计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(10):9−19 [doi: 10.15888/j.cnki.csa.007493]

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

基于多神经网络混合的短文本分类模型①

侯雪亮[1,2], 李　新[1], 陈远平[1]

[1](中国科学院 计算机网络信息中心, 北京 100190)
[2](中国科学院大学, 北京 100049)
通讯作者: 李　新, E-mail: lixin@cnic.cn

摘　要: 文本分类指的是在制定文本的类别体系下, 让计算机学会通过某种分类算法将待分类的内容完成分类的过程. 与文本分类有关的算法已经被应用到了网页分类、数字图书馆、新闻推荐等领域. 本文针对短文本分类任务的特点, 提出了基于多神经网络混合的短文本分类模型 (Hybrid Short Text Classical Model Base on Multi-neural Networks). 通过对短文本内容的关键词提取进行重构文本特征, 并作为多神经网络模型的输入进行类别向量的融合, 从而兼顾了 FastText 模型和 TextCNN 模型的特点. 实验结果表明, 相对于目前流行的文本分类算法而言, 多神经网络混合的短本文分类模型在精确率、召回率和 *F*1 分数等多项指标上展现出了更加优越的算法性能.

关键词: 深度学习; 短文本分类; 关键词提取; 特征重构; 神经网络; FastText; TextCNN

引用格式: 侯雪亮,李新,陈远平.基于多神经网络混合的短文本分类模型.计算机系统应用,2020,29(10):9–19. http://www.c-s-a.org.cn/1003-3254/7493.html

Short Text Classification Model Based on Multi-Neural Network Hybrid

HOU Xue-Liang[1,2], LI Xin[1], CHEN Yuan-Ping[1]

[1](Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China)
[2](University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: Text classification refers to the process of letting a computer learn to complete the classification of content by some classification algorithm under the classification system of text. Algorithms related to text classification have been applied to web classification, digital libraries, news recommendation, and other fields. Based on the characteristics of short text classification tasks, this study proposes a hybrid short text classical model based on multi-neural networks. By reconstructing the text features of the keywords extracted from the short text content, and using the vector fusion as the input of the multi-neural network model, the characteristics of the FastText model and the TextCNN model are taken into account. The experimental results show that compared with the current popular text classification algorithms, the multi-neural network hybrid short text classification model shows more superior algorithm performance on multiple indicators such as accuracy, recall, and *F*1 score.

Key words: deep learning; short text classification; keyword extraction; feature reconstruction; neural network; FastText; TextCNN

在 NLP 领域中, 与文本分类算法相关的研究受到了越来越多的关注. 文本分类的目的是通过对文本信息的挖掘与处理, 来解决信息紊乱的问题, 从而能帮助用户更加精准地定位所需信息, 并可根据已定的类别

① 基金项目: 中国科学院信息化建设专项 (XXH13504-01)
Foundation item: Special Project of Informatization of Chinese Academy of Sciences (XXH13504-01)
收稿时间: 2019-12-05; 修改时间: 2020-01-03; 采用时间: 2020-01-14; csa 在线出版时间: 2020-09-30

集合将其判定为已知的某一类. 目前, 在新闻推荐、情感评价、邮件分类等领域受到广泛应用. 文本分类的特点在于能在文本数据量不明确和复杂的情况下, 根据预先确定的类别信息, 输出成有价值的信息.

随着 machine learning 和 deep learning 等研究方法的不断完善, 文本分类问题的解决路径也从先前的向量空间模型 (VSM) 逐步转移至机器学习和深度学习结合的方法上来[1]. 在深度学习网络中, 卷积神经网络 CNN 可以识别出文本中预言性的 n 元语法; 卷积结构支持有相似成分的 n 元语法共同分享其预测行为, 即使是在预测的过程中遇见未曾登录过的特定 n 元语法也是可以的; 而层次化 CNN 每层则着眼在句子中更长的 n 元语法, 这样模型还可以对非连续的 n 元语法更加敏感, 可以对文本分类的效果产生显著的影响[2].

但是目前的主要研究均集中在长文本和富文本上, 对于短文本的研究较少. 短文本相比较于长文本和富文本而言, 其所含的信息量较少, 无法采用统计学观点进行词向量分析, 传统的神经网络对短文本的分类效果较差, 尤其是对于标签值较少的短文本而言, 比如新闻标题分类、知识库分类等, 其词汇量往往不到 50 个, 而标签值默认只有一个 (相互独立的分类标签)[3]. 所以, 本文提出了一种应用于短文本分类的模型, 利用 TextRank 算法提取短文本中的关键词, 然后进行值序列化分析和特征重构, 然后在基于 FastText 和 TextCNN 的混合短文本分类模型中进行类别向量的输出. 对比实验表明, 在短文本的分类实验上, 该模型较传统方法而言, 在准确率和处理速度上均有明显提升.

1 相关概念与基本理论

关于文本分类的模型主要分 3 大类: 基于概率统计、基于几何和基于统计, 如神经网络、贝叶斯方法、KNN、决策树 (Decision Tree)、支持向量机 (Support Vector Machines, SVM) 等.

1.1 基于概率的模型

基于概率的文本模型的核心思想是, 假设待分类的文本是 D, 它所属的类别集是$C=\{c_1,c_2,c_3,\cdots,c_m\}$, 基于概率模型的文本分类就是对 $1\le i\le n$ 计算出对应的条件概率$(c_i|d)$, 并且将条件概率中出现的最大的类别看作是该待分类文本的类别. 当前, 基于概率的文本分类模型应用最广的是朴素贝叶斯分类器[4].

朴素贝叶斯分类器 (NaiveBayes) 是一种较简单的分类算法, 它的思想基础是对于指定的待分类项, 求解出该项出现条件下, 类别出现的概率 P, 哪个类别的概率最大, 就视为该待分类项归属于哪个[5,6]. 基于贝叶斯分类器的贝叶斯规则如式 (1):

$$P(C|D)=\frac{P(D|C)P(C)}{P(D)}=\frac{P(D|C)P(C)}{\sum_{c_i\in C}P(D|C=c_i)P(C=c_i)} \tag{1}$$

其中, C 和 D 是随机变量.

