计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(8):213−218 [doi: 10.15888/j.cnki.csa.008007]
©中国科学院软件研究所版权所有.
E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于 FastText 和关键句提取的中文长文本分类①

汪家成, 薛　涛

(西安工程大学 计算机科学学院, 西安 710048)
通讯作者: 薛　涛, E-mail: 267478680@qq.com

摘　要: FastText 是一种准确高效的文本分类模型, 但直接应用在中文长文本分类领域存在准确度不高的问题. 针对该问题, 提出一种融合 TextRank 关键子句提取和词频-逆文本频率 (Term Frequency-Inverse Document Frequency, TF-IDF) 的 FastText 中文长文本分类方法. 该方法在 FastText 模型输入阶段使用 TextRank 算法提取文本的关键子句输入训练模型, 同时采用 TF-IDF 提取文本的关键词作为特征补充, 从而在减少训练语料的同时尽可能保留文本分类的关键特征. 实验结果表明, 此文本分类方法在数据集上准确率达到 86.1%, 比经典的 FastText 模型提高了约 4%.

关键词: 文本分类; FastText; TextRank; 词频-逆文本频率

引用格式: 汪家成,薛涛.基于 FastText 和关键句提取的中文长文本分类.计算机系统应用,2021,30(8):213–218. http://www.c-s-a.org.cn/1003-3254/8007.html

Chinese Long Text Classification Based on FastText and Key Sentence Extraction

WANG Jia-Cheng, XUE Tao

(School of Computer Science, Xi'an Polytechnic University, Xi'an 710048, China)

Abstract: FastText is a precise and efficient text classification model, but the precision is low when it is directly applied to Chinese long text classification. Regarding this problem, this study proposes a FastText method for Chinese long text classification, which combines TextRank key clause extraction with Term Frequency-Inverse Document Frequency (TF-IDF). Firstly, TextRank is used to extract the key clauses of the text as input features. Secondly, key words of the text are extracted by TF-IDF as a feature supplement. Finally, the extracted text features are input into the FastText model, which can preserve the key features of the target text while reducing the training corpus. The experimental results show that the accuracy of the proposed method on the datasets is 86.1%, which is about 4% higher than the classic FastText model.

Key words: text classification; FastText; TextRank; Term Frequency-Inverse Document Frequency (TF-IDF)

1　引言

随着互联网技术的高速发展, 网络中每天都会产生海量的数据, 从杂乱的信息中获取有效信息已成为业界的研究热点[1]. 文本分类任务是自然语言处理 (NLP) 领域中最基础的任务之一, 其不仅能有效的筛选信息, 而且在信息检索、情感分类和自动文摘等方面有着重要的应用. 随着人工智能行业的兴起, 文本分类也有了更为广泛的应用, 如人机通信, 问答系统等[2].

文本分类最初使用的是基于规则的方法[3], 由相关领域的专家根据知识和经验制定相应的规则, 然后根据这些规则进行文本分类. 基于规则的文本分类方法虽然在某些领域上有很好的效果, 但是制定分类规则会耗费大量的人力成本, 且如果出现了新的分类标签需要制定新的规则, 因此基于规则的文本分类方法适

① 基金项目: 陕西省 2020 年技术创新引导专项 (基金)(2020CGXNG-012)
Foundation item: Year 2020, Technology Innovation Guidance Special Project (Fund) of Shaanxi Province (2020CGXNG-012)
收稿时间: 2020-11-12; 修改时间: 2020-12-14; 采用时间: 2020-12-18; csa 在线出版时间: 2021-07-31

© 中国科学院软件研究所 http://www.c-s-a.org.cn

用性较差[4].

近年来, 机器学习算法在文本分类中的应用成为自然语言处理研究热点, 机器学习算法中文本分类任务采用的是有监督学习[5], 主要包含模型训练和结果预测两个过程. 在数据进入模型训练之前, 需要对文本进行表示, 将其转化成计算机能够处理的形式. 文本的表示方法大多基于词袋模型和向量空间模型[6], 词袋模型将文本看成词的集合, 文本中的词越多, 词袋表示的文本向量维度就越大, 且词袋模型不考虑词的语义和语序, 会损失一些语义上的特征信息; 为了克服词袋模型无法表示文本语义的缺陷, Mikolov 等[7] 提出了 Word2Vec, 它将每个词转化成词向量, 文本内容的处理便转化为向量空间中的向量运算. 目前已有多种机器学习算法应用在文本分类, 文献 [8] 采用了加权 Word2Vec 和 KNN 的文本分类方法, 在文本分类时获得较好的分类效果; 文献 [9] 采用 LDA 模型主题分布相似度文本分类方法, 补充了文本中的主题特征; 文献 [10] 采用了基于网络新词改进的 TF-IDF 算法, 再结合 SVM 模型以提高分类准确率.

