计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(11):54−62 [doi: 10.15888/j.cnki.csa.008180]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于局部注意力机制的弱监督长文档分类①

马雯琦[1], 何　跃[2]

[1](中国科学技术大学 管理学院, 合肥 230026)
[2](四川大学 商学院, 成都 610065)
通讯作者: 马雯琦, E-mail: marchyvt@mail.ustc.edu.cn

摘　要: 自然语言处理中的文档分类任务需要模型从低层级词向量中抽取高层级特征. 通常, 深度神经网络的特征抽取会利用文档中所有词语, 这种做法不能很好适应内容较长的文档. 此外, 训练深度神经网络需要大量标记数据, 在弱监督情况下往往不能取得良好效果. 为迎接这些挑战, 本研究提出应对弱监督长文档分类的方法. 一方面, 利用少量种子信息生成伪文档以增强训练数据, 应对缺乏标记数据造成的精度难以提升的局面. 另一方面, 使用循环局部注意力学习, 仅基于若干文档片段抽取出摘要特征, 就足以支撑后续类别预测, 提高模型的速度和精度. 实验表明, 本研究提出的伪文档生成模型确实能够增强训练数据, 对预测精度的提升在弱监督情况下尤为显著; 同时, 基于局部注意力机制的长文档分类模型在预测精度上显著高于基准模型, 处理速度也表现优异, 具有实际应用价值.

关键词: 文档分类; 深度学习; 弱监督学习; 伪文档; 局部注意力机制

引用格式: 马雯琦,何跃.基于局部注意力机制的弱监督长文档分类.计算机系统应用,2021,30(11):54–62. http://www.c-s-a.org.cn/1003-3254/8180.html

Weakly-Supervised Long Document Classification Based on Local Attention Mechanism

MA Wen-Qi[1], HE Yue[2]

[1](School of Management, University of Science and Technology of China, Hefei 230026, China)
[2](Business School, Sichuan University, Chengdu 610065, China)

Abstract: The task of document classification in natural language processing requires the model to extract high-level features from low-level word vectors. Generally, the feature extraction of deep neural networks uses all the words in the document, which is not well suited for documents with long content. In addition, training deep neural networks requires massive labeled data, which often fails to achieve satisfied results under weak supervision. To meet these challenges, this research proposes a method to deal with weakly-supervised long document classification. On the one hand, a small amount of seed information is used to generate pseudo-documents to enhance training data to deal with the situation where accuracy is difficult to improve due to the lack of labeled data. On the other hand, using recurrent local attention learning to extract summary features based on only a few document fragments is sufficient to support subsequent category prediction and improve the model's speed and accuracy. Experiments show that the pseudo-document generation model can indeed enhance the training data, and the improvement in prediction accuracy is particularly significant under weak supervision. At the same time, the long document classification model based on the local attention mechanism performs significantly better than benchmark models in prediction accuracy and processing speed, with practical application value.

Key words: document classification; deep learning; weakly-supervised learning; pseudo-document; local attention mechanism

① 基金项目: 国家自然科学基金 (71571174)
Foundation item: National Natural Science Foundation of China (71571174)
收稿时间: 2021-01-30; 修改时间: 2021-03-05; 采用时间: 2021-03-16; csa 在线出版时间: 2021-10-22

文本分类是自然语言处理中一类重要任务[1], 包括主题标注[2]、情感分类[3]、垃圾邮件检测[4] 等应用. 文档是文本的一种常见形式, 现实世界中, 存在大量长文档, 例如学术期刊论文、商业报告和书籍等. 文档分类可用于满足检索、查询等需求, 也可用于识别并过滤暴力、色情、种族歧视等劣质内容. 从内容结构上看, 文档具有层级结构, 即词语组成句子, 句子组成文档[1]. 文档的特征提取和向量表示是文档分类中的关键[5]. 传统的文本分类方法, 例如词袋模型[6]、n-Grams[7] 使用统计学方法把文档表示为 N 维特征向量, 并交由 SVM[8] 或朴素贝叶斯[9] 模型来做分类. 这些方法简单稳健, 但是忽略了文档的结构信息, 丢弃了部分句子间和词语间的关系.

近年来, 基于深度学习的方法展示出相比于传统分类方法的优越性. Kim[10] 使用卷积神经网络 (CNN) 的变体模型自动地从由词向量表示的句子中抽取特征, 再把特征输入到分类网络中预测句子类别. Lee 等人[11] 提出一种同时使用循环神经网络 (RNN) 和 CNN 的模型, 能够序列地对抽取到的卷积特征进行编码, 该模型在短文本分类任务上取得最高水平的 (state-of-the-art) 结果. 这些模型在进行特征抽取时, 使用整个文本的内容, 在处理句子或者影评、短信等形式的短文本时, 计算开销是可以容忍的. 然而, 如果直接套用在长文档上, 一方面, 由于长文档的内容量往往是普通短文本的成百上千倍, 表示长文档的特征向量将是超高维的, 极大增加内存压力; 另一方面, 参照人类阅读习惯, 人类仅根据少数重要的文档片段便能判断文档类别, 所有词语或句子对文档分类结果的贡献并不是均等的[12], 那么使用整个文档内容构建表示向量就是不必要的. 从这个角度出发, 如何从文档中选取出最重要的、包含判别信息的文档片段, 是设计可行且高效的文档分类方法的关键. Yang 等人[1] 提出用于文档分类的层级注意力网络 (HAN), 与文档的层级结构相对应, HAN 也具有层级结构, 它的两级注意力机制分别关注句子层面和词语层面, 使模型在构造文档表示时对重要性强和重要性弱的内容做不同程度的关注. 与 Yang 等人的思想相类似, He 等人[12] 同样使用注意力机制筛选重要的文档信息, 模型性能显著高于基准模型 Kim[10]. 然而, He 等人[12] 中用于抽取文本特征的 CNN 与基准模型 Kim[10] 的设计完全相同, 侧重提取短语范围内的词语关系, 可能不擅长处理包含长句的文档片段; 此外, He 等人[12] 使用不可微分的硬注意力机制[13], 大大增加模型训练难度. 本研究对 He 等人[12] 的模型做出改进, 提出基于局部注意力[14]的长文档分类 (Local-Attention-based Long Document Classification, LALDC) 模型, LALDC 利用循环注意力学习, 序列地预测重要文档片段的位置, 并抽取该位置附近的文本特征, 经过若干次循环, 抽取到的特征向量被整合起来作为文档的摘要特征, 输入后续分类网络, 获得预测结果. LALDC 的设计结构使它能够很好地适应长文档分类任务, 原因在于, LALDC 没有使用文档所有的细节信息[15], 仅仅使用少数文档片段构造判别特征, 就足以支撑分类任务.

