

文章编号: 1003-0077(2019)05-0031-08

融合语言特征的卷积神经网络的反讽识别方法

卢欣, 李阳, 王素格

(山西大学 计算机与信息技术学院, 山西 太原 030006)

摘要: 面对某些热点事件, 微博评论者经常使用反讽来表达对于该事件的看法, 以往的情感分析任务往往忽略这一语言现象。为了提高微博情感分析的准确率, 该文对反讽识别开展了研究。通过分析中文文本的语言现象和社交网络的特性, 归纳了中文微博反讽的语言特征, 提出了一种融合语言特征的卷积神经网络(CNN)的反讽识别方法。该方法将反讽特征和句子分别采用 Word Embedding 作为输入, 再卷积、池化后, 将其全连接融合, 构建了新的卷积神经网络模型。实验结果表明, 该方法在反讽识别的性能上优于传统的基于机器学习的方法。

关键词: 微博; 反讽识别; 卷积神经网络; 语言特征

中图分类号: TP391 **文献标识码:** A

Linguistic Features Enhanced Convolutional Neural Networks for Irony Recognition

LU Xin, LI Yang, WANG Suge

(School of Computer and Information Technology, Shanxi University, Taiyuan, Shanxi 030006, China)

Abstract: Irony is popular in Weibo comments, which is less addressed in past sentiment analysis community. To improve the accuracy of Weibo sentiment analysis, we study irony recognition in this paper. By analyzing the characteristics of Chinese language and social networks, irony linguistic features are summarized. Combining these linguistic features with convolutional neural networks (CNN), a novel method is proposed for recognizing irony. The proposed method combines irony features representation and sentences representation using word embedding as the input of a convolution network. The experimental results indicate that the proposed method is superior to classical machine learning methods for irony recognition.

Keywords: Weibo; irony recognition; convolutional neural networks; linguistic features

0 引言

微博作为新兴的媒体, 语言使用一般比较随意, 面对一些热点话题, 作者常常使用反讽来表达对该话题强烈情感倾向。反讽作为说话或写作时采用的一种带有讽刺意味的语气或写作手段^[1-2], 通常以间接的方式表达真实的意图。因此, 不论在何种语言环境下, 反讽的字面意思和真实意图之间均存在较大的差异^[3], 对其使用的识别是一个复杂的认知过程^[2]。面对微博中出现的反讽表达, 如何对其进行自动识别, 成为研究者关注的课题。依据微博中情感的指向性, 反讽可以划分为两类: ①表达积极情

感的反讽, 主要目的在于以幽默的手法表达对某些人和事物的赞赏。例如, “我家木匠的手艺太差劲了! 他做的门窗硬是找不到缝, 他刨的木板连苍蝇也落不住。”“门窗找不到缝”是在说门窗做得严丝合缝, “苍蝇落不住”是在说木板表面光滑, 表面上是说木匠的手艺差, 实际上是在以幽默的方式表达对木匠手艺的赞扬。②表达消极情感的反讽, 主要目的是以反讽的手法表达对某些人和事物的强烈不满。例如: 在针对铁路火车票降价的微博评论中, “全铁路火车票降价五毛是年度最牛的降价!”, 表面上是在赞扬, 实际表达了作者对此次降价的不满情绪。如何识别这些文本中具有情感的反讽表达, 以提高微博情感分析的准确率, 同时正确地进行舆情分析,

收稿日期: 2019-01-25 定稿日期: 2019-02-28

基金项目: 国家自然科学基金(61632011, 61672331, 61573231, 61432011, 61603229); 山西省重点研发计划(201803D421024)

成为研究者关注的热点问题。

中文反讽识别研究目前主要面临以下的困难:

①没有权威的中文反讽数据,反讽语料只能靠各研究单位通过手工标注,然而受主观因素的影响,语料质量难以保证;②微博是一种比较自由的媒体,文本数据一般比较短,且口语化,难以获取上下文的语义信息。而且反讽的字面意义和实际意义往往存在冲突,需要获取其深层的语义信息;③由于反讽与语言习惯有关,不同语言的语法结构存在差异,一些英文反讽的方法并不能直接应用在中文反讽识别上。和英文反讽识别相比,中文的语法结构和语义更加复杂,一些反讽甚至连人工也难以准确鉴别,中文中的谐音词和语气词也使得中文反讽识别比英文反讽识别难度更大。