首先计算文本 d 在每个给定类别下的概率$P(c_i|d)$, 其后根据概率值的比较, 概率最大的对应的类别即为文本 d 所属类别. 我们对文本 d 的计算如下:

$$Class(d)=\operatorname{argmax}_{c_i\in C}P(c_i|d)=\operatorname{argmax}_{c_i\in C}\frac{P(d|c_i)P(c_i)}{\sum_{c_i\in C}P(d|c_i)P(c_i)} \tag{2}$$

先验概率$P(c_i)$的计算如式 (3):

$$P(c_i)=\frac{N(c_i)}{N} \tag{3}$$

其中, $N(c_i)$是 train 样本中文本类别c_i的样本数量, N为 train 样本的总数, $P(c_i)$表示类别c_i在训练集中所占的比例.

概率$P(d|c_i)$是基于贝叶斯假设得出的: 文本 d 中各词组间是相互独立的. 因为表示被简化, 所以这就是“朴素”的缘由. 事实上, 词组并不是真正意义上的相互独立, 实验只是假设存在独立性. 在实际应用中, 朴素贝叶斯仍可在文本分类的处理中有较好的分类效果. 该假设给出了计算联合概率的方法, 并且联合概率可用条件概率乘积来表示.

$P(d|c_i)$的计算公式如式 (4):

$$P(d|c_i)=\prod_{k=1}^{n}P(t_k|c_i) \tag{4}$$

其中, t_k表示含有 n 个词组的表 V_i 中的某一词组, 所以, 计算$P(d|c_i)$转为计算词组表 V 中每个词组在每个类别下的概率$P(t_k|c_i)$.

1.2 基于几何的模型

基于几何的模型方法是以向量空间模型为基本出发点, 把文本表示为多维的空间向量, 其为多维空间里的一个点. 构建一个拥有分类性质的超平面, 它能把划分出各个类别. 其中最经典的分类器是支持向量机, 而最简单的 SVM 本质上就是一个二分类器, 可以用来区分常见的正反例数据. SVM 可以构建一个区分类别的

N 维空间的决策超平面[6,7], 并且该超平面可以平行移动, 且不会造成错误的分类. 因此, 为了保证预测时的稳定性和鲁棒性, 我们会希望该超平面与样本的距离足够大, 即该超平面能够在边界区域的中界上. 这个理论最初是由 Vapnik 等人提出的, 并被发展成经典的统计机器学习理论, 在二分类问题上拥有非常高效的处理能力.

SVM 中采用了结构风险最小原则, 这项原则的提出是以 support vector 为基础, 在多维空间里确定一个能把样本分成两个类别, 且具有最大边缘值, 达到最大化的分类精确度, 这样的超平面为最优超平面[8].

1.3 基于统计的模型

在 NLP 研究领域中, 基于统计学习的方法称为了主流, 其中最典型的就是 K 最近邻分类算法 (KNN). 它是一种基本的分类与回归方法. 其基本思想是, 给定测试用例, 用距离去度量找出训练数据中和它最近的 K 个实例, 然后用这 K 个最近邻的实例来做预测处理, 最终确定测试用例的类别[6–9].

KNN 算法的过程描述:

1) 计算测试用例和各个训练用例间的距离;

2) 对上述计算得出的距离进行排序;

3) 选择距离绝对值最小的 K 个用例;

4) 计算这 K 个用例对应类别的频率;

5) 选择频率最高的类别作为该测试用例的类别.

2 基于多神经网络混合的短文本分类模型

2.1 关键词提取与特征重构

从短文本中对关键词进行提取, 首先需要考虑到短文本与长文本的不同之处在于, 在短文本里每个关键词的频数较小, 传统的关键词提取模型如 SKE、RAKE、LDA 和 TF-IDF 等则需对大量文本采用统计学分析, 从而得到相应的词频向量. 所以在短文本中关键词的提取方法可以用 TextRank 进行处理.

TextRank 模型时主要用于文本的排序算法, 它是基于图特征构建的, 并有 Google 的 PageRank 模型演化而来. 其主要思想就是构建一个特征图, 把文本里出现的词看作节点, 边与边连接, 并且节点和权重一一对应, 权重高的节点可以作为关键字[10,11].

TextRank 的计算公式如下:

$$WS(V_i)=(1-d)+d*\sum_{V_j\in In(V_i)}\frac{\omega_{ji}}{\sum\limits_{V_k\in Out(V_j)}WS\left(V_j\right)} \quad (5)$$

其中, d 是阻尼系数, 取值范围 0 到 1, 它表示从图中某一点指向任意点的概率, 一般取 0.85. 用 TextRank 算法估计各点的得分时, 需给图中的点初始化任意的数值, 并且递归计算达到收敛, 即图中任意点的误差率小于极限值时就视为达到收敛, 一般情况极限值取 0.0001[12].

TextRank 在短文本中关键词提取的算法如下:

1) 将短文本 T 进行分割:

$$T=[S_1,S_2,S_3,\cdots,S_{m-1},S_m] \quad (6)$$

2) 进行分词和处理词性标注, 将停用词等过滤掉, 只留下词性的部分词:

$$S_i=\left[t_{i,1},t_{i,2},t_{i,3},\cdots,t_{i,n-1},t_{i,n}\right] \quad (7)$$

3) 构建词图, 候选关键词构成节点集合 V. 实验采用了共现关系连接任意两个节点, 并约定两节点间有边当且仅当其对应的词在长度是 K 的窗口里共现.

4) 由 TextRank 公式, 进行迭代传播各个节点的权重值.

5) 将权重倒序排列, Top N 的词作为待选 keyword.

6) 在上个步骤中获取 T 个词后, 标记原始的文本, 若发现产生相邻词组, 则组合为多词关键词.

我们对关键词进行向量化处理, 并采用了相关性分析, 以此能够对以上产生的关键词特征向量重构. 首先在相关性分析方法的选择上, 采用斯皮尔曼相关系数 (Spearman's rank correlation coefficient), 又叫秩相关系数, 对变量间的秩次作分析, 对变量的分布不做规定, 是非参数统计方法. 与之类似的还有皮尔逊相关系数 (Pearson correlation coefficient), 但相比之下, 斯皮尔曼相关系数拥有更广泛的适用范围, 并且更契合本文的目的, 因此我们用 Spearman 秩系数来分析统计特征间的相关性[13].

Spearman's 系数ρ的取值范围为 [−1,1]. 当ρ=−1 时, 两种统计特征负相关, 当ρ=1 时, 两种统计特征正相关, 当ρ=0 时, 两种统计特征不存在相关性.

Spearman 相关系数的计算公式如下:

$$\rho_{XY}=\frac{\sum\limits_{i=1}^{N}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum\limits_{i=1}^{N}(x_i-\overline{x})^2\sum\limits_{i=1}^{N}(y_i-\overline{y})^2}} \quad (8)$$

其中, N表示样本总量, $\overline{x}$和$\overline{y}$分别表示X,Y的均值.

