Deep Learning for Extreme Multi-label Text Classification

用于极端多标签文本分类的深度学习

 概括：本文提出结合CNN的XML-CNN模型来解决大规模的多标签文本分类问题。

一、简介

1.标题和作者

摘要内容

1）解释极端多标签文本分类：极端多标签文本分类 （XMTC） 是指从非常大的标签集合中为每个文档分配其最相关的类标签子集的问题，其中标签的数量可能达到数十万或数百万个。

2）面临问题：巨大的标签空间带来了数据稀疏性和可扩展性等研究挑战。

3）其他方法有局限，本文方法及优势：结合CNN的XML-CNN模型

2.极端多标签文本分类综述及其研究意义（1 引言部分内容）

二、背景

1.传统文本分类方法及其局限性综述（1 引言部分内容）

2.文本分类的深度学习方法简介（1.1，1.2，1.3）

3.在极端多标签场景中需要高效准确的分类（1.4）

三、传统的六个分类方法（2部分）

六个方法（着重解释CNN-Kim方法和Bow-CNN方法）

四、本文建议的方法（XML-CNN）（3部分）重点

1.本文方法提出的亮点：

  1）动态最大池化（dynamic max pooling）

  2）二进制交叉熵损失函数（binary cross-entropy loss function）

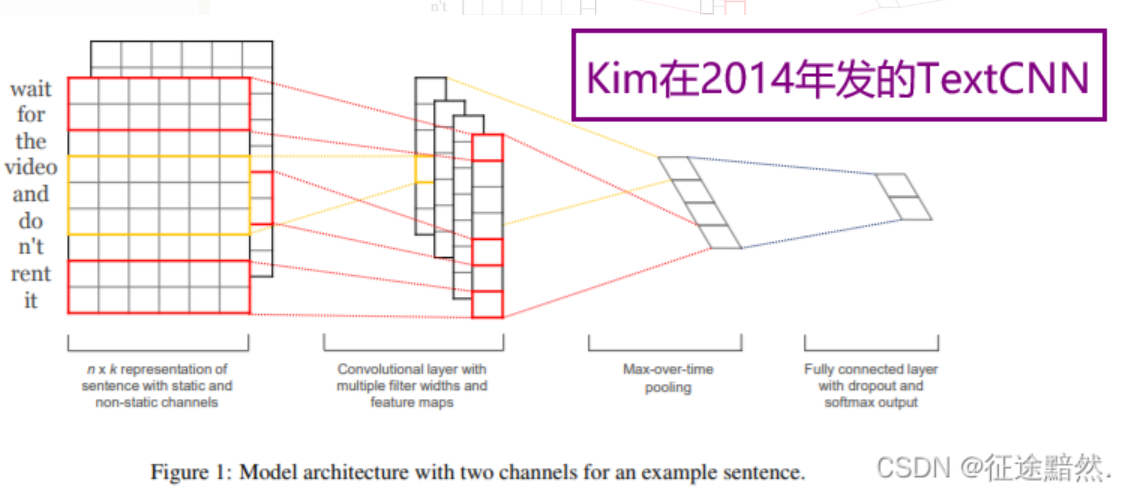
  3）隐藏的瓶颈层（hidden bottleneck layer）

1. 介绍基于CNN-Kim的XML-CNN多类文本分类体系结构

（内容在3部分）

3.动态最大池化

 对比最原始的TextCNN模型，如下图：

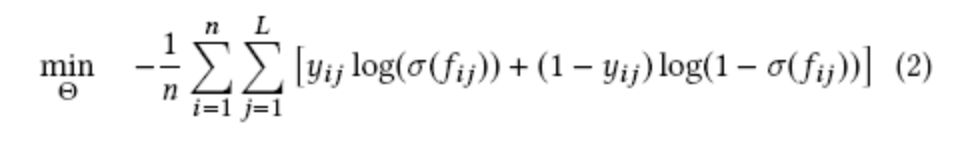


  传统做池化的方法是，一个filter卷积的数据（图中卷积层的一个竖条）池化为一个特征（即一个数字）。

   但是本论文提出的思想是： 当一个文本太大，原始的TextCNN算法的每个filter经过卷积池化后，把文本变成一个特征（数字），这样会丢失太多信息。所以应该把filter卷积的数据（图中卷积层的一个竖条）分成p个块，按照每个块进行最大池化，池化后会剩下p个特征，这样既保留了重要特征，也保留了位置信息，叫做动态最大池化。

4.二进制交叉熵损失函数

多标签分类最常用的损失函数。使用binary cross-entropy, 而不是 softmax cross-entropy



5.隐藏的瓶颈层

  在池化层与输出层之间增加一个全连接的隐含层，该隐含层有h个单元，称为隐藏的瓶颈层，其隐藏单元的数量h远小于pooling层和output层。

  这样做可以：1）降低计算量。假设pooling层有100个参数，output有100个参数，不加隐藏层会有10000个参数参与计算。如果加一个有h个神经元的隐藏层，参与计算的参数是200\*h个，h稍小一点就可以降低很多计算量。2）没有这个隐藏层，模型只有一个非线性的隐藏层，这不够学习良好的文档表示和分类器。

五、比较评估（4 实验部分）

1.所有方法进行比较评估实验

2.实验结果

2.强调XML-CNN可行性

六、结论（5 部分）

七、参考资料

# [1] Kush Bhatia, Himanshu Jain, Purushoam Kar, Manik Varma, and Prateek Jain.

# 2015. Sparse local embeddings for extreme multi-label classification. In Advances

# in Neural Information Processing Systems. 730–738.

# [2] Wei Bi and James Tin-Yau Kwok. 2013. Efficient Multi-label Classification with

# Many Labels.. In ICML (3). 405–413.

# [3] Sayan Ghosh, Eugene Laksana, Stefan Scherer, and Louis-Philippe Morency.

# 2015. A multi-label convolutional neural network approach to cross-domain

# action unit detection. In Affective Computing and Intelligent Interaction (ACII),

# 2015 International Conference on. IEEE, 609–615.

# [4] Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey

# Ioffe. 2013. Deep convolutional ranking for multilabel image annotation. arXiv

# preprint arXiv:1312.4894 (2013).

# [5] Farbound Tai and Hsuan-Tien Lin. 2012. Multilabel classification with principal

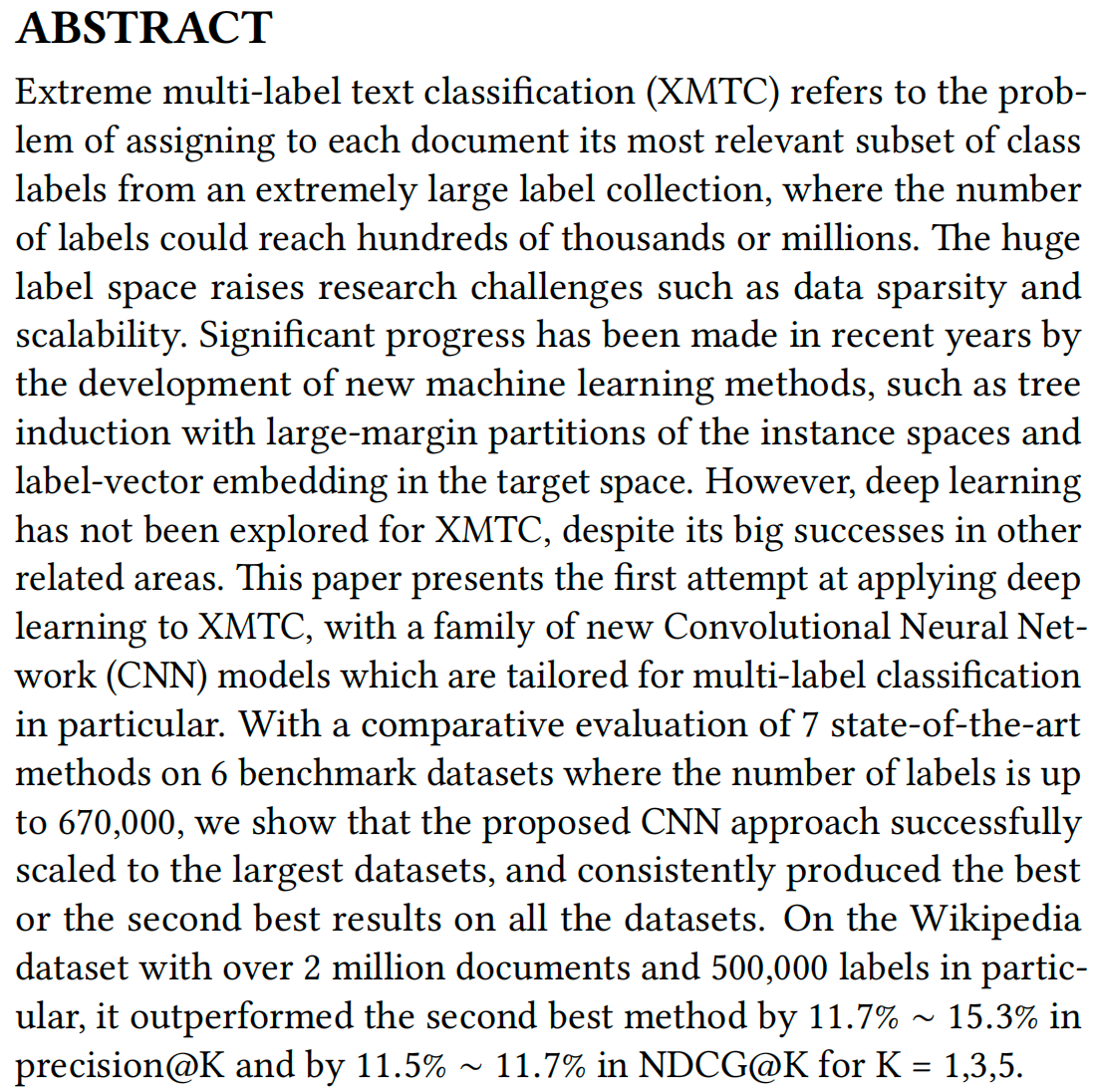
# label space transformation. Neural Computation 24, 9 (2012), 2508–2542.

