# BeNet: BERT Doc-Label Attention Network for Multi-Label Text Classification

**Bo Li**

Baidu Inc., Beijing, 100193, China

libo15@baidu.com

**Abstract.** Multi-label Text Classification (MLTC) holds significant importance and serves as a foundational aspect in Natural Language Processing (NLP), which aims at assigning multiple labels for a given document. Many real-world tasks can be viewed as MLTC, such as tag recommendation, information retrieval, etc. Nevertheless, researchers are faced with numerous challenging issues regarding the establishment of linkages between labels or the differentiation of comparable sub-labels. To address this issue, we provide a novel approach known as the **BE**RT Doc-Label Attention **Net**work (**BeNet**) in this paper, which consist of the BERTdoc layer, the label embeddings layer, the doc encoder layer, the doc-label attention layer and the prediction layer. We apply the powerful technique of BERT to pretrain documents to capture their deep semantic features and encode them via Bi-LSTM to obtain a two-directional contextual representation of uniform length. Then we create label embeddings and feed them together with encoded-pretrained-documents to the doc-label attention mechanism to obtain interactive information between documents and their corresponding labels, finally using MLP to make predictions. We carry out experiments on two real-world datasets, and the empirical results demonstrate that our proposed model outperforms all state-of-the-art MLTC benchmarks. Furthermore, we have undertaken a case study to effectively illustrate the practical implementation of our method.

**Keywords:** Multi-Label Text Classification, Natural Language Processing, BERT Doc-Label Attention Network, Label embedding.

## 1. Introduction

Text classification is a basic data mining task in Natural Language Processing (NLP), including multi-class text classification and multi-label text classification. Multi-class classification only assigns one label to a given document with over two labels in the whole document, while multi-label text classification divides a document into different topics at the same time. Multi-label text classification is a more flinty issue in text classification because it allows multiple labels to exist in a single document with each label representing an aspect of the document content. Therefore, the overall semantic information of the entire document is composed of multiple or hierarchical components. Table 1 exemplifies the instances: the sentence “Young boys are playing football” can be categorized as topic “Youth” and “Sports”, while a news report such as “The cultural industry will become the pillar industry of the national economy in 2023” belongs to either “Economy” or “Culture” as well as the movie “Twilight City” which is classified as a romance movie and a fantastic magic movie.

MLTC aims at exploring multiple best-matched document label pairs according to a specific document and its several corresponding labels, which has many practical scenarios, such as tag recommendation [1] , information retrieval[2] , etc. For example, it always appears on the homepage of news websites, social platforms such as Weibo and Twitter, introductions and reviews of books or movies, and online shopping malls such as Taobao and Jingdong. It principally devotes itself to reducing hunting zones progressively, facilitating humans to select their required information precisely and improving the quality of automatic recommendations in the background, so as to provide a fast retrieval for users to efficiently search for target information while filtering out redundant and irrelevant counterparts.

**Table 1.** The example of Multi-Label Text Classification

| **Text** | **Label** |
|---|---|
| **Sentence**: *Young boys are playing football* | *Youth,Sports* |
| **News**: *The cultural industry will become the pillar industry of the national economy in 2023* | *Economy,Culture* |
| **Movie**: *Twilight City* | *Romance, Magic* |

However, enormous difficulties impede our progress in solving the MLTC task accurately. Several difficult problems of MLTC can be summarized as follows: i) The number of labels for a given text is unknown, because some samples may have only one label while others may belong to dozens or even hundreds of topics; ii) The content of some documents is not rich enough to accurately predict the labels, because these documents belong to three or more labels.

MLTC methods can be broadly classified into two primary categories: traditional multi-label classification algorithms and deep learning-based algorithms. Traditional machine learning algorithms contains BR[3] , ML-DT[4] , Rank-SVM[5] , LP[6] , ML-KNN[7] and CC[8] , difficult to solve high-level label correlation. Deep neural networks such as CNN-based or RNN-based methods[9] also fail to capture high-order dependency between labels or distinguish similar sub-labels. Seq2Seq[10] model is a milestone in MLTC but relying on strict label orders limits its performance, and with the pre-trained model such as BERT[11] , significant improvements in classification performance have been achieved. Although there has been a large amount of research on MLTC, the results are still not very satisfying.

To address the challenges mentioned above, we propose a novel BERT Doc-Label Attention Network (***BeNet***) for MLTC, consisting of the BERTdoc layer, the label embeddings layer, the doc encoder layer, the doc-label attention layer and the prediction layer. We do pre-training via BERT to fully capture semantic information in documents and use GloVe[12] to represent labels as embedding vectors. Then, we convert documents with BiLSTM to obtain a two-directional contextual representation of uniform length. Afterwards, we apply the doc-label attention mechanism to extract interactive information between documents and their corresponding labels, which is then fed into a MLP classifier to do final prediction.

The main contributions of this work can be summarized as follows:

(1) We adopt BERT to provide pre-training for documents to fully capture their semantic features. With token embedding, segment embedding and position embedding, we obtain detailed information of each word as well as the sequential relationship of words and sentences respectively in documents and then feed them into a transformer containing multi-head self-attention, a dense layer and intermediate layer to aggregate scattered features which manage to extract deeper information.

(2) For the reason that a single document belongs to several labels, it's a necessity to establish connection between documents and their corresponding labels. Therefore, we apply doc-label attention mechanism to obtain interactive information between encoded pre-trained document representation and label embeddings.

