Hyperbolic Capsule Networks for Multi-Label Classification

Boli Chen, Xin Huang, Lin Xiao, Liping Jing
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University, Beijing, China
{18120345, 18120367, 17112079, lpjing}@bjtu.edu.cn

Abstract

Although deep neural networks are effective at extracting high-level features, classification methods usually encode an input into a vector representation via simple feature aggregation operations (*e.g.* pooling). Such operations limit the performance. For instance, a multi-label document may contain several concepts. In this case, one vector can not sufficiently capture its salient and discriminative content. Thus, we propose Hyperbolic Capsule Networks (HYPERCAPS) for Multi-Label Classification (MLC), which have two merits. First, hyperbolic capsules are designed to capture fine-grained document information for each label, which has the ability to characterize complicated structures among labels and documents. Second, Hyperbolic Dynamic Routing (HDR) is introduced to aggregate hyperbolic capsules in a label-aware manner, so that the label-level discriminative information can be preserved along the depth of neural networks. To efficiently handle large-scale MLC datasets, we additionally present a new routing method to adaptively adjust the capsule number during routing. Extensive experiments are conducted on four benchmark datasets. Compared with the state-of-the-art methods, HYPERCAPS significantly improves the performance of MLC especially on tail labels.

1 Introduction

The main difference between Multi-Class Classification (MCC) and Multi-Label Classification (MLC) is that datasets in MCC have only serval mutually exclusive classes, while datasets in MLC contain much more correlated labels. MLC allows label co-occurrence in one document, which indicates that the labels are not disjointed. In addition, a large fraction of the labels are the infrequently occurring *tail labels* (Bhatia et al., 2015), which is also referred as the power-law label distribution.

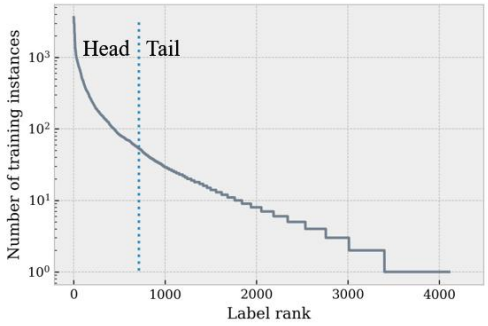


Figure 1: The power-law label distribution of EUR-LEX57K with Y-axis on log-scale. Division is based on average number of training instances.

Figure 1 illustrates the label distribution of EUR-LEX57K (Chalkidis et al., 2019). A multi-label document usually has serval head and tail labels, and hence contain several concepts about both its head and tail labels simultaneously.

Recent works for text classification, such as CNN-KIM (Kim, 2014) and FASTTEXT (Joulin et al., 2017), focus on encoding a document into a fixed-length vector as the *distributed document representation* (Le and Mikolov, 2014). These encoding based deep learning methods use simple operations (*e.g.* pooling) to aggregate features extracted by neural networks and construct the document vector representation. A Fully-Connected (FC) layer is usually applied upon the document vector to predict the probability of each label. And each row in its weight matrix can be interpreted as a label vector representation (Du et al., 2019b). In this way, the label probability can be predicted by computing the dot product between label and document vectors, which is proportional to the scalar projection of the label vector onto the document vector as shown in Figure 2. For example, label "*movie*" should have the largest scalar projection onto a document about "movie". However, even

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 3115–3124

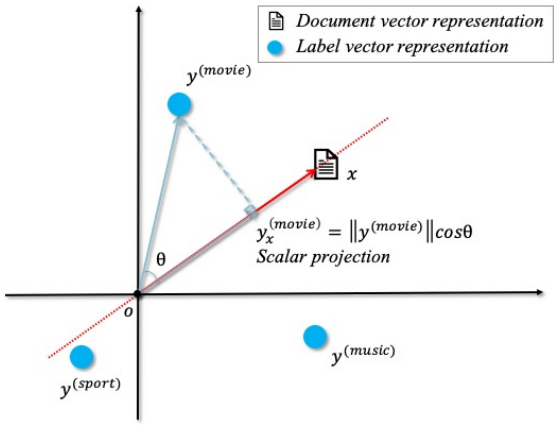


Figure 2: Illustration of the FC layer in the encoding based methods.

the learned label representation of "*music*" can be distinguished from "*movie*", it may also have a large scalar projection onto the document.

Moreover, multi-label documents always contain several concepts about multiple labels, such as a document about "sport movie". Whereas the document vector representation is identical to all the labels, and training instances for tail labels are inadequate compared to head labels. The imbalance between head and tail labels makes it hard for the FC layer to make prediction, especially on tail labels. In this case, one vector can not sufficiently capture its salient and discriminative content. Therefore, the performance of constructing the document vector representation via simple aggregation operations is limited for MLC.

Capsule networks (Sabour et al., 2017; Yang et al., 2018a) has recently proposed to use dynamic routing in place of pooling and achieved better performance for classification tasks. In fact, capsules are fine-grained features compared to the distributed document representation, and dynamic routing is a label-aware feature aggregation procedure. (Zhao et al., 2019) improves the scalability of capsule networks for MLC. However, they only use CNN to construct capsules, which capture local contextual information (Wang et al., 2016). Effectively learning the document information about multiple labels is crucial for MLC. Thus we propose to connect CNN and RNN in parallel to capture both local and global contextual information, which would be complementary to each other. Nevertheless, Euclidean capsules necessitate designing a non-linear squashing function.

