第5章

基于不同表征网络集成的极端多标签学习

极端多标签学习相较于传统多标签学习的显著区别，在于其涉及的标签数量极为庞大，这一特性在极端多标签文本分类任务中极为凸显，特别是在如Wikipedia标注任务中，标签数量高达数百万。面对这一挑战，本章将聚焦极端多标签文本分类，主要面临的挑战包括：一是庞大的标签集合极大地增加了处理过程中的时间和空间开销，从而限制了传统多标签学习方法的有效应用；二是巨大的标签空间带来了数据稀疏和可扩展性问题，如何设计有效的网络结构，既能兼顾可扩展性又能提升预测性能显得尤为重要。本章基于深度网络强大的表征能力，研究基于不同表征网络集成的极端多标签文本分类：一是基于CNN和RNN不同表征能力，提出了自适应空时表征集成的HybridRCNN框架，该算法集成了词、短语、标签三者之间交互注意力，有效地提升了分类器对极端多标签的判别能力，但该方法仅能适应中间量级多标签文本分类（100～30000），并不能适应标签数量极端的学习任务，使得该算法仍然存在局限；二是本章利用了多种Transformer模型的独特表征能力，如BERT[141]、RoBERTa[142]、XLNet[143]等，提出了Multi-V-Transformer框架，该框架集成了多视图的Transformer表征。该算法通过高效地对海量标签进行聚类处理，有效缓解了由于标签数量庞大而引发的可扩展性问题，同时，借助多视图注意力表征机制、极端多标签聚类学习策略和简化的标签集嵌入学习技术，Multi-V-Transformer框架显著提升了模型在复杂场景下的泛化能力。此外，针对多样化的标签量级任务，本章所设计的HybridRCNN与Multi-V-Transformer算法能够形成互补优势，协同应用。实验结果显示在处理极端多标签文本分类任务时，HybridRCNN和Multi-V-Transformer均展现出了优异的性能，表明了这两种算法在该领域内的有效性和实用性。

5.1 引言

极端多标签学习旨在从巨大标签集中找出与问题最相关的标签子集来标记数据样本，如 Wikipedia 文本分类任务，有超过 100 万个标签，需要从这个巨大标签集中找出相关标签来标注新文章或者网页；然而，要同时处理大量的标签、维度和训练样本，使得极端多标签学习变得非常具有挑战性。与传统多标签学习任务相比，极端多标签学习需要解决两个问题：一是巨大的时间和空间开销；二是标签稀疏和可扩展性。为解决上述的问题，存在的极端多标签学习方法主要有四类：一是传统的 1-vs-All 方法。该方法是把多标签分类转化为多个二分类，该类方法并未考虑标签之间的相关性，并且当标签量大的时候难以训练与标签等量的模型。二是基于树的集成方法[144-145]。该方法与传统的决策树学习方法相似，将实例空间或子空间递归划分为树状结构，并在每个非叶子节点上建立基分类器，只关注该节点上的少数活动标签，代表性的算法如 FastXML[146]。三是基于嵌入的方法[147-148]。该方法旨在减少标签的有效数量，通过对较大的标签空间进行低秩假设，使其线性嵌入到低维标签向量中，从而使训练和预测过程变得容易，在预测阶段，通过将嵌入的标签向量映射到高维标签空间，代表性的算法如 SLEEC[43] 和 DXML[21]。四是基于深度学习的方法。该方法主要利用 CNN、RNN 等强大的深度网络表征能力来实现文本的分类，如 TextRNN[149]、AttentionXML[150]、DRNN[151]、Transformer[152] 等。

在深度文本分类任务中，注意力机制被广泛使用，如 Transformer[152]，主要通过探索词与词之间的关系来提升网络的性能，然而在多标签文本分类中，不仅需要考虑词与词之间的关系，还需要考虑短语与短语、词与标签、短语与标签之间的关系，因为有些标签就是一个短语，并且标签之间存在着较强的依赖关系，因此许多研究者致力于联合 CNN 和 RNN 强大的表征能力来探索词、短语、标签之间的关系，如 RCNN[153]、DRNN[151]、CRAN[154]、GRA[155]、LAHA[156]、AttConvNet[157] 等。然而，这些方法只是考虑词、短语、标签三者之间局部的关系，而没有全部考虑三者之间的关系，因此基于注意力机制，我们提出了自适应空时表征集成的 HybridRCNN 框架，同时性地考虑了词与词、短语与短语、词与标签、短语与标签之间的关系。

尽管 HybridRCNN 对不是非常极端的多标签文本分类任务取得不错的结果（通常标签数量为 100～30000），然而当标签量非常极端时，网络模型带来的时间和空间开销使得模型可扩展性差。为解决极端多标签问题，XML-CNN[36] 和 MACH[158] 通过基于嵌入的方式约简标签进行极端多标签学习，AttentionXML[150] 通过对标签集

进行聚类，然后使用注意力机制完成对极端多标签的学习。进一步，提出了许多基于阶段式的极端多标签学习方法，DeepXML[159]提出了一种四阶段的分解任务来解决极端多标签学习，X-Transformer[160]把极端多标签学习任务分解为标签聚类和标签排序两个阶段。这些方法存在两个缺点：一是都需要阶段式的训练，并未实现极端多标签的端到端学习；二是这些方法并未考虑标签之间的关系，也没有根据标签簇的学习得分进行排序来约简标签。因此，我们提出了一种改进的基于多视图 Transformer 表征集成的 Multi-V-Transformer，该算法不仅可以端到端地解决极端多标签的学习场景，而且通过多视图注意力表征、极端多标签聚类学习和约简的标签集嵌入学习来提升模型的泛化性能，有效地弥补了 HybridRCNN 使用局限。Multi-V-Transformer 与 HybridRCNN 存在的不同：一是表征网络不同，HybridRCNN 采用 CNN 和 RNN 异构网络集成，而 Multi-V-Transformer 采用 Transformer 同构网络集成；二是 HybridRCNN 探索词与标签、短语与标签关系通过不同注意力模块，而 Multi-V-Transformer 采用多视图注意力和关系增强模块；三是 HybridRCNN 不能适应极端标签量级，而 Multi-V-Transformer 通过聚类约简学习少量的模型参数以适应极端标签量级；四是 HybridRCNN 采用传统的二值交叉熵损失，而 Multi-V-Transformer 考虑了标签的不平衡，采用了不平衡 Focal 损失[161]进行训练。

综上所述，本章可以概括如下：

(1) 本章详尽阐述了 HybridRCNN 框架，该框架集成了自适应空时表征技术，同时考虑了词间、短语间、词与标签间，以及短语与标签间的多维度关联。HybridRCNN 框架通过实施一种高效的自适应加权集成策略，成功融合了卷积神经网络(CNN)与循环神经网络(RNN)各自的优势信息，从而显著增强了分类器的识别精度与性能。

(2) 我们提出集成 Transformer 多视图表征结构的 Multi-V-Transformer 框架，该算法通过聚类排序模块能有效适应极端标签量级分类任务，并且通过多视图注意力表征、极端多标签聚类学习和约简的标签集嵌入学习来提升模型的泛化性能。

(3) HybridRCNN 和 Multi-V-Transformer 可以互补使用，并且实验在大量的多标签文本分类任务上验证了提出方法的有效性。

5.2　问题描述

在多标签文本分类任务中，令 $D=\{(\boldsymbol{x}_1,\boldsymbol{y}_1),(\boldsymbol{x}_2,\boldsymbol{y}_2),\cdots,(\boldsymbol{x}_N,\boldsymbol{y}_N)\}$ 表示一个原生文档，其中 N 表示训练的文档数，且每个文档 $\boldsymbol{y}_i$ 有 k 个标签(k 的标签集有

上百万个),每个文档有 n 个词,并且每个词可以用 word2vec 技术表示为 d 维的词嵌入向量,即 $\boldsymbol{e}_t \in \mathbb{R}^d, t=\{1,2,\cdots,n\}, \boldsymbol{y}_i \in \{0,1\}^k$ 是对应文档 $\boldsymbol{x}_i=(\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_n)$ 的标签。如果 i-th 文档与 j-th 个标签相关,则 $y_{ij}=1$;否则,$y_{ij}=0$。我们的目标是学习一个函数 $f(\boldsymbol{x}_i) \in \mathbb{R}^k$ 给所有的标签打分,f 需要给标记为 $y_{il}=1$ 的 l 标签较高的分数,因此能通过 $f(\boldsymbol{x}_i)$ 获得一个 top-k 的预测标签集。