(3) We carry out experiments on different types of datasets, with the results indicating our proposed model outperforms all state-of-the-art MLTC models. Additionally, a case study is undertaken to illustrate the practical implementation of our method.

## 2. Related work

The models used to solve the multi-label text classification task can be divided into three categories: problem transformation methods, algorithm adaptation methods and neural network models. Problem transformation methods convert the MLTC task into multiple single-label text classification tasks, such as BR[3] ignoring label dependencies and building a separate classifier for each label, LP creating a binary classifier for each label combination, and CC[8] converting the MLTC task into a binary classification problem chain.

Algorithm adaptive methods aim at modifying specific algorithms to solve MLTC, including local methods and global methods. Local methods such as ML-DT[4] which constructs a decision tree based on multi-label entropy without considering hierarchical structure information, RankSVM [5] which uses SVM similar to a learning system, ML-KNN[7] which applies the k-nearest neighbor algorithm and the maximum posterior probability to determine the label set of each sample, CBM [13] which simplifies the task by transforming it into multiple binary problems. Global methods such as Clus-HMC [14] that uses a single decision tree to process the entire hierarchical category structure, HMCLMLP [15] that trains a set of neural networks with each neural network predicting a given level of categories, CML [16] which aims at encoding label correlation as constraint conditions based on the principle of maximum entropy, joint learning algorithm [17] that allows labels to make back propagation from the next classifiers to the current counterpart. However, the above-mentioned work mainly focuses on the local or global structure to capture low-order label correlation, ignoring the hierarchical dependencies between different levels of labels, facing thorn difficulties when computing higher-order label correlation.

In recent years, neural networks have made significant improvement in MLTC. For example, BPMLL [9] applies a fully connected network and pairwise ranking loss to perform classification. Nam et al. [18] further replaced pairwise ranking loss with a cross-entropy loss function. Kurata, Xiang and Zhou [19] proposed an initialization method, using neurons to model label correlation. Chen et al. [20] proposed a joint approach combined with CNN and RNN to capture local and global semantic information. Bahdanau et al. [21] proposed a method to train a neural network to generate sequences using the actor-critic method. Besides, as the significant appearance of Seq2Seq model, more endeavor has been done in MLTC based on Seq2Seq structure, such as SGM [22], MDC [23], HBLA [24] and R-Transformer_BiLSTM [25].

In addition, some label correlation detection methods show prominent performance to capture relationship between labels. DXML [26] establishes a clear label co-occurrence map to explore label embeddings in lowdimensional space. EXAM [27] introduces an interactive mechanism to incorporate word-level matching signals into text classification tasks. GILE [28] proposes a joint input-label embedding model for neural text classification. However, the above-mentioned label correlation detection methods fail to perform well when the semantic information of labels is highly similar.

## 3. Method

### 3.1. Task description

The MLTC task in this research can be summarized as a tuple set $S = \{(d_i , l_i )\}_{i=1}^{N}$ with $d_i$ and $l_i$ represents the $i$-th document denoted as $D = \{d_i|d_i =\{d^1, d^2, \cdots , d^n\}$ and its corresponding label sets denoted as $L = \{l_i|l_i =\{l^1, l^2, \cdots , l^m\}$. $N$, $n$ and $m$ are the total number of documents, the length of the $i$-th document and the number of labels of the $i$-th document, respectively. Our proposed BeNet model aims at assigning all suitable labels to its corresponding documents based on the conditional probability $Pr(l_i|d_i)$ to solve the MLTC task.

### 3.2. Overview of proposed model

Our proposed BeNet model consists of five layers, i.e, the BERTdoc layer, the label embeddings layer, the doc encoder layer, the doc-label attention layer and the prediction layer shown in Figure 1. The BERTdoc layer refers to pre-train documents via BERT to extract their semantic features while the label embeddings layer means map each label to a high-dimensional space with GloVe[12] . The doc encoder layer denotes encoding each pre-trained word in documents via Bi-LSTM to obtain text representation forward and backward of uniform length. The doc-label attention layer means an interactive strategy capturing mutual features of encoded pre-trained document representation and label embeddings, which then feed into prediction layer (MLP) to complete final multi-label classification. The overall proposed model is trained end-to-end.

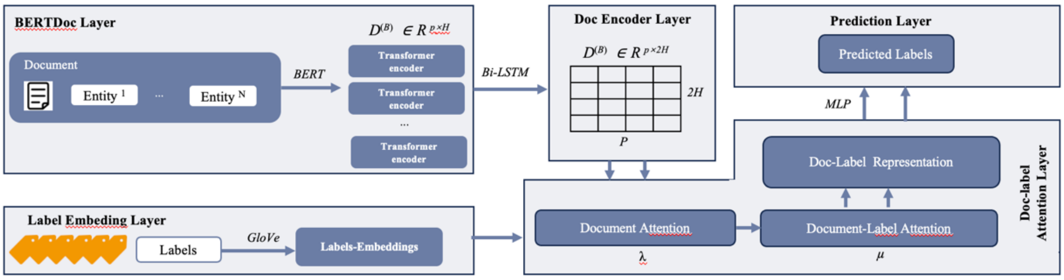


**Figure 1.** Architecture of our proposed BeNet model

### *3.3. BERTdoc layer*

In this layer, we use base-BERT with 12 transformer blocks, 768 dimension of hidden state, 12 head per layer of multihead attention and 110M parameters to pre-train documents to capture their deep information. We preprocess documents as BERT input representation, which are the sum of the token embeddings aiming at different words, the segmentation embeddings distinguishing each sentence in a paragraph and the position embeddings outputing position of words, then pass them to transformers mechanism in BERT. Each embedding of BERT input representation is differentiated via slicing which then fed into BERT model to output pre-trained contextual representation of documents. The process of document pre-training can be elaborated as follows:

$$\text{token, seg, pos} = BERT_{\text{encoder}}\left(D, BERT_{\text{tokenizer}}\right) \tag{1}$$

$$D^{(B)} = BERT_{\text{model}}\left(\text{token}, \text{seg}, \text{pos}\right) \tag{2}$$