Inspired by the hyperbolic representation learning methods which demonstrate that the hyperbolic space has more representation capacity than the Euclidean space (Nickel and Kiela, 2017; Ganea et al., 2018a), Hyperbolic Capsule Networks (HYPERCAPS) is proposed. Capsules are constrained in the hyperbolic space which does not require the squashing function. Hyperbolic Dynamic Routing (HDR) is introduced to aggregate hyperbolic capsules in a label-aware manner. Moreover, in order to fit the large label set of MLC and improve the scalability of HYPERCAPS, adaptive routing is presented to adjust the number of capsules participated in the routing procedure.

The main contributions of our work are therefore summarized as follows:

- We propose to connect CNN and RNN in parallel to simultaneously extract local and global contextual information, which would be complementary to each other.
- HYPERCAPS with HDR are formulated to aggregate features in a label-aware manner, and hyperbolic capsules benefits from the representation capacity of the hyperbolic space.
- Adaptive routing is furthermore presented to improve the scalability of HYPERCAPS and fit the large label set of MLC.
- Extensive experiments on four benchmark MLC datasets demonstrate the effectiveness of HYPERCAPS, especially on tail labels.

2 Preliminaries

In order to make neural networks work in the hyperbolic space, formalism of the *Möbius gyrovector space* is adopted (Ganea et al., 2018b).

An n-dimensional *Poincaré ball* $\mathcal{B}^n$ is a *Riemannian manifold* defined as $\mathcal{B}^n = \{\boldsymbol{x} \in \mathbb{R}^n \mid \|\boldsymbol{x}\| < 1\}$, with its *tangent space* around $\boldsymbol{p} \in \mathcal{B}^n$ denoted as $T_{\boldsymbol{p}}\mathcal{B}^n$ and the *conformal factor* as $\lambda_{\boldsymbol{p}} := \frac{2}{1-\|\boldsymbol{p}\|^2}$. The *exponential map* $\exp_{\boldsymbol{p}} : T_{\boldsymbol{p}}\mathcal{B}^n \rightarrow \mathcal{B}^n$ for $\boldsymbol{w} \in T_{\boldsymbol{p}}\mathcal{B}^n \setminus \{\boldsymbol{0}\}$ is consequently defined as

$$\exp_{\boldsymbol{p}}(\boldsymbol{w}) = \boldsymbol{p} \oplus (\tanh(\frac{\lambda_{\boldsymbol{p}}}{2}\|\boldsymbol{w}\|)\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}). \quad (1)$$

To work with hyperbolic capsules, Möbius operations in the Poincaré ball also need to be formulated.

Möbius addition for $\boldsymbol{u}, \boldsymbol{v} \in \mathcal{B}^n$ is defined as

$$\boldsymbol{u} \oplus \boldsymbol{v} = \frac{(1+2\langle \boldsymbol{u},\boldsymbol{v}\rangle+\|\boldsymbol{v}\|^2)\boldsymbol{u}+(1-\|\boldsymbol{u}\|^2)\boldsymbol{v}}{1+2\langle \boldsymbol{u},\boldsymbol{v}\rangle+\|\boldsymbol{u}\|^2\|\boldsymbol{v}\|^2}, \quad (2)$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product.

Thus *Möbius summation* can be formulated as

$$\mathop{\mathfrak{M}}_{i=m}^{n} \boldsymbol{p}_i = \boldsymbol{p}_m \oplus \cdots \oplus \boldsymbol{p}_n, p_i \in \mathcal{B}^n. \quad (3)$$

Möbius scalar multiplication for $k \in \mathbb{R}$ and $\boldsymbol{p} \in \mathcal{B}^n \setminus \{\mathbf{0}\}$ is defined as

$$k \otimes \boldsymbol{p} = \tanh(k \tanh^{-1}(\|\boldsymbol{p}\|)) \frac{\boldsymbol{p}}{\|\boldsymbol{p}\|}. \quad (4)$$

And $k \otimes \boldsymbol{p} = \mathbf{0}$ when $\boldsymbol{p} = \mathbf{0} \in \mathcal{B}^n$.

The definition of *Möbius matrix-vector multiplication* for $\boldsymbol{M} \in \mathbb{R}^{m \times n}$ and $\boldsymbol{p} \in \mathcal{B}^n$ when $\boldsymbol{M}\boldsymbol{p} \neq \mathbf{0}$ is as follows

$$\boldsymbol{M} \otimes \boldsymbol{p} = \tanh(\frac{\|\boldsymbol{M}\boldsymbol{p}\|}{\|\boldsymbol{p}\|} \tanh^{-1}(\|\boldsymbol{p}\|)) \frac{\boldsymbol{M}\boldsymbol{p}}{\|\boldsymbol{M}\boldsymbol{p}\|}. \quad (5)$$

And $\boldsymbol{M} \otimes \boldsymbol{p} = \mathbf{0}$ when $\boldsymbol{M}\boldsymbol{p} = \mathbf{0}$.

HDR is developed based on these operations.

3 Local and Global Hyperbolic Capsules

Neural networks are generally used as effective feature extractors for text classification. Kernels of CNN can be used to capture local n-gram contextual information at different positions of a text sequence, while hidden states of RNN can represent global long-term dependencies of the text (Wang et al., 2016). Hence, we propose to obtain the combination of local and global hyperbolic capsules by connecting CNN and RNN in parallel, which would be complementary to each other.

Given a text sequence of a document with T word tokens $\boldsymbol{x} = [x_1, \ldots, x_T]$, pre-trained w-dimensional word embeddings (*e.g.* GLOVE (Pennington et al., 2014)) are used to compose word vector representations $\boldsymbol{E} = [\boldsymbol{e}_1, \ldots, \boldsymbol{e}_T] \in \mathbb{R}^{T \times w}$, upon which CNN and RNN connected in parallel are used to construct local and global hyperbolic capsules in the Poincaré ball. Figure 3 illustrates the framework for HYPERCAPS.