# [6] Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to

# large vocabulary image annotation. (2011).

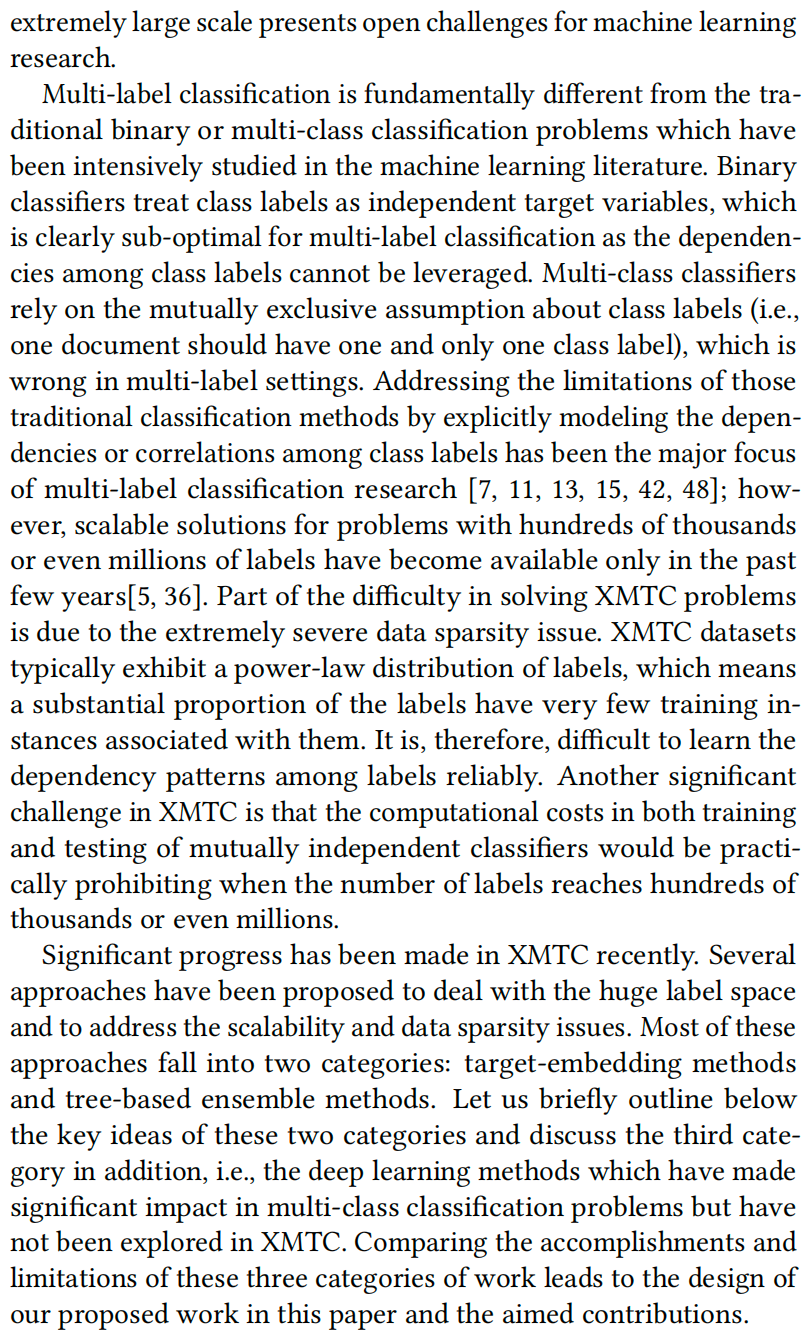
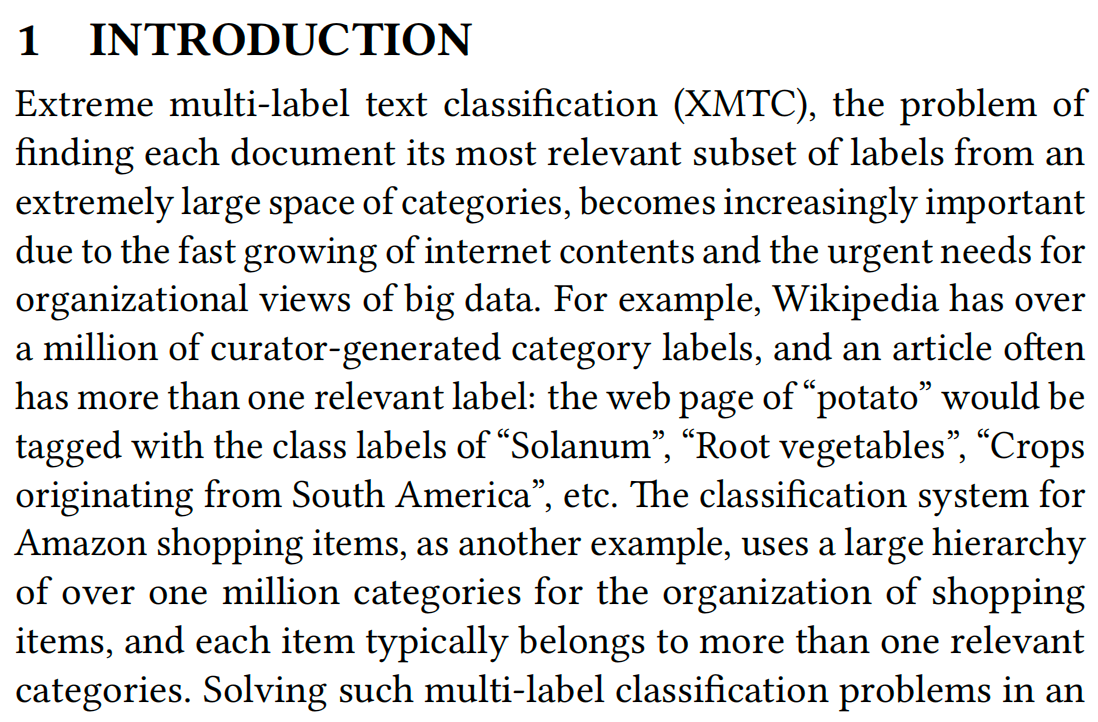
# [7] Deep Learning for Extreme Multi-label Text Classification：https://developer.aliyun.com/article/1170816

# [8] Deep Learning for Extreme Multi-label Text Classification阅读笔记：https://blog.csdn.net/sjyttkl/article/details/103960500



## 摘要

极端多标签文本分类 （XMTC） 是指从非常大的标签集合中为每个文档分配其最相关的类标签子集的问题，其中标签的数量可能达到数十万或数百万个。巨大的标签空间带来了数据稀疏性和可扩展性等研究挑战。近年来，通过开发新的机器学习方法，例如具有实例空间大边距分区的树归纳和在目标空间中嵌入标签向量的树感应，已经取得了重大进展。然而，尽管XMTC在其他相关领域取得了巨大成功，但深度学习尚未被探索。本文介绍了将深度学习应用于XMTC的第一个方法，并提出了一系列新的卷积神经网络（CNN）模型，这些模型特别针对多标签分类而定制。通过对 6 个标签数量高达 670,000 个基准数据集的 7 种最先进方法的比较评估，我们表明所提出的 CNN 方法成功地扩展到最大的数据集，并在所有数据集上始终产生最佳或次优结果。在拥有超过 200 万个文档和 500,000 个标签的维基百科数据集上，当 K = 1,3,5 时，它在 precision@K中比第二好的方法高出11.7% ∼ 15.3%，在 NDCG@K上比第二好的方法高出11.5% ∼ 11.7%。



# 1 引言

极端多标签文本分类（XMTC），是由于互联内容的快速增长和对大数据组织视图的迫切需求，从极大的类别空间中查找每个文档最相关的标签子集的问题变得越来越重要。例如，维基百科有超过一百万个策展人生成的类别标签，一篇文章通常有多个相关标签：“马铃薯”的网页会被标记为“茄属”、“根茎类蔬菜”、“来自南美洲的作物”等类标签。另一个例子是，亚马逊购物商品的分类系统使用超过一百万个类别的大型层次结构来组织购物商品，每个商品通常属于多个相关类别。解决此类极大规模多标签分类问题给机器学习研究带来了严峻的挑战。

多标签分类与机器学习文献中深入研究的传统二元或多类分类问题有着根本的不同。

二元分类器将类标签视为独立的目标变量，这对于多标签分类来说显然是次优的，因为无法利用类标签之间的依赖关系。多类分类器依赖于关于类标签的互斥假设（即一个文档应该有一个且只有一个类标签），这在多标签定义中是错误的。通过显式建模类标签之间的依赖性或相关性来解决这些传统分类方法的局限性一直是多标签分类研究的主要焦点[7,11,13,15,42,48];然而，针对数十万甚至数百万个标签问题的可扩展解决方案直到最近几年才可用[ 5，36]。解决XMTC问题的困难部分是由于极其严重的数据稀疏性问题。XMTC 数据集通常表现出标签的幂律分布，这意味着很大一部分标签具有与之关联的训练实例很少。因此，很难可靠地学习标签之间的依赖关系。XMTC 的另一个重大挑战是，当标签数量达到数十万甚至数百万时，训练和测试相互独立的分类器的计算成本实际上令人望而却步。

XMTC最近取得了重大进展。已经提出了几种方法来处理巨大的标签空间并解决可扩展性和数据稀疏性问题。这些方法大多分为两类：目标嵌入方法和基于树的集成方法。下面我们简单介绍一下这两类的主要思想，并讨论第三类，即深度学习方法，这些方法在多类分类问题中产生了重大影响，但在XMTC中尚未得到探索。比较这三类工作的成就和局限性，可以得出本文提出的工作设计和目标贡献。

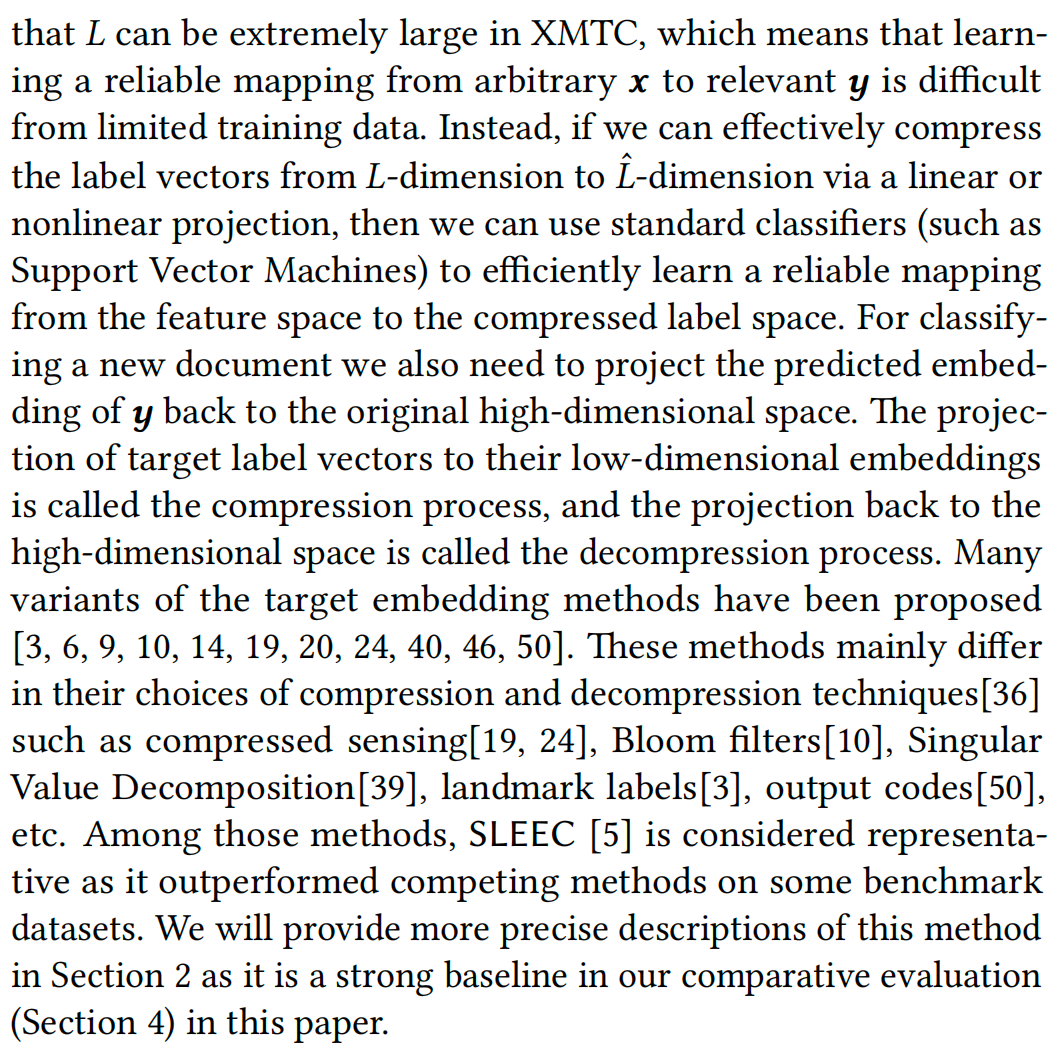
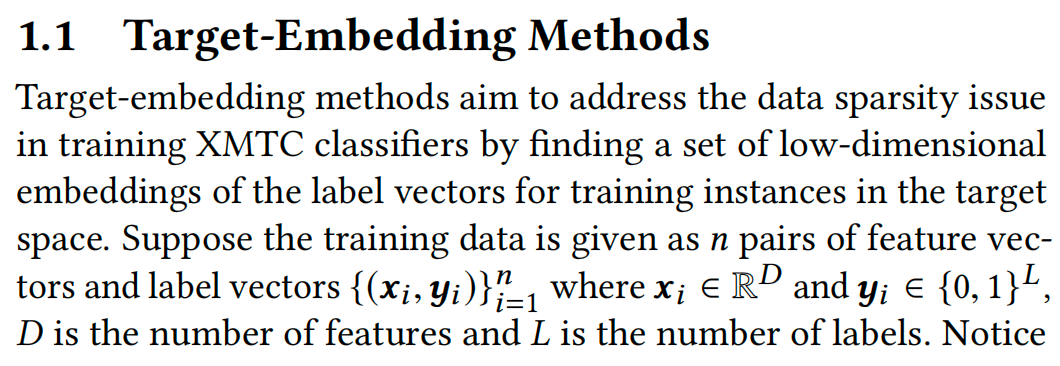
1. 进行极端多标签文本分类目的：第一段
2. 结合机器学习，传统的二元分类器将类标签视为独立的目标变量，这对于多标签分类不友好，因为类标签之间的联系不能被利用。