### *3.4. Doc encoder layer*

To obtain forward and backward contextual representations of given documents, we adopt bidirectional LSTM (Bi-LSTM) to encode pre-trained documents as $2H$-dimensional vectors. Through the encoder layer, we also unify the length of documents to get encoded pre-trained representation $D^{(B)} \in R^{p\times 2H}$ where $p$ means the maximum length of each input document of BERT. The hidden state $h_t \in R^H$ is randomized. The specific equations are shown as follows:

$$\overrightarrow{D_t^{(B)}} = LSTM\left(\overrightarrow{D_{t-1}^{(B)}}, h_t\right) \tag{3}$$

$$\overleftarrow{D_t^{(B)}} = LSTM\left(\overleftarrow{D_{t-1}^{(B)}}, h_t\right) \tag{4}$$

$$D_t^{(B)} = \left[\overrightarrow{D_t^{(B)}}; \overleftarrow{D_t^{(B)}}\right] \tag{5}$$

$$D^{(B)} = \left\{D_t^{(B)}\right\}_{t=1}^{T} \tag{6}$$

### 3.5. Label embeddings layer

For the reason that each label contains latent semantic information besides documents, we convert labels $L = \{l_i|l_i = \{l^1, l^2, \cdots, l^m\}$ to embedding vectors $L^{(G)} \in R^{M\times d}$ via GloVe[12] with M representing the total number of labels , fully establishing contextual relationship among labels.

$$L^{(G)} = Embedding_{label}(L) \in R^{M\times d} \tag{7}$$

### 3.6. Doc-label attention layer

In the MLTC task, a single document belongs to several labels and vice versa, so it's intuitive and vital to capture interactive features between documents and their corresponding labels. Therefore, we adopt a doc-label attention mechanism to fuse information between documents and labels. The details can be described as follows:

Firstly, we apply self-attention mechanism on documents to obtain an independent weight vector λ which implies contribution of documents in doc-label pairs:

$$A_D = \text{softmax}\left(W_1'\tanh\left(W_1 D^{(B)^T}\right)\right) \tag{8}$$

$$\lambda = \sigma\left(\left(A_D D^{(B)}\right)W_1''\right) \tag{9}$$

Then we apply doc-label attention mechanism to get attention label representation $L^{(A)}$ and its independent weight vector $\mu$:

$$A_L = \left(W_2 L^{(G)}\right)\left(W_2' D^{(B)^T}\right) \tag{10}$$

$$L^{(A)} = A_L D^{(B)} \tag{11}$$

$$\mu = \sigma\left(L^{(A)} W_2''\right) \tag{12}$$

The final doc-label representation $S^{(A)}$ is calculated by multiplying dependent label weight vector $\mu_{dep}$ via normalization:

$$\mu_{dep} = \frac{\mu}{\mu + \lambda} \tag{13}$$

$$S^{(A)} = \mu_{dep} L^{(A)} \tag{14}$$

Here, $W_1$ , $W_1'$, $W_1''$, $W_2$ , $W_2'$, $W_2''$ are trainable parameters. $\sigma$ is sigmoid activation function (the same below).

### 3.7. Prediction layer

Finally, a MLP classifier in the prediction layer is used for the final doc-label representation $S^{(A)}$ to make multi-label text classification:

$$\hat{y} = \sigma\left(W_p'\tanh\left(W_p S^{(A)}\right)\right) \tag{15}$$

where $W_p$ , $W_p'$ are trainable parameters.

We adopt cross-entropy loss as the loss function in our work which has been proved suitable for the MLTC task:

$$\min_{\Theta} \sum_{i=1}^{N} \sum_{j=1}^{M} y^{(ij)} \log\left(\sigma\left(\hat{y}^{(ij)}\right)\right) + \left(1 - y^{(ij)}\right) \log\left(1 - \sigma\left(\hat{y}^{(ij)}\right)\right) \tag{16}$$

where $y^{(ij)} \in \{0, 1\}$ denotes the $i$-th ground truth label of the $i$-th document while $y^{\hat{}(ij)} \in [0, 1]$ indicates the predicted probability of the above-mentioned doc-label pairs.

## 4. Experiments setup

### *4.1. Datasets*

In this research, we utilize two multi-label text datasets with the detailed statistics shown in Table 2. Specifically, $W$, $N_{train}$, $N_{test}$ and $M$ denote the number of total words, training documents, test documents and total unique labels, respectively.

**RCV1-V2 [29]** contains 804,414 newswire stories, including 643,531 training documents and 160,883 test ones. Each story belongs to several topics with the total number of labels 103.

**AAPD [22]** is a combination of 55,840 ab stracts and their corresponding topics in the field of computer science from Arxiv in 2018, which consists of 54,840 abstracts as training data and 1,000 ones as test data.