3.1 Local Hyperbolic Capsule Layer

N-gram kernels $\boldsymbol{K} \in \mathbb{R}^{k \times w}$ with different window size k are applied on the local region of the word representations $\boldsymbol{E}_{t:t+k-1} \in \mathbb{R}^{k \times w}$ to construct the local features as

$$l_t = \varphi(\boldsymbol{K} \circ \boldsymbol{E}_{t:t+k-1}), \quad (6)$$

where $\circ$ denotes the element-wise multiplication and φ is a non-linearity (*e.g. ReLU*). For simplicity, the bias term is omitted.

With totally d channels, the local hyperbolic capsules at position t can be constructed as

$$\boldsymbol{l}_t = \exp_{\mathbf{0}}([l_t^{(1)}, \ldots, l_t^{(d)}]) \in \mathcal{B}^d. \quad (7)$$

Therefore, a k-gram kernel with 1 stride can construct $T-k+1$ local hyperbolic capsules. The local hyperbolic capsule set is denoted as $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_L\}$.

3.2 Global Hyperbolic Capsule Layer

Bidirectional GRU (Chung et al., 2014) is adopted to incorporate forward and backward global contextual information and construct the global hyperbolic capsules. Forward and backward hidden states at time-step t are obtained by

$$\begin{aligned} \overrightarrow{\boldsymbol{h}_t} &= \mathrm{GRU}(\overrightarrow{\boldsymbol{h}_{t-1}}, \boldsymbol{e}_t), \\ \overleftarrow{\boldsymbol{h}_t} &= \mathrm{GRU}(\overleftarrow{\boldsymbol{h}_{t+1}}, \boldsymbol{e}_t). \end{aligned} \quad (8)$$

Each of the total $2T$ hidden states can be taken as a global hyperbolic capsule using the exponential map, *i.e.* $\overrightarrow{\boldsymbol{g}_t} = \exp_{\mathbf{0}}(\overrightarrow{\boldsymbol{h}_t})$, and equally for the backward capsules. The global hyperbolic capsule set is denoted as $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_G\}$.

3.3 Hyperbolic Compression Layer

As discussed in (Zhao et al., 2019), the routing procedure is computational expensive for a large number of capsules. Compressing capsules into a smaller amount can not only relieve the computational complexity, but also merge similar capsules and remove outliers. Therefore, hyperbolic compression layer is introduced. Each compressed local hyperbolic capsule is calculated as a weighted Möbius summation over all the local hyperbolic capsules. For instance,

$$\boldsymbol{u}_l = \mathop{\mathfrak{M}}_{\boldsymbol{u}_k \in \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_L\}} r_k \otimes \boldsymbol{u}_k \in \mathcal{B}^d, \quad (9)$$

where r_k is a learnable weight parameter. And likewise for compressing global hyperbolic capsules.

Let set $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_P\}$ denote the compressed local and global hyperbolic capsules together, which are then aggregated in a label-aware manner via HDR.

4 Hyperbolic Dynamic Routing

The purpose of Hyperbolic Dynamic Routing (HDR) is to iteratively aggregate local and global hyperbolic capsules into label-aware hyperbolic capsules, whose activations stand for probabilities of the labels.

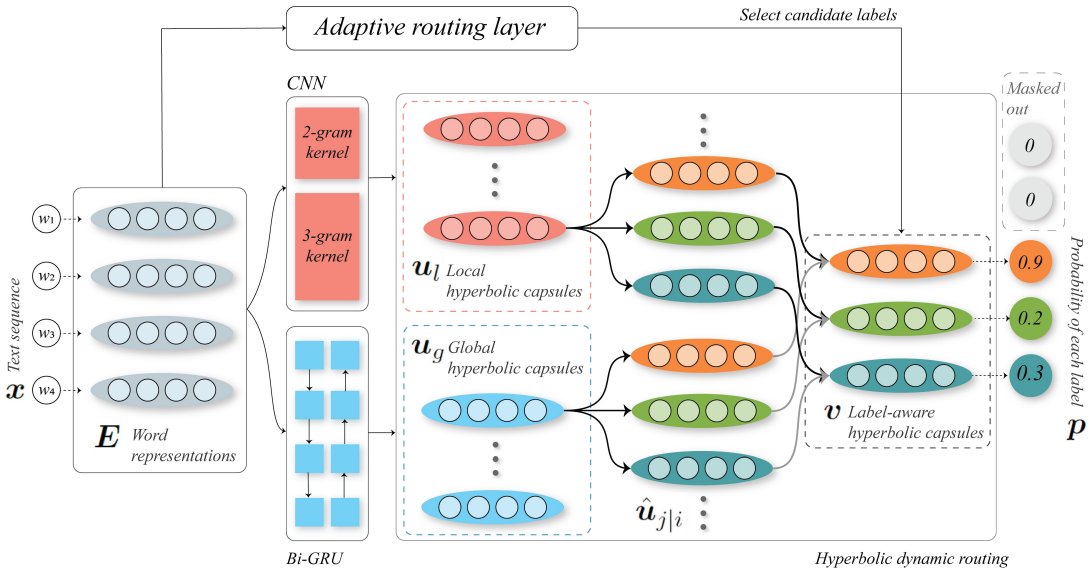


Figure 3: Illustration of HYPERCAPS framework.

