UNIVERSITÀ
DEGLI STUDI
FIRENZE

Correlation Networks for Extreme Multi-label Text Classification

[*Guangxu Xun, Kishlay Jha, Jianhui Sun, Aidong Zhang*]

Department of Computer Science, University of Virginia, Charlottesville, VA, USA

Lorenzo Gianassi

Machine Learning course Project held by Prof. Paolo Frasconi

Introduction

Introduction

Extreme multi-label text classification (XMTC) is a Natural Language Processing (NLP) task, which aims to tag a given text with the most relevant subset of labels from an extremely large label set.

Current XMTC models can be grouped into four categories:

1. **one-vs-all models** which learn a separate classifier for each label
2. **embedding based models** which represent labels in a low-dimensional embedding space
3. **tree based models** which learn a label hierarchy to improve model efficiency
4. **deep learning based models** which employ deep learning techniques.

A deep multi-label text classification model is normally composed of two components:

- The first component extracts information from the text sequence.
- The second component converts the extracted features into label predictions.

The first component normally fall into three categories: **RNN, CNN** and **BERT**.
For the second component the most common choice is a **Fully Connected Layer**

Using a fully connected layer is simple, but it fails to take full advantage of the correlations among different labels.
This correlation help us to obtain more accurate label predictions.

Therefore, a new network architecture, named **Correlation Networks** (*CorNet*) is proposed.
It works as an add-on enhancer module to existing deep XMTC models, and is able to improve the original architecture by allowing it to promote correlated predictions.

It can be said *CorNet* is an independent module that can be added after the last label prediction layer of a deep XMTC model.

To sum up, *CorNet* has the following advantages:

- *CorNet* is able to exploit the correlation information among different labels
- *CorNet* is a general and independent architecture that can be directly integrated with any deep XMTC model without changing the model.
- *CorNet* models consistently achieve significant improvements over the state-of-the-art deep XMTC models on benchmark datasets.
- *CorNet* models converge faster than the original models during training.

UNIVERSITÀ DEGLI STUDI FIRENZE

CorNet

Cornet

A CorNet block is a computational unit which maps raw label predictions to enhanced label predictions based on label correlations.

Formally, a CorNet building block is defined as:

$$y = F(x)+x$$

F stands for the underlying mapping function, and

- *x* denotes the raw label predictions before the CorNet block
- *y* denotes the enhanced label predictions after the CorNet block

It is also added an identity mapping between raw predictions *x* and correlation enhanced predictions *y*.

F is the correlation enhancing function to be learned.
The most straightforward design is one fully connected layer:

$$F(x)=Wx$$

where *W* denotes the weight matrix of the layer.
The i^{th} enhanced prediction y_i is a linear combination of all raw predictions $\{x_1, x_2, \ldots, x_i, \ldots\}$ that represents all possible linear correlations between the i^{th} label and other labels.

The structure of the *F* function is impractical to handle and keep in memory.

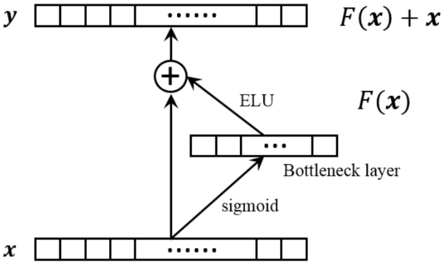
Let us now consider the structure of the matrix *W*.

Let V be the number of distinct labels, then W would be a V-by-V matrix. As a result, there are downsides:

- V is usually huge in XMTC tasks, it is hard for W to fit in the GPU memory.
- Waste of the expressive power of the network, hence most elements in W would be 0s (most labels are uncorrelated).
- One fully connected layer only allows us to exploit linear label correlations.

A bottleneck layer between X and Y is inserted.
Let R denote the dimension of the bottleneck layer with $R \ll V$.

Doing so, the model size is significantly reduced and more complex correlations can be captured by the additional layer.



Therefore, our design for function F can be formally defined as:

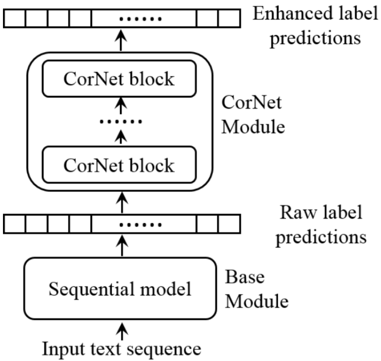
$$F(x) = W_2\delta(W_1\sigma(x) + b_1) + b_2$$

where W_1, W_2 are the weight matrices, b_1, b_2 are the biases, and σ,δ are the sigmoid activation function and the ELU activation function respectively.