针对反讽识别存在的问题,当前主流研究方向主要关注有效的显式反讽特征的选取。然而,由于反讽表达的复杂性,使得传统特征选择方法无法挖掘句子深层语义,单纯通过显式特征的方法难以达到较高的精度。卷积神经网络(CNN)是目前较流行的深度学习技术,能够将底层的特征经过卷积池化后获取抽象的特征,用于挖掘句子的深层语义信息,进而学习到有用的隐式特征,提高分类准确性。本文在借鉴英文反讽识别的基础上,结合中文特有的语言现象和微博的特性,归纳了反讽的几种显式语言特征,再利用这些特征融入卷积神经网络中,对反讽进行识别,以弥补传统特征方法无法挖掘句子深层语义信息的不足。

1 相关工作

针对英文反讽识别研究,主要从特征选取的研究角度出发。Gonzalez-Ibanez等^[4]通过词汇特征来识别反讽,构建了反讽词汇和“@〈用户〉”标签的反讽特征体系,实验结果表明单纯通过词汇特征无法准确有效地识别反讽。Reyes等^[5]选取了n元文法、词性的n元文法、幽默指数、词汇极性、情感复杂度和愉快程度六种特征,实验结果表明以上特征对于特定领域的反讽识别有明显的作用。Konstantin等^[6]在各种分类模型下结合各种反讽特征进行研究,实验结果表明人工选取的特征在提高准确率的同时降低了召回率,然而实验发现词袋模型就能解决这一问题。Edwin^[7]在社交媒体的数据集上结合了消极情感信息和感叹词数量等特征,在最大熵模型上的识别精确率达到了78.4%。David等^[8]在

Twitter数据集上发现上下文信息对反讽识别有重要的意义。Aditya等^[9]介绍了当前基于特征选择方法选取的反讽特征和分类模型。以上所有模型均使用传统的机器学习模型,未能挖掘到反讽的深层语义信息,导致反讽识别的精度不高。Aniruddha Ghosh等^[10]首先采用CNN和LSTM来识别反讽,实验效果表明深度学习方法优于传统的机器学习方法。Yu-Hsiang Huang等^[11]在Word Embedding的帮助下采用三种深度学习模型来识别反讽:卷积神经网络(CNN)、循环神经网络(RNN)和带注意力机制的循环神经网络(Attentive RNN),实验表明Attentive RNN在Twitter数据集上得到了最好的效果。Abhijit Mishra等^[12]利用认知NLP系统,即通过眼球的运动、脑电图信号、脑成像等技术扩充传统的基于文本的特征提取。Lotem Peled等^[13]采用单语机器翻译(MT)技术将反讽的英文翻译为非反讽的英文,通过对非反讽英文情感倾向进行打分来判别是否是反讽,文献公布了3 000条反讽的实验语料。

中文反讽识别研究目前还处于探索阶段。Tang等^[14]构建了一个繁体字的反讽语料库,人工总结了反讽常用的句式结构,并验证了选取结构的有效性。邓钊等^[15]通过中文的特征来识别反讽,构建了基本词汇情感、谐音词、连续的标点符号、微博的长度、动词被动化和双引号内外情感模糊度等六种特征,从新浪微博中标记了300条反语和28 545条非反语构建了实验数据集,实验获得了最高76.74%的准确率。罗婷^[16]引入两组语言学规则——歇后语规则来识别反讽、违反常识规则,对于特定领域的反讽有一定的意义。孙晓等^[17]提出了一种多特征融合的混合神经网络模型,使用了CNN和LSTM进行深度融合,构建了1 000条反讽和1 000条非反讽的数据集。

2 中文微博反讽的语言特征

2.1 语言特征

在相关工作的基础上结合中文微博自身的特点,本文构建了如下的反讽语言学特征体系。

(1) 搭配规则。反讽是一种特殊的修辞方式,经常会同时出现一些固定的词组,例如,“很好,今天路上堵车,我又迟到了。”在反讽中,“很好……又……”这一固定搭配经常出现。

(2) 带强烈情感的副词。微博中,反讽经常使用带强烈情感的副词来表达作者强烈的负面情绪,例如,“你真的是太聪明了,连这个问题都不会。”作者用“真的是”和“太”来表达强烈的负面情感。