随着计算机性能的增强, 深度学习也算法也被广泛的应用在文本分类中. 循环神经网络擅长捕获长的序列信息[11], 因此在长文本分类任务上有良好的表现; Yoon Kim 等[12] 提出了 TextCNN, 将卷积神经网络 CNN 应用到文本分类任务. Facebook 在 2016 年开源了快速文本分类算法 FastText, 该算法使用 n-grams 来缩小与深度模型之间的准确度差距, 能够取得与深度学习分类器相近的准确率, 并且在训练效率上要比深度学习分类器快[13].

虽然 FastText 文本分类方法取得了较为显著的效果, 但应用于中文长文本分类时仍存在不足, 长文本相对于短文本可以提取更多的特征, 但也有更多的冗余词语, 这些词语多是对分类结果没有正向影响的无关词语, 容易影响分类准确率.

针对上述问题, 本文提出一种结合 TF-IDF 和 TextRank 关键子句提取的 FastText 分类方法 (简称 KS-FastText). 该方法使用 TextRank 提取长文本的关键子句, 将文本的关键子句标上相应的标签作为独立句子输入 FastText 模型中训练, 以减少文本中无关词的影响程度; 之后采用 TF-IDF 算法提取文本的关键词, 将关键词词组作为模型的补充特征输入模型训练. 在结果预测时, 对目标文本也进行关键子句提取, 并对各个子句的预测标签加权综合判断目标文本的分类.

2 相关工作

2.1 FastText 模型

FastText 是一个快速准确的文本分类算法, 该算法主要用于解决有监督的文本分类问题. FastText 的结构如图 1 所示, 其结构可以简化为 3 层, 分别为数据输入层、隐含层和输出层[13]. FastText 的模型结构与 CBOW 架构很类似, 不同的是 FastText 通过上下文的词来预测标签, 而 CBOW 是利用上下文的词来预测中间词.

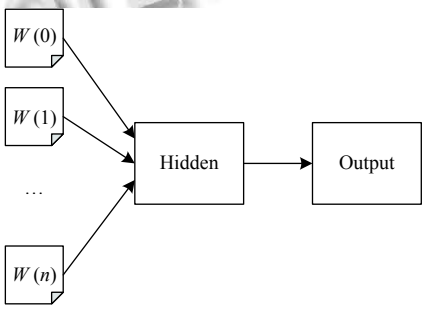

图 1 FastText 模型结构

当训练集中有多种分类标签时, 传统的线性分类器计算压力非常大, 所以 FastText 使用了分层 Softmax 技巧, 这是一个基于哈夫曼树的多分类器, 树形结构中的叶子节点代表了训练集中的标签, 能在多标签分类时有效的减少算法预测目标数量, 以此提高模型的效率.

2.2 TF-IDF

TF-IDF 是一种用以评估字词对于一个文件集或一个语料库中的其中一份文件的重要程度的统计方法. 在文本分类任务中, 词的重要性与它在文件中出现的频率成正比, 与它在数据集中出现的频率成反比. 因此可以使用 TF-IDF 评分作为筛选作为关键词的依据.

TF (Term Frequency) 代表词频, 指的是某一个特定的词语在该文件中出现的次数. 这个数字通常会被归一化, 以防止它偏向长的文件, 词频的计算公式如下:

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{1}$$

式中, $n_{i,j}$ 表示词语 i 在文档 j 中出现的次数, 分母表示文档 j 中总的词语数, TF 值为词语在文档中的出现次数与文档总词数的比值, TF 体现的是词语在文档内的重要程度.