Any number of CorNet blocks can be stacked to form a deep CorNet module and the output of each block is a correlation enhancement over the output of the previous block.

This means more complicated label correlations can be captured by the deep CorNet module.

CorNet is a general architecture and it can be concatenated after any standard deep XMTC architectures.



- The base module could be seen as a black box which is responsible for converting a text sequence into raw label predictions.
- The CorNet module then enhances the raw predictions with label correlations and outputs the enhanced label predictions.

UNIVERSITÀ
DEGLI STUDI
FIRENZE

CorNet Istantiations

CorNet Istantiations

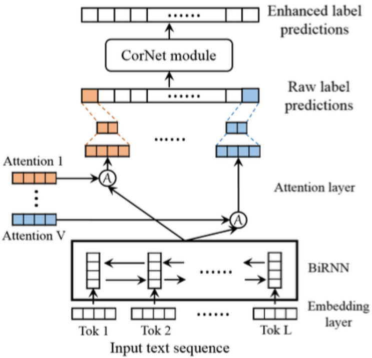
Depending on how raw label prediction x is derived, CorNet models can have different structures.

This flexibility of CorNet allows it to employ different sequence modeling styles as the base module.

To illustrate this point, we develop CorNet models by integrating CorNet modules with several standard deep XMTC architectures.

CorNetAttentionXML

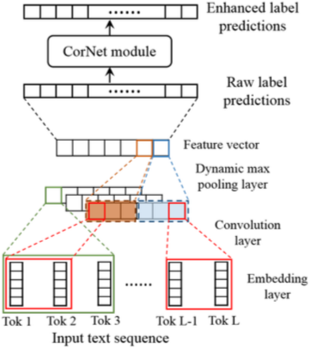
AttentionXML extracts a feature vector for every distinct label.
It has a self-attention for every label and each attention generates a feature vector to predict the corresponding label.



- It models text sequences with bidirectional RNNs.
- Suffers from expensive computational cost.
- The number of self-attentions in AttentionXML is equivalent to the size of the label set V, which could be in the millions.
- The final multi-label prediction vector is obtained by concatenating all individual label predictions.

CorNetXML-CNN

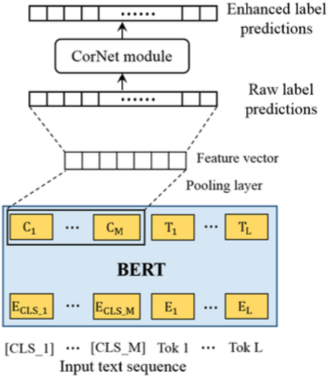
The XML-CNN model can achieve feature vector by applying a set of 1D convolution filters and a dynamic max pooling on the input word embeddings.



- This fixed-dimensional feature vector is used as the aggregate sequence representation.
- The feature vector is proportional to the number of the convolution filters.
- One can adjust the number of convolution filters based on the size of features needed.

CorNetBertXML

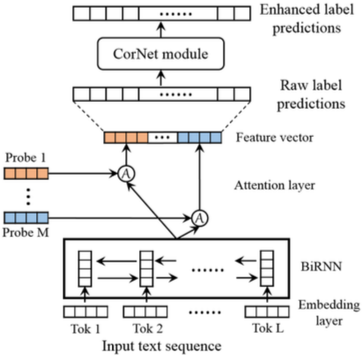
For classification tasks, BERT always adds a special symbol *[CLS]* in front of every input sequence and uses the final hidden state corresponding to this token as the aggregate sequence representation.


Enhanced label predictions
CorNet module
Raw label predictions
Feature vector
Pooling layer
C_1 ··· C_M T_1 ··· T_L
BERT
E_{CLS_1} ··· E_{CLS_M} E_1 ··· E_L
[CLS_1] ··· [CLS_M] Tok 1 ··· Tok L
Input text sequence

- The solution proposed is to have multiple special tokens to accommodate to extreme classification tasks
- The multi-head self-attention mechanism of BERT allows each special token to go over the entire input sequence and the final hidden state of each special token can be deemed as a particular feature vector.