3.XMTC面临两大问题：

  1）解决XMTC问题的困难部分是由于极其严重的数据稀疏问题。相当大比例的标签只有很少的训练实例与它们相关联。因此，学习标签之间的联系是很困难的。

  2）XMTC中的另一个重大挑战是，当标签的数量达到数十万甚至数百万时，训练和测试相互独立的分类器的计算成本难以接受。

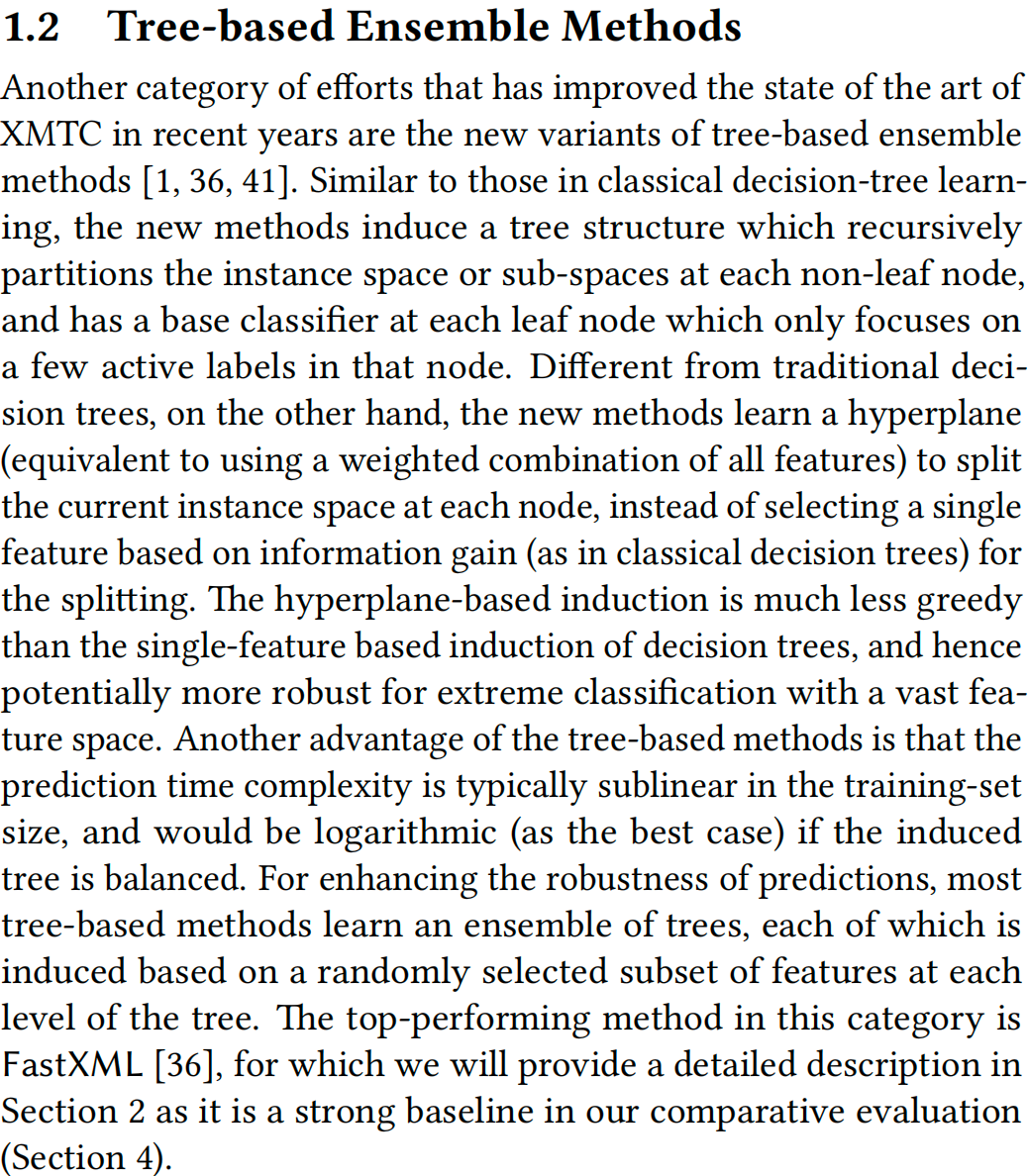
4.三种处理巨大的标签空间问题的方法：1.1目标嵌入方法、1.2基于树的集成方法、1.3深度学习方法。（1.4本文提出的新方法暂不讲解，详细在3



### 1.1 目标嵌入方法

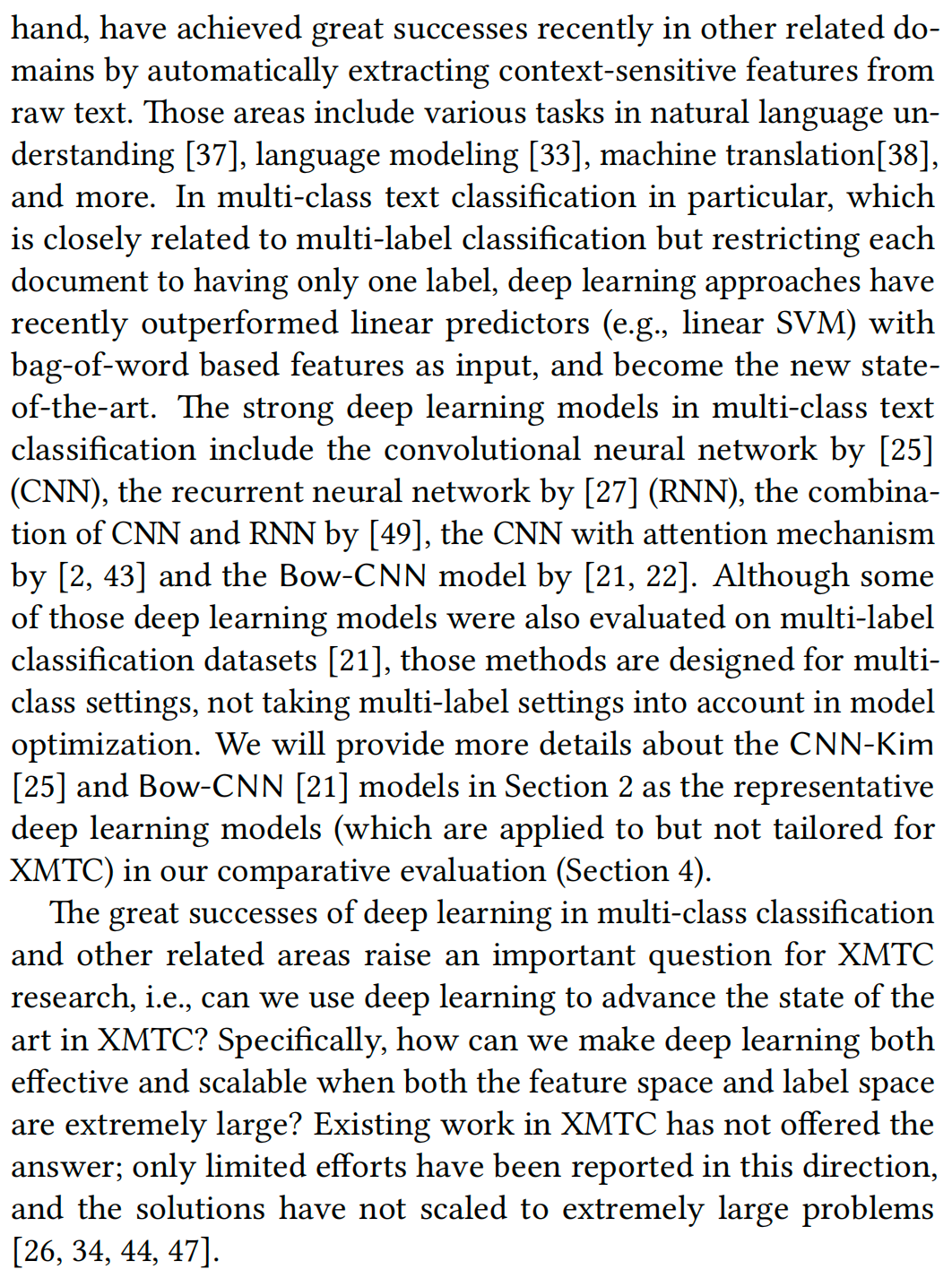
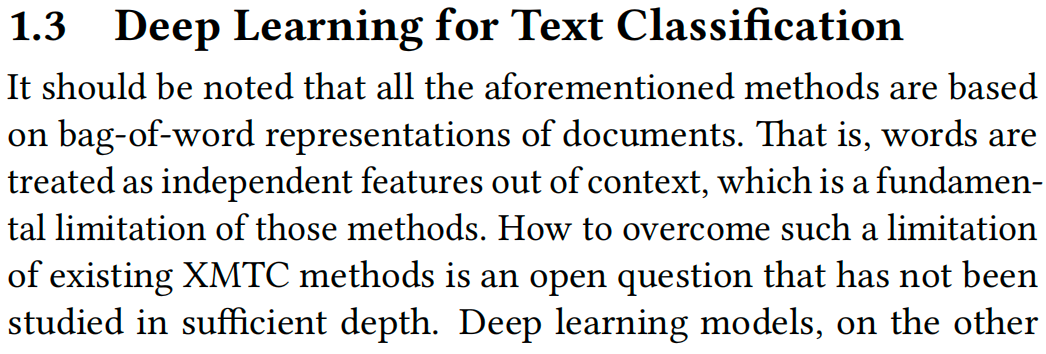
目标嵌入方法旨在通过查找一组用于在目标空间中训练实例的标签向量的低维嵌入来解决训练 XMTC 分类器中的数据稀疏性问题。假设训练数据以 n 对特征向量和标签向量 {（x，y）} 的形式给出，其中 x∈ Rand，y∈ {0，1}，D 是特征数，L 是标签数。已知