© 中国科学院软件研究所 http://www.c-s-a.org.cn

IDF (Inverse Document Frequency) 是逆向文档频率, 用于度量一个词语的普遍重要性. 某特定词语的 IDF, 可以由所有文档的数目除以包含该词语之文件的数目, 再将得到的结果取对数得到, 逆向文档频率的计算公式如下:

$$idf_i = \lg \frac{|D|}{\left|\left\{j : t_i \in d_j\right\}\right|} \tag{2}$$

式中, $|D|$表示文档的数目, t_i 表示包含字词 i 的文档数目.

TF-IDF 值由 *tf* 值与 *idf* 值相乘得到, 其公式为:

$$TF-IDF(i,j) = tf_{i,j} \cdot idf_{i,j} \tag{3}$$

2.3 TextRank

TextRank 算法源于 Google 公司提出来的 PageRank 算法, PageRank 算法通过将网页与其链接的网页之间构成图关系, 每一个网页作为一个节点, 而链接作为边, 通过迭代计算筛选权值大的节点, 也就是链接比较多的网页, 一般用于网站排名. TextRank 算法将文本中的词或者句子类比成 PageRank 算法中的网页, 构建词或者句子之间的图关系, 通过类似的迭代计算可以得到相应文本中句子的重要度排名, 因此可以很方便的得出句子中的关键子句[14,15].

TextRank 在构建图的时候将节点由网页改成了句子, 并为节点之间的边引入了权值, 其中权值表示两个句子的相似程度, 本质上构建的是一个带权无向图, 其计算公式如下:

$$WS(v_i) = (1-d) + d \cdot \sum_{j \in In(v_i)} \frac{w_{ij}}{\sum_{v_k \in Out(v_j)} w_{jk}} WS\left(v_j\right) \tag{4}$$

式中, $WS(v_i)$ 表示节点 v_i 的权重值, d 为阻尼值, 用于做平滑, 表示在图结构中从一个节点跳到另一个节点的概率值. $In(V_i)$ 表示节点 V_i 的前驱节点集合, $Out(v_j)$ 表示节点 V_j 的所有后继节点集合. w_{ij} 为节点 v_i 和节点 v_j 间的权值.

从给定文本中提取关键句时, 将文本中的每个句子分别看作一个节点, 如果两个句子有相似性, 则认为这两个句子对应的节点之间存在一条无向有权边, 衡量句子之间相似性的公式如下:

$$Sim\left(s_i, s_j\right) = \frac{\left|\left\{w_k | w_k \subset s_i \& w_k \subset s_j\right\}\right|}{log\left(|s_i|\right) + log\left(\left|s_j\right|\right)} \tag{5}$$

式中, s_i 和 s_j 表示句子, w_k 表示句子中的词, 分子部分的意思是同时出现在两个句子中的词的数量, 分母是对句子中词的个数求对数后求和, 这样可以遏制较长的句子在相似度计算上的优势. 根据以上相似度计算公式循环计算任意两个节点之间的相似度, 设置阈值去掉两个节点之间相似度较低的边连接, 构建出节点连接图, 然后迭代计算每个节点的 TextRank 值, 排序后选出 TextRank 值最高的几个节点对应的句子作为关键句.

3 中文长文本分类方法

3.1 KS-FastText 基本思想

在中文长文本分类中, 文本中词容量比较大且文本中存在大量冗余数据, 如果全部作为文本的特征输入, 不但耗时较长, 并且分类效果也比较差, 可以通过提取长文本关键特征的方法保留关键特征, 同时减少无关词语的占比. 长文本的特征可以从关键子句和关键词两个方面提取. 关键子句可以有效的保留文本的中心特征句和特征句子词之间的联系, 关键词词组则保留了关键子句忽略的特征词语, 可以作为特征的补充.

使用 TextRank 算法提取文本的关键子句, TextRank 属于无监督学习算法, 无需额外数据训练, 算法通过构建文本子句的图模型并迭代计算每个子句节点的边权重值对子句进行排序, 取评分靠前的 3 条关键子句, 将各个关键子句的分类标签标记为当前文本的分类, 作为输入数据使用. 关键子句容易丢失关键子句外的关键词信息, 因此采用关键词词组对特征进行补充.