CorNetMeSHProbeNet

MeSHProbeNet was originally proposed for biomedical document annotation, so tagging biomedical documents with relevant Medical Subject Headings (MeSH) terms (*Contains extra journal information*).



- We first need to remove journal related components.
- It models text sequences with bidirectional RNNs.
- Each MeSH probe is a self-attention that extract related info from the RNN hidden states and output a fixed-dimensional feature vector.
- Label prediction *x* is calculated based on the concatenated feature vector.

Experiments

Dataset

We use two benchmark datasets, including one small-scale dataset EUR-Lex, one medium-scale dataset AmazonCat-13K. The dataset statistics are summarized in Table

Dataset	*Ntrain*	*NTest*	L	$\bar{L}$	$\tilde{L}$
EUR-Lex	15,449	3,865	3,956	5.30	20.79
AmazonCat-13K	1,186,239	306,782	13,330	5.04	448.57
AmazonCat-13K (Reduced)	100,000	30,000	10,202	4.96	48.69

- For each dataset, the vocabulary size is limited to 500,000 words;
- Word embeddings are initialized with the 300-dimensional pretrained GloVe embeddings.
- All models are trained by the Adam optimizer with a learning rate of 1e-3.

Model Configurations

In order to make the performance comparison as fair as possible, we assign the same embedding dimension for all models.

CorNetAttentionXML

(EUR-Lex):

- dim. of embeddings: 300;
- hidden size: 256;
- number of RNN layers: 1;
- dim. of FC layers: 256;
- dropout rate: 0.5.

(AmazonCat-13K)

- dim. of embeddings: 300;
- hidden size: 200;
- number of RNN layers: 1;
- dim. of FC layers: 100;
- dropout rate: 0.5.

Evaluation Metrics

Two instance-based ranking metrics to evaluate the models are adopted:

- *precision at top k* (**precision@k**)
- *normalized Discounted Cumulative Gain at top k* (**nDCG@k**)

Let $z \epsilon \left\{0, \quad 1\right\}^L$ denote the ground truth label vector of an instance and $\hat{z} \epsilon \mathbb{R}^L$ denote the model predicted score vector for the same instance

$$DCG@k = \sum_{l \epsilon r_k(\hat{z})} \frac{z_l}{log(l+1)}$$

$$precision@k = \frac{1}{k} \sum_{l \epsilon r_k(\hat{z})} z_l$$

$$nDCG@k = \frac{DCG@k}{\sum_{l=1}^{min(k, \|z\|_0)} \frac{1}{log(l+1)}}$$

where

- $r_k(\hat{z})$ is the ground truth indices corresponding to the top k indices of the model predicted rank list,
- $\|z\|_0$ counts the number of ground truth labels for this instance.

EUR-Lex

Model	P@1	P@3	P@5	N@1	N@3	N@5	#GPUs	#hours
XML-CNN	76.81	62.79	51.56	76.81	66.44	60.47	1	0.08
CorNetXML-CNN	78.60	64.22	53.07	78.60	67.81	61.90	1	0.08
BertXML	77.80	64.57	53.25	77.80	67.97	62.10	1	0.25
CorNetBertXML	79.02	65.49	53.94	79.02	68.98	62.97	1	0.29
MeSHProbeNet	79.92	66.52	55.13	79.92	69.98	64.13	1	1.08
CorNetMeSHProbeNet	83.47	70.50	58.73	83.47	73.86	67.95	1	1.09
AttentionXML	85.43	73.30	60.99	85.43	76.54	70.45	1	0.95
CorNetAttentionXML	86.39	73.30	61.72	86.39	77.10	71.24	1	0.96
CorNetAttentionXML (reproduced)	**86.15**	**73.77**	**61.95**	**86.15**	**77.03**	**71.29**	**1**	**0.9**

- *CorNet* favors larger datasets because there exist more label correlations to utilize in larger datasets.
- The improvement is more significant for k=5 than for k=1.
  That is because k=1 represents the most confident label prediction.
- *CorNetAttentionXML* is able to outperform *AttentionXML* and achieve the new state-of- the-art results by incorporating label correlations.