考虑到我们需对词向量中各个特征之间的相关性关系进行分析, 所以我们采用了 Spearman's 的秩相关系数, 并且, 更进一步设置了各特征之间的系数阈值, 规定短文本数据新特征由系数小于阈值的统计特征决定.

本文实验中根据短文本的数据特征值进行关键词提取, 并创建相关的词向量, 采用 Spearman 相关系数进一步分析各个统计特征间的关联性, 最后, 我们基于相关性分析的结果组成新的词向量特征. 这种基于关键词提取算法创建短文本的特征空间, 可以生成低密度的文本向量, 对短文本的分类效果起到显著的影响.

2.2 多神经网络混合的短文本分类模型框架

在本次实验中, 我们提出了一种基于多神经网络混合的短文本分类模型[14], 融合了 TextCNN 和 FastText 两种神经网络.

FastText 模型分为 3 方面: 模型架构、层次 Softmax 和 n-gram 特征. 模型架构是一个线性模型. 它首先需要输入词序列到输入层中, 再把字向量级别的 n 元语法向量组成模型的新特征向量; 从 inputlayer 到 hiddenlayer 的过程中, 模型把词信息用文本来完成表示, 反馈到线性模型分类中, 在 outputlayer 中采用 Softmax 完成类的概率分布计算[15].

原始的 FastText 模型有输入层、隐藏层以及输出层, 我们的目的是想让其应用在短文本内容的分类任务上, 我们对该模型的输入层进行了改进, 添加了计算模型, 即先使用 TextRank 算法对短文本内容进行关键词提取处理, 然后计算关键词向量化间的相关系数, 这一步采用 Spearman 进行实现. 最后, 根据 Spearman's 秩相关系数分析词向量内部各个特征之间的相关性, 并选取若干个特征向量作为输入. 这部分输入后续会经过 n-gram 处理得到新的词序列特征, 传到隐藏层中. 改进后的 FastText 模型架构图如图 1.

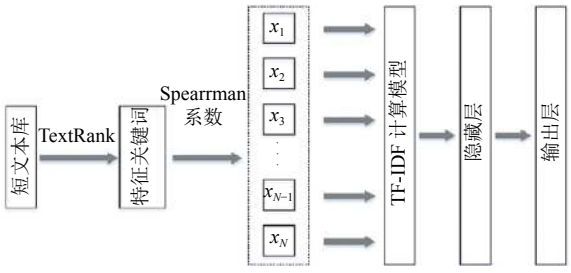


图 1 改进后的 FastText 模型架构图

图 1 中, 模型的输入$(x_1, x_2, x_3, \cdots)$是词序列或短文本, 单层的神经网络作为 hiddenlayer, 最终模型的输出是输入数据所属不同类别的概率向量.

在实际实验中, 由于 n-gram 易产生冗余词条, 为此, 我们将词条进行条件过滤. 在 inputlayer 中加入 TF-IDF 模型, 这样能保存有价值的词向量, 构建新的特征向量. 通过映射到 hiddenlayer 中, 在求解完最大似然函数后做 Softmax 处理, 构建哈夫曼树[16].

在这个过程中, 叶子节点表征着类别, 在构建非叶子节点时要选择左右分支, 若用逻辑回归表示概率, 如下式所示:

$$\sigma_P = \frac{1}{1+e^{-X_i\theta}} \tag{9}$$

$$\sigma_N = 1-\sigma_P \tag{10}$$

其中, σ_P表示正类别的概率, σ_N表示负类别概率, θ为模型中哈夫曼树节点的一个参数, 构建的哈夫曼树中每个类别均有一条路径, 即训练样本的特征向量X_i和类别标签Y_i在哈夫曼树均会有对应的路径进行表示. 换句话说, 我们对X_i的类别进行预测, 本质上就是计算X_i属于标签Y_i的概率, 正如下式所示:

$$P(Y_i|X_i) = \prod_{j=1}^{l} P(w_j|X_j,\theta_j) \tag{11}$$

其中, l是样本词向量数量, w_j是词向量.

在 TextCNN 模型的中, 我们对其进行了改进[17]. 由于短文本中句子的长度有所不同, 因此, 我们使用 padding 操作获得固定长度 n. 这样我们可以保证对于句子里的每个标记信息, 都可以拥有固定的维数 d. 因此, 我们的输入是一个二维矩阵: (n, d).

首先, 我们将对输入进行卷积运算. 卷积是过滤器和输入之间的特殊运算, 可将其视为逐元素相乘的运算过程. 在模型中, 我们有 k 个滤波器, 而且大小为二维矩阵 (f,d). 现在输出将是 k 个列表. 每个列表的长度为 $n–f+1$. 列表中的元素均为标量, 第二维采用了词潜入的维. 我们使用不同大小的过滤器, 以从文本输入中获取丰富的功能.

其次, 实验对模型的卷积运算输出 max-pooling 操作. 对于 k 个列表, 我们将得到 k 个标量.

第三, 我们将连接标量以形成最终特征. 它是一个固定大小的向量. 它与我们使用的过滤器的大小无关.

最后, 我们将使用线性层将这些特征投影到每个

定义的标签上.

多模型神经网络框架的提出主要是根据分治原则用来解决复杂的问题. 前人主要利用它解决分类问题, 包括脸部的图像分类和非平衡问题. 它将原始集分为多个集合, 最后进行策略合并来输出分类的结果. 本文提出引入的多神经网络的目的与之类似, 主要采用不同类型的神经网络特点弥补单一类型神经网络的缺点.

改进后的 TextCNN 模型结构图如图 2.

