采用稀疏自注意力机制和 BiLSTM 模型的细粒度情感分析

曹卫东 潘红坤*
(中国民航大学计算机科学与技术学院 天津 300300)

摘 要 使用 Word2vec 训练词向量、循环神经网络和注意力机制进行情感分析时,存在着文本特征提取不全面、计算资源消耗过多、计算时间较长的问题。为解决这些问题,提出新的 CBSA 网络模型。该模型使用 Cw2vec 预训练的词向量作为输入,双向长短期记忆网络(BiLSTM)来对这些具有时序信息的文本进行全面特征的提取;使用分解后的稀疏自注意力机制(Sparse Self-Attention)再次对这些文本特征进行权重赋予;由 Softmax 对文本进行情感的分类。实验结果表明,使用 Cw2vec 训练的词向量相比 Word2vec,F1-Score 大约提高 0.3%;CBSA 模型相比未分解的自注意力机制(Self-Attention),内存消耗减少了大约 200 MB,训练时间缩短了 210 s。

关键词 Cw2vec 细粒度情感分析 循环神经网络 双向长短期记忆网络 稀疏自注意力机制

中图分类号 TP391 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2022.12.028

FINE-GRAINED SENTIMENT ANALYSIS USING SPARSE SELF-ATTENTION MECHANISM AND BILSTM MODEL

Cao Weidong Pan Hongkun*
(*School of Computer Science and Technology, Civil Aviation University of China, Tianjin 30030, China*)

Abstract When using Word2vec to train word vectors, recurrent neural networks and attention mechanisms for sentiment analysis, there are problems of incomplete text feature extraction, excessive consumption of computing resources, and long computing time. To solve these problems, this paper proposes a new CBSA network model. The model used word vectors pre-trained by Cw2vec as input, and used bidirectional long-short-term memory network (BiLSTM) to extract comprehensive features of these texts with time-series information. The decomposed sparse self-attention mechanism (Sparse Self -Attention) was used to give weight to these text features again. Softmax was used to classify the sentiment on the text. The experimental results show that the word vector trained using Cw2vec has an F1-Score of about 0.3% higher than Word2vec. The CBSA model reduces memory consumption by about 200 MB and reduces training time by 210 s compared with the undecomposed self-attention mechanism (Self-Attention).

Keywords Cw2vec Fine-grained sentiment analysis Recurrent neural network Bidirectional long-short-term memory network Sparse self-attention mechanism

0 引 言

随着社交媒体的迅速发展,在线提供、搜索或共享意见已成为我们日常生活中的一项普遍活动。如此庞大的数据包含了对于内容提供商、社会机构和卖家等而言非常有价值的信息。例如,消费者在确定要预订某家酒店之前,可以查看许多在线评论,从而做出合适的选择。公司还可以直接从网络收集大量的公开信息,而不用对其服务或产品进行民意调查。因此,从此类数据中提取意见至关重要,这将有助于更多地了解用户的偏好或意图。

情感分析作为一种提取观点的技术,包含着许多相关任务,例如情感词典构建[1]、文档级情感分类[2]、方面级情感分类[3]和细粒度情感分析[4]。在这些任务中,细粒度情感分析是主要的和有价值的,因为它可

收稿日期:2020-06-16。民航科技创新重大专项(MHRD20160109);民航安全能力项目(TRSA201803);国家自然科学基金民航联合基金项目(U1833114)。**曹卫东**,副教授,主研领域:数据库与数据挖掘。**潘红坤**,硕士生。

以为许多下游任务提供感性短语的先验知识,例如,“酒店的环境真的不错,从室内的陈列到窗外的景色,以及酒店内花园都显得华贵而清幽”是积极短句,“门面特别小而且差,很不起眼。到前台办手续又发现态度特别差,房间也很一般”是消极短句,因此,细粒度情感分析对于评价的分析以及情感态度的分类具有较高的研究价值。

1 相关工作

细粒度情感分析在建设智能化城市中(如聊天机器人[5])起着至关重要的作用。然而计算机只能处理数字信息,要想完成情感分类的任务,首先需要将文本向量化。与传统的 one-hot 表示法不同,低维分布式词表示法(如 Word2vec[6])能够更好地捕获自然语言词的语义。鉴于汉字和汉字的内部结构丰富,每个中文字符通常比英文单词传达更多的语义信息,设计并学习中文单词的表示方法是至关重要的。Chen 等[7]提出了利用字符级别信息的方法来学习汉字嵌入。除了某些自定义的规则用于提取信息(偏旁部首[8]、组件[9]),还有基于像素学习字符的模型[10]。这些针对中文字符进行向量化的方法虽然比原始 Word2vec 效果好,但仍存在着字符语义信息提取不准确的问题。为更好地建模单词的语义,Cao 等[11]提出基于汉字笔画的 Cw2vec 模型,此模型能够自动获取中文单词间潜在的语义表示,为下游任务(情感分析)提供语义丰富的词向量。