使用 TF-IDF 特征提取关键词, TF-IDF 算法提取每个文档中相对于整体文档区分度高的词, 既考虑词频又考虑了逆文档频率, 如果一个词的词频高且只出现在小部分文档中, 就说明这个词有很强的区分能力, 该词可以作为文本的关键词. 在使用 TF-IDF 方法提取文本的关键词后, 将该篇文档的关键词组成关键词词组并打上该文档的分类标签, 与关键子句一同作为输入数据使用.

在分类预测时, 也使用 TextRank 算法提取对应文本中关键子句, 并综合考虑各个子句的预测标签和概率, 最终得出文本的预测标签.

3.2 KS-FastText 模型框架

经典的 FastText 由输入层、隐藏层和输出层组成. 本文在 KS-FastText 模型的输入层中添加计算模型, 即

© 中国科学院软件研究所 http://www.c-s-a.org.cn

先使用 TextRank 算法提取输入文本的关键子句, 同时使用 TF-IDF 筛选文本特征词词组作为输入数据的特征补充; 之后将得到的关键子句和特征词词组标记为当前文本的分类标签分别送入隐藏层计算. KS-FastText 的模型结构如图 2 所示.

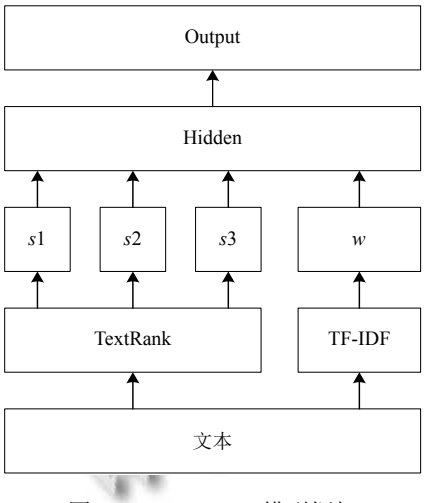

图 2 KS-FastText 模型框架

图 3 中, $s1$、$s2$、$s3$ 是文本经过 TextRank 方法提取的关键子句, w 是通过 TF-IDF 提取的文本关键词词组, 二者均标记为当前文本的标签类型, 作为 FastText 训练模型的输入. 输入的关键子句和关键词词组在输入到隐藏层前会被转换为各自对应词序列的特征向量, 特征向量通过线性变换映射到隐藏层, 该隐藏层通过求解最大似然函数后进行层次 Softmax 计算, 得到最终的输出.

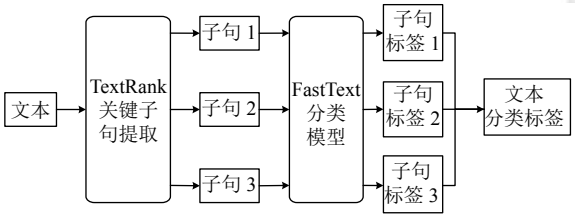

图 3 文本标签的预测流程

3.3 文本预测流程

文本标签的预测流程如图 3 所示. 文本在预测标签时也会采用 TextRank 算法提取关键子句, 之后将各个关键子句分别送入训练好的 FastText 模型中得到每个子句的标签和对应标签的概率, 最后综合各个子句的标签和对应的概率, 选择子句预测结果中概率最大的那个标签作为文本的标签.

4 实验分析

4.1 实验环境

实验环境为 Intel Core i7-8750H 处理器、主频 2.20 GHz、内存 16 GB、1 TB 的 PC 机. 操作系统为 Windows10, 编程语言使用 Python 3.7, 编译环境为 PyCharm 2019.

4.2 实验数据

本文实验采用了搜狐新闻分类数据集, 总共包含 3 万 6 千条数据, 平均每篇新闻字数为 2432 字, 在数据预处理阶段首先去除新闻文本中的图片链接, 同时除去纯图片、视频新闻, 之后按照标签、新闻标题、新闻内容的顺序整合成实验数据集. 数据集中的数据按照 7:3 的比例划分为训练集和测试集. 数据在送入模型训练之前采用 jieba 工具进行分词. 数据类别包括娱乐、财经、房地产、旅游、科技、体育、健康、教育、汽车、新闻、文化和女性 12 个类别, 数据组成如表 1 所示.

表 1 实验数据组成

类别	数量	类别	数量
娱乐	2934	财经	2877
房地产	2872	旅游	2998
科技	2988	体育	2978
健康	3000	教育	2979
汽车	2996	新闻	2935
文化	2995	女性	2997

4.3 评价方法

本文采用的评价指标包括准确率、精确率、召回率和 F 值.