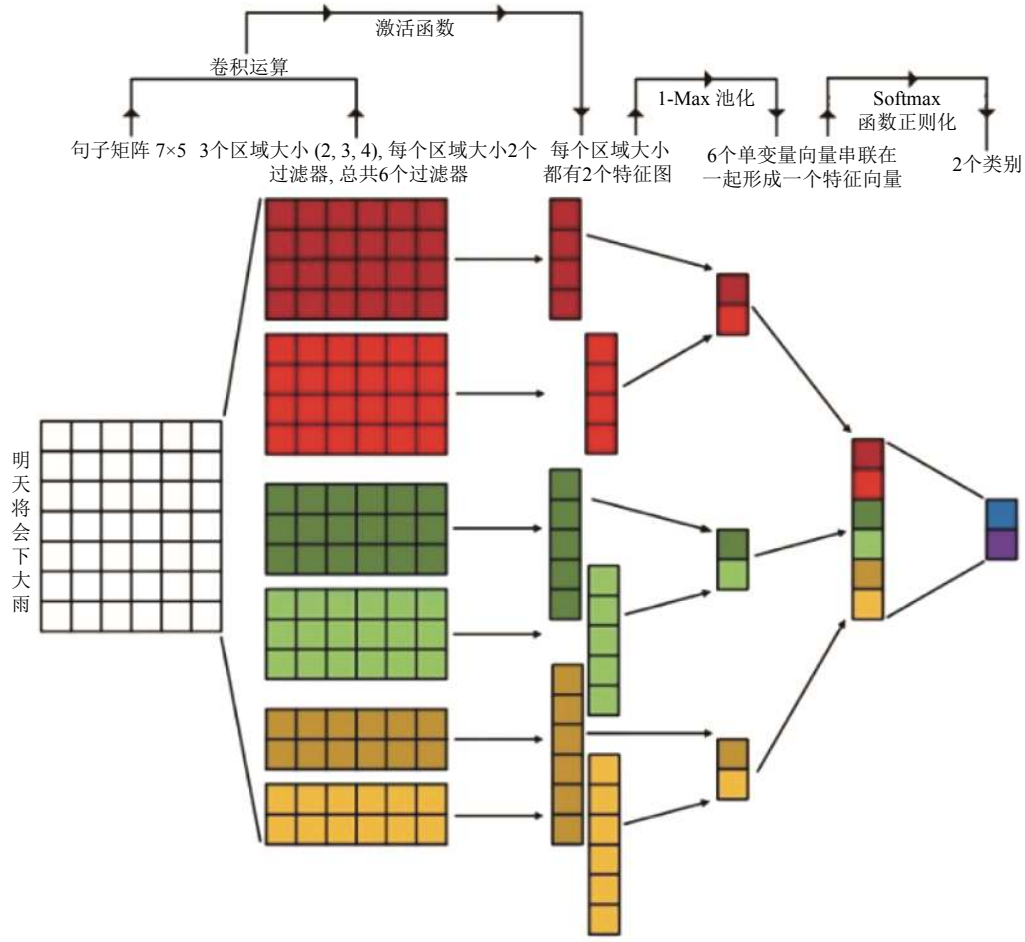


图 2 改进后的 TextCNN 模型结构图

给定一个输入$input1 = \{C_i|w_1, w_2, w_3, \cdots, w_n\}$, 即每个输入由一个类别$C_i$和若干词$w_n$组成的句子构成, 其中$C_i$是该输入数据的目标类别, 每个输入仅有一个, 对应这个输入的一个方面. 比如输入“国际| 美国总统特朗普 9 月 11 日宣布国会议员人选将不再限制绿卡持有量”, 其中“国际”代表着该输入的目标类别, 象征着该输入的一个特定属性. 而本文主要解决的问题就是在未知类别属性下, 通过对语句的识别与理解, 将该短语正确的划分到给定的类别中. 但是由于短文本分类中存在很多棘手的问题, 比如输入信息不足, 类别向量太少, 噪音太大等, 导致短文本分类一直处于较低的准确水平. 我们尝试将原始输入数据进行关键词特征重构, 通过对短文本的词义分析进行挖掘重要特征, 并采用更适合多分类的模型进行向量输出. 在已知原始输入$input1 = \{C_i|w_1, w_2, w_3, \cdots, w_n\}$的情况下, 通过特征重构得到$input2 = \left\{C_i|w'_1, w'_2, w'_3, \cdots, w'_m\right\}$, 其中$w'_m$是通过对词义分析而得到的特征词, 用以表征该输入的多个类别信息, 用作对C_i的有效补充. 在上述输入例句中, 可将“特朗普”、“美国”、“总统”、“国会”、“绿卡”等关键词特征进行重构提取.

在区域r_i中, 其长度为h, 我们将每个词映射成m维的值向量, 从而可将区域表示成$r_i = \{x_1, x_2, x_3, \cdots, x_h\}$. 把文本里每个区域作为 TextCNN 的输入矩阵, 并对该区域进行卷积运算[18], 卷积核的长度为l:

$$c_i = f(w \cdot r_i + b) \tag{12}$$

式中,$w \in R^{m \times l}$是模型的卷积核权重值, 而$b \in R$则是模型的偏置项数据, 在这个模型中通过卷积层, 每个区域均可获取区域r_i对应的特征图$c \in R^{h-l+1}$:

$$c = \{c_1, c_2, c_3, \cdots, c_{h-l+1}\} \tag{13}$$

在区域 r_i 中, 采用 max-over-time pooling 对局部特征进行采样, 提取特征图的有效信息[18], 即$\hat{c} = \max\{c\}$, 下采样的信息图可表示为:

$$\hat{c} = \{\widehat{c_1}, \widehat{c_2}, \widehat{c_3}, \cdots, \widehat{c_k}\} \tag{14}$$

我们假设了两种文本分类算法可以应用于不同的处理场景. 如上文所述, TextCNN 和 FastText 都可以用来进行文本分类, 但解决的问题存在区别, TextCNN 在短文本分类中较重视词向量, 作文本分析或者主题分析较合适, 但由于 TextCNN 需对词向量进行多次卷积运算, 导致运行时间较长. 所以从模型结构角度来说, FastText 采用了更加简单而又高效的文本分类以及表征学习方法, 所以在文本分类中的多标签值问题中更为合适, 且运行耗时很短, 但是在短文本中由于缺少更多信息的补充, 导致算法的准确度大幅降低.

因此, 我们利用 TextCNN 来处理单标签分类与短文本工作, 用 FastText 来处理多关键词的主题分类工作. 优化后的神经网络框架如图 3 所示, 不同于传统的单一任务处理工作, 本模型将原始集进行分而治之, 分别采用 TextCNN 和 FastText 进行文本的处理工作, 并在合并层进行类别向量的融合:

$$f(C_1, C_2) = \beta C_1 + (1-\beta) C_2 \tag{15}$$

在多神经网络混合的短文本分类模型框架中, 在输入层部分, 数据来源有两个类型: “标签值+多关键词”作为多分类, “标签值+短文本”作为文本分析. 简言之, 这个模型的输入主要有多关键词和短文本构成. FastText 模型负责处理多关键词特征, 而 TextCNN 负责处理短文本内容.