D 是特征的数量，L 是 L 在 XMTC 中可以非常大的标签数量，这意味着从有限的训练数据中学习从任意 x 到相关 y 的可靠映射是很困难的。相反，如果我们可以通过线性或非线性投影有效地将标签向量从 L 维压缩到 ˆL 维，那么我们可以使用标准分类器（例如支持向量机）有效地学习从特征空间到压缩标签空间的可靠映射。为了对新文档进行分类，我们还需要将预测的 y 嵌入投影回原始高维空间。将目标标签向量投影到其低维嵌入称为压缩过程，将投影投影回高维空间称为解压缩过程。已经提出了许多目标嵌入方法的变体[3,6,9,10,14,19,20,24,40,46,50]。这些方法的主要区别在于压缩和解压缩技术的选择[36]，如压缩传感[19,24]，Bloom滤波[10]，奇异值分解[39]，地标标签[3]，输出代码[50]等。在这些方法中，SLEEC [5]被认为具有代表性，因为它在一些基准数据集上优于竞争方法。我们将在第 2 节中对这种方法进行更精确的描述，因为它是我们本文比较评估（第 4 节）的有力基线。



### 1.2 基于树的集成方法

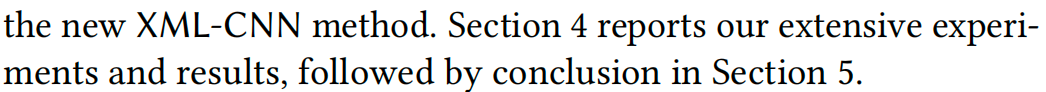
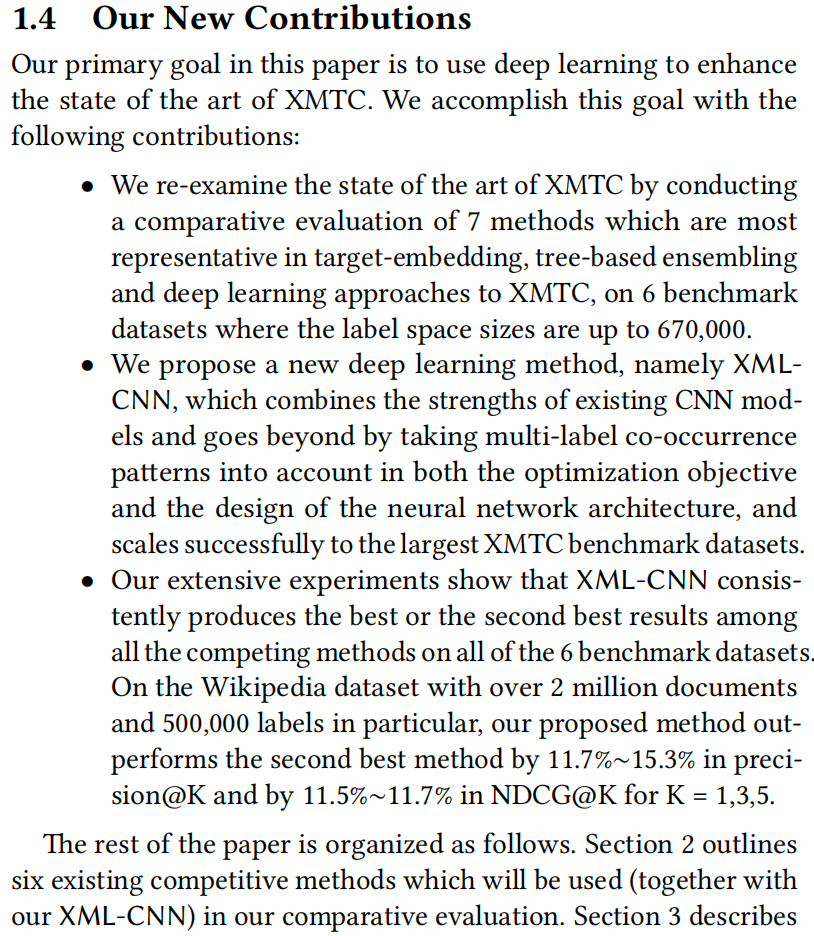
近年来改进XMTC技术水平的另一类工作是基于树的集成方法的新变体[1,36,41]。与经典决策树学习中的方法类似，新方法诱导了一种树结构，该结构以递归方式对每个非叶节点上的实例空间或子空间进行分区，并在每个叶节点上有一个基分类器，该分类器仅关注该节点中的几个活动标签。另一方面，与传统的决策树不同，新方法学习一个超平面（相当于使用所有特征的加权组合）来拆分每个节点的当前实例空间，而不是根据信息增益选择单个特征（如经典决策树）进行拼接。与基于单特征的决策树归纳相比，基于超平面的归纳法的贪婪程度要低得多，因此对于具有广阔特征空间的极端分类来说，它可能更健壮。基于树的方法的另一个优点是，预测时间复杂度在训练集大小中通常是亚线性的，如果诱导树是平衡的，则预测时间复杂度将是对数的（作为最佳情况）。为了增强预测的稳健性，大多数基于树的方法都会学习一组树，每个树都是根据树的每个级别随机选择的特征子集来诱导的。此类别中表现最好的方法是 FastXML [ 36 ]，我们将在第 2 节中对此进行详细说明，因为它是我们比较评估（第 4 节）的强基线。



### 1.3 深度学习在文本分类中的应用

应该注意的是，上述所有方法都是基于文档的词袋表示。也就是说，单词在上下文之外被视为独立的特征，这是这些方法的根本局限性。如何克服现有XMTC方法的这种局限性是一个悬而未决的问题，尚未得到足够深入的研究。深度学习模型，在另一方面，最近通过自动从原始文本中提取上下文相关特征，在其他相关领域取得了巨大成功。这些领域包括自然语言理解[37]、语言建模[33]、机器翻译[38]等各种任务。特别是在多类文本分类中，它与多标签分类密切相关，但将每个文档限制为只有一个标签，深度学习方法最近优于线性预测器（例如，线性 SVM），以基于词袋的特征作为输入，并成为新的最先进的技术。多类文本分类中的强深度学习模型包括 [ 25 ] （CNN） 的卷积神经网络、[ 27 ] （RNN） 的递归神经网络、[ 49 ] CNN 和 RNN 的组合、[ 2，43] CNN 和 [ 21， 22 ] 的 Bow-CNN 模型。尽管其中一些深度学习模型也在多标签分类数据集上进行了评估[ 21 ]，但这些方法都是为多类而设计的，在模型优化中没有考虑多标签。我们将在第 2 节中提供有关 CNN-Kim [ 25 ] 和 Bow-CNN [ 21 ] 模型的更多详细信息，作为我们的比较评估（第 4 节）中的代表性深度学习模型（适用于 XMTC，但不是为 XMTC 量身定制的）。

深度学习在多类分类和其他相关领域的巨大成功为XMTC研究提出了一个重要问题，即我们能否利用深度学习来推进XMTC的最新技术？具体来说，当特征空间和标签空间都非常大时，我们如何才能使深度学习既有效又可扩展？XMTC的现有工作尚未提供答案;在这个方向上只报道了有限的努力，并且解决方案尚未扩展到非常大的问题[26,34,44,47]。



### 1.4 我们的新贡献

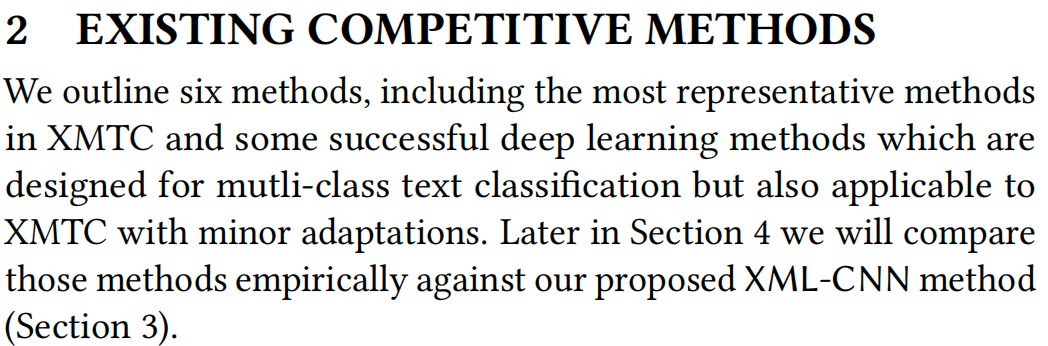
本文的主要目标是使用深度学习来增强 XMTC 的最新技术水平。我们通过以下贡献来实现这一目标：

- 我们通过对 XMTC 的目标嵌入、基于树的组装和深度学习方法中最具代表性的 7 种方法进行比较评估，重新审视了 XMTC 的最新技术，这些方法在 6 个基准数据集上标签空间大小高达 670,000。

- 我们提出了一种新的深度学习方法，即XMLCNN，它结合了现有CNN模型的优势，并在优化目标和神经网络架构的设计中考虑了多标签共现模型，并成功扩展到最大的XMTC基准数据集。

- 我们广泛的实验表明，在所有6个基准数据集上，XML-CNN 在所有竞争方法中始终产生最佳或次优的结果。在拥有超过 200 万个文档和 500,000 个标签的维基百科数据集上，我们提出的方法在 K = 1,3,5 时，precision@K 11.7%∼15.3% 和 11.5%∼11.7% 的NDCG@K优于第二好方法。