4.1 Label-Aware Hyperbolic Capsules

With the acquirement of the compressed local and global hyperbolic capsule set $\{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_P\}$ in layer ℓ, let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_Q\}$ denote the label-aware hyperbolic capsule set in the next layer $\ell+1$, where Q equals to the number of labels.

Following (Sabour et al., 2017), the compressed hyperbolic capsules are firstly transformed into a set of prediction capsules $\{\hat{\boldsymbol{u}}_{j|1}, \ldots, \hat{\boldsymbol{u}}_{j|P}\}$ for the j-th label-aware capsule, each of them is calculated by

$$\hat{\boldsymbol{u}}_{j|i} = \boldsymbol{W}_{ij} \otimes \boldsymbol{u}_i \in \mathcal{B}^d, \tag{10}$$

where $\boldsymbol{W}_{ij}$ is a learnable parameter.

Then $\boldsymbol{v}_j$ is calculated as a weighted Möbius summation over all the prediction capsules by

$$\boldsymbol{v}_j = \mathop{\mathfrak{M}}_{\hat{\boldsymbol{u}}_{j|i} \in \{\hat{\boldsymbol{u}}_{j|1}, \ldots, \hat{\boldsymbol{u}}_{j|P}\}} c_{ij} \otimes \hat{\boldsymbol{u}}_{j|i}, \tag{11}$$

where c_{ij} denotes the coupling coefficient that indicates the connection strength between $\hat{\boldsymbol{u}}_{j|i}$ and $\boldsymbol{v}_j$.

The coupling coefficient c_{ij} is iteratively updated during the HDR procedure and computed by the *routing softmax*

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}, \tag{12}$$

where the logits b_{ij} are the log prior probabilities between capsule i and j, which are initialized as 0.

Once the label-aware hyperbolic capsules are produced, each b_{ij} is then updated by

$$b_{ij} = b_{ij} + \mathfrak{K}(d_{\mathcal{B}}(\boldsymbol{v}_j, \hat{\boldsymbol{u}}_{j|i})), \tag{13}$$

where $d_{\mathcal{B}}(\cdot, \cdot)$ denotes the *Poincaré distance*, which can be written as

$$d_{\mathcal{B}}(\boldsymbol{u}, \boldsymbol{v}) = \cosh^{-1}(1 + \frac{1}{2}\lambda_{\boldsymbol{u}}\lambda_{\boldsymbol{v}}\|\boldsymbol{u} - \boldsymbol{v}\|^2). \tag{14}$$

And $\mathfrak{K}$ is a *Epanechnikov kernel function* (Wand and Jones, 1994) with

$$\mathfrak{K} = \begin{cases} \gamma - x, & x \in [0, \gamma) \\ 0, & x \geq \gamma \end{cases} \tag{15}$$

where γ is the maximum Poincaré distance between two points in the Poincaré ball, which is $d_{\mathcal{B}}(\boldsymbol{p}, \boldsymbol{0})$ with $\|\boldsymbol{p}\| = 1 - \epsilon$ ($\epsilon = 10^{-5}$) to avoid numerical errors.

HDR is summarized in Algorithm 1. Different from the routing procedure described in (Sabour et al., 2017), HDR does not require the squashing function since all the hyperbolic capsules are constrained in the Poincaré ball.

4.2 Adaptive Routing

The large amount of labels in MLC is one major source of the computational complexity for the routing procedure. Since most of the labels are unrelated to a document, calculating the label-aware hyperbolic capsules for all the unrelated labels is redundant. Therefore, encoding based adaptive routing layer is used to efficiently decide the candidate labels for the document.

The adaptive routing layer produces the candidate probability of each label by

$$\boldsymbol{c} = \sigma(\boldsymbol{W}_c \frac{1}{T} \sum_{\boldsymbol{e}_i \in \boldsymbol{E}} \boldsymbol{e}_i + \boldsymbol{b}_c), \tag{16}$$

Table 1: Statistics of the datasets: N_{train} and N_{test} are the numbers of training and test instances, W_{train} and W_{test} are their average word numbers, L is the average label number per instance, I is the average number of training instances per label, $\#H$ and $\#T$ are the numbers of head and tail labels, H and T are their average number of training instances respectively.

Dataset	N_{train}	N_{test}	W_{train}	W_{test}	L	I	$\#H$	H	$\#T$	T
AAPD	49,356	6,484	163.34	164.14	2.41	2,199.03	17	5,002.23	37	911.08
RCV1	23,149	781,265	259.47	269.23	3.21	715.50	27	2,209.44	76	184.76
ZHIHU	2,699,969	299,997	38.14	35.56	2.32	3,165.92	442	7,144.31	1,557	2,036.54
EUR-LEX57K	51,000	6,000	726.46	725.37	5.06	53.45	711	273.72	3,560	9.46