作为句子分类的典型子问题,情感分类不仅需要理解单个话语的句子,还需要从整个会话中获取上下文信息。Pang 等[12]运用这些机器学习的方法在对英文电影评论的情感分析中取得较好的效果。Tripathy 等[13]在影评数据集上分别使用 NB(Naive Bayesian)、SVM(Support Vector Machine)方法,实验结果表明,SVM 的效果好于 NB。Srujan 等[14]通过人工构造文本特性如词性、情感,运用机器学习的方法来完成情感分析的任务,虽然取得了不错的成绩,但太多的人工标注文本特征,使得模型的实时性比较差。Kim[15]第一次将 CNN 卷积神经网络用在处理英文短文本情感分析,实现了句子级的分类任务,验证了深度学习网络在情感分析中的可靠性。但 CNN 在文本处理过程中并没有考虑上下文信息,针对一些具有时序信息的句子,效果不佳。Vania 等[16]使用卷积神经网络、长短期记忆网络对文本进行分析,使用预训练好的单词级别词向量和字符级别词向量作为特征输入,并证明字符级别词向量能够比词级别词向量学习到更好的特征。Zhang 等[17]使用 RNN 对中文的微博语料进行情感分析,训练带有词语信息和句子信息向量特征,最终证明计算句子向量的方式可以帮助学习句子的深层结构。Liang 等[18]将长短期记忆神经网络(LSTM)用于中文微博文本情感分析,解决了 RNN 梯度弥散问题。为了获取更加全面的句子特征,Xiao 等[19]提出双向长短期记忆神经网络(BiLSTM)的中文情感分析方法,把单向的 LSTM 网络反方向扩展,能够更好地利用文本前后的信息。无论是 CNN 还是 RNN,对文本的特征提取都是不全面的,无法区分不同的词,不同的句子对情感倾向的不同作用。基于此,通常会在这些网络之上构建额外的注意力层[20],以便将更多的注意力放在最相关的单词上,从而更好地理解句子。

传统的注意力机制关注每个单词与整个文本的联系,单词对全部序列具有依赖性,计算量较大。例如 Transformer[21]中的 Self-Attention 包含两次序列自身的矩阵乘法、计算量和显存占用量都为(N^2)级别的(N 代表句子的长度)。如果处理的序列较长(N),就会浪费太多的时间和内存开销(N^2)。Huang 等[22]提出一种交织稀疏自注意力机制,该机制的主要创新之处就是把紧密相似矩阵拆分成两个稀疏相似矩阵的乘积,用这两个连续的稀疏矩阵分别估算出一个相似的矩阵,第一个注意力机制用来估算长距离的相似性,第二个注意力机制用来估算短距离的相似性。类似于局部注意力机制,计算单词的权重时分别考虑不同的长度,该机制虽然节省了大量的内存和计算,但对于一些特长距离依赖的句子,效果不是很理想。考虑到长距离的依赖性,Child 等[23]提出一种跳跃式的注意力机制(StrideSelf-Attention),即每个单词只考虑与它距离为倍数的单词的关系,该方法虽然能大大缩短计算时间,降低内存消耗,但对一些近距离相关性较强的情感分类任务,准确率却相比自注意力机制低了的许多。基于此,本文对稠密的自注意力机制进行分解得到 Sparse Self-attention,使用 Cw2vec 预训练好的词向量作为输入,BiLSTM + Sparse Self-Attention 对文本进行特征提取。实验证明 Cw2vec + BiLSTM + SparseSelf - Attention 的组合模型(CBSA)在情感分析的任务中,不仅准确率较高,而且占用更少的内存,大大减低了神经网络模型的训练时间,实现了语句的局部紧密和远程相关的特性。本文的创新点主要有:

(1) 创新地使用 Cw2vec 训练词向量,获取中文的语义信息。

(2) 使用 BiLSTM 提取语句的上下文信息,获取全面的文本特征。

(3) 对稠密的 Self-Attention 进行分解,并发运行

多个注意力机制,节省内存,降低模型收敛时间。

2　模型介绍

本文设计的 CBSA 模型结构如图 1 所示,模型的整体计算流程为:

步骤 1　首先对评论数据集进行预处理,去除标点符号、无意义的高频词。

步骤 2　获取基于 Cw2vec 的词向量。

步骤 3　将得到的词向量作为 BiLSTM 的输入,获取语句上下文相关特征。

步骤 4　SparseSelf-Attention 对 BiLSTM 得到的特征重新进行权重分配。

步骤 5　最后由 Softmax 实现对情感极性的判断。

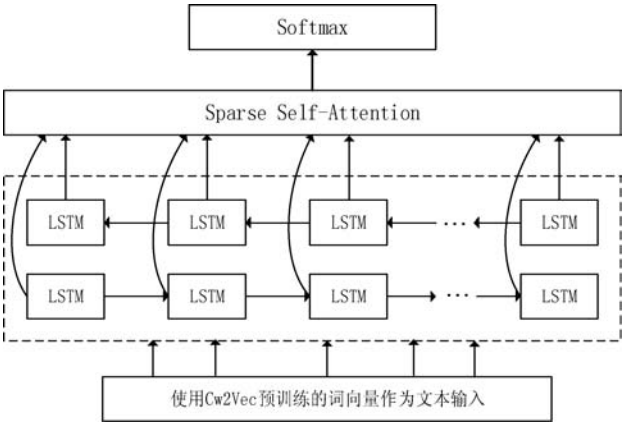

图 1　CBSA 模型结构图

2.1　Cw2vec 词向量

现有的词向量模型主要集中在英语、西班牙语和德语等欧洲语言上,这些语种在书写系统中采用的拉丁文字,与中文字符结构完全不同。而单个中文字符都包含着丰富的语义信息,使用 Word2vec 预训练中文词向量往往不能够全面地捕获语义信息。基于此,本文采用 Cw2vec 预训练词向量。