此外, 训练深度神经网络需要大量标记数据, 而获取标记数据往往成本高昂, 缺乏标记数据造成的弱监督是阻碍模型性能提升的瓶颈. 目前, 对弱监督学习还没有严格的定义. Zhou[16] 把弱监督分为 3 种类型: (1) 不完全监督, 训练数据中仅有一小部分带有标签; (2) 不确切监督, 训练数据的标签是粗粒度的; (3) 不精确监督, 训练数据的标签不总是真实的. 本研究面临的弱监督属于第 1 种类型. Meng 等人[17] 提出一种弱监督方法, 解决使用深度学习做文本分类任务时经常面临的数据缺失问题. 该方法由两个模块构成: (1) 伪文档生成模块, 利用种子信息 (例如少量标记文档、类别关键词、类别名称等) 生成伪标记文档, 用于对基于 CNN 或 RNN 的模型做预训练; (2) 自训练模块, 在真实的未标记数据上通过 Bootstrap 方法对模型做参数微调. Meng 等人[17] 的方法能够灵活处理不同类型的弱监督信号, 并容易整合到现有深度神经网络模型中. 然而, 使用该方法生成的伪文档中, 词语之间没有前后语义关系, 并不符合真实文档的情况. 本研究借鉴 Meng 等人的伪文档生成模块, 并做出实质性改进, 提出伪文档生成 (Pseudo-Document Generation, PDG) 模型, 利用弱监督信号生成接近真实文档的伪标记文档, 能够增强训练数据, 提升模型精度.

综上所述, 本研究的贡献主要在两方面: PDG 模型用于生成伪文档, 缓解数据不足造成的模型精度难以提升的问题; LALDC 模型用于长文档分类, 解决过往的深度神经网络模型难以适应长文档的问题, 在模型精度、训练和推断速度上均具有实际应用的价值.

1 研究设计

本研究可拆分为伪文档生成、长文档分类两部分. 首先使用 PDG 模型生成伪文档以增强训练数据, 然后

使用 LALDC 模型预测文档所属类别. 图 1 展示了整个流程.

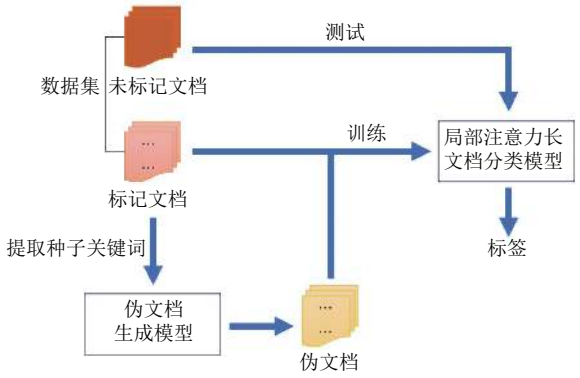

图 1 弱监督长文档分类的流程

1.1 伪文档生成模型

Meng 等人[17] 利用种子信息捕捉词语、文档、类别之间的语义相关性, 假设词语和文档共享一个联合语义空间, 基于这个空间学习出能够为每个类别生成伪文档的生成模型. 具体而言, 使用 Skip-Gram 模型[18] 学习出语料库中所有词语的p维向量表示并做归一化, 使得所有词向量都处在p维单位球空间上, 也就是联合语义空间. 对于标记文档形式的种子信息, 先使用 TF-IDF 加权法提取每个类别的种子关键词, 再从语义空间中寻找与种子关键词平均相似度最高的前 t 个类别关键词, 使用它们去拟合一个 von Mises Fisher (vMF) 分布[19], 也就是该类别在R^p中单位球上的语义模型.

伪文档生成过程中, Meng 等人[17] 使用了生成式混合模型: 从由背景分布 (整个语料库的分布) 和特定类别的 vMF 分布构成的混合分布中重复做单词采样来生成伪文档. 这个方法看似高效稳定, 但由于每次采样都是独立的, 导致生成的伪文档中词语乱序, 不符合一篇文档在意思表达上应该连贯通顺的认知. 为使生成的词语之间具有时序上的联系, 本研究使用基于门控循环单元 (GRU)[20] 的循环神经网络生成具有意义的伪文档. 具体而言, 基于整个语料库训练 GRU 语言模型, 对于每个类别, 从该类别的 vMF 分布中采样 1 个词嵌入向量, 并使用嵌入空间中最接近这个向量的词语作为序列的第 1 个词语, 然后把这个序列输入到 GRU 模型中生成下一个词语, 把它拼接到序列末尾并重复上个步骤, 直到序列长度等于一个事先设定好的参数, 最终的序列就作为该类别的一篇伪文档. 图 2 展示了整个流程.