**Table 2.** Statistics of two datasets

| Dataset | $W$ | $N_{train}$ | $N_{test}$ | $M$ |
|---|---|---|---|---|
| RCV1-V2 | 47,236 | 23,149 | 781,265 | 103 |
| AAPD | 69,399 | 54,840 | 1,000 | 54 |

### *4.2. Baseline*

We compare our proposed model with the following nine benchmarks:

**BR [3]** establishes multiple binary classifiers for each label, ignoring dependency between labels.

**LP [6]** creats a multi-class classifier for all unique label combinations.

**CC [8]** converts the MLTC task into a chain of binary classification problems with consideration of highorder label correlation.

**S2S [10]** is the pure sequence-to-sequence model which can be used on the MLTC task.

**CNN-RNN [20]** utilizes a combination of CNN and RNN to capture global and local semantic features as well as label correlation.

**HBLA [24]** is a hybrid neural network model to simultaneously take advantage of both label semantics and fine-grained text information

**R-Transformer_BiLSTM [25]** model based on label embedding and attention mechanism for multi-label text classification.

**CNN [30]** adopts multiple convolution kernals to extract contextual information with activation function to ouput probability distribution.

**S2S+Attn [31]** adds attention mechanism on the basis of RNN-oriented Seq2Seq model.

### *4.3. Evaluation metrics*

Inspired by the previous work [7,20], we evaluate our proposed model and other nine benchmarks with Hamming Loss, micro-Precision, micro-Recall and micro-F1.

**Hamming loss (HL)** calculates the percentage of mislabeled documents whose predicted labels are not adequate or irrelevant. The equation can be elaborated as follows:

$$HL = \frac{1}{N}\sum_{i=1}^{N} \frac{XOR\left(y^{(ij)}, \sigma\left(\hat{y}^{(ij)}\right)\right)}{M} \qquad \textbf{(17)}$$

where $N$, $M$, $y^{(ij)}$, $y^{\wedge(ij)}$ have explained before. $XOR$ is exclusive-or logic function with $XOR(0, 1) = XOR(1, 0) = 1$ and $XOR(0, 0) = XOR(1, 1) = 0$.

**micro-Precision (mP)** interprets global precision with True Positives and False Positives of the $i$-th given label, i.e $FP_i$ and $TP$

$$mP = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} TP_i \times \sum_{i=1}^{M} FP_i} \qquad \textbf{(18)}$$

**micro-Recall (mR)** describes global recall with True Positives and False Negatives of the $i$-th given label, i.e., $FP_i$ and $FN_i$

$$mR = \frac{\sum_{i=1}^{M} TP_i}{\sum_{i=1}^{M} TP_i \times \sum_{i=1}^{M} FN_i} \qquad \textbf{(19)}$$

**micro-F1 (mF1)** weights the global precision and recall of the total categories which can be represented as follows:

$$mF1 = \frac{2 \times mP \times mR}{mP + mR} \qquad \textbf{(20)}$$

### *4.4. Hyper parameters and training*

We carry out our experiments on NVIDIA TESLA V100 GPU with Pytorch. In the BERTdoc layer and the label embeddings layer, we set the maximum length of each document as 500 in the pre-training process with BERT and adjusted the embedding size of labels as 300. As for the doc encoder layer, the dimension of hidden state in BiLSTM is set to 300. When it comes to training process, we use Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The batch size is adjusted to 128 and the learning rate is initialized to 0.0001. We evaluate model performance on test sets after 200 epochs with early stopping when the validation loss stops decreasing by 10 epochs.

## 5. Experimental results

### *5.1. Model comparison*

We compare our proposed BeNet model with other nine benchmarks on two datasets evaluated with *HL*, *mP* , *mR*, *mF* 1 shown in Table 3. Moreover, (+) in Table 3 means the higher the value is, the better performance of the model, such as *mP* , *mR* and *mF* 1 while (−) indicates the opposite, such as *HL*.

The nine benchmarks can be divided into three categories referred to as machine learning methods (i.e., BR, CC, LP), conventional deep learning models (i.e., CNN, CNN-RNN) and Seq2Seq-based approaches (i.e., S2S, S2S+Attn, R-Transformer_BiLSTM, HBLA). As shown in Table 3, we can see that generally conventional deep learning methods outperform machine learning models on RCV1-V2 and AAPD, which strongly demonstrates conventional deep learning model are superior in extracting deep semantic information than feature-engineering driven traditional machine learning methods dependent on burdensome handcrafts. Surprisingly, CNN performs best on the above-mentioned two datasets with *mP* possibly due to the function of convolution kernels which exactly manage to capture accurate features but needing validation on more datasets.