给定一个 d 维的词嵌入向量 $\boldsymbol{e}_t \in \mathbb{R}^d, t=\{1,2,\cdots,n\}$,输入 $\boldsymbol{x}_i=(\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_n)$ 可以表示为维度为 $d \times n$ 的一个特征图,在文本分类中可以通过使用 CNN 和 RNN 获取短语级的表征和词级的表征。

(1) 短语级表征 CNN。给定文档表示 $\boldsymbol{x}_i \in \mathbb{R}^{d \times n}$,应用卷积核 $\boldsymbol{W}_i \in \mathbb{R}^{\omega \times d}$ 和偏差项 $\boldsymbol{b}_i$ 学习 ω-grams 短语级的表征,令向量 $\boldsymbol{c}_i$ 表示词$(\boldsymbol{e}_{i-\omega+1},\cdots,\boldsymbol{e}_i)$的联合,则特征 $\boldsymbol{p}_i$ 表示为

$$\boldsymbol{p}_i=\sigma(\text{Conv1D}(\boldsymbol{W}_i,\boldsymbol{c}_i)+\boldsymbol{b}_i) \tag{5.1}$$

式中:σ 为激活函数;Conv1D(a,b)是 1 维卷积操作,a 为卷积核,b 为输入。$\boldsymbol{p}=[\boldsymbol{p}_1,\boldsymbol{p}_2,\cdots,\boldsymbol{p}_{n-\omega+1}] \in \mathbb{R}^{n-\omega+1}$ 能被产生通过每个词级窗口,最后通过 CNN 获得一个短语级的表征 $\boldsymbol{P} \in \mathbb{R}^{2r \times l}$,其中 $2r$ 为核数,l 为词序列的长度。

(2) 词级表征 RNN。给定一个文档 $\boldsymbol{x}_i \in \mathbb{R}^{d \times n}$,使用 Bi-GRU[162]模型学习双向的词级信息,Bi-GRU 的输出可以表示为

$$\boldsymbol{H}=[\boldsymbol{H}^f;\boldsymbol{H}^b]$$

式中

$$\boldsymbol{H}^f=(\overrightarrow{\boldsymbol{h}}_1,\cdots,\overrightarrow{\boldsymbol{h}}_n), \quad \boldsymbol{H}^b=(\overleftarrow{\boldsymbol{h}}_1,\cdots,\overleftarrow{\boldsymbol{h}}_n) \tag{5.2}$$

其中:$\overrightarrow{\boldsymbol{h}}_t \in \mathbb{R}^r$ 和 $\overleftarrow{\boldsymbol{h}}_t \in \mathbb{R}^r$ 分别表示一个 r 维的前向和后向词级表。整个输出 $\boldsymbol{H} \in \mathbb{R}^{2r \times n}$ 表示词级表征。下面分别介绍 HybridRCNN 和 Multi-V-Transformer 方法。

5.3 HybridRCNN 框架

如图 5.1 所示,我们提出的自适应空时表征集成的 HybridRCNN 网络结构分为两个分支,通过不同的注意力机制进行融合,最后输出文档的空时表征,整个网络是一个端到端的框架。下面分别介绍空间语义信息表征和时序语义信息表征两个分支。

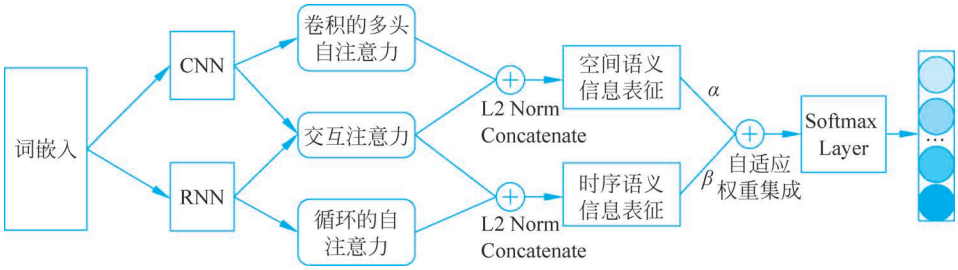


图 5.1　HybridRCNN 网络结构

5.3.1　空间语义信息表征

尽管式(5.1)可以表示短语级信息，但它只是简单地考虑输出结果的激活，而忽略语言关系和输出结果之间的细粒度信号。我们通过混合的注意力机制，包括多头的自注意力和交互注意力，最终得到更好的空间语义信息表征。该表征不仅考虑了短语与短语关系，还考虑了短语与标签之间的关系。

1. 卷积的多头自注意力

基于点积的注意力，如图 5.2(a)为卷积的多头自注意力的示意图，$\boldsymbol{Q}\in\mathbb{R}^{2r\times l}$、$\boldsymbol{K}\in\mathbb{R}^{2r\times l}$和$\boldsymbol{V}\in\mathbb{R}^{2r\times l}$分别表示 query、key、value 三个嵌入矩阵，注意力输出矩阵表示为

$$\text{Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{2r}}\right)\boldsymbol{V} \tag{5.3}$$

其中 query 根据相应的 key 计算权重指派值，权重矩阵被$\sqrt{2r}$尺度化，根据式(5.1)，矩阵$\boldsymbol{Q}\in\mathbb{R}^{2r\times l}$、$\boldsymbol{K}\in\mathbb{R}^{2r\times l}$和$\boldsymbol{V}\in\mathbb{R}^{2r\times l}$可计算如下：

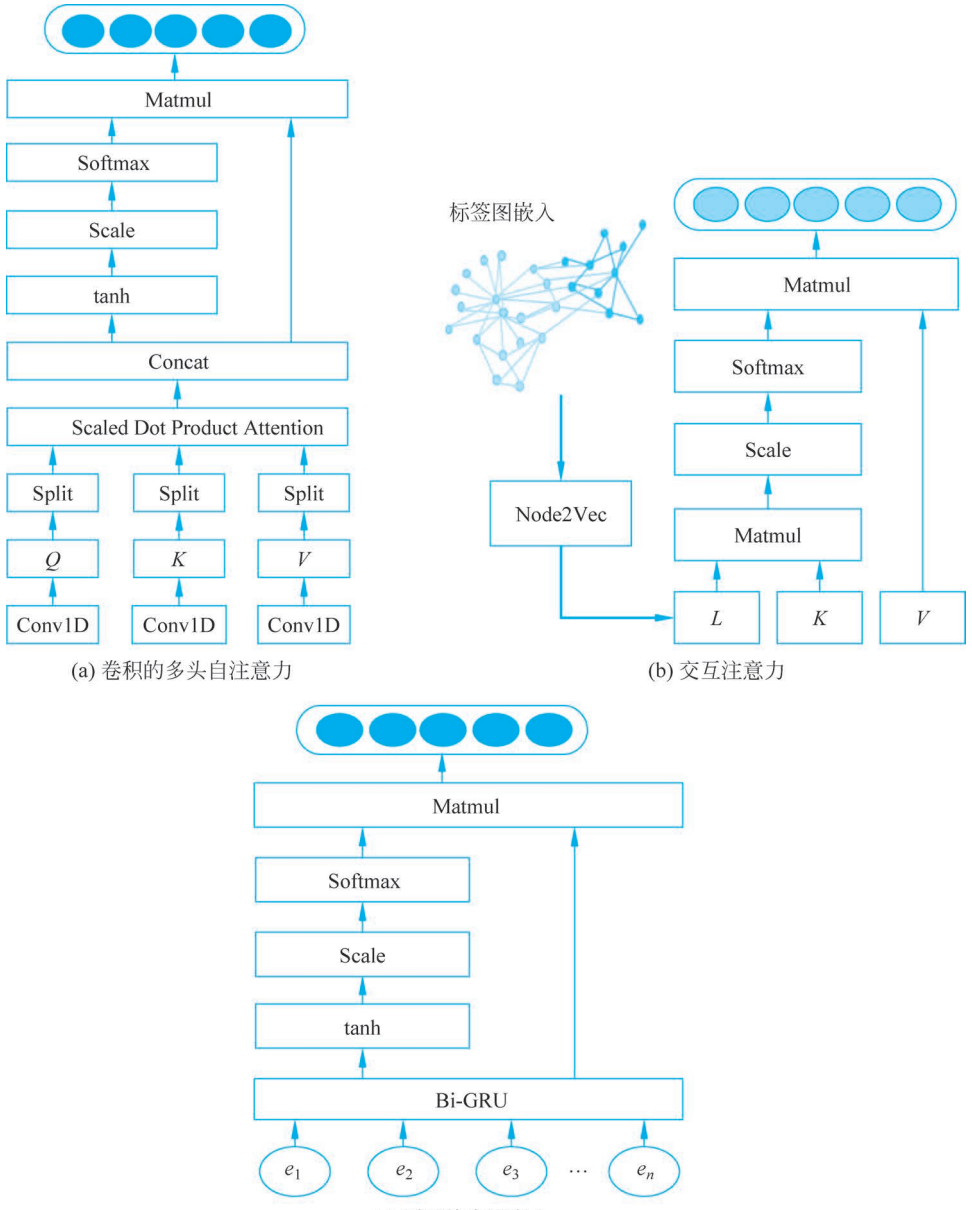
$$\begin{aligned}\boldsymbol{Q}&=\sigma(\text{Conv1D}(\boldsymbol{W}^{q},\boldsymbol{c})+\boldsymbol{b}^{q})\\ \boldsymbol{K}&=\sigma(\text{Conv1D}(\boldsymbol{W}^{k},\boldsymbol{c})+\boldsymbol{b}^{k})\\ \boldsymbol{V}&=\sigma(\text{Conv1D}(\boldsymbol{W}^{v},\boldsymbol{c})+\boldsymbol{b}^{v})\end{aligned} \tag{5.4}$$