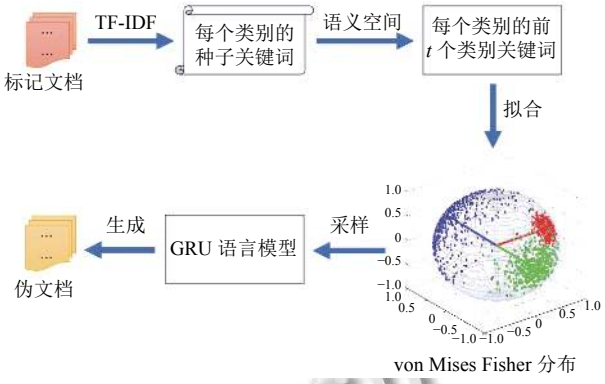

图 2 使用 PDG 模型生成伪文档的流程

1.2 基于局部注意力机制的长文档分类模型

LALDC 由 3 个子网络和 1 个初始特征组成: 摘要网络、定位网络、分类网络和初始文档表示, 模型架构如图 3 所示. 在每个时间步, 给定由定位网络输出的位置pos_t, 摘要网络从pos_t前后的文档片段所包含的词语中抽取局部特征. LALDC 的关键点在于能否准确找到包含判别信息的文档片段所在的位置, 定位网络通过循环注意力机制完成这一任务, 它基于观察到的上下文信息, 决定要把注意力放在文档的哪个片段, 使得摘要网络抽取到的信息足以支撑后续分类任务. 初始文档表示包含文档的初始上下文信息, 为定位网络提供了第 1 个输入, 使得定位网络在第 1 个时间步就能准确找出位置. 几轮特征抽取后, 得到原文档的摘要特征, 随后输入到分类网络中预测文档类别. 此外, LALDC 使用 Embedding 层把文档中的词语转换为词向量 (由预训练好的 GloVe[21] 表示).

(1) 摘要网络

在第 t 个时间步, 摘要网络从由位置pos_t决定的文档片段中抽取特征. 通过卷积和池化操作, 把词向量编码成特征向量, 并连结位置特征p_t, 一起映射为长度固定的特征s_t, 作为定位网络的输入. Kim[10] 提出用于句子分类的 CNN 模型, 抽取文本特征的卷积核窗口大小为 3、4、5, 模型在多个数据集上表现优异. Zhang 等人[22] 对用于句子分类的 CNN 做了全面实验分析, 指出对于非常长的句子, 应当尝试加入更大窗口的卷积核, 以捕捉更大范围内词语间的关系. He 等人[12] 使用与文献 [10] 中同样窗口大小的卷积核抽取文档片段的特征. 考虑到文档片段在长度上是超过普通句子的, 可能包含多个长句, 参考文献 [22], 本研究设计的摘要网络同样使用多核 CNN, 卷积核窗口大小为 3、5、7、10, 全面捕捉短语和长句的特征, 如图 4 所示.

(2) 定位网络

定位网络的核心是使用 GRU[20] 的循环神经网络和局部注意力机制[14]. 在第 t 个时间步, 定位网络的 GRU 接收由摘要网络生成的特征向量s_t作为输入, 结合隐藏状态h_t产生下一个状态向量h_{t+1}. 接下来, 注意力层根据状态向量h_{t+1}计算下一个应该被摘要网络“看到”的文档片段的位置pos_{t+1}. 通过这样的循环计算, LALDC 在若干次特征抽取后形成包含判别信息的摘要特征, 作为类别预测的依据. 提升模型性能的关键在于, 训练定位网络, 使它总是把注意力放在重要的文档片段.