Cw2vec 是一种学习汉语单词嵌入的新方法,通过使用笔画 N-gram 设计了一种极简方式来学习文本特征,该笔画可以来获取汉字的语义和层次信息。中文向量化的过程如下:

(1) 语句的切分。基于字符级别切分成单个中文汉字。例如:"位置"切分成"位""置"。

(2) 字符笔画信息的获取。从每个字符获取笔画信息并把它们拼接起来。

位:撇、竖、点、横、点、撇、横。

置:竖、横折、竖、竖、横、横、竖、竖、横折、横、横、横、横。

位置:撇、竖、点、横、点、撇、横、竖、横折、竖、竖、横、横、竖、竖、横折、横、横、横、横。

(3) 笔画序列数字化:为每个笔画分配一个整数 ID,分别为 1 到 5,如表 1 所示。

表 1　笔画和 ID 的对应关系

笔画	横、提	竖、竖钩	撇	捺、点	折钩
ID	1	2	3	4	5

(4) 设置滑动窗口大小为 n,生成笔画 N-gram 特征,如图 2 所示。以"地理位置方便"为例,分词结果为"地理"、"位置"和"方便"。中心词为"位置",上下文单词为"地理","方便"。

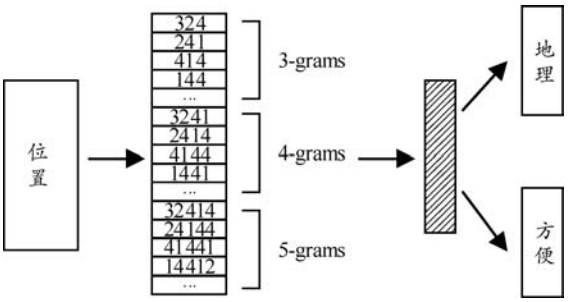

图 2　N-gram 特征图

Cw2Vec 模型考虑到单词与上下文之间的相关性,相似性定义为:

$$sim(w,c) = \sum_{q \in S(w)} \boldsymbol{q} \cdot \boldsymbol{c} \tag{1}$$

损失函数为:

$$L = \sum_{w \in D} \sum_{c \in T(w)} \log\sigma(sim(w,c)) + \lambda E_{\bar{c} \sim P}[\log\sigma(-sim(w,\bar{c}))] \tag{2}$$

最后使用 Softmax 函数对给定 w 的 c 预测模型进行建模:

$$p(c \mid w) = \frac{\exp(sim(w,c))}{\sum_{\bar{c} \in V} \exp(sim(w,\bar{c}))} \tag{3}$$

式中:w、c 分别为当前单词和上下文单词;$S(w)$ 为单词 w 的笔画 N-grams 的集合;q 为 $S(w)$ 的笔画元素,$\boldsymbol{q}$ 为 q 的向量嵌入;$T(w)$ 为所有具有窗口大小的上下文单词给定的当前单词的集合,D 是训练语料库中所有单词的集合;λ 是负样本的数量;$E_{\bar{c} \sim P}$ 是期望项,选定的负样本 $\bar{c}$ 服从 P 分布。

2.2　BiLSTM 上下文信息提取

假设输入文本用 $\boldsymbol{X}$ 表示,由 L 个单词组成。使用 Cw2vec 进行训练,得到词嵌入表示特征 $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_L\}$,其中 $\boldsymbol{v}_i$ 为每个单词的向量属于 $\mathbf{R}^k$,k 表示每个词的维度,在本文中与训练的词向量维度为 300。则一条评论的向量化为:

$$\boldsymbol{X} = \{\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_L\} \tag{4}$$

循环神经网络在处理时序信息的问题上相比于卷

积神经网络具有很好的优势。RNN 利用激活函数序列输入特征表示 $\boldsymbol{X}_t$ 和前一时段的隐藏层输入值 h_{t-1}，并转化为当前隐藏状态的输出值 h_{t-1}：

$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, \boldsymbol{X}_t) \tag{5}$$

LSTM 则可以解决 RNN 梯度弥散问题。LSTM 的优势在于具有三种特殊的门函数，即输入门、遗忘门、输出门，如图 3 所示[18]。

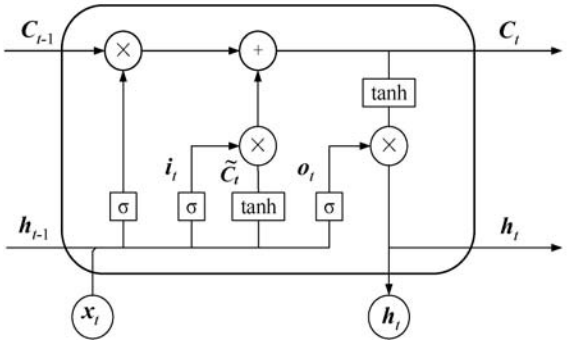

图 3 单个 LSTM 计算过程示意图

遗忘门：

$$\boldsymbol{f}_t = \sigma(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_f) \tag{6}$$