我们采用 Softmax 函数处理文本的类别输出:

$$y = Softmax(W \cdot C + b) \tag{16}$$

其中, W 为权重矩阵, b 为偏置项, 我们用最小化的交叉熵优化混合模型, 交叉熵计算如下:

$$loss = \sum_{i \in D} \sum_{j \in C} \widehat{y_i^j} \log y_i^j + \lambda \|\theta\|^2 \tag{17}$$

式中, D 为训练集文本数据, C 为数据的类别树, y 是待分类短文本句子的预测性类别, $\hat{y}$为短文本的实际类别, $\lambda\|\theta\|^2$为式子的交叉熵正则项.

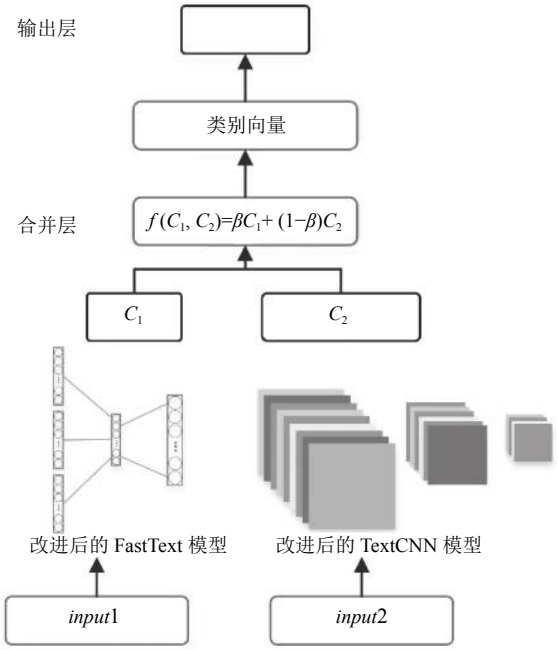


图 3 多神经网络混合的短文本分类模型框架

两种神经网络模型的输出在合并层进行融合, 形成短文本类别的向量输出, 合并层的具体融合方式如式所示执行, C_1和C_2分别表示两种分类算法的中间输出向量, 代表混合短文本分类模型中关键词和短文本各自的影响力, 由于C_1和C_2都是用来表征文本类别向量, 所以它们的维数是一致的. 它们按照影响比例进行相加, 其中β参数负责控制 FastText 和 TextCNN 模型对最终类别向量输出的影响效力, β的数值越大, 即表示混合模型中 FastText 模型的类别输出向量对短文本类别向量的影响力就越高; 反之, 混合模型中 TextCNN 模型的类别输出向量对短文本类别向量的影响力就越高.

2.3 Dropout 防止混合模型的过拟合

在文本分类模型框架中加入 Dropout 的目的是将其应用于词向量的分类中, 能够使得模型降低对于噪音的敏感度, 防止因噪音导致过拟合[19]. 在某种程度上, Dropout 可作为文本数据增强的方法, 将词向量修剪得更为准确.

2.4 模型对比

传统的短文本处理方法有 RNN、LSTM、GRU、RCNN 等模型, 一般通过 Doc2Vec 或者 lDA 模型将文

本转换成一个固定纬度的特征向量, 然后再基于抽取的特征训练一个分类器.

FastText 模型相比于其他模型, 在训练速度和算法准确率上有了更好的表现, 主要做了如下改进:

层次 Softmax: 针对有大量类别的数据集或者是类别不均衡分布的数据集, FastText 模型的层次 Softmax 技巧通过对标签信息进行编码, 可以缩小模型预测的目标数量. 该技巧建立在 Huffman 编码的基础上, 用来表征类别的树形结构, 当类别数为 K, word embedding 大小为 d 时, 计算复杂度可以从$O(Kd)$降到$O(d\log(K))$, 使得模型的计算效率更高.

n-gram: 相比于传统方法采用的词袋模型不能考虑词之间的顺序, FastText 模型加入了 n-gram 特征, 通过对词序的记录进行区分词的信息, 使得模型的可读性更强, 提高了短文本的语义理解能力. 模型把所有的 n-gram 哈希到 buckets 数量的桶中, 并共享一个向量, 这样既保证了在查找时$O(1)$的效率, 也把内存消耗控制在$O(buckets*dim)$范围内.

Subwords: subwords 可以理解为是字级别的 n-gram, 它丰富了词表示的层次, 对相同字信息能够有更好的语义相似性识别能力. 这种技巧对低频词生成的词向量效果会更好, 并且对于训练词库之外的单词, 也可以构建词向量.

FastText 模型的架构中只有一层的隐藏层和输出层, 保证了模型在 CPU 上能实现分钟级的训练, 更容易在工程应用上进行部署. 但是也正因此, 模型只能对单个标签进行分类, 在做多标签处理时, 模型也仅会随机选一个 label 进行模型的更新, 导致其他信息被忽略, 不适合做多标签的分类应用.

TextCNN 模型是 CNN 模型的一个变种, 可以充分发挥 CNN 的并行计算能力, 训练速度更快, 在保留了原始 CNN 的特点之外, 还加入了对文本特征的抽取能力. TextCNN 采用一维卷积来获取句子的 n-gram 特征表示, 具有很强的文本浅层特征抽取能力. 模型可以识别出任务里具有语言性的 n 元语法, 在预测过程中遇见未登录的特定 n 元语法时, 它的卷积结构还可以让有相似元素的 n 元语法分享预测的行为, 并且层次化的 CNN 的每一层都关注着句子里更长的 n 元语法, 使得模型可以对非连续的 n 元语法更加敏感. TextCNN 通过调整卷积核的高度, 可以对综合词汇的多种时序信息进行灵活处理, 提高了模型对文本的解读能力.

2.5 混合模型的特点

模型对数据的整体处理流程为: 文本序列中的词和词组构成特征向量, 特征向量通过线性/非线性变换映射到中间层, 中间层再映射到标签数据上, 输出文本对应的标签值. 本文中混合模型其实是兼顾了文档向量和特征抽取的能力, 保证了模型在遇到复杂多变的数据集时的多态性.

FastText 模型的输入层到隐藏层部分, 主要生成用来表征文本的向量; 模型的隐藏层到输出层是一个 Softmax 线性的多类别分类器. 它更适用于分类类别非常大且数据集足够大的情况, 当类别数较小或数据集较小时容易导致过拟合. 而 TextCNN 采用了 max pooling 操作, 可以保证文本的特征的位置与旋转不变性, 强特征的提取效果不受其位置所限制. 此外, max pooling 能够减少模型参数数量, 有利于减少过拟合问题. 但是这也容易导致 TextCNN 丢失特征的位置信息, 并且无法记录同一特征的强度信息.