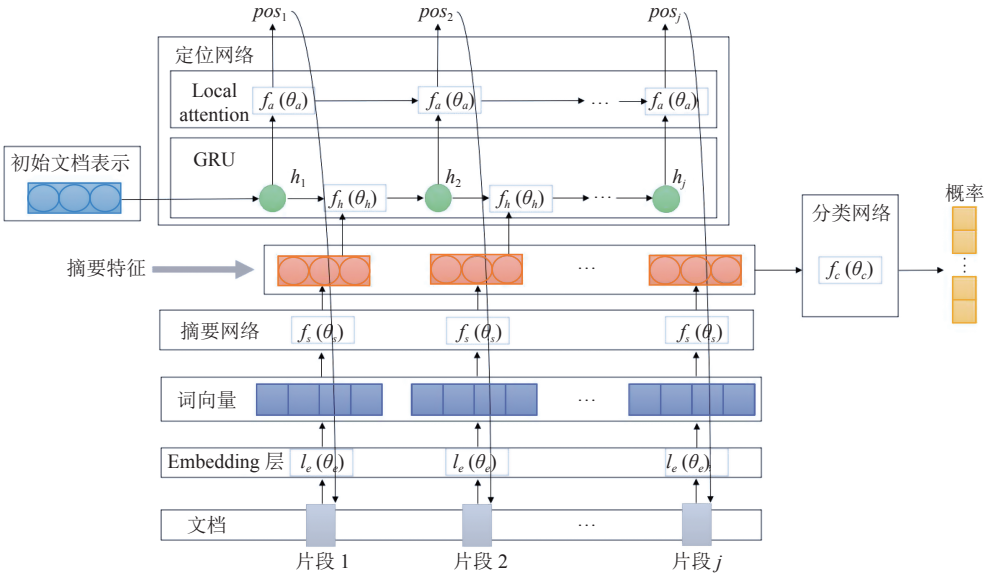

图 3 LALDC 的模型架构

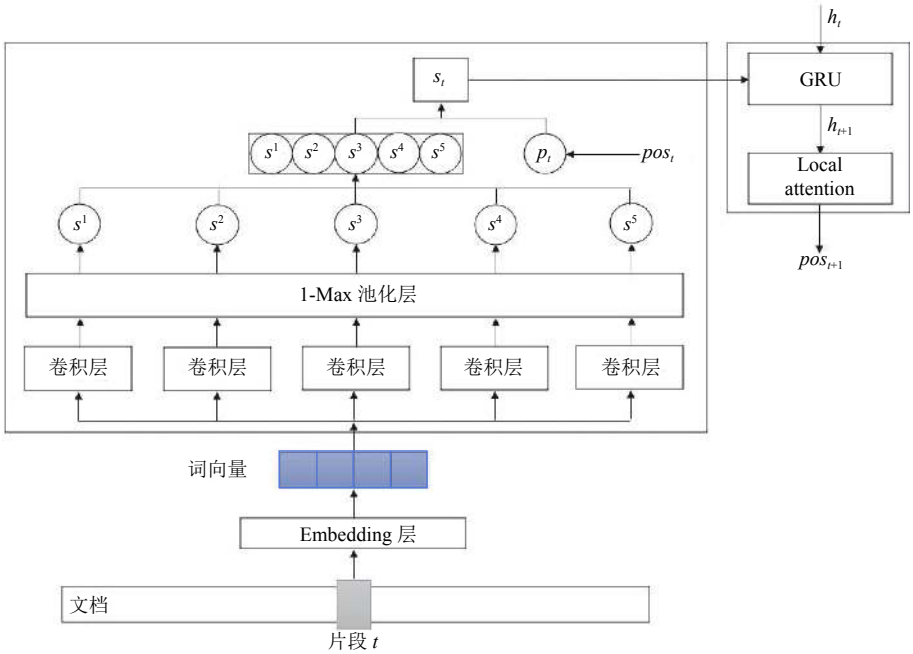

图 4 摘要网络

软注意力 (Soft Attention)[23] 针对文档中的每个句子或每个单词, 计算权重向量作为注意力概率分布, 非常适合短文本任务, 比如情感分析、句子分类等. 然而, 对于长文档, 由于句子和单词的数量过多, 为一整篇文

档计算注意力概率的开销过大, 不具有实践意义[14]. He 等人[12] 使用硬注意力 (Hard Attention) 机制[18] 来避免软注意力机制造成的庞大计算开销, 在每个时间步, 硬注意力直接从整个文档中随机采样一组词语, 提高了计算效率. 然而, 由于硬注意力的选取过程是离散的, 导致模型不可微分, 需要更复杂的技术 (比如减方差、强化学习) 来训练[14]. 局部注意力[14] 可以看作是软注意力和硬注意力的混合体, 它在计算上比软注意力开销更少, 同时, 与硬注意力不同, 局部注意力几乎处处可微, 更易于实施和训练. LALDC 的定位网络设计为循环局部注意力模型, 如图 5 所示.

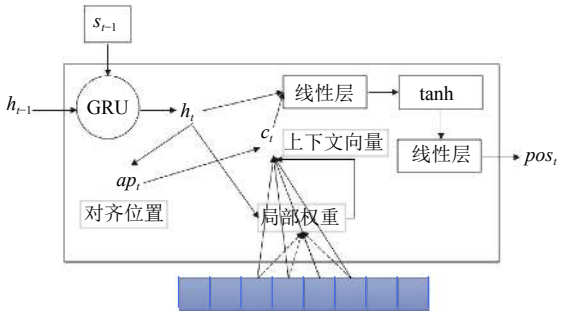

图 5 定位网络

每次循环局部注意力有选择性地关注一个较小的上下文窗口. 具体而言, 在第 t 个时间步, 模型根据隐藏状态h_t生成对齐位置 (aligned position) ap_t, 计算窗口 $[ap_t-D, ap_t+D]$内对象的局部权重 (注意力分布), D 根据经验来选择. 随后, 上下文向量c_t计算为该窗口内对象的加权平均 (这一步类似于软注意力). 最后, 预测接下来应该抽取特征的片段位置pos_t.

(3) 初始文档表示

由于定位网络需要文档的上下文信息来预测注意力的位置, 仅使用片段信息可能不够, 此外, 定位网络作出第 1 次预测之前, GRU 需要接收初始文档信息作为输入来产生第 1 个隐藏状态. 因此, 对原始文档做随机采样, 并编码成代表初始文档信息的向量, 作为定位网络的第 1 个输入. 具体而言, 从原始文档中随机采样 K 组词语 (每次采样的窗口大小相同), 每组词语由 Embedding 编码成词向量, 再输入到摘要网络中抽取特征, 初始文档表示s_0计算为 K 个特征向量的加和, 这一过程如图 6 所示.

(4) 分类网络

分类网络接收摘要网络经过若干轮循环后生成的摘要特征作为输入, 目标是正确预测文档所属类别, 该网络设计为典型的全连接层+Softmax 层的结构.

