine Learning Research*, vol. 5, pp. 361–397, 2004.
[44] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit*, vol. 37, no. 9, pp. 1757–1771, Sep. 2004, doi: 10.1016/j.patcog.2004.03.009.
[45] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier Chains for Multi-label Classification," *Mach Learn*, vol. 85, pp. 333–359, 2011, doi: 10.1007/978-3-642-04174-7_17.
[46] C. Fan, W. Chen, J. Tian, Y. Li, H. He, and Y. Jin, "Accurate Use of Label Dependency in Multi-Label Text Classification Through the Lens of Causality," *Applied Intelligence*, vol. 53, no. 19, pp. 21841–21857, Oct. 2023. http://arxiv.org/abs/2310.07588