(3) 谐音词。谐音现象在汉语的使用中非常普遍,汉语中有很多词的读音相同或相似,使用者经常使用谐音词表达某种意味^[18],例如,“杯具”是“悲剧”的谐音词,“鸭梨”是“压力”的谐音词,微博作为一种简练随意的网络媒体,很多用户使用谐音词表达反讽的情感倾向。

(4) 动词被动化。中文中某些动词本身没有负面的含义,前面加上“被”的话含义会发生转变。例如,“每年看统计数据觉得自己生活水平很高,结果还不是被平均!”就表达了负面的情感。

(5) 特定的标点符号。主要指引号、感叹号和问号。微博作者故意对某些词语加上引号来暗示读者另外一层意思,同样,用户经常使用感叹号和问号来表达强烈的情感倾向。例如,春晚的现场观众真的是全国“低笑点大赛”的佼佼者。

(6) 网络词汇。微博作为网络新兴媒体,含有大量的网络用语,有些网络用语本身就带有强烈的反讽意味,例如,“醉了”“仆街”“废柴”等。

(7) 特定的语气词。“呵呵”“哈哈”“嘿嘿”这三个语气词带有一种嘲讽的意味,在反讽的微博中经常出现。

2.2 语言特征选取方法

(1) 搭配规则

选取反讽微博中经常发现的两个词的搭配,并对所有的搭配进行 χ^2 统计,最后选择同现统计值较高的词语搭配作为反讽的搭配规则,选取的部分解析计算结果如表1所示,本文总结了20个常用的搭配规则。

(2) 词汇特征

在传统的分类方法中, χ^2 统计量进行特征选择是比较高效的方法。对于强烈的情感副词、中文谐音词和网络词汇,均采用人工收集的办法。计算分词后所有不重复词的 χ^2 值,人工从中分别收集前20个情感词、谐音词和网络词汇,结果如下:

① 强烈情感词:极好、太、极其、非常、真的是、最、极度、绝、完全、无比、牛、异常、之至、分外、何等、很、满、多么、格外、实在。

② 谐音词:河蟹、杯具、灰常、内牛满面、餐具、

木有、神马、亚历山大、有木有、笑屎了、无盐、鸭梨、围脖、尼玛、鸡冻、童鞋、稀饭、表、虾米、肿么了。

表1 搭配规则统计值

编号	规则	评分
1	可以……一点	2.530
2	可以……再	2.456
3	再……一点	2.783
4	很好……但是	3.146
5	很好……又	4.257

③ 网络词汇:涨姿势、醉了、五毛、打酱油、冏、仆街、上天台、拍砖、咔嚓、逗比、靠、东东、废柴、小白、晕、蒜你狠、抓狂、亮骚、心塞、逼格。

3 反讽识别模型的构建

3.1 词嵌入向量(Word Embedding)

将自然语言文本通过词向量进行向量表示,可以为特定任务下的算法进行预处理。One-hot是一种最基本的表示词向量的方法,然而,对于微博这种短文本数据,利用这种表示方法构造的词向量往往比较稀疏,而且难以获取词汇之间的上下文信息。由于One-hot表示方法存在缺陷,Hinton^[19]率先提出了Word Embedding方法,将高维数据降为低维数据,在解决向量稀疏问题的同时通过低维数据的位置关系来反映上下文信息。Mikolov等^[20]提出了Skip-gram模型来训练特定领域Word Embedding词向量,Skip-gram模型通过当前词汇的词向量来预测上下文词向量,假设某一词组序列为 $s_1, s_2, s_3, \dots, s_N$,实验的目标是式(1)的值最小化。

$$E = \frac{1}{N} \sum_{n=1}^N \sum_{-a \leq i \leq a, i \neq 0} -\log p(s_{n+i} | s_n) \quad (1)$$

其中, a 是以当前词语为中心的窗口大小,表示选取当前词 s_n 的前 a 个词和后 a 个词。 $p(s_{n+i} | s_n)$ 表示在词 s_n 出现条件下词 s_{n+i} 出现的概率。基本的Skip-gram模型计算条件概率,如式(2)所示。

$$p(s_{n+i} | s_n) = \frac{\exp(v_{s_{n+i}}^T v_{s_n})}{\sum_{t=1}^{|V|} \exp(v_t^T v_{s_n})} \quad (2)$$