**Table 3.** Comparisons of KeNet and fourteen baselines on AAPD, RCV1-V2. (+) means the higher the value is the better performance of the model. (−) indicates the opposite.

| **Datasets** | **AAPD** | | | | **RCV1-V2** | | | |
|---|---|---|---|---|---|---|---|---|
| **Metrics** | **HL(-)** | **mP(+)** | **mR(+)** | **mF1(+)** | **HL(-)** | **mP(+)** | **mR(+)** | **mF1(+)** |
| BR | 0.0316 | 0.664 | 0.648 | 0.646 | 0.0086 | 0.904 | 0.816 | 0.858 |
| CC | 0.0306 | 0.657 | 0.651 | 0.654 | 0.0087 | 0.887 | 0.828 | 0.857 |
| LP | 0.0323 | 0.662 | 0.608 | 0.634 | 0.0087 | 0.896 | 0.824 | 0.858 |
| CNN | 0.0256 | **0.849** | 0.545 | 0.664 | 0.0089 | **0.922** | 0.798 | 0.855 |
| CNN-RNN | 0.0280 | 0.718 | 0.618 | 0.664 | 0.0085 | 0.889 | 0.825 | 0.856 |
| S2S | 0.0255 | 0.743 | 0.646 | 0.691 | 0.0082 | 0.883 | 0.849 | 0.866 |
| S2S+Attn | 0.0261 | 0.720 | 0.639 | 0.677 | 0.0081 | 0.889 | 0.848 | 0.868 |
| R-Transformer_BiLSTM | 0.0240 | 0.762 | 0.689 | 0.718 | 0.0070 | 0.910 | 0.890 | 0.893 |
| HBLA | **0.0223** | 0.768 | **0.722** | **0.744** | **0.0063** | 0.906 | **0.892** | **0.899** |
| **BeNet(ours)** | **0.0236** | **0.822** | **0.674** | **0.741** | **0.0068** | **0.925** | **0.894** | **0.909** |

A milestone of the MLTC task is sequence-to-sequence (Seq2Seq) model, followed by a bundle of Seq2Seq-based models like S2S+Attn, R-Transformer_BiLSTM, HBLA, etc. The average results of Seq2Seq-based models show an advantage over that of conventional deep learning models on RCV1-V2 and AAPD, undoubtedly indicating that Seq2Seq-based models are capable of exploring latent label orders with global embedding which beat conventional deep learning solutions overwhelmingly. Akin to the comparison between conventional deep learning models and machine learning methods, conventional deep learning models perform just plain better than Seq2Seq-based models with *mp* on these two datasets, which needs more corpora for interpretation.

Most importantly, the experiment results show that our model BeNet has the best performance on all two datasets, outperforming the current state-of-the-art model on RV1-V2 and AAPD. Specially, the pre-trained model BERT uses token embedding, segment embedding and position embedding to capture different angle of semantic features and applies transformer with multi-head self-attention, dense layer and intermediate layer to extract deeper and hierarchical contextual information of documents. Encoding of documents with BiLSTM further takes contextual relationship among documents into consideration forward and backward as well as carries out documents-cutting that unifies the length of documents. Furthermore, label embeddings integrate all unique labels in order to capture latent connections between each label-pair, finding semantic similarity between labels, then seeking out the combination of labels corresponding to a certain document. Doc-label attention mechanism is capable of establishing relationship between documents and their corresponding labels to learn interactive information between encoded pre-trained document representation and label embeddings.

### *5.2. Ablation study*

To analyze the contributions of each component of our proposed model, we carry out ablation study of five derived models which remove or change any layer on RV1-V2 shown in Table 4. Because of similar tendency on the other two datasets, we only take results on RV1-V2 as an example.

Specifically, w/o BERTdoc and BERTdoc to EMBdoc represents derived models without pre-training on documents with BERT and applying traditional GloVe technique to establish document embeddings instead of BERT, respectively, both affecting the performance compared with proposed BeNet model by a wide margin by 22.06% in *HL*, 6.77% in *mP*, 8.43% in *mR* and 2.66% in *mF*1 as well as by 11.76% in *HL*, 0.80% in *mP*, 5.75% in *mR* and 3.31% in *mF*1, which indicate the powerful capabilities of BERT in capturing deep semantic information. With multiple embeddings such as token embeddings, segment embeddings and position embeddings as well as transformers containing multihead attention, the pre-training model BERT manages to extract global semantic information of documents undoubtedly. When we remove the doc-label attention layer away from the final model named w/o Doc-label attention, the results also decrease by 26.47% in *HL*, 2.93% in *mP* , 9.54% in *mR* and 6.29% in *mF* 1, demonstrating its function of extracting interactive features between documents and their corresponding labels via establishing contextual connection of the two parts, which also clarifies attention mechanism is able to model long sequences, fully finding semantic interaction of document-label pairs at any distance. W/o Label embeddings means feeding only encoded pre-trained documents to the doc-label attention layer without label embeddings, which also has a negative effect on model performance by 29.41% in *HL*, 5.53% in *mP* , 4.32% in *mR* and 4.92% in *mF* 1, because label embeddings take all unique labels into consideration, establishing relationship among labels which aims at exploring latent combinations of labels corresponding to given documents. For the derived model without BiLSTM encoder for documents named w/o Doc encoder, we can see that the performance has also a large distance with the proposed BeNet model by 33.82% in *HL*, 8.20% in *mP* , 6.90% in *mR* and 7.53% in *mF* 1, possibly because the doc encoder layer further takes the contextual information of documents into consideration, enhancing the global semantic interaction.

Above all, each component of the proposed model BeNet has indispensable abilities separately and the organic combination of these layers jointly make tremendous contributions to its state-of-the-art performance.