其中激活函数σ设置为 ELU[163]，基于 Transformer，多头注意力能得到比单头注意力更好的结构，因此使用多头注意力表达不同部分的信息：

$$\begin{aligned}&\boldsymbol{P}=\text{Muti-head Attention}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{Concat}(\text{head}_1,\text{head}_2,\cdots,\text{head}_h)\\ &\text{where head}_i=\text{Attention}(\boldsymbol{Q}_i,\boldsymbol{K}_i,\boldsymbol{V}_i)\end{aligned} \tag{5.5}$$

其中联合输出$\boldsymbol{P}\in\mathbb{R}^{2r\times l}$通过$h$个并行的注意力层扩展了单头注意力的能力。在多标签文本分类任务中，由于每个文档能被指派到多个标签，因此使用多标签注意力机制来聚焦不同的标签关系，基于联合矩阵$\boldsymbol{P}\in\mathbb{R}^{2r\times l}$，最后输出多标签注意力$\boldsymbol{S}_j(j=1,2,\cdots,k)$表示为

$$\boldsymbol{S}_j=\sum_{i=1}^{n}\alpha_{ij}\boldsymbol{P}_i,\quad \boldsymbol{T}_i=\tanh(\boldsymbol{P}_i\boldsymbol{W}_j^{(1)}),\quad \alpha_{ij}=\text{softmax}(\boldsymbol{T}_i\boldsymbol{W}_j^{(2)}) \tag{5.6}$$



(a) 卷积的多头自注意力　　(b) 交互注意力

(c) 循环的自注意力

图 5.2　HybridRCNN 子模块结构图

式中：$\boldsymbol{W}_j^{(1)} \in \mathbb{R}^{2r}$ 和 $\boldsymbol{W}_j^{(2)} \in \mathbb{R}^{2r}$ 为需要学习的参数；α_{ij} 为标准化的第 j 个标签的权重，整个的 $\boldsymbol{S} \in \mathbb{R}^{2r \times k}$ 是在卷积多头自注意力下的空间语义信息表征。

2. 交互注意力

为了充分利用标签的关系信息，可以通过交互注意力捕捉标签和短语之间的

细粒度交互信息，如图 5.2(b)所示。使用标签共现图探索标签的结构信息，即每个标签可以看作一个节点，若任意两个标签共同出现在一个文档中，则标签之间有边相连。基于随机游走，使用 Node2Vec[164] 图嵌入技术捕捉高阶的标签依赖关系，每个标签能被表达为 $2r$ 维向量，即 $\boldsymbol{L}_j \in \mathbb{R}^{2r}(j=1,2,\cdots,k)$ 表示第 i 个标签，因此整个标签嵌入表示为 $\boldsymbol{L} \in \mathbb{R}^{k\times 2r}$。基于矩阵 $\boldsymbol{K} \in \mathbb{R}^{2r\times l}$ 和 $\boldsymbol{V} \in \mathbb{R}^{2r\times l}$，交互注意力下的空间语义信息表征表示为

$$\boldsymbol{I}_1 = \boldsymbol{V} \times \text{softmax}(\boldsymbol{LK})^{\mathrm{T}} \tag{5.7}$$

式中：$\boldsymbol{K} \in \mathbb{R}^{2r\times l}$ 和 $\boldsymbol{V} \in \mathbb{R}^{2r\times l}$ 根据式(5.4)能得到；交互矩阵 $\boldsymbol{LK}$ 表达了标签嵌入和短语表征之间的交互信息。

基于 CNN 结构，可得到矩阵 $\boldsymbol{S} \in \mathbb{R}^{2r\times k}$ 和 $\boldsymbol{I}_1 \in \mathbb{R}^{2r\times k}$，$\boldsymbol{S} \in \mathbb{R}^{2r\times k}$ 聚焦于短语语义，$\boldsymbol{I}_1 \in \mathbb{R}^{2r\times k}$ 聚焦于标签关系语义，最后空间的语义信息表征通过联合可表示为

$$\boldsymbol{C} = \text{Concat}(\boldsymbol{S}, \boldsymbol{I}_1) \tag{5.8}$$

5.3.2　时序语义信息表征

尽管式(5.2)在文本分类任务中取得了很大的成果，但是它自然地忽略了细粒度的词级线索(因为一个文档中的单词对不同的标签有不同的贡献)，因此使用混合的注意力机制捕捉时序的语义信息表征，包括循环的自注意力和交互注意力。

1. 循环的自注意力

为了更好地建模上下文词级依赖关系，使用加权自注意机制来关注文档的不同方面，这不仅可以学习长期的时间依赖性，还可以捕获文档的各种密集部分，如图 5.2(c)所示。类似于式(5.6)，循环的自注意力 $\boldsymbol{U} \in \mathbb{R}^{2r\times k}$ 可描述为

$$\boldsymbol{T} = \tanh(\boldsymbol{W}_1 \boldsymbol{H}), \quad \boldsymbol{A} = \text{softmax}(\boldsymbol{W}_2 \boldsymbol{T})^{\mathrm{T}}, \quad \boldsymbol{U} = \boldsymbol{HA} \tag{5.9}$$

式中：$\boldsymbol{A} \in \mathbb{R}^{n\times k}$ 为注意力得分矩阵；$\boldsymbol{W}_1 \in \mathbb{R}^{d\times 2r}$ 和 $\boldsymbol{W}_2 \in \mathbb{R}^{k\times d}$ 为可学习的参数；$\boldsymbol{U} \in \mathbb{R}^{2r\times k}$ 为循环自注意力下的时序文档表示。

2. 交互注意力

与 CNN 中的交互注意力类似，引入交互注意力来捕获细粒度的词级信号，计算单词和标签之间的匹配分数，根据式(5.7)，基于 RNN 的交互注意力可描述为

$$\boldsymbol{I}_2 = \boldsymbol{H} \times \text{softmax}\left([\boldsymbol{L}, \boldsymbol{L}] \begin{bmatrix} \boldsymbol{H}^f \\ \boldsymbol{H}^b \end{bmatrix}\right)^{\mathrm{T}} \tag{5.10}$$

式中：$\boldsymbol{I}_2 \in \mathbb{R}^{2r\times k}$ 通过联合矩阵 $\boldsymbol{H}$ 可以计算得到，表示交互注意力下的时序语义信息。

基于 RNN 结构，可得到矩阵 $\boldsymbol{U} \in \mathbb{R}^{2r\times k}$ 和 $\boldsymbol{I}_2 \in \mathbb{R}^{2r\times k}$，最后空间的语义信息表征通过联合可表示为

$$\boldsymbol{R} = \text{Concat}(\boldsymbol{U}, \boldsymbol{I}_2) \tag{5.11}$$

5.3.3 自适应权重集成预测

基于集成学习思想，我们设计了一种自适应加权集成策略，自然地集成两种互补信息以实现最终的空时文档表征。当 $\boldsymbol{C}\in\mathbb{R}^{2r\times k}$ 和 $\boldsymbol{R}\in\mathbb{R}^{2r\times k}$ 获得之后，首先使用 l_2 标准化它们，然后通过一个 MLP 层和全连接层转换 $\boldsymbol{C}\in\mathbb{R}^{2r\times k}$ 和 $\boldsymbol{R}\in\mathbb{R}^{2r\times k}$ 到权重 $\alpha\in\mathbb{R}^{k\times 1}$ 和 $\beta\in\mathbb{R}^{k\times 1}$：

$$\begin{cases}\alpha=\sigma(\boldsymbol{W}_1^{\alpha}\tanh(\boldsymbol{W}_2^{\alpha}\boldsymbol{C}+\boldsymbol{b}^{\alpha}))\\ \beta=\sigma(\boldsymbol{W}_1^{\beta}\tanh(\boldsymbol{W}_2^{\beta}\boldsymbol{R}+\boldsymbol{b}^{\beta}))\end{cases}\tag{5.12}$$