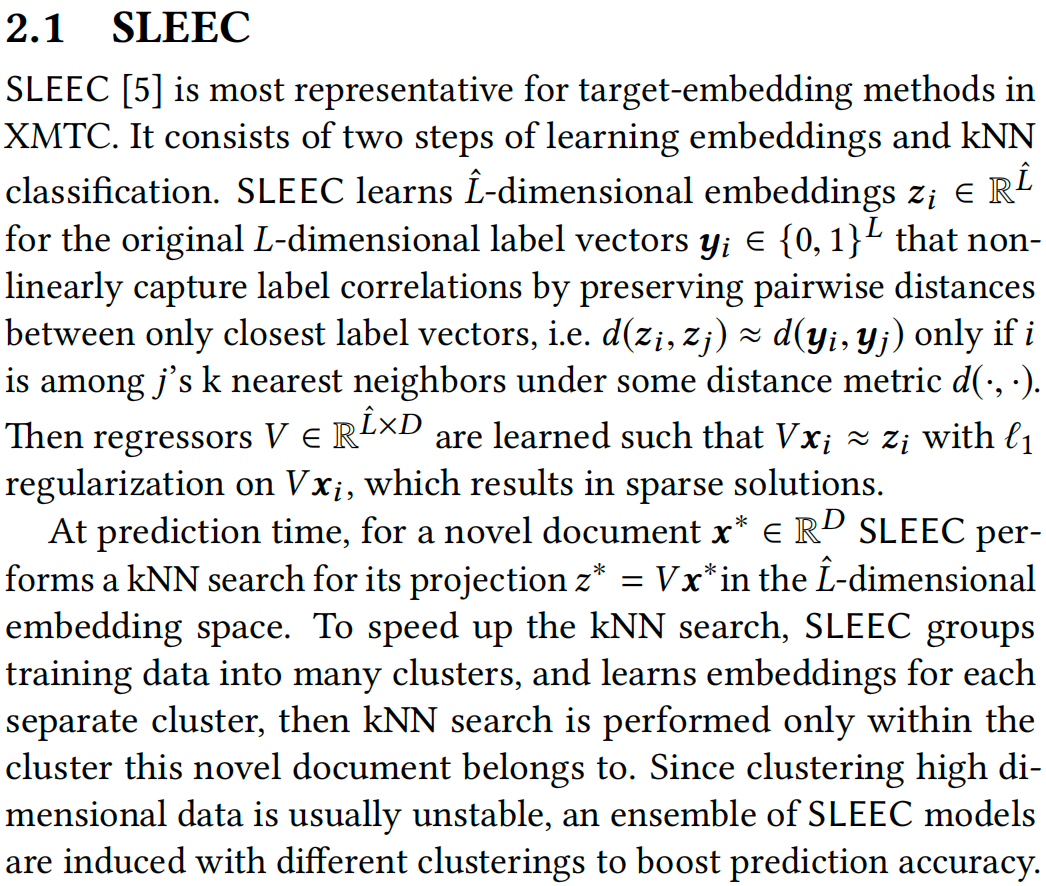
本文的其余部分组织如下。第 2 节概述了六种现有的竞争方法，这些方法将（与我们的 XML-CNN 一起）用于我们的比较评估。第 3 节描述第 2 节概述了六种现有的竞争方法，这些方法将（与我们的 XML-CNN 一起）用于我们的比较评估，即新的 XML-CNN 方法。第 4 节报告了我们广泛的实验和结果，然后是第 5 节的结论。



## 2 现有的竞争方法

我们概述了六种方法，包括 XMTC 中最具代表性的方法和一些成功的深度学习方法，这些方法专为多类文本分类而设计，但也适用于 XMTC，但稍作调整。在第 4 节的后面，我们将根据经验将这些方法与我们提出的 XML-CNN 方法（第 3 节）进行比较。

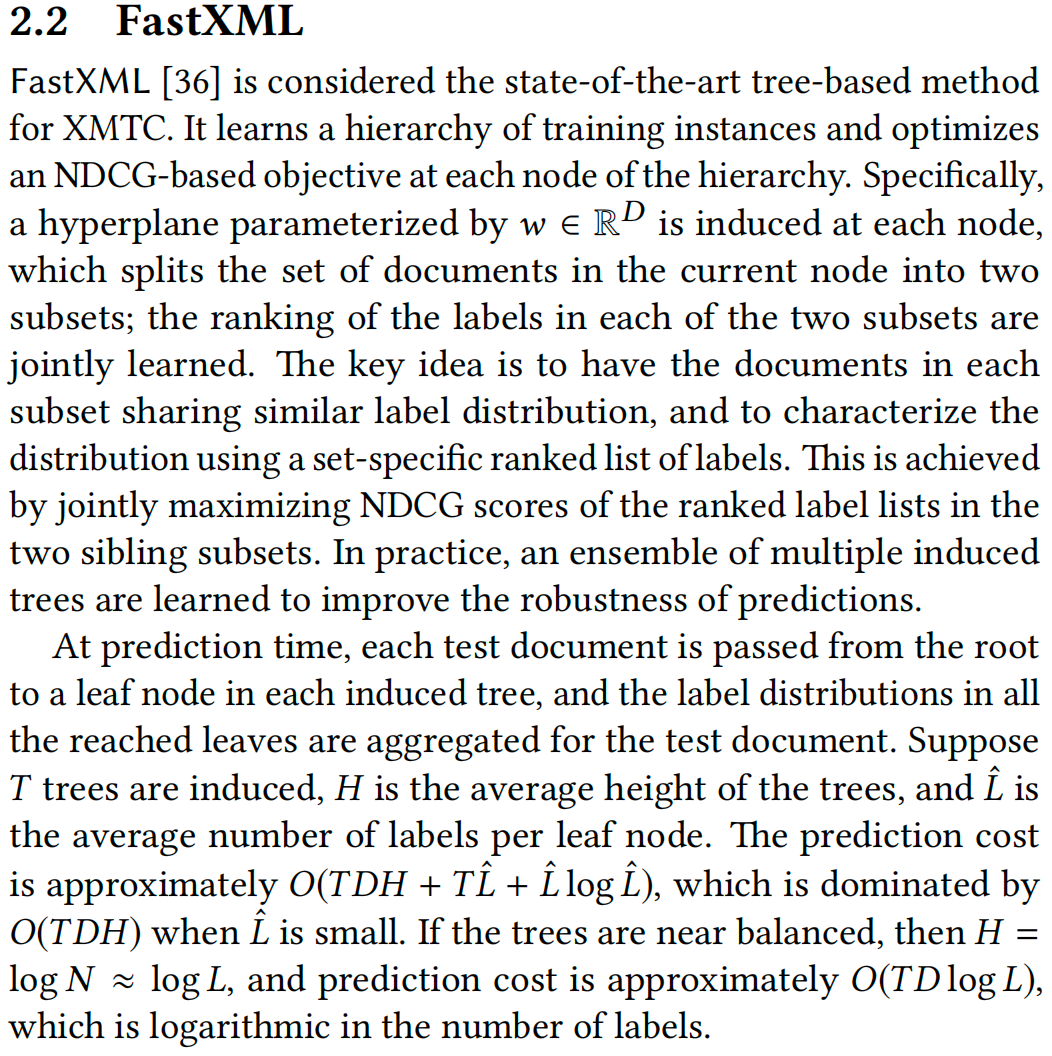
1.六种现有其他方法如下



### 2.1 SLEEC

SLEEC [ 5 ] 是 XMTC 中目标嵌入方法最具代表性的。它由学习嵌入和 kNN 分类两个步骤组成。SLEEC 学习原始 L 维标签向量 y 的 ˆL 维嵌入 z∈ R∈ {0， 1}，该嵌入通过仅保留最近标签向量之间的成对距离来非线性捕获标签相关性，即 d（z，z） ≈ d（y， y） 仅当 i 在某个距离度量 d（·，·） 下位于 j 的 k 最近邻之间时。然后学习回归器 V ∈ R，使得 V x≈ z 在 V x 上进行 ' 正则化，从而产生稀疏解。

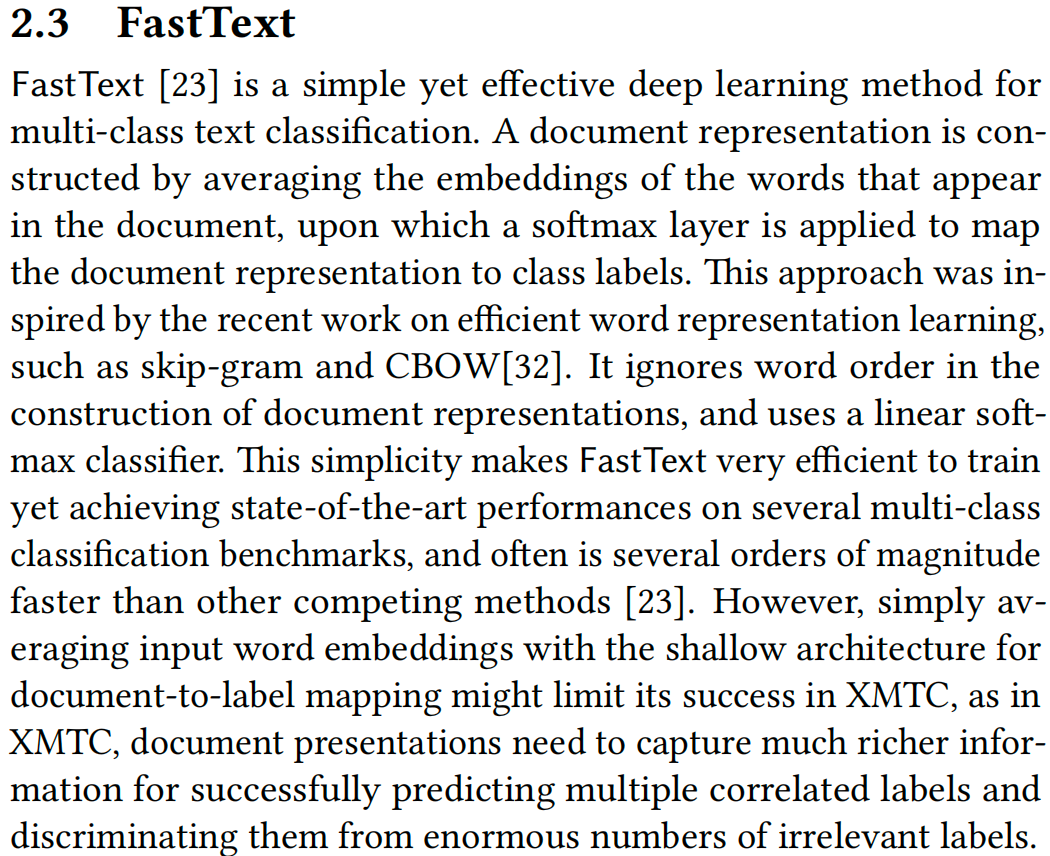
在预测时，对于新文档 x∈RSLEEC 在 ˆL 维嵌入空间中对其投影 z= V xx 执行 kNN 搜索。为了加快 kNN 搜索速度，SLEEC 将训练数据分组到多个集群中，并学习每个单独集群的嵌入，然后仅在此新文档所属的集群内执行 kNN 搜索。由于聚类高维数据通常是不稳定的，因此用不同的聚类诱导SLEEC模型的集合，以提高预测精度。