输入门：

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_i) \tag{7}$$

$$\tilde{\boldsymbol{C}}_t = \tanh(\boldsymbol{W}_c \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_c) \tag{8}$$

$$\boldsymbol{C}_t = \boldsymbol{f}_t \times \boldsymbol{C}_{t-1} + \boldsymbol{i}_t \times \tilde{\boldsymbol{C}}_t \tag{9}$$

输出门：

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{x}_t] + \boldsymbol{b}_o) \tag{10}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \times \tanh(\boldsymbol{C}_t) \tag{11}$$

式(3)－式(8)中：$\{\boldsymbol{W}_*, \boldsymbol{b}_*\}$ 为神经网络训练的参数集合；$\tilde{\boldsymbol{C}}_t$、$\boldsymbol{f}_t$、$\boldsymbol{i}_t$、$\boldsymbol{o}_t$ 分别为时刻 t 记忆的输入单元、遗忘门、输入门和输出门的输出值；$\boldsymbol{h}_{t-1}$、$\boldsymbol{x}_t$ 分别表示时刻 t 上一个记忆单元以及当前的记忆单元的输入；$\boldsymbol{C}_t$ 表示时刻 t 记忆单元的内部状态；$\boldsymbol{h}_t$ 表示时刻记忆单元的输出。

考虑到文本分析要用到上下文信息，本文模型选用双向 LSTM(BiLSTM)为基础建模，双向的长短期记忆网络由正向和反向的 LSTM 组成，计算过程如下：

$$\overrightarrow{\boldsymbol{h}}_t = f_{\text{LSTM}}(\boldsymbol{h}_{t-1}, \boldsymbol{X}_t) \tag{12}$$

$$\overleftarrow{\boldsymbol{h}}_t = f_{\text{LSTM}}(\boldsymbol{h}_{t+1}, \boldsymbol{X}_t) \tag{13}$$

$$\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}}_t, \overleftarrow{\boldsymbol{h}}_t] \tag{14}$$

式中：$\overrightarrow{\boldsymbol{h}}_t$ 表示正向的 LSTM 得到的特征表示；$\overleftarrow{\boldsymbol{h}}_t$ 表示反向的 LSTM 得到的特征表示；$\boldsymbol{h}_t$ 为 BiLSTM 在 t 时刻的最终输出。

2.3 多头自注意力机制

Google 在 2017 年提出一种新的注意力机制——多头自注意力[24]。相比于单一的注意力机制而言，多头机制能够从多方面捕获序列的关键信息。每个头通过向量点积进行相似度运算，得到 Attention 值，结构如图 4 所示。

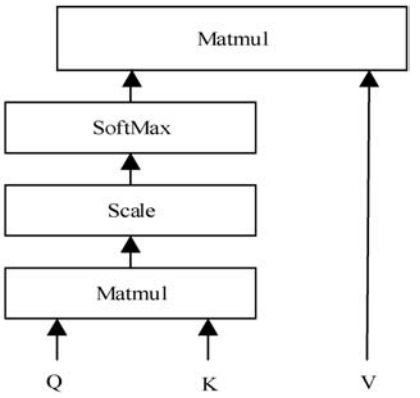

图 4 缩放点积运算(SDA)

$$SDA(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}}{\sqrt{d_K}}\right)\boldsymbol{V} \tag{15}$$

式中：查询向量序列 $\boldsymbol{Q} \in \mathbf{R}^{l \times d_k}$，键向量序列 $\boldsymbol{K} \in \mathbf{R}^{m \times d_k}$，值向量序列 $\boldsymbol{V} \in \mathbf{R}^{m \times d_v}$，输入文本的长度为 L，为了避免内积过大，增加缩放因子 $\frac{1}{\sqrt{d_k}}$，用于调和。

多头自注意力机制就是将(Q、K、V)通过线性转换送入到 SDA，再重复运算 h 次，最后拼接所有的 Attention，缩小每个 head 的尺寸，其计算成本和具有全维度的单个 Attention 机制相当。结构示意图如图 5 所示。

$$head_i = SDA(\boldsymbol{Q}\boldsymbol{W}_i^{Q}, \boldsymbol{K}\boldsymbol{W}_i^{K}, \boldsymbol{V}\boldsymbol{W}_i^{V}) \tag{16}$$

$$Head = MultiHead(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = Concat(head_1, head_2, \cdots, head_h)\boldsymbol{W}^{O} \tag{17}$$

式中：$\boldsymbol{W}_i^{Q} \in \mathbf{R}^{d_k \times \tilde{d}_k}$，$\boldsymbol{W}_i^{K} \in \mathbf{R}^{d_k \times \tilde{d}_k}$，$\boldsymbol{W}_i^{V} \in \mathbf{R}^{d_v \times \tilde{d}_v}$，$\boldsymbol{W}_i^{V} \in \mathbf{R}^{\tilde{d}_v \times d_k}$，$d_K = h\tilde{d}_v$，最后的输出序列 $Head \in \mathbf{R}^{l \times (h\tilde{d}_V)}$。本文对自注意力机制进行分解，即 $\boldsymbol{Q} = \boldsymbol{K} = \boldsymbol{V} = \boldsymbol{X}$ 的情况。