其中, v_t 和 v'_t 分别是词 s_t 的输入和输出向量, $|V|$ 是词典的大小。

在实验中,只要训练足够大的语料库,选择恰当的窗口,Skip-gram就能训练出高质量的词向量。

3.2 结合反讽特征和 Word Embedding 的双输入 CNN 模型

为了将反讽本身的特征和深度学习进行融合, 本文结合反讽特征和 Word Embedding 构建了双输入卷积神经网络模型。该模型引入反讽特征, 用于

对句子进行语义拓展, 然后使用词向量建立特征矩阵和句子矩阵的双输入矩阵, 经过卷积和池化操作自动提取文本的局部特征向量, 在连接融合层进行串接融合, 最后将融合后的特征向量输入 softmax 分类器中, 以实现对句子的反讽识别。模型结构如图 1 所示。

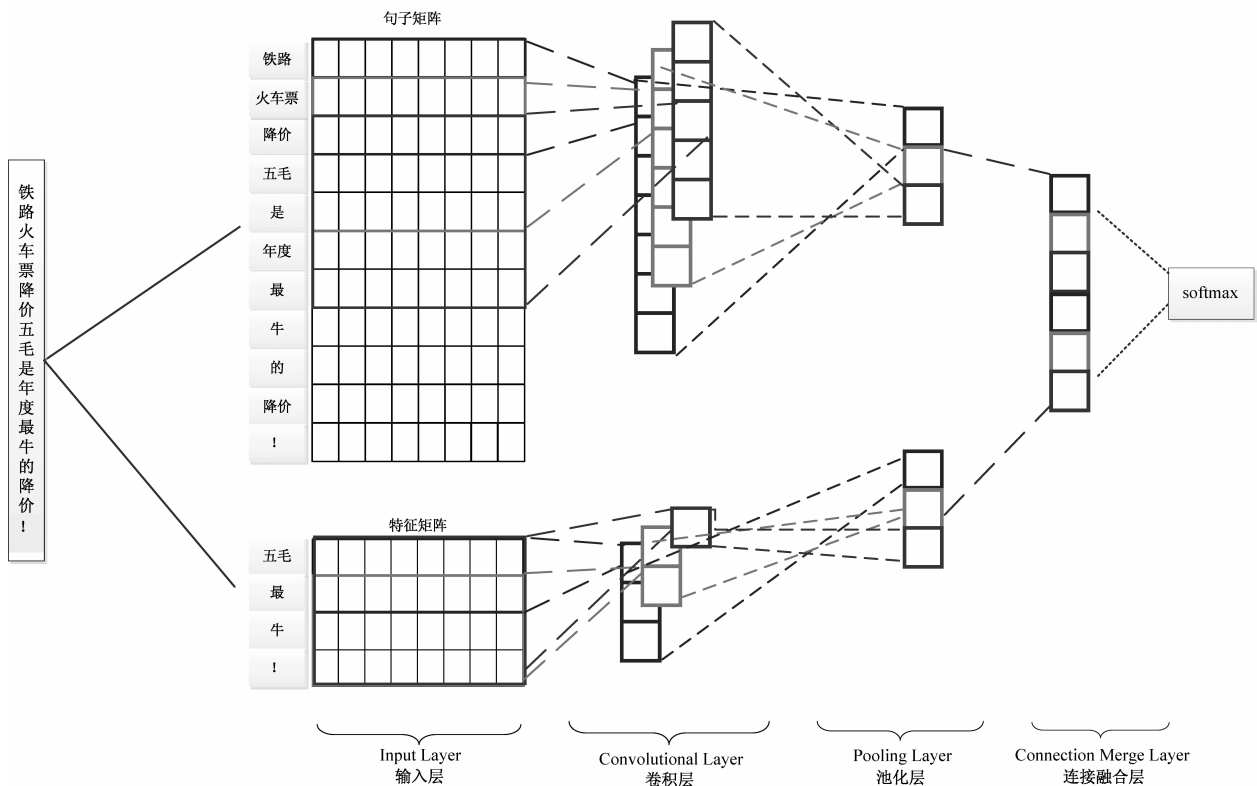


图 1 融合反讽特征的双输入 CNN 模型结构图

(1) 输入层

本文面向的对象是微博中的句子, 因此, 首先通过微博中的句号将微博文本以句子为单元存储到数据集中, 然后将句子表示成二维数据矩阵形式, 作为模型的输入。微博中的每个词经过 Word Embedding 训练之后转化为 v 维向量。假设数据集中最长的句子包含 n 个词, 由 n 个 v 维向量组成的 $n \times v$ 的矩阵作为卷积神经网络的输入矩阵, 将其余的句子用 0 补齐, 变成 $n \times v$ 的矩阵。