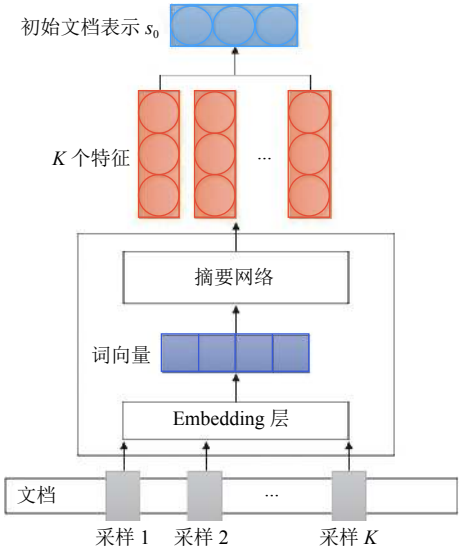

图 6 初始文档表示

(5) 模型优化

由于 LALDC 各个子网络都是可微分的, 因此使用典型的有监督学习训练方法, 通过最小化一个平滑交叉熵损失函数来优化参数, 如算法 1 所示.

算法 1. LALDC 参数优化算法

1. Input: 文档 *Doc*, 最大迭代次数 *maxIter*, 定位网络循环次数 T
2. for i = 1, 2, ⋯, *maxIter* do
   初始化摘要特征 S
   获取文档 *Doc* 的初始文档表示s_0, 输入到定位网络以预测pos_1
3. for t = 1, 2, ⋯, T do
   摘要网络在pos_t附近的文档片段上抽取特征s_t, 更新 S
   if $t < T$
   定位网络接收s_t作为输入, 预测pos_{t+1}
   else
   分类网络接收 S 作为输入, 预测文档 *Doc* 的类别
4. end for
5. 更新参数
6. end for
7. Output: 预测标签

2 实验

2.1 arXiv 数据集

常用的文档分类基准数据集, 例如 IMDB review、Yahoo Answer、Amazon review 等, 它们的平均单篇文档所含单词数都不超过 350 个词语, 也就是说这些数据集大部分是短文档, 并不符合一般长文档 (比如学术论文、书籍等) 的字数情况, 因此使用这些数据集来检

验本研究的模型是不适合的. He 等人[12] 使用自主收集的 arXiv 论文数据集来评估长文档分类模型, 为便于与基准模型做比较, 本研究同样使用该数据集.

arXiv 是一个收集物理学、数学、计算机科学和生物学等领域预发表论文的网站. arXiv 数据集包含 31 363 篇论文, 分为 10 个类别, 通过 arXiv sanity preserver 程序[24] 收集. 注意到某些类别属于同一个大类, 比如 cs.AI、cs.CE、cs.DS 等都属于计算机科学, 大大增加了正确区分不同类别论文的难度. 对数据的清洗包括: 只保留有意义的英语单词, 丢弃词数少于 1000 的文档, 过长的文档被截断为前 10 000 个词. 表 1展示了数据集的详细信息.

表 1 arXiv 数据集的统计信息

类别	文档数量	平均词数
cs.AI (Artificial Intelligence)	2995	6212
cs.CE (Computational Engineering)	3005	5777
cs.DS (Data Structures)	4136	7439
cs.IT (Information Theory)	3233	5938
cs.NE (Neural and Evolutionary)	3012	5856
cs.PL (Programming Languages)	2901	7012
cs.SY (Systems and Control)	3106	5948
math.AC (Commutative Algebra)	2885	5984
math.GR (Group Theory)	3065	6642
math.ST (Statistics Theory)	3025	6983

2.2 基准模型

使用两组基准模型. 一是 Kim[10] 中做句子分类的 CNN 模型, 为简洁, 称作 TextCNN; 二是 He 等人[12] 基于硬注意力机制的长文档分类模型, 称作 HeLDC. 为使 TextCNN 适合长文档分类, 对它稍做调整, 具体如下: 与本研究的 LALDC 基本思想类似, 从原始文档中随机抽取 K 个长度为 w 的片段, 把每个片段分别作为 TextCNN 的输入, 卷积池化后得到特征向量, 把这些向量做加和, 再输入到后续的分类网络做预测.

2.3 实验设置

(1) 文档表示

使用开源的 GloVe[21] 作为初始化词向量, 向量维数为 100. 所有不在 GloVe 中的词语都被随机初始化, Embedding 层与后续网络一起联合训练 (jointly-train).

(2) 训练参数

使用 TensorFlow 搭建模型, 定位网络中 GRU 大小为 256, 局部注意力窗口 D 为 12; 摘要网络使用 3、5、7、10 四种窗口大小的卷积层, 每层有 100 个核; 使用 ADAM 优化算法[25], 初始学习率为 0.001, dropout 概率为 0.5, batch 大小为 64.

(3) 评价指标

arXiv 数据集各类别文档数量相差较小, 可采用精度 (accuracy) 评估模型性能.

2.4 结果分析

(1) 实验 1

为了验证弱监督情况下, 生成伪文档能够增强训练数据, 同时评估 LALDC 与基准模型的性能, 设计以下实验. 从 arXiv 数据集中选取 0.2 比例的文档作为训练集 TrainSet, 剩余文档作为测试集 TestSet. 使用 PDG 模型为每个类别生成 200、300、400、500 篇伪文档, 和 TrainSet 一起构成 5 组训练数据, 即 TrainSet、TrainSet+200 篇伪文档、TrainSet+300 篇伪文档等依此类推. 使用这 5 组数据训练 LALDC 和基准模型, 每次抽取特征的文档片段窗口大小 w 为 40, 抽取次数为 14. 使用 TestSet 检验模型性能, 实验结果如图 7 所示.