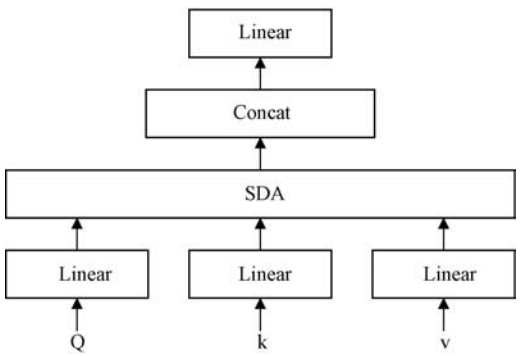

图 5 Multi-head Attention 结构

2.4 分解 Self-Attention

多头注意力机制关注每个元素对短语的影响，图 6 所示为 x_i 与 X 序列中的每个单词计算相关度，其计

算量为 $O(n^2)$,其中 n 为输入短语的长度。

图 6　Self-Attention

可以看出 Self-Attention 无论是内存消耗量还是计算量上都是十分庞大的。为改善这一状况,一个基本的思想就是减少语句相关性的计算,设定每个元素只跟短语中的一部分元素有关。常见的有 Local Self-Attention,即放弃全局关联,规定每个元素只与前后 c 个元素以及自身有关,相对距离超过 c 的注意力直接设为 0,如图 7 所示(其中 c 被设置为 2 的情况)。

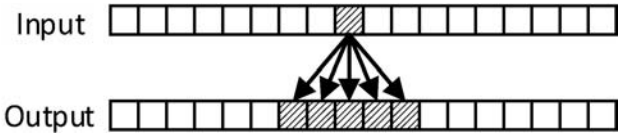

图 7　Local Attention

局部注意力机制(Local Self-Attention)虽然节约了内存,缩短计算时间,但忽略了远程的相关性。为获取长程关联性,本文提出新稀疏自注意力机制(Sparse Self-Attention),该注意力机制拥有 p 个独立的注意力头,每个独立的注意力只关注于特定位置的元素。设想每个元素只与它局部相当距离不超过 c 的,且远程距离为 k,$2k$,$3k$,…的元素相关(c、k 为提前设置好的参数),强行设置其他位置的元素注意力为 0。如图 8 所示(c、k 分别被设置为 2、5 的情况)。

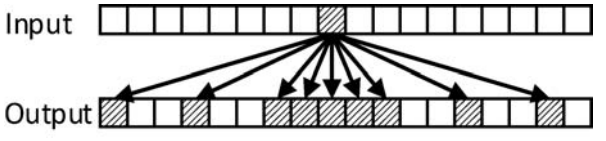

图 8　Sparse SelfAttention

稀疏自注意力机制(Sparse Self-Attention)将 BiLSTM 输出的 $\boldsymbol{h}_t$ 矩阵映射为输出矩阵,并由全连接模式 $S=\{S_1,S_2,\cdots,S_n\}$ 参数化,其中 S_i 为第 i 个输入向量参与其中的索引值,n 为输出序列的长度。

$$Attend(\boldsymbol{h}_t,S)=(a(h_{ti},S_i))_{i\in\{1,\cdots,n\}} \tag{18}$$

$$A_i^{(m)}=S_i \tag{19}$$

$$A_i^{(1)}=\{i-c,\cdots,i-1,i,i+1,\cdots,i+c\} \tag{20}$$

$$A_i^{(2)}=\{j,\cdots,i-k,i,i+k,\cdots,l\} \tag{21}$$

$$attention(\boldsymbol{h}_t)=\boldsymbol{W}_p(attend(\boldsymbol{h}_t,A)^{(i)})_{i\in\{1,2,\cdots,p\}} \tag{22}$$

式(18)中:$(i-j)\bmod k=0$,$(l-i)\bmod k=0$。其中 $A_i^{(m)}$ 代表第 m 个注意力头 A 在第 i 位置时关注的元素索引,$\boldsymbol{W}_p$ 为权重矩阵。新的机制既实现了局部紧密性,又实现远程相关性。与传统 Self-Attention 相比,无论是在内存消耗还是计算时间都有着很大的优势。

2.5　情感分类

本文使用 softmax 分类器来完成文本的情感分析,其中 $\boldsymbol{W}$、$\hat{\boldsymbol{X}}$、$\boldsymbol{b}$ 分别为权重矩阵、文本特征向量和偏置。输出分类概率为

$$y=\text{softmax}(\boldsymbol{W}\times\hat{\boldsymbol{X}}+\boldsymbol{b}) \tag{23}$$

为了防止过拟合,在 Softmax 之前添加了 Dropout[24],随机丢弃一些网络节点,能够显著地提高模型的泛化能力。使用反向传播算法,采用的交叉熵损失函数为:

$$loss=-\sum_i\sum_j\hat{y}^j\log y_i^j+\lambda\|\theta\|^2 \tag{24}$$

式中:i 为训练数据集;j 为情感分类的类别;$\hat{y}$ 为预测的情感类别;y 为实际的情感类别;$\lambda\|\theta\|^2$ 为 L2 正则化。

3　实验与结果分析

本文实验测试为 64 位 Ubuntu 操作系统,开发环境为 Python3.7,Keras 2.3.1,后端为 TensorFlow 2.0,开发工具为 PyCharm。