式中：$\boldsymbol{W}_1^{\alpha}$、$\boldsymbol{W}_2^{\alpha}$、$\boldsymbol{W}_1^{\beta}$ 和 $\boldsymbol{W}_2^{\beta}$ 为可学习的参数；$\boldsymbol{b}^{\alpha}$ 和 $\boldsymbol{b}^{\beta}$ 为偏置项。

标准化权重获得最后的空时文档表征：

$$\alpha=\frac{\alpha}{\alpha+\beta},\quad \beta=\frac{\beta}{\alpha+\beta}$$
$$\boldsymbol{T}=\alpha\times\boldsymbol{C}+\beta\times\boldsymbol{R}\tag{5.13}$$

式中：$\boldsymbol{T}\in\mathbb{R}^{2r\times k}$ 表示最后的空时语义表征，通过权重 α 和 β 不仅可以表达空间语义信息和时间语义信息表征的重要性，而且大大地拓宽了传统 CNN 和 RNN 表征范围的限制。当得到 $\boldsymbol{T}\in\mathbb{R}^{2r\times k}$ 之后，可以全连接层建立分类器，获得预测：

$$\hat{\boldsymbol{Y}}=\sigma(\boldsymbol{W}_1^{Y}\mathrm{relu}(\boldsymbol{W}_2^{Y}\boldsymbol{T}))\tag{5.14}$$

式中：$\boldsymbol{W}_1^{Y}\in\mathbb{R}^{1\times r}$ 和 $\boldsymbol{W}_2^{Y}\in\mathbb{R}^{r\times 2r}$ 为预测层的参数；σ 为 sigmoid 函数。

使用二值交叉熵损失为多标签文本分类：

$$\mathcal{L}_{\mathrm{loss}}=-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k}[y_{ij}\log(\hat{Y}_{ij})+(1-y_{ij})\log(1-\hat{Y}_{ij})]\tag{5.15}$$

在 5.5 节，HybridRCNN 大量的实验表明了 HybridRCNN 具有较好性能。但是，当标签量达极端量级时，由于标签空间的增加，该模型的可扩展性存在局限。我们提出改进的 Multi-V-Transformer 框架来弥补可扩展性的问题。

5.4 改进的 Multi-V-Transformer 框架

近年来，Transformer 已经受到了广泛的关注，不管是在文本领域还是图像领域，Transformer 都取得了比 CNN 和 RNN 好的结果，因此使用不同 Transformer 表征结构代替 CNN 和 RNN 来提取文档表征，如 BERT[141]、RoBERTa[142]、XLNet[143]等；另外，当应用到图像领域的多标签学习任务时，只需要把特征提取器变换为视觉 Transformer 即可，如 Vision Transformer[165]、DETR[166]、Image GPT[167]等。在极端多标签学习任务中，HybridRCNN 仅能适应标签量在万级的，并不能适应标签量在百万级的。因此，我们提出 Multi-V-Transformer 框架来解决这种过度极端的多标签学习任务，如图 5.3 所示。

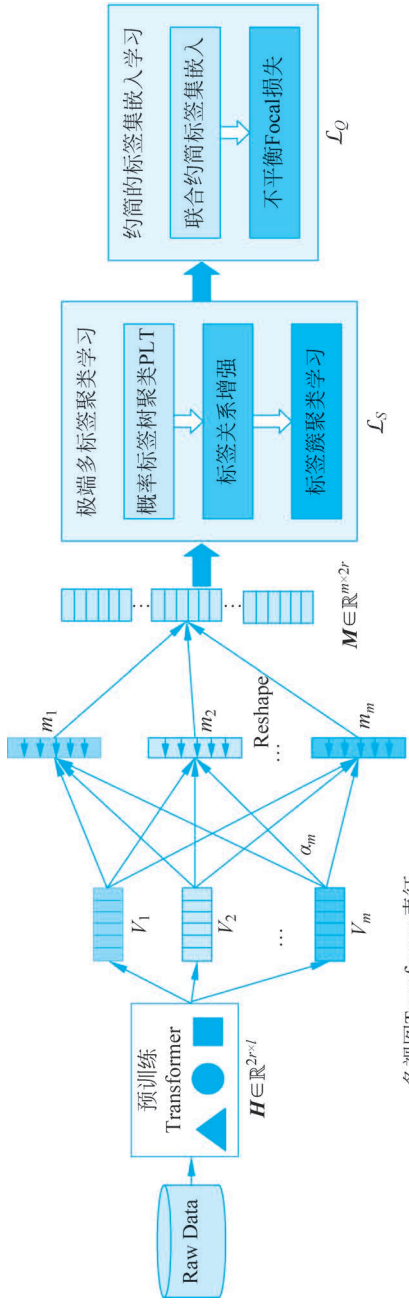


图 5.3　Multi-V-Transformer 网络结构

5.4.1 多视图注意力 Transformer 表征

在 NLP 任务中，Transformer 模型具有较好的表征性能，为了适应标签到百万的量，没有使用较大的 Transformer 模型(24 层，且 1024 的隐藏维度)，仅使用基本的 Transformer 模型(12 层，且 768 的隐藏维度)，即 $r=768$，输入序列长度 l 设置为 128，为了更好地表达富的文本信息，联合最后输出的 2 层 Transformer，即 Transformer 输出矩阵 $\boldsymbol{H}\in\mathbb{R}^{2r\times l}$。

通常情况下，在极端多标签文本分类中，标签信息源自不同的分析视角，因此使用多视角注意力提取文本表征，即每个视图表征文本的一个特定领域，描述如下：

$$\boldsymbol{M}=\sum_{t=1}^{T}\alpha_m\boldsymbol{H}^{\mathrm{T}},\quad \alpha_m=\mathrm{softmax}(V_m\boldsymbol{H}^{\mathrm{T}})=\frac{\exp(V_m h_t)}{\sum_{t=1}^{T}\exp(V_m h_t)} \tag{5.16}$$

式中：$\boldsymbol{M}\in\mathbb{R}^{m\times 2r}$ 表示多视图 Transformer 表征；V_m 表示第 m 个视图，在实验中设置 $m=3$，也就是说从 3 个视角提取文本信息。

5.4.2 极端多标签聚类学习

在极端多标签文本分类中，标签较为稀疏，如果完全按照传统正、负样本训练方式，将带来很大的时间和空间开销，导致模型可扩展性差，因此需要使用合适的方式对标签集进行约简，以满足实际需要。如图 5.3 所示，通过对标签集进行聚类来约简标签，多标签聚类学习模块分为三个步骤：一是概率标签树聚类；二是标签关系增强；三是标签簇聚类学习。

1. 概率标签树聚类

首先将包含有标签的稀疏文本特征和该标签文本特征进行内积求和，然后标准化得到每个标签的特征表示，再基于 AttentionXML[150] 算法中的概率标签树(PLT)[140-150]，使用平衡 k-均值($k=2$)进行递归的聚类，直到满足条件：给定每个簇的最大标签量，要求将标签划分到 S 个簇中，每个标签簇中包含的标签量满足小于最大标签量或者大于最大标签量的一半。得到 S 个簇时，基于式(5.16)得到的表征 $\boldsymbol{M}\in\mathbb{R}^{m\times 2r}$，可以通过全连接层映射 $\boldsymbol{M}$ 到 S 维的向量 $\boldsymbol{P}$：

$$\boldsymbol{P}=\sigma(\boldsymbol{W}_p\boldsymbol{M}+\boldsymbol{b}_p) \tag{5.17}$$

式中：$\boldsymbol{P}$ 返回一个 S 维的向量表征，表示 S 个标签簇的得分；$\boldsymbol{W}_p$、$\boldsymbol{b}_p$ 为可学习参数；$\sigma(\cdot)$为 sigmod 函数。

2. 标签关系增强

在多标签分类中，标签之间存在着较强的依赖关系，HybridRCNN 框架通过

探索混合的词、短语和标签之间的依赖关系来提升模型的性能，然而式(5.17)忽视了不同簇之间、标签之间的关系，因此传达标签关系通过在原生的预测 $\boldsymbol{P}$ 基础上增加 bottleneck 层来实现标签增强，如图 5.4 所示。

$$\begin{cases}\hat{\boldsymbol{P}}=F(\boldsymbol{P})+\boldsymbol{P}\\ F(\boldsymbol{P})=\boldsymbol{W}_2\delta(\boldsymbol{W}_1\sigma(\boldsymbol{P})+\boldsymbol{b}_1)+\boldsymbol{b}_2\end{cases}\tag{5.18}$$