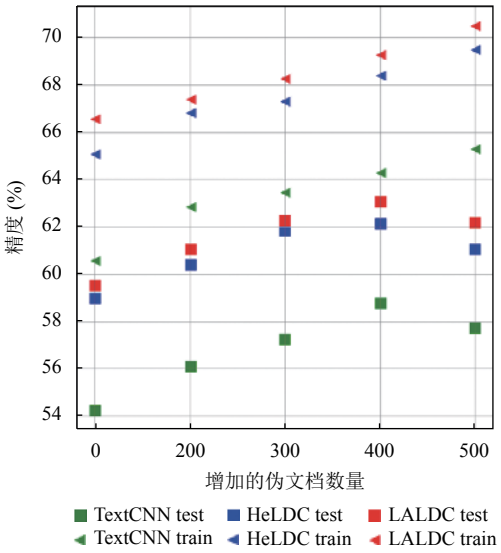

图 7 不同数量的伪文档对训练数据的增强作用

结果表明, 伪文档确实能够增强训练数据, 提升模型训练效果. 此外, 如果生成的伪文档数量过少, 则没有充分利用种子信息, 对模型性能提升较小, 如果生成的伪文档数量过多, 则会携带过多噪声, 降低泛化能力, 在本次实验中, 生成 400 篇伪文档是比较好的选择. 比较 3 种模型, TextCNN 明显比 HeLDC 和 LALDC 逊色很多. 从模型结构分析, 同样是使用 CNN 抽取文本特征, 影响性能的关键在于正确选择出包含判别信息的文档片段. 对于 TextCNN, 实验设计为随机从原始文档中抽取片段作为输入, 而并没有去“预测”重要的片段应该在的位置. HeLDC 和 LALDC 都使用循环注意力

机制去序列地预测下一个应被看到的文档片段, 毫无疑问, 这种选取方式更可能捕捉到判别信息, 支撑后续分类任务. LALDC 的表现略优于 HeLDC, 原因在于本研究设计的 CNN 层 (摘要网络) 能够同时抽取短语和长句范围内的词语关系, 而 HeLDC 只能捕捉短语级特征; 此外, 本研究使用局部注意力机制, 注意力向量是窗口内对象的加权平均, 相比于 HeLDC 使用的硬注意力机制, 局部注意力向量包含更大范围的信息, 有助于更准确地预测重要文档片段的位置.

(2) 实验 2

为了评估本研究基于 GRU 语言模型的 PDG 与 Meng 等人[17] 的生成式混合模型 (以下简称为 GM 模型) 在伪文档质量上的差异, 同时观察不同监督强度下伪文档的数据增强作用, 设计以下实验. 从 arXiv 数据集中选取 0.2–0.8 之间不同比例的文档作为训练集 TrainSet, 剩余文档作为测试集 TestSet, 构成不同标记率的数据集. 参考实验 1 的结果, 对于每个数据集, 分别使用 PDG 模型和 GM 模型为每个类别生成 400 篇伪文档, 和 TrainSet 一起构成 3 组训练数据, 即 TrainSet、TrainSet+PDG 生成的伪文档、TrainSet+GM 生成的伪文档. 使用这 3 组数据训练 LALDC 和基准模型, 每次抽取特征的文档片段窗口大小 w 为 40, 抽取次数为 14. 使用 TestSet 检验模型性能, 实验结果如图 8 所示.

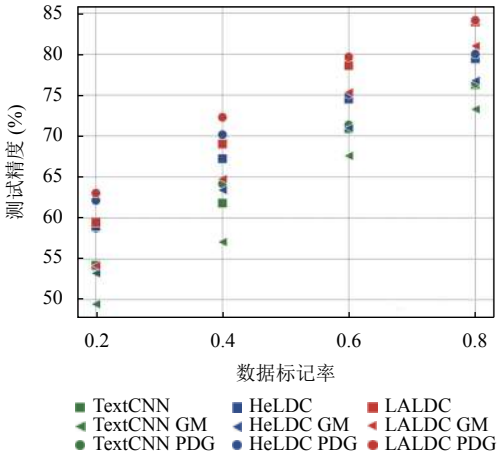

图 8 不同数据标记率下两种伪文档对测试精度的影响

结果表明, 在数据标记率较低时 (0.2 和 0.4), 使用 PDG 生成的伪文档对测试精度具有显著提升效果, 而在所有数据标记率下, 使用 GM 生成的伪文档反而降低了测试精度. 从模型结构分析, PDG 和 GM 都使用 vMF 分布建立语义模型, 差别在于后续生成伪文档的方式, PDG 使用 GRU 语言模型生成的伪文档语句间具有前后语义关系, 更接近真实文档, 对训练数据有切实的增强作用; 而 GM 通过从混合分布中重复做词语采样来生成伪文档, 实质上是词语的无序堆砌, 类似于真实文档被打乱语序后的样子, 反而会给训练数据带来噪声. 因此, PDG 生成的伪文档质量更高, 更适合于本研究的分类任务. 此外, 观察到随着数据标记率提高, PDG 伪文档对测试精度的提升幅度逐渐减弱, 也就是说, 强监督情况下, 训练集本身已能够提供充足信息, 没有伪文档并不影响模型训练, 弱监督情况下, 伪文档能够发挥更明显的作用, 缓解缺乏训练数据造成的困扰.