首先通过分词工具将数据集中的每个句子 S 进行分词。将 S 作为分词工具的输入, 输出的 S 被分为 n 个词语 $s_1, s_2, s_3, \dots, s_n$, 将 S 表示为如式(3)所示。

$$S: \{s_1, s_2, s_3, \dots, s_n\} \quad (3)$$

从训练好的词向量中获取 S 的每个词语 s_i 的词表达, 将 s_i 表示为 $R^{1 \times v}$ 空间中的向量, 如式(4)所示。

$$s_i = (s_{i1}, s_{i2}, \dots, s_{iv}) \quad i \in [1, n] \quad (4)$$

S 的矩阵表示是将 S 的词向量按原来的顺序排

列, 如式(5)所示。

$$S = s_1 \oplus s_2 \oplus \dots \oplus s_n \quad (5)$$

因此句子 S 被转换为向量矩阵, 如式(6)所示。

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1v} \\ s_{21} & s_{22} & \dots & s_{2v} \\ \vdots & \vdots & \vdots & \vdots \\ s_{n1} & s_{n2} & \dots & s_{nv} \end{bmatrix} \quad (6)$$

将向量矩阵作为 CNN 的输入, 经过卷积和池化操作后获得抽象的句子表示。

同理, 反讽的特征是句子中的符号和词集, 因此, 能够获得特征的矩阵表示 T 作为模型的另一个输入, l 为句子中出现的特征词个数。

$$T = \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1v} \\ t_{21} & t_{22} & \dots & t_{2v} \\ \vdots & \vdots & \vdots & \vdots \\ t_{l1} & t_{l2} & \dots & t_{lv} \end{bmatrix} \quad (7)$$

以反讽句子为例：“铁路火车票降价五毛是年度最牛的降价!”经过分词后结果为“铁路/火车票/降价/五毛/是/年度/最/牛/的/降价/!”。句子经过分词,共有 11 个词向量,具体表示如式(8)所示。

$$\mathbf{S} = \begin{bmatrix} s_{11} & s_{12} & \cdots & s_{1v} \\ s_{21} & s_{22} & \cdots & s_{2v} \\ \vdots & \vdots & \vdots & \vdots \\ s_{11v} & s_{112} & \cdots & s_{11v} \end{bmatrix} \quad (8)$$

同样因为包含“五毛/最/牛/!”等四个特征,“五毛”是网络用词,“最/牛”在情感词表中。“!”也作为特征,因此,得到特征的矩阵表示 \mathbf{T} ,如式(9)所示。

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1v} \\ t_{21} & t_{22} & \cdots & t_{2v} \\ \vdots & \vdots & \vdots & \vdots \\ t_{41} & t_{42} & \cdots & t_{4v} \end{bmatrix} \quad (9)$$

(2) 卷积层

在图像处理的卷积层操作中,像素矩阵的行列两个方向移动卷积核都有意义,但是将 CNN 应用到微博的反讽分类时,以 \mathbf{S} 矩阵作为输入,在矩阵的行方向移动卷积核窗口没有意义,由此卷积核和词具有相同的维度。同时由于 \mathbf{S} 矩阵的列序是原先句子的次序,卷积窗口在矩阵的列方向的移动能够获取到词汇的上下文信息。因此,可以利用 CNN 实现句子特征自动提取。

在卷积层中对每个不同大小的窗口都设置了 m 个卷积核 $C_1, C_2, C_3, \dots, C_m$,不同大小的卷积核能够获取到不同的信息,增加了信息的全面性,其中任意卷积核 $\mathbf{C} \in R^{k \times v}$ 的形式如式(10)所示。

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1v} \\ c_{21} & c_{22} & \cdots & c_{2v} \\ \vdots & \vdots & \vdots & \vdots \\ c_{k1} & c_{k2} & \cdots & c_{kv} \end{bmatrix} \quad (10)$$