式中：$\boldsymbol{W}_1$、$\boldsymbol{W}_2$ 为权重矩阵；$\boldsymbol{b}_1$、$\boldsymbol{b}_2$ 为偏置项；σ、δ 分别为 sigmoid 和 ELU 函数。

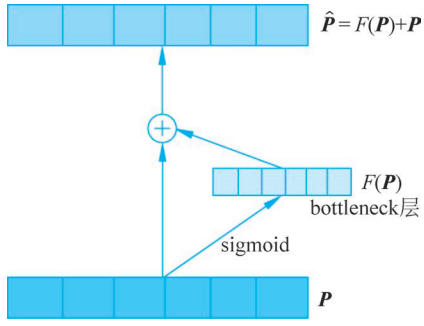


图 5.4　标签关系增强

3. 标签簇聚类学习

为了更好地学习标签表征，基于聚类得到的 S 个簇索引，构造簇标签 $\boldsymbol{y}^S\in\{0,1\}^S$ 为二值 one-hot 编码，基于二值交叉熵对学习聚类簇表征：

$$\mathcal{L}_S=-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{S}[y_{ij}^S\log(\hat{P}_{ij})+(1-y_{ij}^S)\log(1-\hat{P}_{ij})]\tag{5.19}$$

式中：y_{ij}^S 为第 i 个样本属于第 j 个簇；$\hat{P}_{ij}$ 为第 i 个样本属于第 j 个簇的增强预测，可以由式(5.18)得到。

基于式(5.19)训练，选取前 k 个簇对应标签作为标签的约简集，记为 $U=\{l:\ l\in S\}$，即标签 l 属于簇 S，这样大大地约简了原来数百万的标签。

5.4.3　约简的标签集嵌入学习

当得到前 k 个簇之后，基于 k 个簇所含标签得到标签的约简集 U，然后找到这些标签真实所对应的标签 $\boldsymbol{y}^U\in\{0,1\}^U$。

1. 联合约简标签集嵌入

当得到标签集 U 后，基于多视图表征 $\boldsymbol{M}$ 可以得到联合约简标签集嵌入向量 $\boldsymbol{Q}$：

$$\boldsymbol{Q}=\sigma(\boldsymbol{W}_Q\boldsymbol{M}+\boldsymbol{b}_Q)\tag{5.20}$$

式中：$\boldsymbol{W}_Q$、$\boldsymbol{b}_Q$ 为可学习参数。

2. 不平衡 Focal 损失

尽管约简的标签集 U 已经大大地缩小了训练的标签数量，但是正、负样本之间仍然存在着大的不平衡。Focal 损失[161]基于二值交叉熵已经被广泛使用，其旨在降低简单负样本权重让模型重点关注更难分的样本，然而 Focal 损失在二值交叉熵基础上使用相同的参数 γ。而在多标签问题中，正、负样本之间存在极度的不平衡，使用不平衡 Focal 损失对约简的标签集进行学习[168]：

$$\mathcal{L}_Q=-\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{U}\begin{cases}(1-Q_k)^{\gamma+}\log(Q_k), & y_k^U=1\\ (Q_k)^{\gamma-}\log(1-Q_k), & y_k^U=0\end{cases} \tag{5.21}$$

式中：y_k^U 为约简标签集 U 中样本对应的真实标签；Q_k 为使用式(5.20)得到的预测；γ^+、γ^- 表达了不同正、负样本权重的贡献，通常情况下，$\gamma^->\gamma^+$，我们的实验设置 $\gamma^+=0$，$\gamma^-=1$。

5.4.4 集成的 Multi-V-Transformer 预测

在 Multi-V-Transformer 中，使用端到端的训练方式联合损失 $\mathcal{L}_S$ 和 $\mathcal{L}_Q$，从而训练损失如下：

$$\mathcal{L}=\mathcal{L}_S+\mathcal{L}_Q \tag{5.22}$$

为了提高预测精度，使用集成学习思想，根据不同的预训练模型使用多数投票的集成策略进行模型最终的预测。我们的实验选择的预训练模型为 BERT[141]、RoBERTa[142]、XLNet[143]。

5.5 中间量级多标签文本实验分析

选择中间量级(100～30000)多标签文本分类数据集，验证我们提出的 HybridRCNN 方法的有效性，HybridRCNN 采用并行混合注意力机制的方式集成了 CNN 和 RNN 结构，因此比较 HybridRCNN 和相关的 CNN-RNN 网络结构，如串行结构 RCNN[153]、DRNN[151]，并行结构 CRAN[154]，混合结构 GRA[155]。此外，我们的方法也和使用基于注意力机制网络结构比较，如 TextCNN[169]、TextRNN[149]、DPCNN[170]、Transformer[152]、AttConvNet[157] 和 LAHA[156] 等。

5.5.1 实验设置

1. 实验数据集

我们采用了 5 个基准数据集来全面验证 HybridRCNN 方法的有效性和性能。

这些数据集的详细信息如表 5.1 所示，其中包含了训练样本数与测试样本数、特征总数、总的标签数、每个文档平均对应的标签数，以及每个标签平均对应的文档数。这些指标全面揭示了数据集规模、特征丰富度及标签分布情况。

表 5.1　中间量级多标签数据集详细信息

Datasets	N_{trn}	N_{tst}	D	L	$\tilde{L}$	$\hat{L}$
Rcv1	23149	7965	47236	102	3.18	649.85
Ydata	29999	18968	146248	414	2.39	86.85
Yelp	196507	33620	744607	508	2.96	810.73
Eurlex_4k	15449	3865	186104	3956	5.30	20.79
Wiki10_31k	14146	6616	101938	30938	18.64	8.52

2. 参数设置

在 HybridRCNN 方法中，使用 Node2Vec 技术映射每个标签到一个低维的密集型向量，标签嵌入维度设置为 128，多头数设置为 5，为自注意力机制，注意力维度设置为 16，整个深度学习模型使用 Adam 训练，初始学习率设置为 0.008，batch 大小设置为 64。采用 Glove[171]（300 维度）作为词嵌入向量，Bi-GRU 隐藏层维度设置为 64，CNN 的卷积核数设置为 128。

5.5.2　CNN-RNN 集成结构比较

HybridRCNN 方法使用并行的 CNN-RNN 结构，因此比较 HybridRCNN 和相关的 CNN-RNN 网络集成结构，如串行结构 RCNN[153]、DRNN[151]，并行结构 CRAN[154]，混合结构 GRA[155]。与第 3 章评估方法一样，使用评估指标如 $p@k\{1,3,5\}$ 和 $\text{ndcg}@k\{3,5\}$，详细实验结果如表 5.2 所示。

通过实验结果可知：

（1）与串行 CNN-RNN 结构（RCNN 和 DRNN）相比，HybridRCNN 的性能优于 RCNN 和 DRNN。原因是 RCNN 和 DRNN 只考虑文本中的长期依赖关系和局部信息，忽略了标签语义结构信息。我们的模型不仅利用自注意机制获取长期依赖关系和局部信息，还利用交互机制获取标签语义结构信息。

（2）与并行 CNN-RNN 结构（CRAN）相比，HybridRCNN 的性能优于 CRAN。原因是 CRAN 通过自注意机制简单地结合 CNN 和 RNN 来学习文档表示，而我们的模型采用加权集成注意力融合策略来学习更深层次的表示。