Algorithm 1 Hyperbolic Dynamic Routing

procedure HDR($\hat{\boldsymbol{u}}_{j|i}$, r, ℓ)
  Initialize $\forall i, j : b_{ij} \leftarrow 0$
  for r iterations **do**
    for all capsule i in layer ℓ and capsule j in layer $\ell + 1$:
    $c_{ij} \leftarrow \text{softmax}(b_{ij})$ ▷ Eq. 12
    for all capsule j in layer $(\ell + 1)$:
    $\boldsymbol{v}_j \leftarrow \mathfrak{M}_i\, c_{ij} \otimes \hat{\boldsymbol{u}}_{j|i}$
    for all capsule i in layer ℓ and capsule j in layer $\ell + 1$:
    $b_{ij} \leftarrow b_{ij} + \mathfrak{K}(d_{\mathcal{B}}(\boldsymbol{v}_j, \hat{\boldsymbol{u}}_{j|i}))$
  return v_j

where σ denotes the *Sigmoid* function. $\boldsymbol{W}_c$ and the bias $\boldsymbol{b}_c$ are learnable parameters updated by minimizing the *binary cross-entropy* loss (Liu et al., 2017)

$$\mathcal{L}^c = -\sum_{j=1}^{Q} \big(y_j log(c_j) + (1 - y_j) log(1 - c_j)\big), \quad (17)$$

where $c_j \in [0, 1]$ is the j-th element in $\boldsymbol{c}$ and $y_j \in \{0, 1\}$ denotes the ground truth about label j. The adaptive routing layer selects the candidate labels during test. Label-aware hyperbolic capsules are then constructed via HDR to predict probabilities of these candidate labels.

During the training process, negative sampling is used to improve the the scalability of HYPERCAPS. Let $\mathcal{N}^+$ denote the true label set and $\mathcal{N}^-$ denote the set of randomly selected negative labels, the loss function is derived as

$$\mathcal{L}^f = -\big(\sum_{j \in \mathcal{N}^+} log(a_j) + \sum_{j \in \mathcal{N}^-} log(1 - a_j)\big), \quad (18)$$

where $a_j = \sigma(d_{\mathcal{B}}(\boldsymbol{v}_j, \boldsymbol{0}))$ is activations of the j-th label-aware capsules, which is proportional to the distance from the origin of the Poincaré ball.

5 Experiments

The proposed HYPERCAPS is evaluated on four benchmark datasets with various label number from 54 to 4271. We compare with the state-of-the-art methods in terms of widely used metrics. Performance on tail labels is also compared to demonstrate the superiority of HYPERCAPS for MLC. An ablation test is also carried out to analyse the contribution of each component of HYPERCAPS.

5.1 Experimental Setup

Datasets Experiments are carried out on four publicly available MLC datasets, including the small-scale AAPD (Yang et al., 2018b) and RCV1 (Lewis et al., 2004), the large-scale ZHIHU[1] and EUR-LEX57K (Chalkidis et al., 2019). Labels are divided into head and tail sets according to their number of training instances, *i.e.* labels have less than average number of training instances are divided into the tail label set. Their statistics can be found in Table 1.

Evaluation metrics We use the rank-based evaluation metrics which have been widely adopted for MLC tasks (Bhatia et al., 2015; Liu et al., 2017), *i.e. Precision@k* (*P@k* for short) and *nDCG@k*, which are respectively defined as

$$P@k = \frac{1}{k} \sum_{j \in rank_k(a)} y_j, \quad (19)$$

$$nDCG@k = \frac{\sum_{j \in rank_k(a)} y_j / log(j + 1)}{\sum_{j=1}^{\min(k, \|y\|_0)} 1 / log(j + 1)}, \quad (20)$$

where $y_j \in \{0, 1\}$ denotes the the ground truth about label j, $rank_k(a)$ denotes the indices of the candidate label-aware hyperbolic capsules with k largest activations in descending order, and $\|y\|_0$ is the true label number for the document instance.

[1] https://www.biendata.com/competition/zhihu/data/.

Table 2: Results on all the labels in *P@k* and *nDCG@k*, bold face indicates the best of each line.

Dataset	Metric	FastText	SLEEC	XML-CNN	SGM	RegGNN	NLP-Cap	HyperCaps
AAPD	*P@1*	75.33	75.85	76.31	77.90	79.92	81.75	**85.37**
	P@3	53.83	54.36	54.41	55.76	57.31	59.63	**61.89**
	P@5	37.57	37.89	37.83	38.58	39.50	41.97	**42.51**
	nDCG@3	71.22	71.54	72.12	73.73	75.77	78.40	**81.64**
	nDCG@5	75.78	75.98	76.39	78.05	80.03	83.70	**85.87**
RCV1	*P@1*	95.40	95.35	96.86	95.37	96.53	97.05	**97.10**
	P@3	79.96	79.51	81.11	81.36	81.69	81.27	**82.04**
	P@5	55.64	55.06	56.07	53.06	56.23	56.33	**57.06**
	nDCG@3	90.95	90.45	92.22	91.76	92.28	92.47	**93.03**
	nDCG@5	91.68	90.97	92.63	90.69	92.67	93.11	**93.66**
Zhihu	*P@1*	49.40	50.22	49.68	50.32	50.67	53.73	**56.50**
	P@3	31.50	32.21	32.27	31.83	32.43	33.83	**35.77**
	P@5	23.23	23.81	24.17	23.95	24.23	25.10	**26.27**
	nDCG@3	46.52	47.57	46.65	46.90	47.97	48.89	**50.61**
	nDCG@5	49.16	50.34	49.60	50.47	50.70	51.19	**52.89**
EUR-Lex57k	*P@1*	86.18	89.43	85.33	89.11	90.46	90.83	**91.42**
	P@3	73.18	76.73	74.40	78.03	79.29	80.72	**82.18**
	P@5	60.15	63.59	61.21	65.02	65.83	69.14	**70.53**
	nDCG@3	77.42	80.98	78.59	82.30	83.45	84.13	**86.05**
	nDCG@5	73.21	76.96	74.36	78.50	79.40	81.91	**83.28**

The final results are averaged over all the test instances.