**Table 4.** Ablation study of five derived models on RCV1-V2

| **Datasets** | **RCV1-V2** | | | |
|---|---|---|---|---|
| **Metrics** | **HL(-)** | **mP(+)** | **mR(+)** | **mF1(+)** |
| w/o BERTdoc | 0.0083 | 0.8662 | 0.8247 | 0.8856 |
| w/o Doc encoder | 0.0091 | 0.8547 | 0.8365 | 0.8455 |
| w/o Label embeddings | 0.0088 | 0.8763 | 0.8572 | 0.8666 |
| w/o Doc-label attention | 0.0086 | 0.8985 | 0.8163 | 0.8554 |
| BERTdoc to EMBdoc | 0.0076 | 0.9175 | 0.8456 | 0.8801 |
| **BeNet(ours)** | **0.0068** | **0.9248** | **0.8942** | **0.9092** |

### *5.3. Parameters sensitivity*

To increase the robustness of our proposed BeNet model, we carry out a series of experiments to analyze the impact of the length of input documents in the BERTdoc layer and the dimension of hidden state in the doc encoder layer of our proposed BeNet model on the RV1-V2 dataset with results shown in Figure 2. Due to the similar trend of parameters on two above-mentioned datasets, we just take one as an example.

From Figure 2, it is obvious to discover that the similar trend and small gaps emerge between the training set and test set on the dimension of hidden state and document length, demonstrating the proposed model BeNet manages to avoid overfitting as well as maintain strong generalization ability. Specifically, the turning points of dimension of hidden state (Figure 2(a)) is 300 both on the training set and test set, the larger of the dimension in the doc encoder layer when less than 300, the better performance of the proposed model achieving, with best performance 0.9243 on the training set and 0.9092 on the test set. Moreover,it drops dramatically beyond 300, extrapolating when the dimension of hidden state exceeds a certain limit, it will exert a negative influence on the model performance possibly due to complexity of the model. When it comes to the effect of document length on the proposed model (Figure 2(b)), there are two peaks at 250 and 500, attaining 0.8552 and 0.9243 on the training set as well as 0.8315 and 0.9092 on the test set, respectively. The two curves both see an upward trend below 250 and from 400 to 500 while falling off over 500 as well as oscillating between 250 and 400, indicating that BERT manages to capture more significant information when it learns from more input documents within acceptable limits.

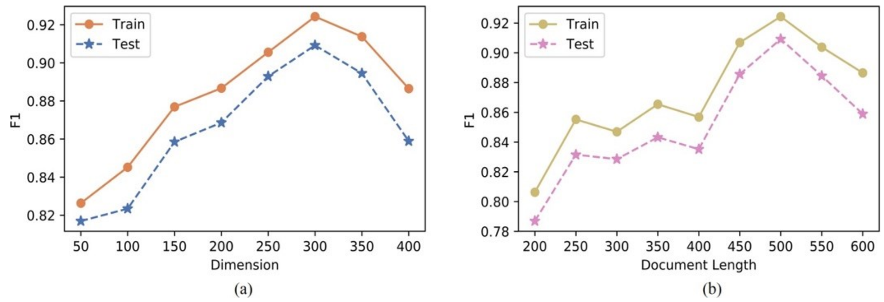
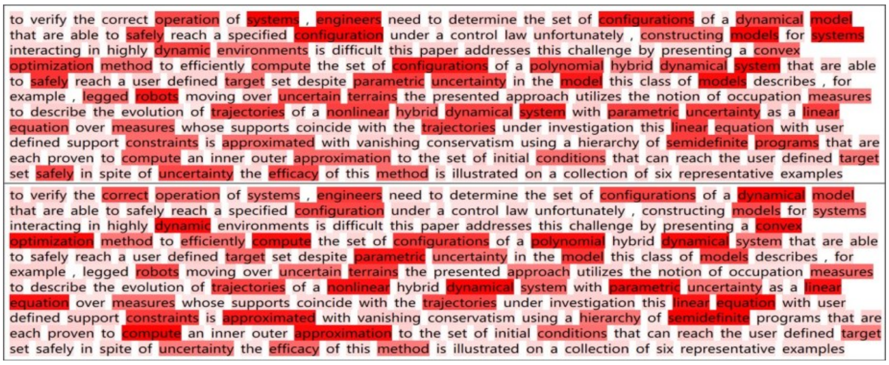


**Figure 2.** Influence of dimension of hidden state and document length on the RV1-V2 dataset

### *5.4. A case study*

Next, we make a case study to further interpret how to classify multi-label documents with our proposed model. Take a certain document from AAPD dataset labeled *cs.sy* and *math.oc* as an example with detailed content shown in Figure 3.



to verify the correct operation of systems , engineers need to determine the set of configurations of a dynamical model that are able to safely reach a specified configuration under a control law unfortunately , constructing models for systems interacting in highly dynamic environments is difficult this paper addresses this challenge by presenting a convex optimization method to efficiently compute the set of configurations of a polynomial hybrid dynamical system that are able to safely reach a user defined target set despite parametric uncertainty in the model this class of models describes , for example , legged robots moving over uncertain terrains the presented approach utilizes the notion of occupation measures to describe the evolution of trajectories of a nonlinear hybrid dynamical system with parametric uncertainty as a linear equation over measures whose supports coincide with the trajectories under investigation this linear equation with user defined support constraints is approximated with vanishing conservatism using a hierarchy of semidefinite programs that are each proven to compute an inner outer approximation to the set of initial conditions that can reach the user defined target set safely in spite of uncertainty the efficacy of this method is illustrated on a collection of six representative examples