表 5.2 CNN-RNN 集成结构实验比较

Datasets	Metrics	Methods				
		RCNN	GRA	DRNN	CRAN	HybridRCNN
Rcv1	p@1	94.64	94.79	88.37	92.08	**94.89**
	p@3	75.18	**77.36**	62.33	72.59	75.53
	p@5	52.03	**53.58**	44.50	50.43	52.66
	ndcg@3	86.63	**88.60**	73.99	83.86	87.00
	ndcg@3	86.63	**88.60**	73.99	83.86	87.00
Ydata	p@1	48.39	50.34	28.38	34.10	**54.04**
	p@3	39.15	39.89	24.39	24.50	**40.19**
	p@5	30.11	30.78	18.90	19.33	**31.34**
	ndcg@3	44.88	46.63	26.14	30.34	**49.66**
	ndcg@5	46.92	47.87	27.43	32.28	**52.29**
Yelp	p@1	84.98	85.01	81.43	80.21	**86.70**
	p@3	51.93	52.83	48.76	50.95	**57.29**
	p@5	36.55	37.03	34.66	36.04	**40.73**
	ndcg@3	67.59	68.84	63.71	65.62	**73.10**
	p@5	68.09	69.93	64.46	66.38	**74.18**
Eurlex_4k	p@1	73.68	74.48	19.73	71.81	**75.79**
	p@3	57.57	58.14	15.97	55.96	**59.66**
	p@5	46.65	47.63	13.59	45.38	**48.54**
	ndcg@3	61.55	62.74	16.85	59.9	**63.62**
	ndcg@5	55.43	56.76	15.26	53.94	**57.38**
Wiki10_31k	p@1	80.37	**81.47**	—	65.96	**81.26**
	p@3	50.69	51.89	—	38.58	**59.47**
	p@5	37.26	39.78	—	29.88	**49.23**
	ndcg@3	57.36	58.34	—	44.38	**64.22**
	ndcg@5	46.25	48.19	—	36.76	**55.86**

(3) 与混合 CNN-RNN 结构(GRA)相比,当标签数量增加时,HybridRCNN 表现得更好。原因是 GRA 采用软对齐机制对短语和词序列之间的关系进行建模,而我们的模型采用交互机制对短语、词序列和标签图结构之间的关系进行建模,特别地,利用标签交互信息是多标签文本分类问题的关键。

5.5.3　注意力机制网络结构比较

由于我们提出的 HybridRCNN 方法采用混合注意力机制,因此实验也和使用注意力机制的模型进行比较,以验证混合注意力机制的有效性。比较的基准方法有 TextCNN[169]、TextRNN[149]、DPCNN[170]、Transformer[152]、AttConvNet[157] 和 LAHA[156],实验结果如表 5.3 所示。

表 5.3　注意力机制网络结构比较实验结果

Datasets	Metrics	Methods						
		TextCNN	TextRNN	DPCNN	Transformer	AttConvNet	LAHA	HybridRCNN
Rcv1	p@1	90.90	93.13	90.58	93.62	91.84	93.22	**94.89**
	p@3	68.84	73.99	67.81	70.43	69.07	74.52	**75.53**
	p@5	47.85	52.31	47.70	49.20	47.97	52.04	**52.66**
	ndcg@3	80.43	86.32	79.02	82.33	80.61	85.63	**87.00**
	ndcg@5	81.19	87.81	80.38	83.43	81.25	86.58	**87.85**
Ydata	p@1	35.07	53.26	37.86	44.29	50.40	45.88	**54.04**
	p@3	25.17	37.88	31.62	37.27	39.96	35.18	**40.19**
	p@5	19.84	30.57	24.75	29.17	30.97	28.06	**31.34**
	ndcg@3	31.24	47.58	35.12	42.11	46.36	42.90	**49.66**
	ndcg@5	33.07	49.88	37.09	44.68	48.59	45.86	**52.29**
Yelp	p@1	81.91	86.66	80.32	77.17	86.36	80.90	**86.70**
	p@3	50.65	58.32	47.11	44.10	**59.22**	50.18	57.29
	p@5	35.75	41.16	33.25	31.23	**41.93**	35.78	40.73
	ndcg@3	65.73	74.06	62.21	58.35	**74.80**	65.02	73.10
	ndcg@5	66.28	74.86	62.84	58.82	**75.72**	65.98	74.18

续表

Datasets	Metrics	Methods						
		TextCNN	TextRNN	DPCNN	Transformer	AttConvNet	LAHA	HybridRCNN
Eurlex_4k	p@1	67.95	68.24	47.03	39.11	52.59	62.86	**75.79**
	p@3	54.02	53.13	35.12	28.98	41.14	49.23	**59.66**
	p@5	43.48	43	28.54	24.48	33.51	40.38	**48.54**
	ndcg@3	57.57	56.88	37.97	31.42	43.95	52.6	**63.62**
	ndcg@5	51.62	51.06	33.94	28.72	39.48	47.61	**57.38**
Wiki10_31k	p@1	79.61	79.17	—	80.48	—	80.02	**81.26**
	p@3	49.52	60.89	—	**61.59**	—	57.83	59.47
	p@5	36.31	49.66	—	**50.75**	—	47.54	49.23
	ndcg@3	56.08	65.04	—	**65.86**	—	62.63	64.22
	ndcg@5	45.10	56.16	—	**57.23**	—	54.24	55.86

根据表 5.3 可得到如下结果：

(1) 与 TextCNN 和 TextRNN 相比，在大多数情况下，HybridRCNN 的性能优于 TextCNN 和 TextRNN。原因是 HybridRCNN 集成 CNN 和 RNN 网络并全面捕获语义空间和时序信息，这两者对于有效的多标签文本分类是必不可少的。

(2) 与不同注意机制模型比较，如 TextCNN 使用 pooling attention 机制、AttConvNet 使用注意力卷积机制、TextRNN 使用自注意力机制、LAHA 使用混合注意力机制、Transformer 使用多头自注意机制，在更多情况下，HybridRCNN 达到了较好的性能。这也说明了对多标签文本分类，提出的混合注意机制是一种有效且灵活的方法。

(3) 与 DPCNN 相比，HybridRCNN 也取得较好的结果。与 DPCNN 通过设计一个深度金字塔 CNN 架构来表示词级信号不同，HybridRCNN 更多地考虑了词级与标签之间的交互信息。

5.5.4 HybridRCNN 消融分析

为了验证 HybridRCNN 每个组件的影响，使用消融对方法不同重要组件进行分析：一是交互注意力模块(IA)；二是混合注意力 CNN 空间文档表征(SR)；三是混合注意力 RNN 时序文档表征(TR)；四是空时文档表征(SR＋TR＋concat)；五是加权的空时文档表征(SR＋TR＋weighted)，实验针对密集数据集 RCV1、Ydata、Yelp 及稀疏数据集 Eurlex_4k。图 5.5 给出了四个数据集在 Accuracy、micro-F1、$p@k\{1,3,5\}$不同评估指标上的结果。