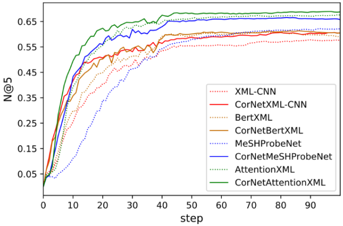
AmazonCat-13K

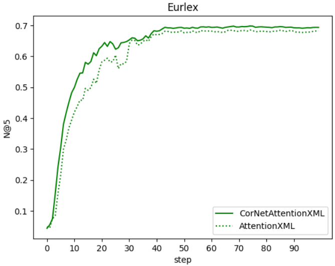
Model	P@1	P@3	P@5	N@1	N@3	N@5	#GPUs	#hours
XML-CNN	94.53	79.12	63.38	94.53	88.19	85.61	1	2.57
CorNetXML-CNN	95.36	80.55	64.83	95.36	89.54	87.11	1	4.17
BertXML	94.78	80.78	65.51	94.78	89.57	87.58	1	3.88
CorNetBertXML	95.22	81.37	66.03	95.22	90.13	88.13	1	3.70
MeSHProbeNet	95.63	81.84	66.47	95.63	90.65	88.69	1	9.08
CorNetMeSHProbeNet	96.10	82.82	67.57	96.04	91.54	89.78	1	9.22
AttentionXML	95.13	81.12	66.10	95.13	89.90	88.13	4	14.73
AttentionXML (reproduced)	**91.12**	**74.97**	**59.62**	**91.02**	**83.17**	**80.27**	**1**	**2.58**
CorNetAttentionXML	96.16	82.81	67.63	96.19	91.54	89.91	4	20.21
CorNetAttentionXML (reproduced)	**92.22**	**76.07**	**60.83**	**92.22**	**84.49**	**81.09**	**1**	**3.50**

- *CorNetBertXML* and *CorNetMeSHProbeNet* also outperformed *AttentionXML* on AmazonCat-13K.
- The *CorNet* enhanced models are able to maintain similar training time consumptions to their original counterpart models, thanks to the simple architecture of *CorNet*.
- We also observe that from *(CorNet)XML-CNN*, *(CorNet)BertXML* to *(CorNet)MeSHProbeNet*, *(CorNet)AttentionXML*, the precision grows higher, but the training time consumption also becomes longer.

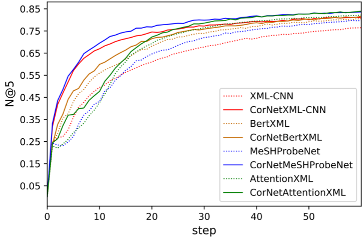
Convergence

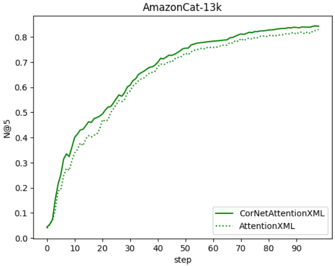
Eur-lex:





AmazonCat-13k:





UNIVERSITÀ DEGLI STUDI FIRENZE

Ablation Analysis

Ablation Analysis

The CorNetMeSHProbeNet model is used as the backbone architecture here.

Model size:

- Since the CorNet module introduces several additional layers into the base module, one might wonder whether the performance improvement is a consequence of the label correlation or simply the additional parameter size.
- To make the comparison more convincing, we include **MeSHProbeNet-large**, which is a wider and deeper version of MeSHProbeNet.
- We can see that although MeSHProbeNet-large achieves minor improvement over MeSHProbeNet with the help of a wider and deeper configuration, CorNetMeSHProbeNet is still able to significantly outperform MeSHProbeNet-large.

Ablation Analysis

To investigate the trade-off between performance and computational cost associated with the depth of the CorNet module, we conduct experiments with different numbers of CorNet blocks.

Number of CorNet Blocks:

- The performance is robust to the number of CorNet blocks, although more CorNet blocks generally indicate a better performance.
- Two CorNet blocks achieve a good balance between performance and complexity and are used as the default setting in the our experiments.

Conclusion

- **CorNet** is a general and independent architecture that can be directly integrated with any deep XMTC models as an add-on enhancer module.
- Results demonstrated the effectiveness of *CorNet*, which is able to advance the state-of-the-art performance on several different multi-label text classification datasets.
- *CorNet* also exhibited the ability to accelerate the convergence rate during training.

Furthermore, we can say that you were able to reproduce the experiments carried out in this paper by obtaining almost the same results