3.1　数据集

本文实验室数据集为中国科学院谭松波博士整理的中文酒店评论数据集,共有 10 000 条评论组成,包含着 7 000 条积极情感和 3 000 条消极情感。情感标签分为两类[0,1],消极情感为 0,积极情感为 1,为平衡数据集,实验选取 6 000 条数据集,正负样本各3 000 条。本文使用 sklearn 中的 train_test_split 随机抽取 90% 作为训练集,10% 作为测试集,进行多次实验,选取平均值作为实验的结果。实验数据集如表 2 所示。

表 2　实验数据集

消极评论	积极评论
房间设备差,几乎就是招待所的标准,周围环境脏乱差,噪声大、烧烤气味刺鼻。	服务质量很高,市中心的环境也很吸引人,还不错的!
早餐很差,check-in 服务也差,我预定的无烟大床,结果进房间发现是 2 张床的房间,要他们换,发现是大床没错,但是是吸烟房间,唉,将就了。后来发现床垫上放屁股处有一个大大的凹陷,绝对是劣质货。	朋友推荐的酒店,位置不错,交通很方便,酒店的服务也很贴心,入住时前台适时地送上一杯冰柠檬茶,让人感受到酒店的细心服务,夏日的夜晚坐在露天啤酒花园喝着啤酒,与朋友畅谈很是惬意。
房间巨小,电视成了摆设,开不了,服务员态度冷漠。以后不会再进这家酒店	服务态度比较好,愿意听取客户所提出意见

3.2 评估标准

本文评估指标主要有这三个:准确率(Precision)、召回率(Recall)和 F1-Score。

$$Precision = \frac{TP}{TP + FP} \tag{25}$$

$$Recall = \frac{TP}{TP + FN} \tag{26}$$

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{27}$$

3.3 实验参数设置

深度学习模型不同的参数设置会直接影响整个实验的分类效果,对情感分析是特别重要的。本文实验中,与 Child[23] 处理长文本序列设置的 $c = 32, k = 128$ 的情况不相同,处理的文本信息长度绝大部分都在 100 个以内,所以 c、k 按照相同的倍数缩小设置为 4、16。注意力头数的设置选取最优的 4 个注意力头,通过多次对比实验,选取效果最好的参数,详细设置如表 3 所示。

表 3 模型参数设置

实验参数	取值
词向量维度	300
BiLSTM 单元个数	64
Multi-Heads	4
c	4
k	16
Dropout	0.4
Loss	softmax_cross_entropy
Optimizer	Adam
学习率	0.001
批处理大小	128
迭代轮数	20

3.4 结果分析

3.4.1 中文词向量的嵌入

本实验使用 Cw2vec 对文本进行词向量的训练,往往能够更好地获得中文字符间联系。如表 4 所示,分别以 Cw2vec 和 Word2vec 作比较。其中余弦相似度越大,单词间关系越紧密,余弦相似度越小,单词间相关性越小。

表 4 词向量余弦相似度比较

词向量	单词	单词	余弦相似度
Word2Vec	昏暗	陈旧	0.351 8
	雾霾	雷雪	0.338 1
	愉悦	愉快	0.549 6
Cw2Vec	昏暗	陈旧	0.381 3
	雾霾	雷雪	0.354 9
	愉悦	愉快	0.556 2

从表 4 中可以明显看出基于笔画预训练 Cw2vec 向量更能获取单词之间的相关性,特别是对一些字符结构相似的,例如“雾霾”“雷雪”能够更好地获取字符特征,相比 Word2vec 预训练词向量的余弦相似度提高了 2%。

实验整体效果而言,分别使用 Word2vec 和 Cw2vec 预训练的词向量作为 BiLSTM + Sparse Self-Attention 组合模型的输入。观察表 5 可以发现,基于 Cw2vec 的词嵌入比基于 Word2vec 的词嵌入整体效果 F1-Score 大约提高 0.3%,验证了基于笔画 Cw2vec 训练词向量作为输入的可靠性。

表 5 词向量对整个实验的影响(%)

词向量	Precision	Recall	F1
Word2vec	**92.80**	92.75	92.77
Cw2vec	91.96	**94.27**	**93.10**

3.4.2 单一模型比较

为了验证本文提出的模型的有效性,笔者选取以下几种模型作为对照实验,全部使用预训练好的 Cw2vec 词向量作为输入,如表 6 所示。

表 6 酒店数据分析结果(%)

模型	Precision	Recall	F1
SVM	81.26	81.52	81.39
CNN	86.23	85.94	86.08
LSTM	87.84	88.09	87.96
Bi-LSTM	88.57	88.89	88.73
CBSA 模型	**91.96**	**94.27**	**93.10**

观察表 6 可以发现,在细粒度情感分类任务中,深度学习算法(CNN、LSTM 和 BiLSTM)明显比传统的机器学习算法 SVM 效果要好,这是因为深度神经网络能够深层次、多维度的自动提取文本特征,在细粒度情感分析任务中,深度学习算法(CNN、LSTM、BiLSTM)明显比传统的机器学习算法 SVM 效果要好,这是因为深度神经网络能够深层次、多维度地自动提取文本特征,

有效地避免了机器学习方法中人工提取特征的缺陷。对比 CNN、LSTM 和 BiLSTM 可以发现,BiLSTM 不仅可以捕获长距离文本的依赖关系,而且还能够获取文本从后往前的信息。而添加注意机制的 CBSA 网络模型则能够更好地给不同的词赋予不同的权重,克服了特征无差别提取的缺点,在准确率、召回率和 F1-Score 都有着明显的提高。实验结果表明,添加注意力机制后的效果大约提高 5%,相比较其他模型在情感分析任务上更有优势。