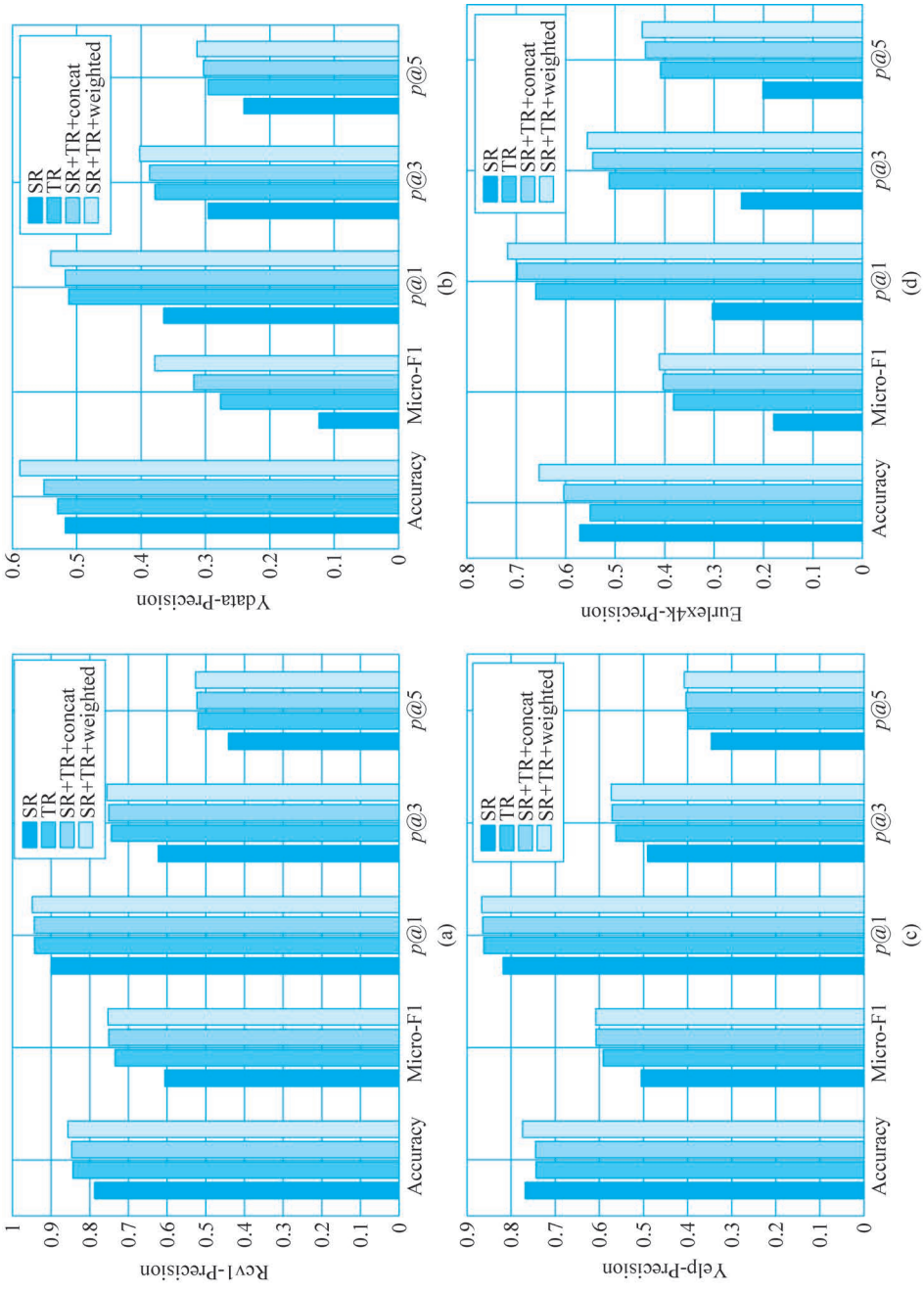


图 5.5　HybridRCNN 消融分析

通过实验结果可知：

(1) 与单组件(SR 和 TR)相比，加权集成组件(SR＋TR＋weighted)在四种数据集中都有较好的性能，说明在处理多标签文本分类时，集成的空时文档表示比单一表示更稳定且更适合。

(2) 与 SR 和 TR 的拼接(SR＋TR＋concat)方法进行了比较，加权集成组件(SR＋TR＋weighted)提高了所有四个数据集的性能，这隐含地表明所提出的加权集成策略能够自适应地集成两种互补信息，大大提高多标签文本分类的识别能力。

5.5.5 HybridRCNN 可视化分析

为了进一步说明 HybridRCNN 的有效性，在 Wiki10_31k 一个实例文档上使用热图可视化来表征空时文档表示，这个实例文档包含 14 个标签，分别为 programming、software、architecture、architect、software-defifinition、occupation、roles、reference、architecture-methodology、wikipedia、computer、IT-related、duties 和 history。如图 5.6(a)所示，通过细粒度的短语级信号，标签“architecture-methodology”可以通过短语“design of the architecture”和“design methodology”捕捉；标签“IT-related”可以通过短语“it related architects”捕捉，但是 CNN 空间文档表征忽视了词级标签“roles”和“duties”。如图 5.6(b)所示，标签“roles”和“duties”能通过词级信息进行捕捉，在图 5.6(c)中可观察到，空时文档表征集成了从子网 CNN 和 RNN 获得的两种互补信息，极大地提高了识别能力，这也说明我们提出的 HybridRCNN 结构是有效的。

5.5.6 HybridRCNN 时间复杂度比较

在表 5.4 中，我们列出了整体运行时间(由训练时间、验证时间及测试时间共同组成)，所有时间均以秒为单位表示，并同时展示了模型训练时所占用的存储空间大小，单位为 GB。在比较的方法中，仅仅有 8 个方法可以扩展到 Wiki10_31k 数据集，在 CNN-RNN 框架中，我们的方法得到了与 RCNN 和 DRNN 相似的时间复杂度，而且我们的方法能够容易地通过使用多 GPU 并行来提升模型的训练时间。

(a) (b) (c)

图 5.6 HybridRCNN 可视化分析

表 5.4 算法整体运行时间(训练时间+验证时间+测试时间)和模型大小比较

Methods	Rcv1		Ydata		Yelp		Eurlex_4k		Wiki10_31k	
	Time/s	Size/GB	Time/s	Size/GB	Time/s	Size/GB	Time/s	Size/GB	Time/s	Size/GB
RCNN	295	0.409	556	0.644	1712	1.89	287	0.146	7869	0.534
GRA	281	0.362	377	0.52	1022	1.03	364	0.149	2584	0.56
DRNN	815	0.406	2080	0.639	6382	1.89	553	0.132	—	—
CRAN	230	0.355	308	0.513	718	1.02	361	0.154	3191	0.649
TextCNN	92	0.341	182	0.494	511	1.41	169	0.133	2701	0.534
TextRNN	237	0.408	383	0.641	1080	1.89	240	0.139	3614	0.479
DPCNN	131	0.406	361	0.64	1036	1.89	259	0.132	—	—
Transformer	293	0.41	779	0.645	2733	1.89	474	0.147	3069	0.535
AttConvNet	268	0.428	662	0.662	2371	1.91	347	0.17	—	—
LAHA	108	0.344	469	0.646	1531	1.89	362	0.143	5632	0.483
HybridRNN	285	0.412	893	0.646	2998	1.89	533	0.125	8459	0.453

5.6 极端量级多标签文本实验分析

当标签量很大时,模型 HybridRCNN 存在使用局限,因此我们提出了改进的 Multi-V-Transformer 框架,使用五个标签量较大的数据集验证 Multi-V-Transformer,并且和 DiSMEC[172]、Parabel[173]、Bonsai[174]、FastXML[175]、SLEEC[43]、XML-CNN[36]、AttentionXML[150]、X-Transformer[160]等极端多标签方法进行了比较。

5.6.1 实验设置

(1) 实验数据集。我们利用 4 个基准数据集来评估 Multi-V-Transformer 方法的有效性,具体数据概览如表 5.5 所示,其中 N_{trn}、N_{tst} 表示训练样本数和测试

样本数，D 是特征总数，L 是总的标签数，$\tilde{L}$ 是每个文档对应的平均标签数，$\hat{L}$ 是每个标签对应的平均文档数。为了进一步分析方法的可扩展性，我们特别选取了 Eurlex_4k 和 Wiki10_31k 这两个数据集进行深入探讨。

表 5.5　极端量级多标签数据集详细信息

Datasets	N_{trn}	N_{tst}	D	L	$\tilde{L}$	$\hat{L}$
Eurlex_4k	15449	3865	186104	3956	5.30	20.79
Wiki10_31k	14146	6616	101938	30938	18.64	8.52
AmazonCat-13K	1186239	306782	203882	13330	5.04	448.57
Amazon-670K	490449	153025	135909	670091	5.45	3.99

（2）参数设置。在 Multi-V-Transformer 方法中，我们使用 Tesla V100 GPU 训练模型，GPU 显存是 16G，对所有的实验，我们使用 3 个视图，初始学习率设置为 0.0001，权重衰减设置为 0.01。

5.6.2　极端多标签实验比较

我们比较提出的 Multi-V-Transformer 和优秀的极端多标签学习方法，包括常用的 1-vs-ALL 方法（如 DiSMEC[172]、Parabel[173]、Bonsai[174]）、基于树的方法（如 FastXML[175]）、基于嵌入的方法（如 SLEEC[43]）、基于深度学习的方法（如 XML-CNN[36]、AttentionXML[150]、X-Transformer[160]），评估指标使用 $p@k\{1,3,5\}$，实验结果见表 5.6。

通过实验结果可知：

（1）与相关的深度极端多标签学习算法相比，如 XML-CNN[36]、AttentionXML[150]、X-Transformer[160]，我们的方法取得了较好的性能，如在 Wiki10_31k 数据集，我们的方法在 $p@1$ 上提高了 X-Transformer 到 0.76%，提高了 AttentionXML 到 1.8%。

（2）相较于 X-Transformer，我们的方法使用极端多标签聚类学习约简标签，能在较为极端的 Amazon-670K 数据集上进行训练，在 $p@1$ 上提高了 AttentionXML 到 1.71%。

表 5.6　极端量级多标签实验比较

Datasets	Metrics	Methods								
		DiSMEC	Parabel	Bonsai	FastXML	SLEEC	XML-CNN	AttentionXML	X-Transformer	Multi-V-Transformer
Eurlex_4k	p@1	83.21	82.12	82.30	76.37	63.40	75.32	87.12	87.22	**87.58**
	p@3	70.39	68.91	69.55	63.36	50.35	60.14	73.99	75.12	**75.64**
	p@5	58.73	57.89	58.35	52.05	41.28	49.21	61.92	62.90	**63.69**
Wiki10_31k	p@1	84.13	84.19	84.52	83.03	85.88	81.42	87.47	88.51	**89.27**
	p@3	74.72	72.46	73.76	67.47	72.98	66.23	78.48	**78.71**	78.64
	p@5	65.94	63.37	64.69	57.76	62.70	56.11	69.37	**69.62**	69.00
AmazonCat-13K	p@1	93.81	93.02	92.98	93.11	90.53	93.26	95.92	96.70	**96.86**
	p@3	79.08	79.14	79.13	78.20	76.33	77.06	82.41	83.85	**84.31**
	p@5	64.06	64.51	64.46	63.41	61.52	61.40	67.31	68.58	**69.09**
Amazon-670K	p@1	44.78	44.91	45.58	36.99	35.05	33.41	47.58	—	**49.29**
	p@3	39.72	39.77	40.39	33.28	31.25	30.00	42.61	—	**44.14**
	p@5	36.17	35.98	36.60	30.53	28.56	27.42	38.92	—	**40.24**