此外, FastText 模型为了保证句子词向量的相似性 (模型要求词向量按照相同的方向更新), 没有添加正则项, 从而导致模型的泛化能力减弱. 而 TextCNN 使用预训练产生的词向量作 embedding layer, 并且在本文实验中采用混合向量 (基于字向量、静态词向量、动态词向量) 进行文本中词信息的表示, 能够捕捉到更多的语义信息, 以此来减少数据集的影响, 提高模型的准确率. 同时, TextCNN 在池化层后加了全脸阶层和 Softmax 层做分类任务, 并且加上了 L2 正则项和 Dropout 方法, 来防止模型的过拟合问题, 最后整体使用梯度法进行参数的更新模型的优化.

3 实验验证

3.1 实验数据

为了确认该短文本分类的混合模型的算法有效性, 同时也为了在长文本中验证其效果, 实验选择了相关的公开数据集:

1) TTNews: 短文本采用今日头条的公开新闻数据集进行实验验证, 数据文件中每行是一条数据, 以_!_ 符号进行分割, 文本格式内容依次为新闻的 ID 编号, 新闻分类编码 code, 新闻分类的名称信息, 新闻标题 title 和新闻关键词 keyword. 数据采集时间为 2018 年 05 月, 共计有 382 688 条, 有 15 种新闻类别, 每种新闻类别对应的数据量大小如表 1 所示.

表 1 TTNews 数据集中类别统计

新闻类别	数据量
news_tech	41 542
news_entertainment	39 396
news_sports	37 568
news_car	35 785
news_game	29 300
news_culture	28 030
news_finance	27 085
news_edu	27 058
news_world	26 909
news_military	24 984
news_travel	21 422
news_agriculture	19 322
news_house	17 672
news_story	6273
stock	340

为了保证是短文本数据集, 本次实验是选用两个关键数据: 新闻分类名称和新闻对应的标题, 比如“news_entertainment 胡歌为林心如澄清娱乐界谣言, 最后发现纯属乌龙”.

2) THUCNews: 长文本实验方面, 实验采用了清华大学的 THUCNews 数据集, 该数据集是由新浪新闻在 2005~2011 年间的所有数据经过滤产生的, 包含了 74 万篇的新闻文档, 均是 UTF-8 纯文本格式. 进一步在原始的分类基础之上, 划分了 14 个待选的类别: Education, Technology, Finance, Stocks, Home, Social, Games, Fashion, Current Affairs, Lottery, Real Estate, Sports, Constellation, Entertainment[20]. 如表 2.

表 2 THUCNews 数据集中类别统计

新闻类别	数据量
Technology	162 928
Stocks	154 397
Sports	131 604
Entertainment	92 631
Current Affairs	63 085
Social	50 848
Education	41 935
Finance	37 097
Home	32 585
Games	24 372
Real Estate	20 049
Fashion	13 367
Lottery	7587
Constellation	3577

3.2 数据处理

首先需要对上述文本数据集进行自动分词, 本文采用中文分词工具 NLPIR 汉语粉刺系统, 调用 GitHub 上开源的工具包进行文本分词, 每个词用空格分开. 并且将数据处理为每条新闻文本为一行, 每行的末尾使用“__label__”特殊字符串.

对分词后的数据进行清洗, 去除无意义的语气词、特殊符号、错词等, 把有效数据进行整理. 我们将 traindata 和 testdata 按照 8:2 的比例划分, 并用 shuffle 函数将原始数据打散.

3.3 评估指标

本次实验我们采用了常用的评估标准, 包括: 召回率 R (Recall)、准确率 P (Precision) 和 $F1$-分数 ($F1$-score).

我们假设文本分类针对数据c_1的类别划分结果如表 3 所示.

表 3 类别划分结果

分类器的分类判断	文本与类别的关系	
	属于	不属于
标记为“是”	a	b
标记为“否”	c	d

实验在类别c_1的召回率计算如下式:

$$R = recall_i = \frac{a}{a+c} * 100\% \tag{18}$$

实验在类别c_1的准确率计算如下式:

$$P = precision_i = \frac{a}{a+b} * 100\% \tag{19}$$

$F1$-score: 也称 $F1$ 分数, 是分类问题中的重要衡量指标. 在计算方法上, 它同时考虑了模型的精确率和召回率两个指标, 所以可将其视为准确率和召回率的调和平均数, 最大是 1, 最小是 0. 计算方式为:

$$F1 = \frac{2RP}{R+P} \tag{20}$$

3.4 实验对比算法模型

本次实验的对比算法包括: CNN、RNN、TextCNN、TextRNN、FastText、Seq2seqAttn、RCNN 等模型.

3.5 实验参数

在混合模型的实际训练中, 模型整体保持了较高的运行效率, 在训练时间上的消耗见实验结果表 3.

FastText 模型中最重的两个参数分别是: 词向量大小维度 dim 和 subwords 范围的大小 subsize. 其中, dim 越大, 模型就能获得更多的信息但同时对训练数据的要求也就更高, 而且训练速度也会降低, 因此本实验中词向量的维度 dim 选 100. Subwords 是单词序列中包含最小 min 到最大 max 之间的所有字符串, 根据本

文实验数据我们设置为 1～3 之间灵活调整, 以便能够从数据中识别出中文特有的属性 (如人名、专有名词等信息). 在训练参数上, 本实验的 epochs 设为 10, 学习率 Lr 设为 0.1. FastText 是基于多线程的, 根据本文实验环境, 对 CPU 核数调整为 16.

TextCNN 模型的输入采用基于 Word2Vec 和 glove 的混合词向量, 具体包括字向量、静态词向量和动态词向量. 因为在处理 NLP 任务时需要对文本进行截取, 在短文本实验中 padding 值为 50, 长文本实验中 padding 值为 300, 卷积核的尺寸 filter_size 范围区间可取 (1, 10), 在短文本实验中取{2,3,4}, 在长文本实验中取{3,4,5}, 每个 filter 的 feature maps 取 100, 因此, 短文本实验中的卷积层大小为 50×100, 长文本实验中卷积层大小为 300×100. 卷积核的数量 filter_nums 范围区间可取 (100, 600), dropout 范围区间可取 (0, 0.5), 在本文实验中统一取 filter_nums 为 150, dropout 取 0.5. 实验中采用 ReLU 激活函数进行处理, 池化层选择了 1–max pooling, 为了进一步检验模型的性能水平, 本文实验采取了交叉验证处理. 在模型的训练参数上, 本实验主要设置 batch_size 为 64, num_epochs 为 10.