3.4.3　分解多头自注意力机制的影响

表 7 为分解自注意力机制的实验结果对比,可以看出分解后的稀疏自注意力机制在 F1 值几乎没下降,并未影响实验分类精度。观察图 9 和图 10 可以发现本文的模型相比未分解的自注意力机制,占用更少的内存开销,减少了大约 200 MB,训练时间降低了210 s,综上可以验证本文提出的稀疏自注意力机制不仅仅能够得到较高的分类结果,而且占用更少的内存开销、降低模型的训练时间。

表 7　分解自注意力机制的结果(%)

模型	Precision	Recall	F1
未分解的 Self-Attention	93.53	93.19	93.16
CBSA 网络模型	**91.96**	**94.27**	**93.10**

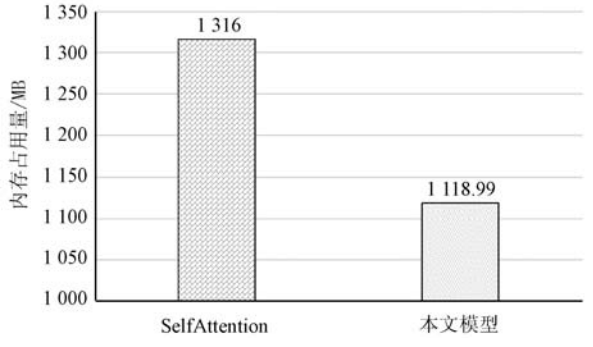

图 9　内存使用量

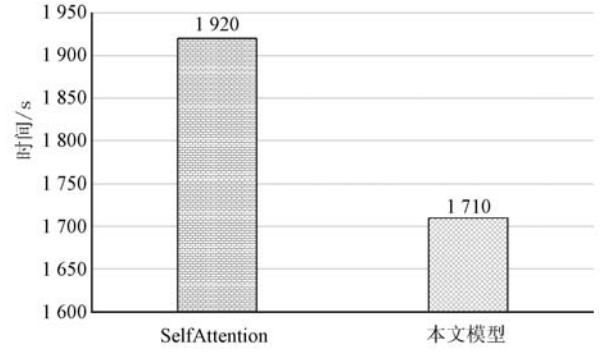

图 10　模型完成 20 轮迭代的训练时间

3.4.4　多头自注意力机制参数的影响

在 Google 提出的 Transformer[13] 中,注意力的头数 $h=8$,词向量的维度 $d_k=512$,每个头的注意力大小为 $\hat{d}_v=d_k/h=64$。在本文中,要适当地调整参数的大小来适应情感分析任务。BiLSTM 输出的词向量维度 $d_k=128$,h 测试四个不同的值:1、2、4、8,每个头的注意力大小 $\hat{d}_v$ 分别为 128、64、32、16,结果如表 8 所示。

表 8　注意力头数对实验结果的影响

模型	h	$\tilde{d}_v$	F1 值/%
Multi-heads	1	128	91.36
	2	64	91.45
	4	32	**93.10**
	8	16	92.90

从表 8 可以看出,多头自注意力机制参数的选择影响着模型的性能,当 $h=4$,$\hat{d}_v=32$ 时,实验效果是最优的。证明了多头自注意力机制的优点:多头并发计算,可以获取序列不同方面的信息。

4　结　语

本文提出一种新的 CBSA 网络模型来进行细粒度情感分析。一方面,使用基于 Cw2vec 预训练的词向量,能够更好地获取语义特征。另一方面,通过对自注意力机制进行分解,实现了局部紧密性、远程稀疏性的特性。实验表明,本文提出的模型能够挖掘丰富的隐藏情感信息,占用更少的内存、时间开销,更加准确地完成情感分析任务。

然而 Cw2vec 基于笔画训练词向量,是一个固定的静态编码表示,例如:“大夫”和“丈夫”的笔画是一样的,这样就会导致语义理解的偏差。未来的工作计划准备采用大语料下预训练的 BERT 模型,每个单词的词向量根据不同的上下文信息动态地表示,这样才能消除 Cw2vec 一词多义的问题。

参　考　文　献

[1] Dong D, Jing L, Yu J, et al. Sparse self-attention LSTM for sentiment lexicon construction[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2019, 11(99): 1777 – 1790.

[2] Wang Z, Lee S Y M, Li S, et al. Emotion analysis in code-switching text with joint factor graph model[J]. IEEE/ACM Transactions on Audio, Speech and Language Processing, 2017, 25(3): 469 – 480.