5.6.3　Multi-V-Transformer 集成消融分析

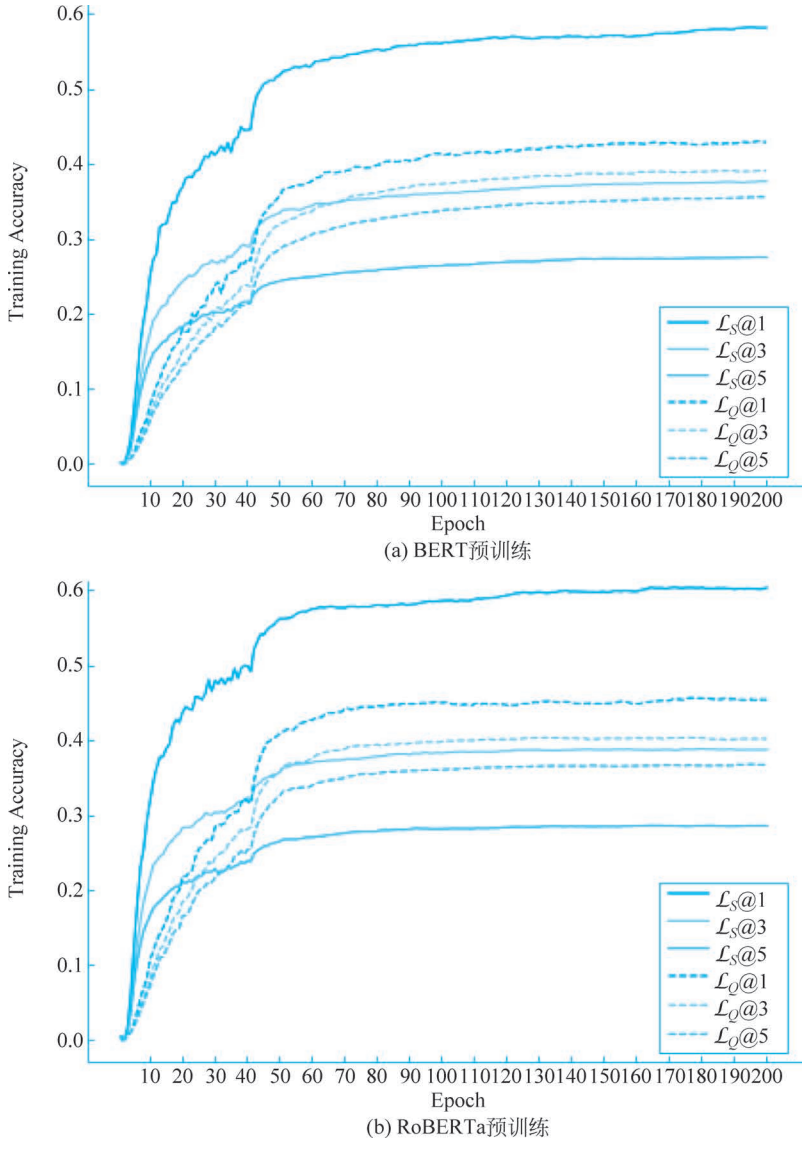
基于预训练模型 BERT、RoBERTa、XLNet，对 Multi-V-Transformer 进行集成消融分析，结果如表 5.7 所示。通过对不同的预训练模型进行集成，我们的方法取得了较好的性能，例如：在 Wiki10_31k 数据集，在 $p@1$ 上，集成提高 BERT 到 1.98%，提高 RoBERTa 到 3.46%，提高 XLNet 到 4.11%；在 Amazon-670K 数据集，集成提高 BERT 到 3.02%，提高 RoBERTa 到 2.02%，提高 XLNet 到 2.31%。

表 5.7　极端量级 Multi-V-Transformer 集成消融分析

Datasets	Metrics	Methods			
		BERT	RoBERTa	XLNet	Ensemble
Eurlex_4k	$p@1$	85.30	85.43	86.88	**87.58**
	$p@3$	73.20	72.56	74.25	**75.64**
	$p@5$	60.67	60.20	61.72	**63.69**
Wiki10_31k	$p@1$	87.29	85.81	85.16	**89.27**
	$p@3$	76.09	73.53	73.28	**78.64**
	$p@5$	65.32	63.22	63.35	**69.00**
AmazonCat-13K	$p@1$	96.57	96.55	96.40	**96.86**
	$p@3$	83.73	83.68	83.34	**84.31**
	$p@5$	68.57	68.47	68.14	**69.09**
Amazon-670K	$p@1$	46.27	47.27	46.98	**49.29**
	$p@3$	41.36	42.29	41.93	**44.14**
	$p@5$	37.57	38.47	38.19	**40.24**

5.6.4　Multi-V-Transformer 聚类学习分析

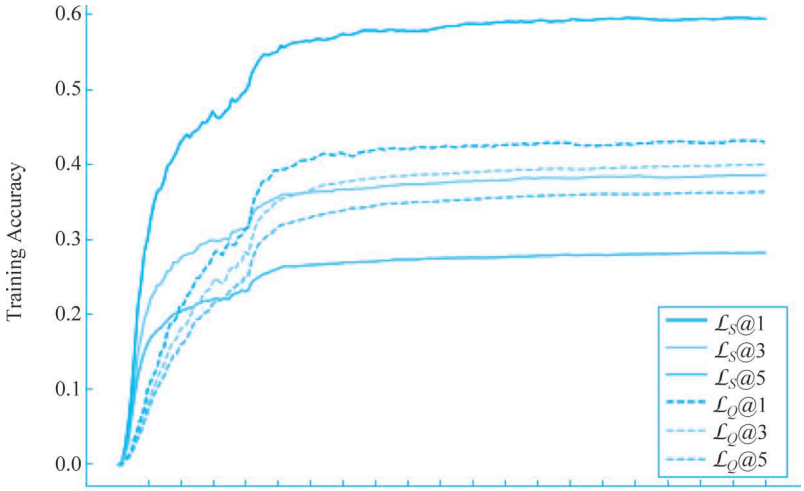
在 Multi-V-Transformer 中，我们使用极端多标签聚类学习来约简标签，随着模型训练迭代次数的增加，观察 Amazon-670K 数据集极端多标签聚类学习（公式 $\mathcal{L}_S$）和约简的标签集嵌入（公式 $\mathcal{L}_Q$）两部分精度变化曲线，如图 5.7 所示。通过使用极端多标签聚类学习，能约简极大的多标签集，然后基于约简的标签集使用不平衡的 Focal 损失对约简的标签集进行学习，最终实现对极端量级的多标签学习。



(a) BERT预训练

(b) RoBERTa预训练

图 5.7 不同预训练模型 Multi-V-Transformer 聚类学习分析



(c) XLNet预训练

图 5.7 （续）

5.7　本章小结

在本章中，针对不同量级的多标签文本分类任务，我们提出了两个网络模型，即 HybridRCNN 模型和 Multi-V-Transformer 模型。为了应对标签数量范围广泛（100～30000）的任务挑战，我们设计了一种创新的 HybridRCNN 网络架构，该架构集成了自适应空时表征技术。此架构能够同时考虑词与词之间、短语与短语之间、词与标签之间以及短语与标签之间的复杂关系。进一步地，通过实施自适应加权集成策略，HybridRCNN 有效地融合了卷积神经网络（CNN）与循环神经网络（RNN）的互补信息，从而实现了分类器识别能力的大幅提升。为了适应极端量级的标签，我们提出了集成 Transformer 多视图表征结构的 Multi-V-Transformer，该网络通过聚类排序模块能有效适应标签量上百万级的分类任务，并且通过多视图注意力表征、极端多标签聚类学习和约简的标签集嵌入学习来提升模型的泛化性能。最后在大量的多标签文本分类任务上与相关的优秀方法进行了比较，并验证了提出方法的有效性。