(3) 实验 3

把重点放在长文档分类模型本身, 为了观察调整窗口 w 的大小、抽取特征的次数会如何影响模型性能, 同时检验 LALDC 的运行效率, 设计以下实验. 从 arXiv 数据集中选取 0.8 比例的文档作为训练集 TrainSet, 剩余文档作为测试集 TestSet, 参考实验 2 的结果, 不使用伪文档. 设计 20、40 两种窗口大小, 10、14、18 三种抽取次数, 共 6 种组合, 观察 LALDC 和基准模型基于看到的 200、280、360、400、560、720 个词语对文档做类别预测的精度, 同时记录模型在训练和推断阶段的用时. 实验结果如图 9、表 2 所示.

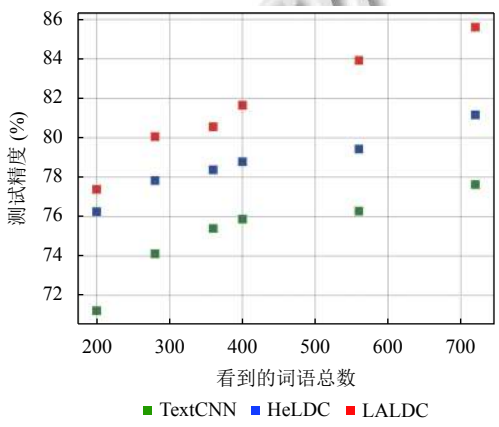

图 9 模型看到的词语总数对测试精度的影响

结果表明, 给定抽取特征的次数, 更大的摘要窗口总是产生更高的测试精度, 总体而言, 模型看到的词语数量越多, 抽取到的特征越丰富, 预测精度越高. 与基准模型相比, LALDC 在看到同样数量词语的情况下, 测试精度显著高于 TextCNN 与 HeLDC. 此外, HeLDC、TextCNN 在看到 720 个词语时分别产生 81.2%、77.6%

的精度, 与 HeLDC 相比, LALDC 仅看到一半数量 (360 个) 词语就能产生 80.5% 的精度, 与 TextCNN 相比, LALDC 仅看到不足 1/3 (即 200 个) 词语就能产生 77.4% 的精度. 也就是说, LALDC 只需看到数量少得多的词语, 就能达到与基准模型同等的预测精度. 观察 LALDC 与基准模型的精度差距, 发现当窗口 w 变大时 (由 20 变为 40), 差距明显增加, 即 LALDC 在使用大窗口时优势更加显著. 这与本研究设计的摘要网络密切相关, 根据前文分析, LALDC 能够抽取长句范围内的词语关系, HeLDC 不具备这个能力. 当窗口大小为 40 时, 定位的文档片段更可能包含完整长句, LALDC 的优势就展现出来了.

表 2 模型运行效率 (窗口大小为 40)

模型 (特征抽取次数)	每批次训练用时 (s)	每批次测试用时 (s)	训练总时长 (min)	测试精度 (%)
TextCNN (10)	0.15	0.12	14.8	75.87
TextCNN (14)	0.17	0.13	16.3	76.28
TextCNN (18)	0.20	0.15	21.4	77.63
HeLDC (10)	0.62	0.47	50.7	78.79
HeLDC (14)	0.68	0.55	58.2	79.43
HeLDC (18)	0.81	0.67	74.6	81.16
LALDC (10)	0.57	0.39	45.6	81.65
LALDC (14)	0.61	0.43	51.5	83.92
LALDC (18)	0.69	0.51	62.9	85.61

最后, 在模型效率和实践性方面, TextCNN 由于模型结构简单, 在训练时收敛速度最快, 但是最多能达到 77.63% 的预测精度. HeLDC 虽然训练用时较长, 但它最高能达到 81.16% 的预测精度, 弥补了训练收敛速度慢的缺陷. 本研究提出的 LALDC 比两个基准模型的预测精度更高, 但在训练用时上介于两者之间, 原因在于, LALDC 相较于 HeLDC, 模型参数数量更少, 因此更易于训练. 例如, 同样是使用循环注意力机制, HeLDC 中的循环神经网络使用 LSTM 单元, 而 LALDC 使用 GRU, 这是因为 GRU 的实验效果与 LSTM 相似, 但是更易于计算[26]. 更进一步地, LALDC 在预测精度最高时, 每批次 (64 个样本) 的训练和测试用时分别为 0.69 s、0.51 s, 意味着 LALDC 在训练和测试阶段每秒钟分别处理 93 篇、125 篇长文档, 这样的计算效率应该是可以被实际应用接受的.

3 结论与展望

本研究提出弱监督情况下基于局部注意力机制的长文档分类方法, 在伪文档生成和长文档分类两方面均做出贡献. PDG 利用少量种子信息产生能够增强训练数据的伪文档, 在弱监督情况下显著提升预测精度. LALDC 使用局部注意力机制, 既能规避软注意力庞大的计算开销, 又比硬注意力更易于训练. 通过循环注意力学习, LALDC序列地预测重要文档片段所在位置并抽取特征, 直到摘要特征能够支撑后续分类任务. 与两个基准模型相比, LALDC 在观察到更少文档信息的情况下, 就达到基准模型的预测精度, 而在观察到充足文档信息时, LALDC 的性能显著高于基准模型. 同时, LALDC 的训练和测试用时处于两个基准模型之间, 具有实际应用的价值.

受欧阳文俊和徐林莉[27]、Yang 等人[1] 提出的层级注意力模型的启发, 在未来研究中, 可以对 LALDC 中的定位网络做进一步改进. 考虑使用层级注意力机制, 同时关注句子层面和单词层面. 这一尝试可能使分类模型进行更细粒度的计算, 不仅定位出文档中重要的句子, 还能从句子中抽取重要词汇的信息.