### 2.2 FastXML

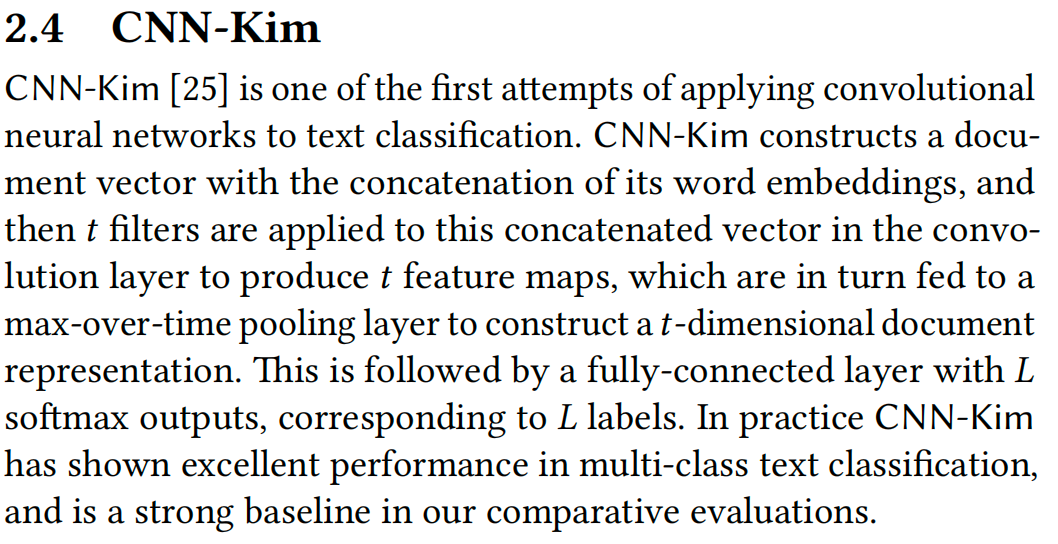
FastXML [ 36] 被认为是 XMTC 最先进的基于树的方法。它学习训练实例的层次结构，并在层次结构的每个节点上优化基于 NDCG 的目标。具体来说，在每个节点上诱导一个由 w ∈ Ris 参数化的超平面，它将当前节点中的文档集拆分为两个子集;两个子集中每个子集中的标签的排名是共同学习的。关键思想是让每个子集中的文档共享相似的标签分布，并使用特定于集合的标签排名列表来表征分布。这是通过共同最大化两个兄弟子集中排名标签列表的 NDCG 分数来实现的。在实践中，学习多个诱导树的集合以提高预测的鲁棒性。

在预测时，每个测试文档从根传递到每个诱导树中的叶节点，并将所有到达的叶子中的标签分布聚合为测试文档。假设诱导了 T 棵树，H 是树的平均高度，ˆL 是每个叶节点的平均标签数。预测成本约为O（T DH + T ˆL + ˆL log ˆL），当ˆL较小时，以O（T DH）为主。如果树接近平衡，则 H = log N ≈ log L，预测成本约为 O（T D log L），这是标签数量的对数。



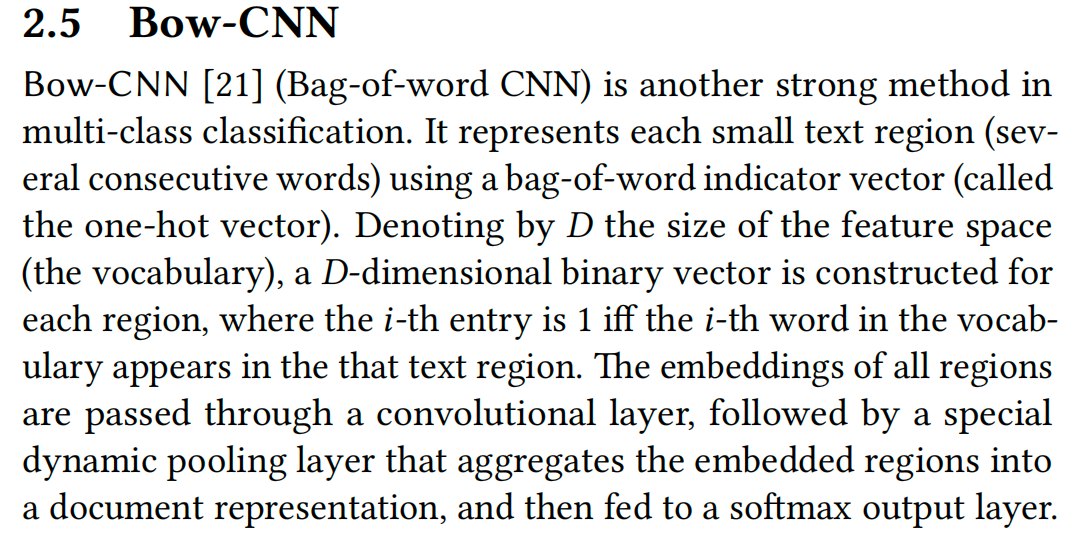
### 2.3 FastText

FastText [ 23] 是一种简单而有效的深度学习方法，用于多类文本分类。文档表示是通过对文档中出现的单词的嵌入进行平均来构造的，并在此基础上应用 softmax 层将文档表示映射到类标签。这种方法的灵感来自最近关于高效单词表示学习的工作，如skip-gram和CBOW[ 32 ]。它在构造文档表示时忽略词序，并使用线性 softmax 分类器。这种简单性使得 FastText 在训练时非常高效，同时在多个多类分类基准上实现了最先进的性能，并且通常比其他竞争方法快几个数量级 [ 23 ]。然而，简单地用文档到标签映射的浅层架构对输入词嵌入进行平均可能会限制其在 XMTC 中的成功，因为在 XMTC 中，文档表示需要捕获更丰富的信息才能成功预测多个相关标签并将它们与大量不相关的标签区分开来。



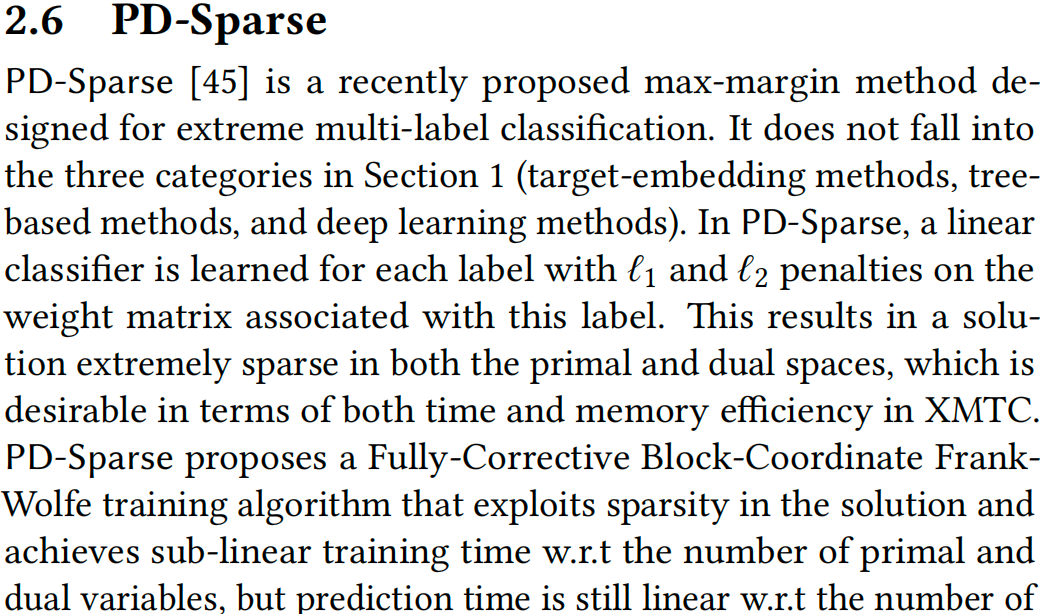
### 2.4 CNN-Kim

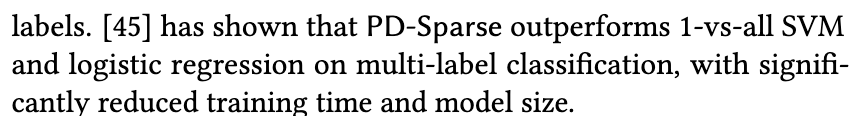
CNN-Kim [ 25 ] 是最早将卷积神经网络应用于文本分类的学者之一。CNN-Kim 用其词嵌入的串联构造一个文档向量，然后在卷积层中将 t 滤波器应用于该级联向量以生成 t 特征图，这些特征图又被馈送到最大随时间推移的池化层以构建 t 维文档表示。接下来是具有 L softmax 输出的全连接层，对应于 L 标签。在实践中，CNN-Kim在多类文本分类中表现出优异的表现，是我们比较评估的有力基线。



### 2.5 Bow-CNN

Bow-CNN [21]（Bag-of-word CNN）是多类分类中的另一种强方法。它使用词袋指示器向量（称为单热向量）表示每个小文本区域（几个连续的单词）。用 D 表示特征空间（词汇表）的大小，为每个区域构造一个 D 维二进制向量，其中第 i 个条目为 1，如果词汇表中的第 i 个单词出现在该文本区域中。所有区域的嵌入都经过卷积层，然后是一个特殊的动态池化层，该池化层将嵌入区域聚合到文档表示中，然后馈送到 softmax 输出层。