数据中, FP 表示实际为负但被预测为正的样本数量, TN 表示实际为负被预测为负的样本的数量, TP 表示实际为正被预测为正的样本数量, FN 表示实际为正但被预测为负的样本的数量.

准确率是分类正确的样本占总样本个数的比例, 准确率 A 的计算公式为:

$$A=\frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

精确率是衡量测试集中预测为正类正确的比率, 由测试样本中预测正确的正例样本数量除以所有预测为正例的样本总数得到, 精确率 P 的计算公式为:

$$P=\frac{TP}{TP+FP} \tag{7}$$

召回率是衡量原有样本中有多少正例被预测, 由

© 中国科学院软件研究所 http://www.c-s-a.org.cn

原有样本中预测为正例的样本数除以样本中总正例的个数得到, 主要包含将样本中的正类预测为正类的数量 TP, 以及将正类预测为负类的数量 FN, 召回率 R 计算公式为:

$$R=\frac{TP}{TP+FN} \tag{8}$$

F 值是评价分类文本的综合指标, 是召回率与精确率的平均值, F 值的计算公式为:

$$F1=\frac{2\cdot P\cdot R}{P+R} \tag{9}$$

4.4 实验结果与分析

本文使用 Python 语言实现了 FastText 分类模型和 KS-FastText 分类模型. 实验过程中, 使用了 FastText 模型的默认参数: 学习速率 lr=0.1, 迭代次数 eporch=10, 词向量的维度 dim=100, 字级别的 n-gram 值设置为 2, 以适应中文的词语组成习惯, 词语的最小出现次数 minCount=1, 损失函数 loss 选用层次 Softmax.

本文实验对 KS-FastText 分类模型、经典 FastText 分类模型和贝叶斯文本分类模型在数据集上进行对比, 分别计算每个分类器综合的准确率、精确率、召回率和 $F1$ 值, 对比结果如表 2 所示.

表 2 各个分类器的实验结果 (%)

方法名称	贝叶斯分类模型	FastText分类模型	KS-FastText分类模型
准确率A	77.7	78.4	86.1
精确率P	81.0	78.7	83.0
召回率R	78.0	78.4	82.6
$F1$值	79.5	78.5	82.7

表 2 中的数据表明, 在本文的数据集中, KS-FastText 分类模型在各个评判参数上都要优于贝叶斯分类模型和经典 FastText 分类模型. KS-FastText 分类模型较贝叶斯分类模型和经典 FastText 分类模型在准确率上分别提高了 8.4% 和 7.7%, 在精确率上分别提高了 1.99% 和 4.28%, 在召回率上分别提高了 4.64% 和 4.26%, 在 F 值上分别提高了 3.24% 和 4.57%.

KS-FastText 分类模型在标签预测时, 对测试文本提取了关键子句, 并综合选择所有子句预测标签中概率最大的那个标签作为预测文本的标签, 本文比较了 KS-FastText 模型在预测时分别以子句 1、子句 2、子句 3 的预测标签和综合值作为文本的标签预测值时的准确率. 如图 4 所示.

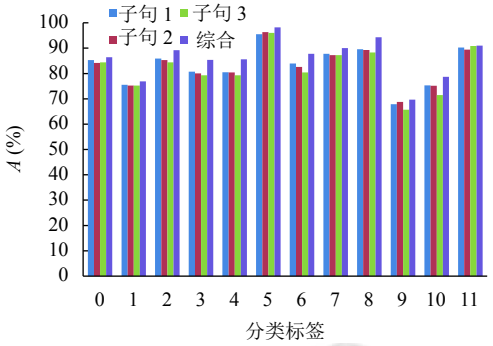

图 4 子句标签准确率情况

图 4 中的数据表明, 使用所有子句中最大概率标签作为预测文本标签时的分类准确率均高于以任何子句标签作为预测文本标签时的准确率. 说明采用综合子句标签判断文本标签的方法对模型的分类准确率有一定的提高.

本文同时也比较了各个分类模型在各个分类标签上的准确率值, 结果如图 5 所示.