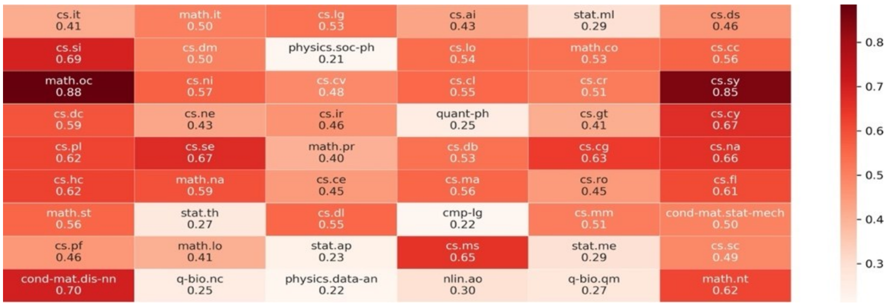
to verify the correct operation of systems , engineers need to determine the set of configurations of a dynamical model that are able to safely reach a specified configuration under a control law unfortunately , constructing models for systems interacting in highly dynamic environments is difficult this paper addresses this challenge by presenting a convex optimization method to efficiently compute the set of configurations of a polynomial hybrid dynamical system that are able to safely reach a user defined target set despite parametric uncertainty in the model this class of models describes , for example , legged robots moving over uncertain terrains the presented approach utilizes the notion of occupation measures to describe the evolution of trajectories of a nonlinear hybrid dynamical system with parametric uncertainty as a linear equation over measures whose supports coincide with the trajectories under investigation this linear equation with user defined support constraints is approximated with vanishing conservatism using a hierarchy of semidefinite programs that are each proven to compute an inner outer approximation to the set of initial conditions that can reach the user defined target set safely in spite of uncertainty the efficacy of this method is illustrated on a collection of six representative examples

**Figure 3.** Visual analysis of our proposed model on a MLTC task with label *cs.sy* (above) and *math.oc* (below)

Above all, we aim to explore different contributions of each word to the whole document displayed in color according to its belonging labels *cs.sy* and *math.oc*, respectively. For the first label *cs.sy*, it's not difficult to find that words such as *systems*, *engineers*, *configurations*, *models*, *robots* and their variants covered with deep red facilitate the proposed model to predict the correct category while words like *operation*, *dynamical*, *safely*, *optimization*, *terrains*, *trajectories*, *programs* and their different forms with less deep red also motivate multi-label text classification, catering for human perception. High contribution words to the second label *math.oc* such as *dynamical*,*convex*, *optimization*, *polynomial*, *nonlinear*, *linear*, *approximated*, *semidefinite* as well as less high correlation words like *correct*, *engineers*, *configurations*, *models*, *robots*, *constraints* are also conducive for predicting the target label from human perspective. Moreover, some auxiliary words, preposition, article such as *to*, *the*, *of*, *in* and other words with wide range of application scenarios like *initial*, *ultilizes*, *address* have little correlation with the corresponding labels.

Next, we reveal different probabilities of all unique labels calculated by our proposed BeNet through a heatmap shown in Figure 4 with the probabilities of correct labels *cs.sy* and *math.oc* obtaining 0.85 and 0.88 which substantially exceed other labels averaged by 0.2 to 0.7. Furthermore, some less related labels prefixed by *cs* and *math* have a probabilities between 0.4 and 0.7 while other almost irrelevant labels such as $physics.soc-ph$, $q-bio.nc$ only occupy 0.2 to 0.3.

From this concrete example, it's intuitionistic for researchers of Natural Language Processing to clarify the mechanism within our propose BeNet model on how to classify multi-label documents into multiple categories.



| | | | | | |
|---|---|---|---|---|---|
| cs.it<br>0.41 | math.it<br>0.50 | cs.lg<br>0.53 | cs.ai<br>0.43 | stat.ml<br>0.29 | cs.ds<br>0.46 |
| cs.si<br>0.69 | cs.dm<br>0.50 | physics.soc-ph<br>0.21 | cs.lo<br>0.54 | math.co<br>0.53 | cs.cc<br>0.56 |
| math.oc<br>0.88 | cs.ni<br>0.57 | cs.cv<br>0.48 | cs.cl<br>0.55 | cs.cr<br>0.51 | cs.sy<br>0.85 |
| cs.dc<br>0.59 | cs.ne<br>0.43 | cs.ir<br>0.46 | quant-ph<br>0.25 | cs.gt<br>0.41 | cs.cy<br>0.67 |
| cs.pl<br>0.62 | cs.se<br>0.67 | math.pr<br>0.40 | cs.db<br>0.53 | cs.cg<br>0.63 | cs.na<br>0.66 |
| cs.hc<br>0.62 | math.na<br>0.59 | cs.ce<br>0.45 | cs.ma<br>0.56 | cs.ro<br>0.45 | cs.fl<br>0.61 |
| math.st<br>0.56 | stat.th<br>0.27 | cs.dl<br>0.55 | cmp-lg<br>0.22 | cs.mm<br>0.51 | cond-mat.stat-mech<br>0.50 |
| cs.pf<br>0.46 | math.lo<br>0.41 | stat.ap<br>0.23 | cs.ms<br>0.65 | stat.me<br>0.29 | cs.sc<br>0.49 |
| cond-mat.dis-nn<br>0.70 | q-bio.nc<br>0.25 | physics.data-an<br>0.22 | nlin.ao<br>0.30 | q-bio.qm<br>0.27 | math.nt<br>0.62 |

**Figure 4.** Weights of all labels of the given document

## 6. Conclusion

In this paper, we propose a novel BERT Doc-Label Attention Network (BeNet), which designed to reliably predict all labels associated with each text. We use BERT to pretrain documents, fully capturing their deep semantic information and establishing connections among words and among sentences, respectively. Then we adopt BiLSTM to explore the contextual information of documents forward and backward. Next, we ultilize GloVe to construct label embeddings to dig latent information between label pairs. Afterwards, we apply doc-label attention mechanism to obtain interactive information between encoded pre-trained document representation and label embeddings via GloVe, followed by a MLP classifier to make final prediction. We carry out experiments on two datasets with four common evaluation metrics, the results demonstrate that our proposed model outperforms all state-of-the-art MLTC models. We also carry out case study to visualize its real applications. In the future, we will generalize our model with more datasets to increase its robustness and extend its applications in more scenarios.