[3] Lee C W, Song K Y, Jeong J, et al. Convolutional attention networks for multimodal emotion recognition from speech and text data[EB/OL]. [2022-11-17]. https://arxiv.org/abs/

1805.06606v22018.
[4] Sahay S, Kumar S H, Xia R, et al. Multimodal relational tensor network for sentiment and emotion classification[EB/OL]. [2022-11-17]. https://arxiv.org/abs/1806.02923v1.
[5] Luo L, Yang H, Chin F Y L. EmotionX-DLC: Self-attentive BiLSTM for detecting sequential emotions in dialogues[EB/OL]. [2022-11-17]. https://arxiv.org/abs/1806.07039v1.
[6] Mikolov T, Chen K, Corrado G S, et al. Efficient estimation of word representations in vector space[EB/OL]. [2022-11-17]. https://arxiv.org/abs/1301.3781? context=cs.CL.
[7] Chen X, Xu L, Liu Z, et al. Joint learning of character and word embeddings[C]//Proceedings of the 24th International Conference on Artificial Intelligence. ACM, 2015:1236 - 1242.
[8] Yin R, Wang Q, Li P, et al. Multi-granularity chinese word embedding[C]// Proceedings of the 24th International Conference on Artificial Intelligence. ACM, 2016:1236 - 1242.
[9] Yu J, Jian X, Xin H, et al. Joint embeddings of Chinese words, characters, and fine-grained subcharacter components [C]// Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.
[10] Su T, Lee H. Learning Chinese word representations from glyphs of characters[EB/OL]. [2022-11-17]. https://arxiv.org/abs/1708.04755.
[11] Cao S, Lu W, Zhou J, et al. cw2vec: Learning Chinese word embeddings with stroke n-gram information[C]// Proceedings of the AAAI Conference on Artificial Intelligence, 2018: 5053 - 5061.
[12] Pang B, Lee L. Opinion mining and sentiment analysis[J]. Foundations and Trends? in Information Retrieval, 2008, 2(1/2): 1 - 135.
[13] Tripathy A, Agrawal A, Rath S K. Classification of sentimental reviews using machine learning techniques[J]. Procedia Computer Science, 2015, 57: 821 - 829.
[14] Srujan K S, Nikhil S S, Rao H R, et al. Classification of amazon book reviews based on sentiment analysis[M]. Information Systems Design and Intelligent Applications. Singapore: Springer, 2018: 401 - 411.
[15] Kim Y. Convolutional neural networks for sentence classification[EB/OL]. [2022-11-17]. https://arxiv.org/abs/1408.5882.
[16] Vania C, Lopez A. From characters to words to in between: Do we capture morphology? [EB/OL]. [2022-11-17]. https://arxiv.org/abs/1704.08352v1.
[17] Zhang Y, Jiang Y, Tong Y. Study of sentiment classification for Chinese microblog based on recurrent neural network[J]. Chinese Journal of Electronics, 2016, 25(4): 601 - 607.
[18] Liang J, Chai Y, Yuan H, et al. Polarity shifting and LSTM based recursive networks for sentiment analysis[J]. Journal of Chinese Information Processing, 2015, 5: 152 - 159.
[19] Xiao Z, Liang P. Chinese sentiment analysis using bidirectional LSTM with word embedding[C]// International Conference on Cloud Computing and Security, 2016: 601 - 610.
[20] Tang J, Lu Z, Su J, et al. Progressive self-supervised attention learning for aspect-level sentiment analysis[EB/OL]. [2022-11-17]. https://arxiv.org/abs/1906.01213v2.
[21] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing, 2017: 5998 - 6008.
[22] Huang L, Yuan Y, Guo J, et al. Interlaced sparse self-attention for semantic segmentation[EB/OL]. [2022-11-17]. https://arxiv.org/abs/1907.12273v1.
[23] Child R, Gray S, Radford A, et al. Generating long sequences with sparse transformers[EB/OL]. [2022-11-17]. https://arxiv.org/abs/1904.10509.
[24] Agarwal A, Negahban S, Wainwright M J. A simple way to prevent neural networks from overfitting[J]. The Journal of Machine Learning Research, 2014, 40(2): 1171 - 1197.

(上接第 186 页)

[16] Kamal A, Abulaish M, Anwar T. Mining feature opinion pairs and their reliability scores from web opinion source [C]//2nd International Conference on Web Intelligence, 2012:15 - 21.
[17] Desai J, Hariya S. Sentiment analysis approach to adapt a shallow parsing based sentiment lexicon[C]//2015 International Conference on Innovations in Information, Embedded and Communication Systems(ICIIECS), 2015: 1 - 4.
[18] 顾正甲,姚天昉.评价对象及其倾向性的抽取和判别[J].中文信息学报,2012,26(4):91 - 97.
[19] Qian Q, Huang M, Lei J, et al. Linguistically regularized LSTMs for sentiment classification[C]//55th Annual Meeting of the Association for Computational Linguistics, 2016: 1117 - 1154.
[20] Mauro D, Andrea G, Tettamanzi B. Propagating and aggregating fuzzy polarities for Concept-Level sentiment analysis [J]. Springer, 2015, 7(2): 186 - 197.
[21] 刘亚桥,陆向艳,邓凯凯,等.摄影领域评论情感词典构建方法[J].计算机工程与设计,2019,40(10):3037 - 3042.
[22] Appel O, Chiclana F, Carter J. A Hybrid approach to sentiment analysis[C]//IEEE Congress on Evolutionary Computation(CEC), 2016:4950 - 4957.
[23] sentiment-analysis[EB/OL]. [2020 - 06 - 23]. https://github.com/AIChallenger/AI_Challenger_2018/sentiment-analysis.
[24] fastText[EB/OL]. (2020 - 04 - 28). https://github.com/facebookresearch/fastText.
[25] Yang H, Yang J, Song Y. A Multi-task learning model for Chinese-oriented aspect polarity classification and aspect term extraction[EB]. arXiv:1912.07976, 2019.