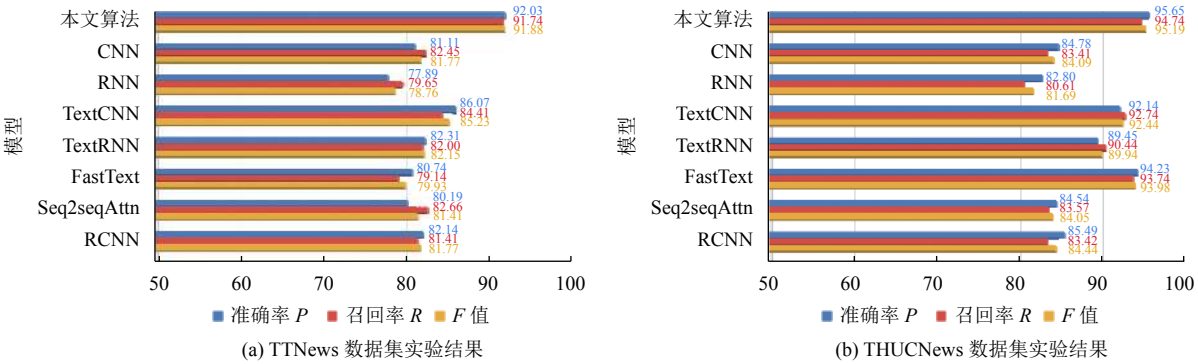
3.6 实验结果分析

本次实验采用短文本数据集 TTNews 对算法模型进行测试验证, 并用长文本数据集 THUCNews 对各算法模型进行比较, 进一步验证所提的多神经网络混合的短文本分类模型在实验数据上的有效性, 实验结果如表 4 和图 4 所示.

从表 4 和图 4 的实验结果可以看出, 本文提出的多神经网络混合的短文本分类模型在 TTNews 数据集和 THUCNews 数据集上都取得了较好的文本分类效果. 尤其是在短文本分类 (TTNews 数据集) 上, 该模型的分类准确率达到了 92.03%, 相比于原始的 FastText 模型、TextCNN 模型分别提升了 13.98%、6.92%; 在长文本分类 (THUCNews 数据集) 上, 该模型的分类准确率达到了 95.65%, 相比于原始的 FastText 模型、TextCNN 模型分别提升了 1.48%、3.81%. 在模型的召回率和 $F1$ 分数等评价指标上, 也能很明显看出混合短文本分类模型的效果相比于原始的 FastText 模型、TextCNN 模型均有明显的提升效果, 从而验证了本文提出模型的有效性.

表 4 不同模型在数据集上的实验结果 (单位: %)

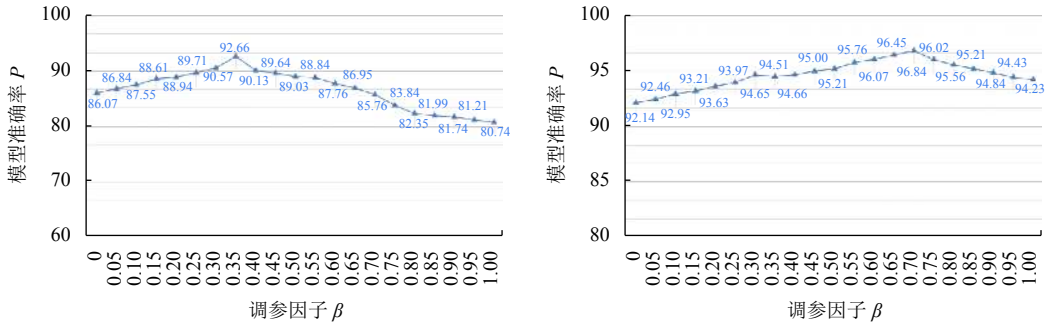
模型	指标	数据集	
		TTNews	THUCNews
本文模型	P	92.03	95.65
	R	91.74	94.74
	$F1$	91.88	95.19
CNN	P	81.11	84.78
	R	82.45	83.41
	$F1$	81.77	84.09
RNN	P	77.89	82.80
	R	79.65	80.61
	$F1$	78.76	81.69
TextCNN	P	86.07	92.14
	R	84.41	92.74
	$F1$	85.23	92.44
TextRNN	P	82.31	89.45
	R	82.00	90.44
	$F1$	82.15	89.94
FastText	P	80.74	94.23
	R	79.14	93.74
	$F1$	79.93	93.98
Seq2seqAttn	P	80.19	84.54
	R	82.66	83.57
	$F1$	81.41	84.05
RCNN	P	82.14	84.49
	R	81.41	83.42
	$F1$	81.77	84.44



(a) TTNews 数据集实验结果　　(b) THUCNews 数据集实验结果

图 4 不同模型在数据集上的准确率、召回率和 $F1$ 均值

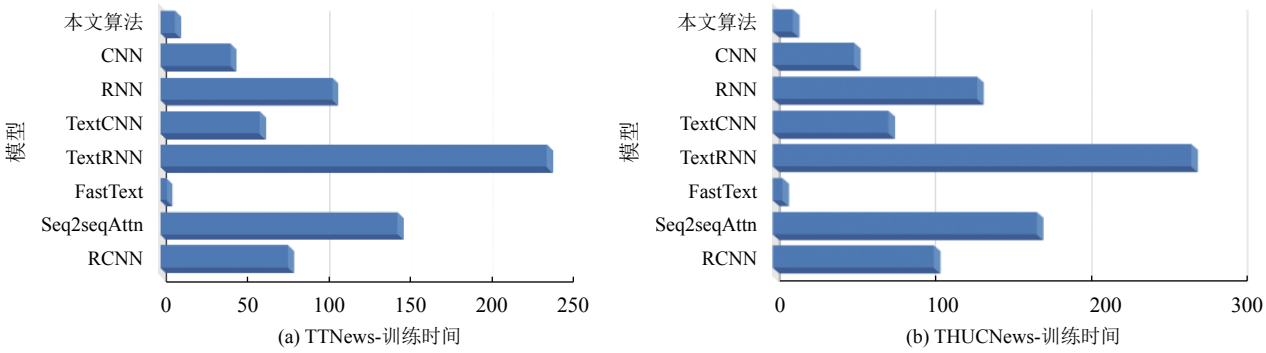
β超参数的主要作用在于调节关键词和短文本内容对混合分类模型最终分类向量输出结果的影响. 因此针对不同的特点的文本分类数据集, β参数的取值会影响到多神经网络最优的输出结果. 在图 5 中我们可以发现, 在短文本数据集 TTNews 下, β在 0.35 取值时整体的文本分类精确度达到了最高, 在长文本数据集 THUCNews 下, β在 0.70 取值时整体的文本分类精确度达到了最高. 产生这一现象是因为, FastText 更适合长文本分类, 且对标签值的依赖性更高, 需要输入多个标签值对文本进行划分. 而 TextCNN 采用了多层神经网络和卷积层, 对文本的要求度不高. 实验结果在某种程度上也证明了神经网络分类模型中对文本大小的关联性.