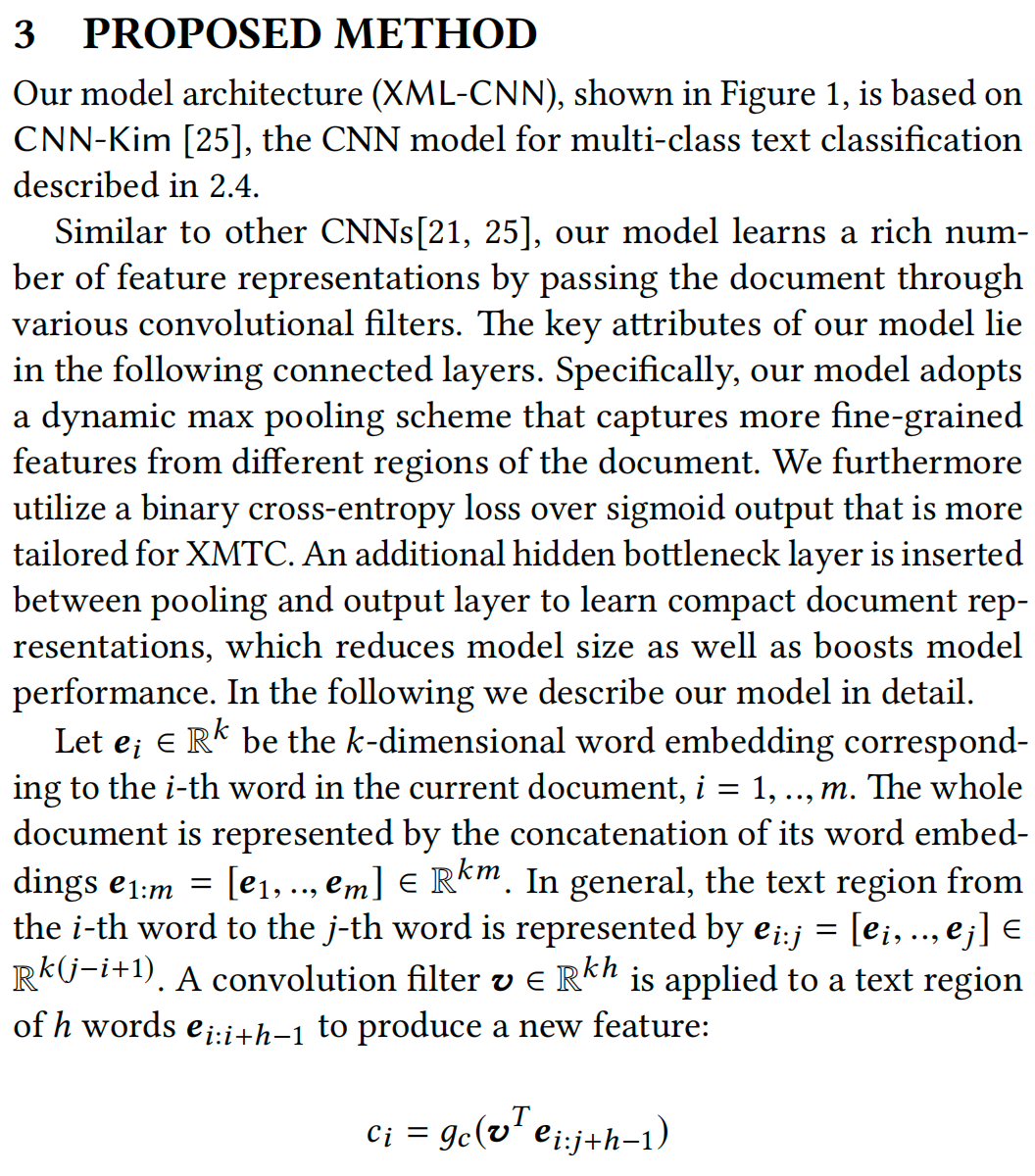


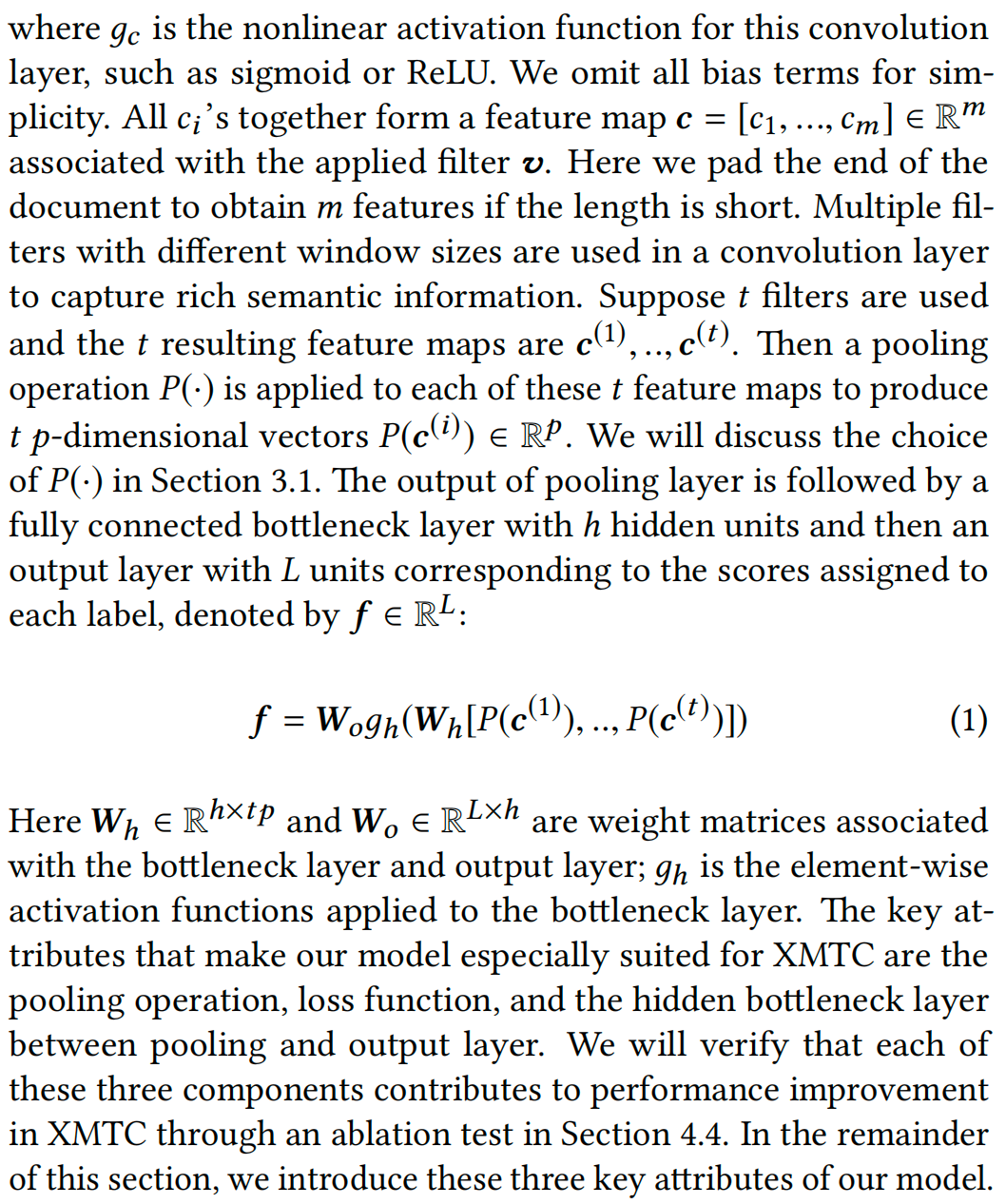


### 2.6 PD-Sparse

PD-Sparse [45] 是最近提出的一种最大边距方法，专为极端多标签分类而设计。它不属于第 1 节中的三个类别（目标嵌入方法、基于树的方法和深度学习方法）。在 PD-Sparse 中，为每个标签学习线性分类器，并在与此标签关联的权重矩阵上进行l1和l2惩罚。这导致解在原始空间和对偶空间中都非常稀疏，这在XMTC的时间和内存效率方面都是可取的。

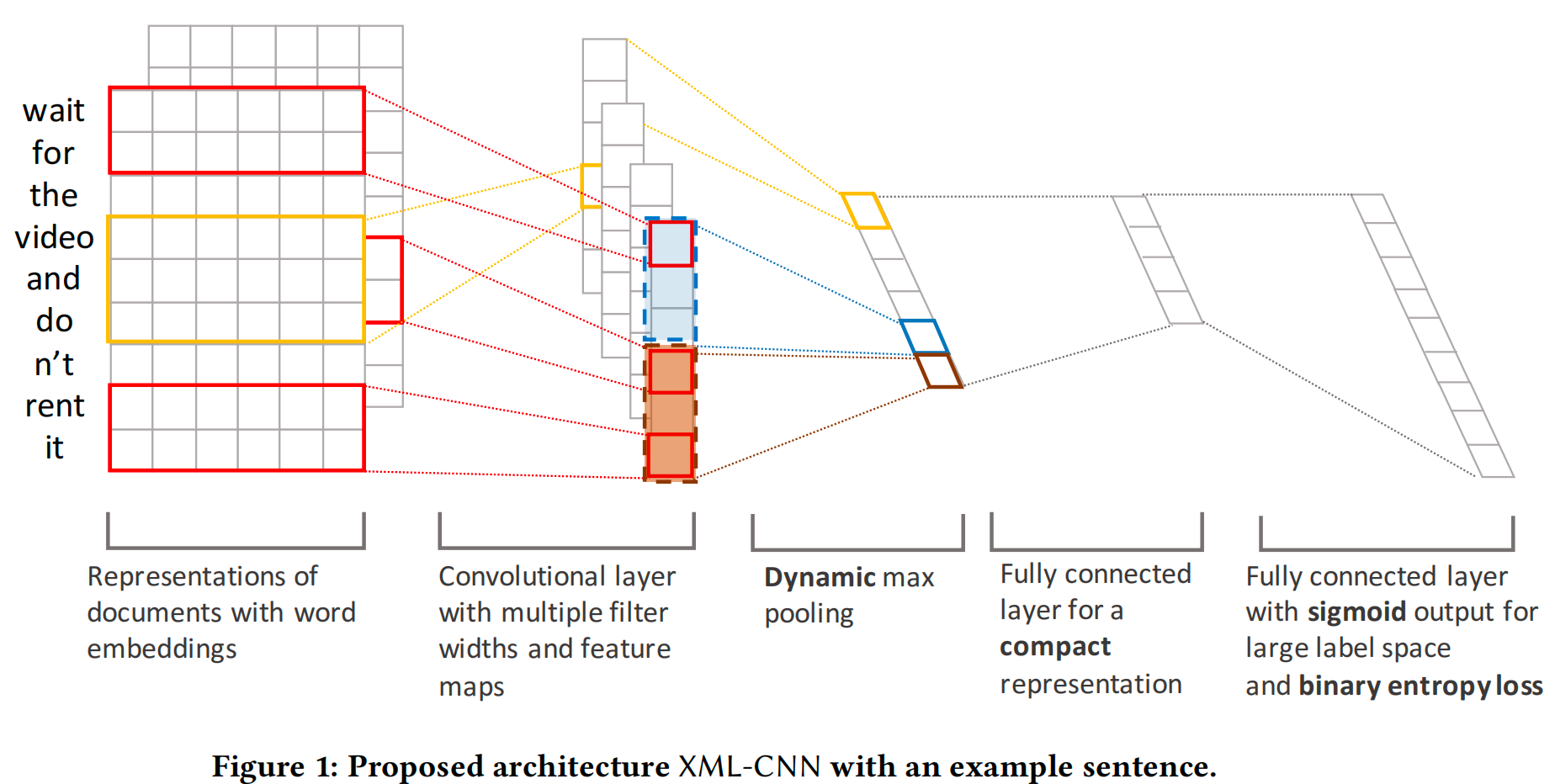
PD-Sparse 提出了一种完全校正块坐标 FrankWolfe 训练算法，该算法利用解中的稀疏性，实现 （μm） 和对偶变量的次线性训练时间，但预测时间仍然是线性的 （w.r.t）标签。[ 45 ] 表明PD-Sparse 在多标签分类上优于 1 对 All SVM 和 logistic 回归，训练时间和模型大小显著减少。





## 3 建议的方法（重点）

我们的模型架构 （XML-CNN） 如图 1 所示，基于 CNN-Kim [25] ，即 2.4 中描述的用于多类文本分类的 CNN 模型。



（表示使用 Word 的文档嵌入）（具有多个滤波器宽度和特征图的卷积层）（动态最大值池）（全连接层为紧凑的表示法）（具有 S 形输出的全连接层，用于大标签空间和二进制熵损失）