参考文献

1 Yang ZC, Yang DY, Dyer C, *et al*. Hierarchical attention networks for document classification. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego: Association for Computational Linguistics, 2016. 1480–1489. [doi: 10.18653/v1/N16-1174]

2 Wang SD, Manning CD. Baselines and bigrams: Simple, good sentiment and topic classification. Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics. Jeju Island: Association for Computational Linguistics, 2012. 90–94.

3 Maas AL, Daly RE, Pham PT, *et al*. Learning word vectors for sentiment analysis. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland: Association for Computational Linguistics, 2011. 142–150.

4 Sahami M, Dumais S, Heckerman D, *et al*. A Bayesian approach to filtering junk E-mail. Proceedings of 1998 AAAI Workshop on Learning for Text Categorization. Madison: AAAI, 1998. 55–62.

5 陈杰, 陈彩, 梁毅. 基于 Word2Vec 的文档分类方法. 计算机系统应用, 2017, 26(11): 159–164. [doi: 10.15888/j.cnki.csa.006055]

6 Wallach HM. Topic modeling: Beyond bag-of-words. Proceedings of the 23rd International Conference on

Machine Learning. Pittsburgh: ACM, 2006. 977–984. [doi: 10.1145/1143844.1143967]
7 Damashek M. Gauging similarity with n-Grams: Language-independent categorization of text. Science, 1995, 267(5199): 843–848. [doi: 10.1126/science.267.5199.843]
8 Joachims T. Transductive inference for text classification using support vector machines. Proceedings of the 16th International Conference on Machine Learning. San Francisco: ACM, 1999. 200–209.
9 McCallum A, Nigam K. A comparison of event models for naive Bayes text classification. Proceedings of 1998 AAAI Workshop on Learning for Text Categorization. Madison: AAAI, 1998. 41–48.
10 Kim Y. Convolutional neural networks for sentence classification. Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing. Doha: Association for Computational Linguistics, 2014. 1746–1751. [doi: 10.3115/v1/D14-1181]
11 Lee JY, Dernoncourt F. Sequential short-text classification with recurrent and convolutional neural networks. Proceedings of 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. San Diego: Association for Computational Linguistics, 2016. 515–520. [doi: 10.18653/v1/N16-1062]
12 He J, Wang LQ, Liu L, *et al*. Long document classification from local word glimpses via recurrent attention learning. IEEE Access, 2019, 7: 40707–40718. [doi: 10.1109/ACCESS.2019.2907992]
13 Xu K, Ba J, Kiros R, *et al*. Show, attend and tell: Neural image caption generation with visual attention. Proceedings of the 32nd International Conference on International Conference on Machine Learning. Lille: ACM, 2015. 2048–2057.
14 Luong MT, Pham H, Manning CD. Effective approaches to attention-based neural machine translation. Proceedings of 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: Association for Computational Linguistics, 2015. 1412–1421. [doi: 10.18653/v1/D15-1166]
15 Zhou BL, Khosla A, Lapedriza A, *et al*. Learning deep features for discriminative localization. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 2921–2929. [doi: 1109/CVPR.2016.319]
16 Zhou ZH. A brief introduction to weakly supervised learning. National Science Review, 2018, 5(1): 44–53. [doi: 10.1093/nsr/nwx106]
17 Meng Y, Shen JM, Zhang C, *et al*. Weakly-supervised neural text classification. Proceedings of the 27th ACM International Conference on Information and Knowledge Management. Torino: ACM, 2018. 983–992. [doi: 10.1145/3269206.3271737]
18 Mikolov T, Sutskever I, Chen K, *et al*. Distributed representations of words and phrases and their compositionality. Proceedings of the 26th International Conference on Neural Information Processing Systems. Lake Tahoe: ACM, 2013. 3111–3119.
19 Banerjee A, Dhillon IS, Ghosh J, *et al*. Clustering on the unit hypersphere using von Mises-Fisher distributions. Journal of Machine Learning Research, 2005, 6: 1345–1382.
20 Cho K, van Merriënboer B, Gulcehre C, *et al*. Learning phrase representations using RNN encoder-decoder for statistical machine translation. Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing. Doha: Association for Computational Linguistics, 2014. 1724–1734. [doi: 10.3115/v1/D14-1179]
21 Pennington J, Socher R, Manning C. Glove: Global vectors for word representation. Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing. Doha: Association for Computational Linguistics, 2014. 1532–1543. [doi: 10.3115/v1/D14-1162]
22 Zhang Y, Wallace BC. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. Proceedings of the 8th International Joint Conference on Natural Language Processing. Taipei, 2017. 253–263.
23 Wang F, Jiang MQ, Qian C, *et al*. Residual attention network for image classification. Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 6450–6458. [doi: 10.1109/CVPR.2017.683]
24 Karpathy. Arxiv sanity preserver. https://github.com/karpathy/arxiv-sanity-preserver. (2019-04-25) [2021-01-01].
25 Kingma DP, Ba J. Adam: A method for stochastic optimization. Proceedings of the 3rd International Conference on Learning Representations. San Diego, 2015. 13.
26 Wang WH, Yang N, Wei FR, *et al*. Gated self-matching networks for reading comprehension and question answering. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: Association for Computational Linguistics, 2017. 189–198. [doi: 10.18653/v1/P17-1018]
27 欧阳文俊, 徐林莉. 基于层级注意力模型的无监督文档表示学习. 计算机系统应用, 2018, 27(9): 40–46. [doi: 10.15888/j.cnki.csa.006533]