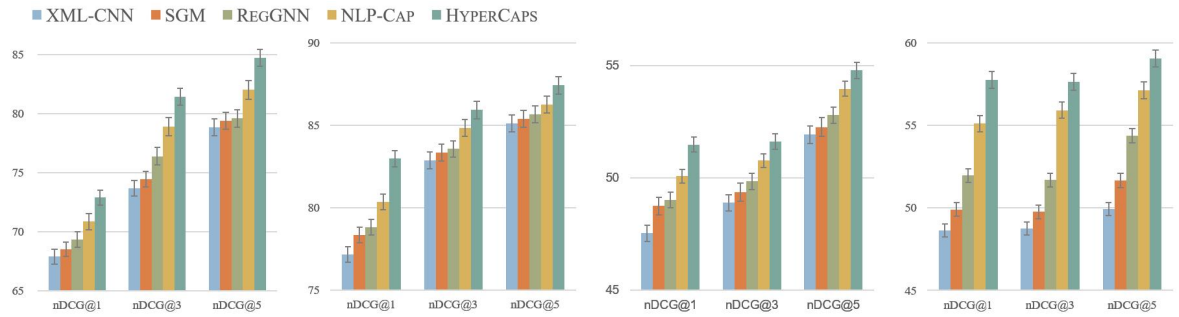
Baselines To demonstrate the effectiveness of HyperCaps on the benchmark datasets, six comparative text classification methods are chosen as the baselines. FastText (Joulin et al., 2017) is a representative encoding-based method which use average pooling to construct document representations and MLP to make the predictions. SLEEC (Bhatia et al., 2015) is a typical label-embedding method for MLC, which uses *k-nearest neighbors* search to predict the labels. XML-CNN (Liu et al., 2017) employs CNN as local n-gram feature extractors and a dynamic pooling technique as aggregation method. SGM (Yang et al., 2018b) applies the *seq2seq* model with attention mechanism, which takes the global contextual information. RegGNN (Xu et al., 2019) uses a combination of CNN and LSTM with a dynamic gate that controls the information from these two parts. NLP-Cap (Zhao et al., 2019) is a capsule-based approach for MLC, which reformulates the routing algorithm. NLP-Cap use only CNN to construct capsules, and it applies the squashing function onto capsules.

Implementation Details All the words are converted to lower case and padding is used to handle the various lengths of the text sequences. Maximum length of AAPD, RCV1 and EUR-Lex57k is set to 500, while maximum length of Zhihu is 50. To compose the word vector representations, pre-trained 300-dimensional Glove (Pennington et al., 2014) word embeddings are used for AAPD, RCV1 and EUR-Lex57k, while Zhihu uses its specified 256-dimensional word embeddings. The dimension of the Poincaré ball is set to 32 with a radius $1 - \epsilon$ ($\epsilon = 10^{-5}$) to avoid numerical errors. Multiple one-dimensional convolutional kernels (with window sizes of 2, 4, 8) are applied in the local hyperbolic capsule layer. The number of compressed local and global hyperbolic capsules is 128. Adaptive routing layer is not applied on the small-scale datasets AAPD and RCV1. The maximum candidate label number is set to 200 for the large-scale datasets Zhihu and EUR-Lex57k. For the baselines, hyperparameters recommended by their authors are adopted.

5.2 Experimental Results

The proposed HyperCaps is evaluated on the four benchmark datasets by comparing with the six baselines in terms of *P@k* and *nDCG@k* with $k = 1, 3, 5$. Results on all the labels averaged over the test instances are shown in Table 2. *nDCG@1* is omitted since it gives the same value as *P@1*. It is notable that HyperCaps obtains competitive results on the four datasets.

The encoding-based FastText is generally inferior to the other baselines as it applies the average pooling on word vector representations, which ig-



(a) AAPD (b) RCV1 (c) ZHIHU (d) EUR-LEX57K

Figure 4: Results on tail labels in *nDCG@k*.

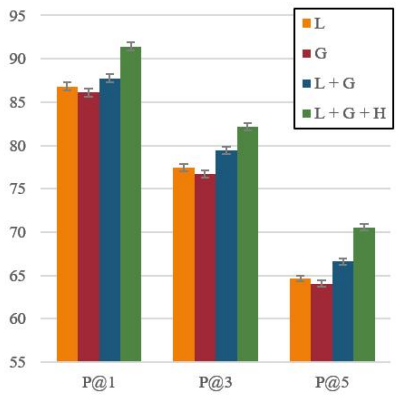


Figure 5: Results of ablation test on EUR-LEX57K in *P@k*. L denotes local capsules, G denotes global capsules, H denotes HDR.

nores word order for the construction of document representations. The typical MLC method SLEEC takes advantage of label correlations by embedding the label co-occurrence graph. However, SLEEC uses TF-IDF vectors to represent documents, thus word order is also ignored. XML-CNN uses a dynamic pooling technique to aggregate the local contextual features extracted by CNN, while SGM uses attention mechanism to aggregate the global contextual features extracted by LSTM. REGGNN is generally superior to both of them as it combines the local and global contextual information dynamically and takes label correlations into consideration using a regularized loss. However, the two capsule-based methods NLP-CAP and HYPERCAPS consistently outperform all the other methods owing to dynamic routing, which aggregates the fine-grained capsule features in a label-aware manner.

Moreover, NLP-CAP only uses CNN to extract the local contextual information, while HYPERCAPS benefits from the parallel combination of local and global contextual information. In addition, NLP-CAP applies the non-linear squashing function for capsules in the Euclidean space, while HDR is designed for hyperbolic capsules, which take advantage of the representation capacity of the hyperbolic space. Therefore, HYPERCAPS outperforms NLP-CAP as expected. This result further confirms that the proposed HYPERCAPS with HDR is effective to learn the label-aware hyperbolic capsules for MLC.