## References

[1] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel text classification for automated tag suggestion," ECML PKDD discovery challenge, vol. 75, p. 2008, 2008.

[2] S. Gopal and Y. Yang, "Multilabel classification with metalevel features," in Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval, 2010, pp. 315–322.

[3] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," Pattern recognition, vol. 37, no. 9, pp. 1757–1771, 2004.

[4] A. Clare and R. D. King, "Knowledge discovery in multilabel phenotype data," in European conference on principles of data mining and knowledge discovery. Springer, 2001, pp. 42–53.

[5] A. Elisseeff and J. Weston, "A kernel method for multilabelled classification," Advances in neural information processing systems, vol. 14, 2001.

[6] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," International Journal of Data Warehousing and Mining (IJDWM), vol. 3, no. 3, pp. 1–13, 2007.

[7] 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.

[8] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," Machine learning, vol. 85, pp. 333–359, 2011.

[9] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," IEEE transactions on Knowledge and Data Engineering, vol. 18, no. 10, pp. 1338–1351, 2006.

[10] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," Advances in neural information processing systems, vol. 27, 2014.

[11] 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, 2018.

[12] 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.

[13] C. Li, B. Wang, V. Pavlu, and J. Aslam, “Conditional bernoulli mixtures for multi-label classification,” in International conference on machine learning. PMLR, 2016, pp. 2482–2491.

[14] C. Vens, J. Struyf, L. Schietgat, S. Dzeroski, and H. Blockeel, “Decision trees for hierarchical multi-label classification,” Machine learning, vol. 73, pp. 185–214, 2008.

[15] R. Cerri, R. C. Barros, A. C. PLF de Carvalho, and Y. Jin, “Reduction strategies for hierarchical multi-label classification in protein function prediction,” BMC bioinformatics, vol. 17, no. 1, pp. 1–24, 2016.

[16] N. Ghamrawi and A. McCallum, “Collective multi-label classification,” in Proceedings of the 14th ACM international conference on Information and knowledge management, 2005, pp. 195–200.

[17] L. Li, H. Wang, X. Sun, B. Chang, S. Zhao, and L. Sha, “Multi-label text categorization with joint learning predictions-as-features method,” in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 835–839.

[18] J. Nam, J. Kim, E. Loza Menc´ıa, I. Gurevych, and J. Furnkranz, “Large-scale multi-label text classification—revisiting neural networks,” in Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14. Springer, 2014, pp. 437–452.

[19] G. Kurata, B. Xiang, and B. Zhou, “Improved neural networkbased multi-label classification with better initialization leveraging label co-occurrence,” in Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 521–526.

[20] G. Chen, D. Ye, Z. Xing, J. Chen, and E. Cambria, “Ensemble application of convolutional and recurrent neural networks for multi-label text categorization,” in 2017 International joint conference on neural networks (IJCNN). IEEE, 2017, pp. 2377–2383.

[21] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, and Y. Bengio, “An actor-critic algorithm for sequence prediction,” arXiv preprint arXiv:1607.07086, 2016.

[22] 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, 2018.

[23] J. Lin, Q. Su, P. Yang, S. Ma, and X. Sun, “Semantic-unitbased dilated convolution for multi-label text classification,” arXiv preprint arXiv:1808.08561, 2018.

[24] L. Cai, Y. Song, T. Liu, and K. Zhang, “A hybrid bert model that incorporates label semantics via adjustive attention for multi-label text classification,” Ieee Access, vol. 8, pp. 152 183–152 192, 2020.

[25] Y. Yan, F. Liu, X. Zhuang, and J. Ju, “An r-transformer bilstm model based on attention for multi-label text classification,” Neural Processing Letters, vol. 55, no. 2, pp. 1293–1316, 2023.

[26] W. Zhang, J. Yan, X. Wang, and H. Zha, “Deep extreme multi-label learning,” in Proceedings of the 2018 ACM on international conference on multimedia retrieval, 2018, pp. 100–107.

[27] C. Du, Z. Chen, F. Feng, L. Zhu, T. Gan, and L. Nie, “Explicit interaction model towards text classification,” in Proceedings of the AAAI conference on artificial intelligence, vol. 33, no. 01, 2019, pp. 6359–6366.

[28] N. Pappas and J. Henderson, “Gile: A generalized inputlabel embedding for text classification,” Transactions of the Association for Computational Linguistics, vol. 7, pp. 139–155, 2019.

[29] D. D. Lewis, Y. Yang, T. Russell-Rose, and F. Li, “Rcv1: A new benchmark collection for text categorization research,” Journal of machine learning research, vol. 5, no. Apr, pp. 361–397, 2004.

[30] Y. Kim, “Convolutional neural networks for sentence classification,” arXiv preprint arXiv:1408.5882, 2014.

[31] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” arXiv preprint arXiv:1409.0473, 2014.