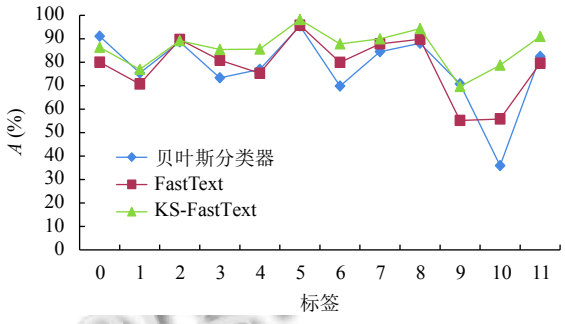

图 5 各个分类器在每一类标签上 A 值比较

图 5 中的数据表明, 在实验数据集中, KS-FastText 模型在大部分标签上分类的准确率优于经典的 FastText 分类模型和贝叶斯分类模型; 同时, KS-FastText 相较于贝叶斯分类器在各个分类标签上的准确率分布也更稳定.

实验结果证明, KS-FastText 分类模型采用关键子句抽取和关键词补充方法, 减少了中文长文本中无关词对分类结果的影响, 更适用于解决中文长文本分类问题.

5 结语

本文对 FastText 模型进行了改进, 以适应中文长文本环境. 在改进过程中, TextRank 用于提取文本关键子句, 以减少无关词语对分类结果的影响. 对于长文本

© 中国科学院软件研究所 http://www.c-s-a.org.cn

的子句按照独立分类的句子输入模型中训练, 而在预测结果的过程中, 文本分类标签取其各个子句预测标签中的概率最大值, 提高分类的准确率. 实验表明, 本文提出的 KS-FastText 方法在中文长文本环境中的效果较经典 FastText 算法有所提高.

参考文献

1 于游, 付钰, 吴晓平. 中文文本分类方法综述. 网络与信息安全学报, 2019, 5(5): 1–8. [doi: 10.11959/j.issn.2096-109x.2019045]

2 牛雪莹, 赵恩莹. 基于 Word2Vec 的微博文本分类研究. 计算机系统应用, 2019, 28(8): 256–261. [doi: 10.15888/j.cnki.csa.007030]

3 段旭磊, 张仰森, 孙祎卓. 微博文本的句向量表示及相似度计算方法研究. 计算机工程, 2017, 43(5): 143–148. [doi: 10.3969/j.issn.1000-3428.2017.05.023]

4 Borgers DP, Heemels WPMH. Event-separation properties of event-triggered control systems. IEEE Transactions on Automatic Control, 2014, 59(10): 2644–2656. [doi: 10.1109/TAC.2014.2325272]

5 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]

6 阴爱英, 吴运兵, 郑一江, 等. 基于 fastText 模型的词向量表示改进算法. 福州大学学报 (自然科学版), 2019, 47(3): 314–319.

7 Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space. arXiv preprint arXiv: 1301.3781, 2013.

8 马思丹, 刘东苏. 基于加权 Word2Vec 的文本分类方法研究. 情报科学, 2019, 37(11): 38–42. [doi: 10.13833/j.issn.1007-7634.2019.11.006]

9 杨萌萌, 黄浩, 程露红, 等. 基于 LDA 主题模型的短文本分类. 计算机工程与设计, 2016, 37(12): 3371–3377. [doi: 10.16208/j.issn1000-7024.2016.12.044]

10 叶雪梅, 毛雪岷, 夏锦春, 等. 文本分类 TF-IDF 算法的改进研究. 计算机工程与应用, 2019, 55(2): 104–109, 161. [doi: 10.3778/j.issn.1002-8331.1805-0071]

11 Liu PF, Qiu XP, Huang XJ. Recurrent neural network for text classification with multi-task learning. Proceedings of the 25th International Joint Conference on Artificial Intelligence. New York, NY, USA. 2016. 2873–2879.

12 Kim Y. Convolutional neural networks for sentence classification. arXiv preprint arXiv: 1408.5882, 2014.

13 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. 2016. 427–431.

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

15 李娜娜, 刘培玉, 刘文锋, 等. 基于 TextRank 的自动摘要优化算法. 计算机应用研究, 2019, 36(4): 1045–1050. [doi: 10.19734/j.issn.1001-3695.2017.11.0786]

© 中国科学院软件研究所 http://www.c-s-a.org.cn