5.3 Performance on Tail Labels

In MLC, tail labels have low occurring frequency and hence are hard to predict compared to head labels. The performance on tail labels of the four benchmark datasets is evaluated in terms of *nDCG@k* with $k = 1, 3, 5$. Figure 4 shows the results of the five deep learning based MLC methods, *i.e.* XML-CNN, SGM, REGGNN, NLP-CAP and HYPERCAPS. *nDCG@1* is smaller than *nDCG@3* on AAPD, RCV1 and ZHIHU since most of their test instances contain less than three tail labels. It is remarkable that HYPERCAPS outperforms all the other methods on tail labels.

REGGNN takes advantage of the local and global contextual information and label correlations, thus it outperforms XML-CNN and SGM. The two capsule-based methods NLP-CAP and HYPERCAPS are both superior to the other methods, which indicates that the label-aware dynamic routing is effective for the prediction on tail labels. In addition, the fact that HYPERCAPS significantly improves the prediction performance compared to NLP-CAP implies that the representation capacity of the hyperbolic space and the combination of local and global contextual information are helpful for learning on tail labels. The results demonstrate the superiority of the proposed HYPERCAPS on tail labels for MLC.

5.4 Ablation Test

An ablation test would be informative to analyze the effect of varying different components of the proposed HYPERCAPS, which can be taken apart as local Euclidean capsules only (denoted as L), global Euclidean capsules only (denoted as G), a combination of the local and global Euclidean capsules (denoted as L + G), and a combination of the local and global hyperbolic capsules (denoted as L + G + H). Euclidean capsules (in L, G and L + G) are aggregated via the origin dynamic routing (Sabour et al., 2017), while hyperbolic capsules (in L + G + H) are aggregated via our HDR.

Figure 5 shows the results on EUR-LEX57K in terms of $P@k$ with $k = 1, 3, 5$. In order to make the comparison fair, the number of total compressed capsules is equally set to 256 for all the four models. Adaptive routing is also applied with the maximum candidate label number set equally to 200. Generally, the proposed combination of local and global contextual information contributes to the effectiveness of the model (L + G). Therefore, it is practical to combine the local and global contextual information via dynamic routing. HDR furthermore improves the performance by making use of the representation capacity of the hyperbolic space. Overall, each of the components benefits the performance of HYPERCAPS for MLC.

In summary, extensive experiments are carried out on four MLC benchmark datasets with various scales. The results demonstrate that the proposed HYPERCAPS can achieve competitive performance compared with the baselines. In particular, effectiveness of HYPERCAPS is shown on tail labels. The ablation test furthermore confirms that the combination of local and global contextual information is practical and HYPERCAPS benefits from the representation capacity of the hyperbolic space.

6 Related Work

6.1 Multi-Label Classification

Multi-label classification (MLC) aims at assigning multiple relevant labels to one document. The MLC label set is large compared to Multi-class classification (MCC). Besides, the correlations of labels (*e.g.* hierarchical label structures (Banerjee et al., 2019)) and the existence of tail labels make MLC a hard task (Bhatia et al., 2015).

As data sparsity and scalability issues arise with the large number of labels, XML-CNN (Liu et al., 2017) employs CNN as efficient feature extractor, whereas it ignores label correlations, which are often used to deal with tail labels. The traditional MLC method SLEEC (Bhatia et al., 2015) makes use of label correlations by embedding the label co-occurrence graph. The seq2seq model SGM (Yang et al., 2018b) uses the attention mechanism to consider the label correlations, while REGGNN (Xu et al., 2019) applies a regularized loss specified for label co-occurrence. REGGNN additionally chooses to dynamically combine the local and global contextual information to construct document representations.

6.2 Capsule Networks

Capsule networks are recently proposed to address the representation limitations of CNN and RNN. The concept of capsule is first introduced by (Hinton et al., 2011). (Sabour et al., 2017) replaces the scalar output features of CNN with vector capsules and pooling with dynamic routing. (Hinton et al., 2018) proposes the EM algorithm based routing procedure between capsule layers. (Gong et al., 2018) proposes to regard dynamic routing as an information aggregation procedure, which is more effective than pooling. (Yang et al., 2018a) and (Du et al., 2019a) investigate capsule networks for text classification. (Zhao et al., 2019) then presents a capsule compression method and reformulates the routing procedure to fit for MLC.

Our work is different from the predecessors as we design the Hyperbolic Dynamic Routing (HDR) to aggregate the parallel combination of local and global contextual information in form of hyperbolic capsules, which are constrained in the hyperbolic space without the requirement of non-linear squashing function. In addition, adaptive routing is proposed to improve the scalability for large number of labels.

6.3 Hyperbolic Deep Learning

Recent research on representation learning (Nickel and Kiela, 2017) indicates that hyperbolic space is superior to Euclidean space in terms of representation capacity, especially in low dimension. (Ganea et al., 2018b) generalizes operations for neural networks in the Poincaré ball using formalism of Möbius gyrovector space. Some works lately demonstrate the superiority of the hyperbolic space for serval natural language processing tasks, such as textual entailment (Ganea et al., 2018a), machine translation (Gulcehre et al., 2019) and word embedding (Tifrea et al., 2019). Our work presents

the Hyperbolic Capsule Networks (HYPERCAPS) for MLC.