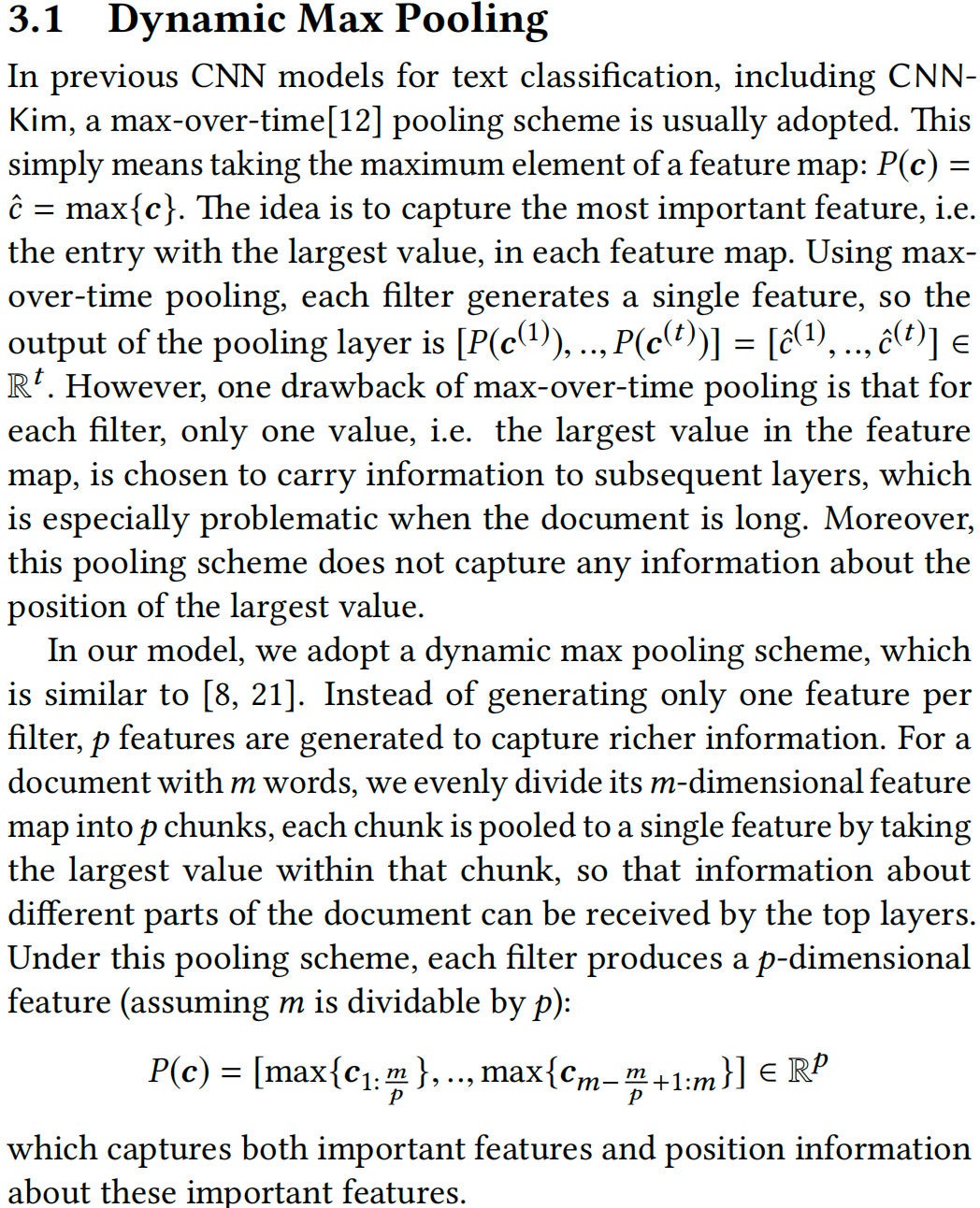
图 1：用一个示例表示建议的架构 XML-CNN

与其他CNN类似[21,25]，我们的模型通过使文档通过各种卷积滤波器来学习丰富的特征表示。我们模型的关键要素在于以下连接层。具体来说，我们的模型采用了动态的最大池化方案，从文档的不同区域捕获了更细粒度的特征。此外，我们还利用了二元交叉熵损耗，而不是更适合XMTC的S形输出。在池化层和输出层之间插入一个额外的隐藏 boleneck 层，以学习紧凑的文档表示，从而减小模型大小并提高模型性能。在下文中，我们将详细描述我们的模型。

设 e∈ Rbe 对应于当前文档中第 i 个字的 k 维词嵌入，i = 1， ..， m。整个文档由其单词嵌入 e= [e， ..， e] ∈ R 的串联表示。通常，从第 i 个单词到第 j 个单词的文本区域由 e= [e， ..， e] ∈ R 表示。将卷积滤波器 v ∈ Ris 应用于 h 个单词的文本区域以产生一个新特征：

其中 gc是该卷积层的非线性激活函数，例如 sigmoid 或 ReLU。为简单起见，我们省略了所有偏见术语。所有 c 共同形成与应用的过滤器 v 关联的特征图 c = [c， ...， c] ∈ R。在这里，我们填充文档的末尾，如果长度较短，则获得 m 个特征。在卷积层中使用具有不同窗口大小的多个过滤器来捕获丰富的语义信息。假设使用 t 过滤器，并且 t 生成的特征图为 c， ..， c。然后，将池化操作 P（·） 应用于这些 t 特征图中的每一个，以生成 t p 维向量 P（c） ∈ R。我们将在第 3.1 节中讨论 P（·） 的选择。池化层的输出之后是一个具有 h 隐藏单元的全连接柱颈层，然后是一个具有 L 个单元的输出层，该输出层对应于分配给每个标签的分数，用 f ∈ R 表示：

这里 W∈ Rand W∈ 与Boleneck层和输出层相关的稀有权重矩阵;GIS是应用于Boleneck层的元素激活函数。使我们的模型特别适合 XMTC 的关键属性是池化操作、损失函数以及池化层和输出层之间的隐藏柱颈层。我们将通过第 4.4 节中的消融测试来验证这三个组件中的每一个都有助于提高 XMTC 的性能。在本节的其余部分，我们将介绍模型的这三个关键参数。



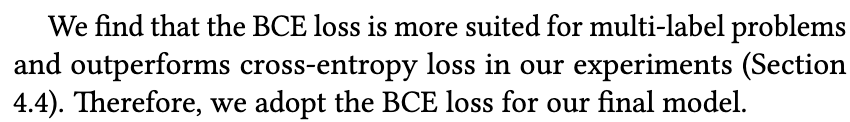
### 3.1 动态最大池化

在以前的CNN文本分类模型中，包括CNNKim，通常采用最大随时间[12]池化方案。这仅仅意味着取特征图的最大元素：P（c） = ˆc = 最大{c}。这个想法是在每个特征图中捕获最重要的特征，即具有最大值的条目。使用 maxover-time 池化时，每个滤波器生成单个特征，因此池化层的输出为 [P（c），..，P（c）] = [ˆc，..， ˆc] ∈ R。但是，最大随时间池化的一个缺点是，对于每个过滤器，只选择一个值，即特征图中的最大值，将信息传送到后续图层，这在文档很长时尤其成问题。此外，此池化方案不会捕获有关最大值位置的任何信息。

在我们的模型中，我们采用了动态最大池化方案，类似于 [8，21 ]。不是每个滤波器只生成一个特征，而是生成 p 个特征来捕获更丰富的信息。对于一个有 m 个字的文档，我们将其 m 维特征图平均分成 p 个块，每个块通过取该块中的最大值来汇集到一个特征中，以便顶层可以接收有关文档不同部分的信息。在这种池化方案下，每个滤波器都会产生一个 p 维特征（假设 m 可被 p 整除）：

它捕获重要特征并定位有关这些重要特征的信息。





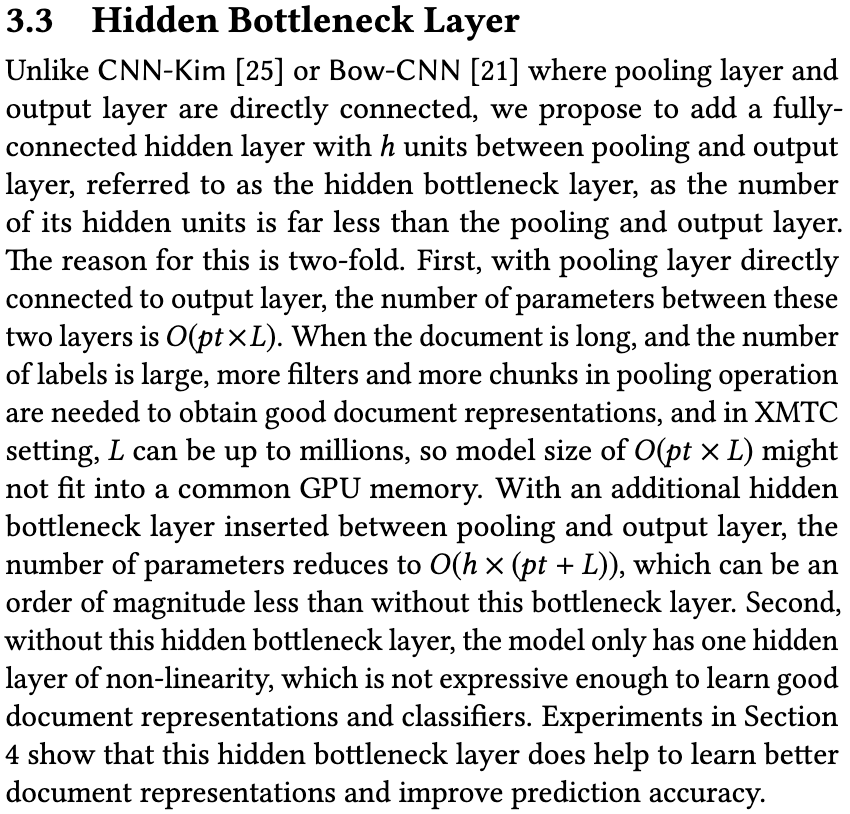
### 3.2 损失函数

从多类分类问题到多标签分类问题最直接的适应是扩展传统的交叉熵损失。具体来说，[ 16-18，25 ] 将扩展的交叉熵损失函数视为

其中 Θ 表示模型参数，y表示实例 i 的相关标签集，ˆpis 通过 softmax 激活对标签 j 上的实例 i 进行模型预测：

XMTC的另一个直观合理的目标是秩损失[47]，它最大限度地减少了相关和不相关标签的错误排序对的数量，即它旨在分配相关标签的分数高于不相关标签。然而，当应用于简单前馈神经网络中的多标签分类数据集时，秩损失已被证明不如二元交叉熵损失 （BCE） 而不是 sigmoid 激活[ 34]。二元交叉熵目标可以表述为：

我们发现BCE损失更适合多标签问题，并且在我们的实验中优于交叉熵损失（第4.4节）。因此，我们采用BCE损失作为最终模型。



### 3.3 隐藏瓶颈层

与池化层和输出层直接连接的 CNN-Kim [ 25 ] 或 Bow-CNN [ 21 ] 不同，我们建议在池化层和输出层之间增加一个具有 h 个单位的全连接隐藏层，称为隐藏 boleneck 层，因为其隐藏单元的数量远远少于池化和输出层。原因有两方面。首先，当池化层直接连接到输出层时，这两层之间的参数数为 O（pt ×L）。当文档较长且标签数量较大时，需要更多的过滤器和更多的块进行池化操作才能获得良好的文档表示，而在 XMTC seing 中，L 可以达到数百万个，因此 O（pt × L） 的模型大小可能不适合常见的 GPU 内存。在池化层和输出层之间插入一个额外的隐藏柱颈层，参数数量减少到 O（h × （pt + L）），这比没有这个柱颈层要小一个数量级。其次，如果没有这个隐藏的boleneck层，模型只有一个隐藏的非线性层，它的表现力不足以学习良好的文档表示和分类器。第 4 节中的实验表明，这种隐藏的boleneck层确实有助于学习文档表示并提高预测准确性。