(a) TTNews-准确率 P 与调参因子 β (b) THUCNews-准确率 P 与调参因子 β

图 5 本文模型的准确率与调参因子β的变化

本文在相同数据集和相同的训练环境下 (保证了各变量的控制条件), 分析了各模型的训练时间, 并对比分析了它们分别在短文本数据集 (TTNews) 和长本文数据集 (THUCNews) 上的实际效果, 结果如图 6.



(a) TTNews-训练时间 (b) THUCNews-训练时间

图 6 不同模型在数据集上的训练时间对比

从图 6 可以看出, 在相同的训练环境下, 本文所提出的多神经网络混合短文本分类模型兼顾了 FastText 模型训练收敛较快的优点. 相比于其他的神经网络模型而言, 混合短文本分类模型能够大幅降低训练时间, 这是因为在处理多关键词的分类时, 将数据灌入了 FastText 模型, 而短文本内容则在 TextCNN 模型中进行分类, 这样能够使训练任务得到均衡分配.

4 结论与展望

本文提出了将短文本内容进行关键词提取并重构短文本特征, 将重构后的特征值作为 FastText 模型的输入, 原始的短文本内容作为 TextCNN 模型的输入, 并引入参数β将类别向量进行调节, 作为融合的输出向量进行类别分析. 在多神经网络混合的短文本分类模型中, 把两种文本分类算法进行深度融合, 针对短文本分类的特点进行实质化改进, 从而兼顾了 FastText 模型和 TextCNN 模型的特点. 最终实验结果表明本文提出的多神经网络混合的短文本分类模型在短文本分类情景下的精确率、召回率和 $F1$ 分数等指标都表现出优越的算法性能, 相比其他的文本分类算法更为突出、

高效.

本文的主要研究关键点在于短文本内容的分类算法和对分类算法的融合应用, 同时也针对短文本内容的特点进行了特征重构, 将分类任务均衡化来解决相应的问题. 但是对于从短文本中提取与主题关联度契合度更高的关键词这一步仍有很大的改进空间. 由于短文本内容往往包含的信息量较少, 如新闻标题、用户评价文本等, 所以如何从信息度较低的短文本内容中提炼出符合要求的关键词是下一步需要研究的工作重点.

参考文献

1 Salton G, Wong A, Yang CS. A vector space model for automatic indexing. Communications of the ACM, 1975, 18(11): 613–620. [doi: 10.1145/361219.361220]

2 Joulin A, Grave E, Bojanowski P, *et al*. Bag of tricks for efficient text classification. Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. Valencia, Spain. 2017. 427–431.

3 常耀成, 张宇翔, 王红, 等. 特征驱动的关键词提取算法综述. 软件学报, 2018, 29(7): 2046–2070. [doi: 10.13328/j.cnki.jos.005538]

4 康卫, 邱红哲, 焦冬冬, 等. 基于搜索的短文本分类算法研究. 电子技术应用, 2018, 44(11): 121–123, 128.

5 McCallum A, Nigam K. A comparison of event models for naive Bayes text classification. Proceedings of AAAI-98 Workshop on Learning for Text Categorization. 1998. 41–48.

6 孙启干. 面向 Web 文本检索的归一化向量分类算法 [硕士学位论文]. 重庆: 重庆大学, 2012.

7 Lin YS, Jiang JY, Lee SJ. A similarity measure for text classification and clustering. IEEE Transactions on Knowledge and Data Engineering, 2014, 26(7): 1575–1590. [doi: 10.1109/TKDE.2013.19]

8 崔伟东, 周志华, 李星. 支持向量机研究. 计算机工程与应用, 2001, 27(1): 58–61. [doi: 10.3321/j.issn:1002-8331.2001.01.019]

9 Tan SB. An effective refinement strategy for KNN text classifier. Expert Systems with Applications, 2006, 30(2): 290–298. [doi: 10.1016/j.eswa.2005.07.019]

10 Rose S, Engel D, Cramer N, *et al*. Automatic keyword extraction from individual documents. In: Berry MW, Kogan J., eds. Text Mining: Applications and Theory. New Jersey: John Wiley & Sons, Ltd, 2010. 1–20.

11 Mihalcea R, Tarau P. Textrank: Bringing order into text. Proceedings of 2004 Conference on Empirical Methods in Natural Language Processing. Barcelona, Spain. 2004. 404–411.

12 何金金, 郭振波, 王开西. 基于 TextRank 的网评产品特征提取方法. 自然科学版, 2018, 31(1): 109–114.

13 Hauke J, Kossowski T. Comparison of values of Pearson's and Spearman's correlation coefficients on the same sets of data. Quaestiones Geographicae, 2011, 30(2): 87–93. [doi: 10.2478/v10117-011-0021-1]

14 张备. 基于多神经网络的混合动态推荐研究 [硕士学位论文]. 重庆: 重庆大学, 2017.

15 古倩. 基于特征向量构建的文本分类方法研究 [硕士学位论文]. 西安: 西安理工大学, 2019.

16 冯勇, 屈渤浩, 徐红艳, 等. 融合 TF-IDF 和 LDA 的中文 FastText 短文本分类方法. 应用科学学报, 2019, 37(3): 378–388. [doi: 10.3969/j.issn.0255-8297.2019.03.008]

17 Kim Y. Convolutional neural networks for sentence classification. Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing. Doha, Qatar. 2014. 1746–1751.

18 刘全, 梁斌, 徐进, 等. 一种用于基于方面情感分析的深度分层网络模型. 计算机学报, 2018, 41(12): 2637–2652. [doi: 10.11897/SP.J.1016.2018.02637]

19 Srivastava N, Hinton G, Krizhevsky A, *et al*. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 2014, 15: 1929–1958.

20 孙茂松, 李景阳, 郭志芃, 等. THUCTC: 一个高效的中文文本分类工具包. 2016.