7 Conclusion

We present the Hyperbolic Capsule Networks (HYPERCAPS) with Hyperbolic Dynamic Routing (HDR) and adaptive routing for Multi-Label Classification (MLC). The proposed HYPERCAPS takes advantage of the parallel combination of fine-grained local and global contextual information and label-aware feature aggregation method HDR to dynamically construct label-aware hyperbolic capsules for tail and head labels. Adaptive routing is additionally applied to improve the scalability of HYPERCAPS by controlling the number of capsules during the routing procedure. Extensive experiments are carried out on four benchmark datasets. Results compared with the state-of-the-art methods demonstrate the superiority of HYPERCAPS, especially on tail labels. As recent works explore the superiority of hyperbolic space to Euclidean space for serval natural language processing tasks, we intend to couple with the hyperbolic neural networks (Ganea et al., 2018b) and the hyperbolic word embedding method such as POINCARÉGLOVE (Tifrea et al., 2019) in the future.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61822601, 61773050, and 61632004; the Beijing Natural Science Foundation under Grant Z180006; National Key Research and Development Program (2017YFC1703506); the Fundamental Research Funds for the Central Universities (2019JBZ110). We thank the anonymous reviewers for their valuable feedback.

References

Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsioutsiouliklis. 2019. Hierarchical transfer learning for multi-label text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6295–6300.

Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. 2015. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems 28*, pages 730–738.

Ilias Chalkidis, Emmanouil Fergadiotis, Prodromos Malakasiotis, and Ion Androutsopoulos. 2019. Large-scale multi-label text classification on EU legislation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6314–6322.

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning*.

Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, Jianxin Liao, Chun Wang, and Bing Ma. 2019a. Investigating capsule network and semantic feature on hyperplanes for text classification. pages 456–465.

Cunxiao Du, Zhaozheng Chin, Fuli Feng, Lei Zhu, Tian Gan, and Liqiang Nie. 2019b. Explicit interaction model towards text classification. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pages 6359–6366.

Octavian Ganea, Gary Becigneul, and Thomas Hofmann. 2018a. Hyperbolic entailment cones for learning hierarchical embeddings. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1646–1655.

Octavian Ganea, Gary Becigneul, and Thomas Hofmann. 2018b. Hyperbolic neural networks. In *Advances in neural information processing systems 31*, pages 5345–5355.

Jingjing Gong, Xipeng Qiu, Shaojing Wang, and Xuanjing Huang. 2018. Information aggregation via dynamic routing for sequence encoding. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2742–2752.

Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. 2019. Hyperbolic attention networks. In *International Conference on Learning Representations*.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. 2011. Transforming auto-encoders. In *International Conference on Artificial Neural Networks*, pages 44–51. Springer.

Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. 2018. Matrix capsules with EM routing. In *International Conference on Learning Representations*.

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

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1746–1751.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1188–1196.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. 2004. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397.

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

Maximillian Nickel and Douwe Kiela. 2017. Poincaré embeddings for learning hierarchical representations. In *Advances in Neural Information Processing Systems 30*, pages 6338–6347.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems 30*, pages 3856–3866.

Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. 2019. Poincare glove: Hyperbolic word embeddings. In *International Conference on Learning Representations*.

Matt P Wand and M Chris Jones. 1994. *Kernel smoothing*. Chapman and Hall/CRC.

Xingyou Wang, Weijie Jiang, and Zhiyong Luo. 2016. Combination of convolutional and recurrent neural network for sentiment analysis of short texts. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 2428–2437.

Yunlai Xu, Xiangying Ran, Wei Sun, Xiangyang Luo, and Chongjun Wang. 2019. Gated neural network with regularized loss for multi-label text classification. In *2019 International Joint Conference on Neural Networks*, pages 1–8.

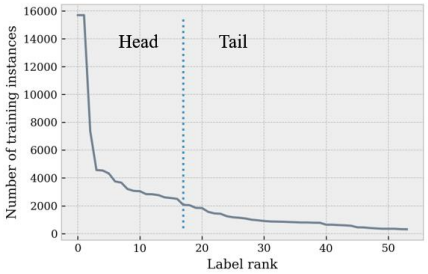
Min Yang, Wei Zhao, Jianbo Ye, Zeyang Lei, Zhou Zhao, and Soufei Zhang. 2018a. Investigating capsule networks with dynamic routing for text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3110–3119.

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

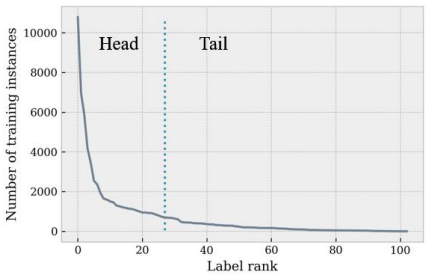
Wei Zhao, Haiyun Peng, Steffen Eger, Erik Cambria, and Min Yang. 2019. Towards scalable and reliable capsule networks for challenging NLP applications. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1549–1559.

Appendix

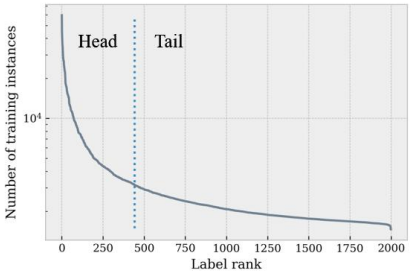
A Label Distributions



(a) AAPD



(b) RCV1



(c) ZHIHU

Figure 6: Label distributions of the other three benchmark datasets. Y-axes of ZHIHU is on log-scale

Figure 1 and Figure 6 show the label distributions of the four benchmark datasets. Head and tail labels are divided based on the average number of training instances (listed in Table 1), *i.e.* labels have less than average number of training instances are tail labels. We observe that this division generally follows the *Pareto Principle*, as nearly 80% of labels are divided into the tail label set.