其中, k 是窗口的大小, v 代表词向量的维度。

卷积的过程是将 \mathbf{S} 矩阵从上到下分为 $n-k+1$ 个子窗口矩阵 $\mathbf{S}' \in R^{k \times v}$,分别与 \mathbf{C} 进行 $(*)$ 运算, $(*)$ 定义如式(11)所示。

$$\mathbf{S}' * \mathbf{C} = f\left(\sum_{i \in (1, k), j \in (1, v)} (s_{ij} \times c_{ij} + b_{ij})\right) \quad (11)$$

其中, b_{ij} 是偏置, f 是激励函数,这里采用 RELU 函数来进行归一化,式中 \mathbf{S}' , \mathbf{C} 都是 $k \times v$ 的矩阵, $s_{ij} \times c_{ij} + b_{ij}$ 即矩阵 \mathbf{S}' 和矩阵 \mathbf{C} 对应元素相乘再加上偏置。因此, \mathbf{S} 矩阵和任意一个卷积核进行了 $n-k+1$ 次运算,按顺序排列得到卷积向量 $\mathbf{Q} \in R^{(n-k+1) \times 1}$ 。同样,将 \mathbf{S} 矩阵与所有的卷积核进行

卷积,得到所有的卷积层输出 $\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3, \dots, \mathbf{Q}_m$,传入池化层中。其中 \mathbf{Q}_i 表示如式(12)所示。

$$\mathbf{Q}_i = (q_{i1}, q_{i2}, q_{i3}, \dots, q_{i(n-k+1)}) \quad i \in (1, m) \quad (12)$$

(3) 池化层

\mathbf{S} 矩阵经 m 个卷积核进行卷积操作后,获得了 m 个 $R^{(n-k+1) \times 1}$ 空间的向量。池化层将获得的向量进行简化,定义的池化操作如下:即选取卷积层输出 $\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3, \dots, \mathbf{Q}_m$ 中每个向量中的最大值,如式(13)所示。

$$p_i = \text{pooling}(\mathbf{Q}_i)$$

$$= \max(q_{i1}, q_{i2}, q_{i3}, \dots, q_{i(n-k+1)}) \quad i \in (1, m) \quad (13)$$

(4) 连接融合层

连接融合层就是将 m 个池化后得到的语义特征向量进行融合得到 \mathbf{M} 矩阵的表示向量,从而得到一个 $R^{m \times 1}$ 的空间的表示向量 \mathbf{P} ,如式(14)所示。

$$\mathbf{P} = (p_1, p_2, p_3, \dots, p_m) \quad (14)$$

同理,能得到 \mathbf{T} 矩阵的表示向量 \mathbf{P}' ,如式(15)所示。

$$\mathbf{P}' = (p'_1, p'_2, p'_3, \dots, p'_m) \quad (15)$$

将 \mathbf{P} 和 \mathbf{P}' 进行拼接得到向量 \mathbf{P}'' ,如式(16)所示。

$$\mathbf{P}'' = \text{mix}(\mathbf{P}, \mathbf{P}')$$

$$= (p_1, p_2, p_3, \dots, p_m, p'_1, p'_2, p'_3, \dots, p'_m) \quad (16)$$

最后将 \mathbf{P}'' 输入到 softmax 函数中进行分类。

4 实验设置以及结果分析

4.1 数据采集

目前,中文反讽的文章较少,其数据也未公开,因此需要人工标注。本文在电影、汽车、奥运会、春晚等领域收集了 20 万条微博数据,人工标注了其中 15 000 条微博。由于有些微博中包含多个句子,因此共标注了 39 822 条带有情感的句子,其中包含反讽的句子有 2 398 个,占带情感总句子数的 6.02%,说明在微博中反讽句子的分析是不可忽视的。

反讽语料的句子标注过程:将反讽看作二分类问题,若为反讽标注标签为 1,否则标注标签为 0。另外,在标注情感的同时直接标注是否为反讽。但标注情感标签时,不区分句子是否包含显式情感词,完全通过人工判定,将其情感类别标注为消极、积极或者中性。为了保证标注的质量,减少人工判定的偏差,在标注过程中,采用交叉检验。存在观点不一致时进行讨论,统一认识。为了防止实验数据集类别倾斜,影响分类的性能,需要将数据的类别平衡化,从非反讽数据集中任意抽取 2 398 条句子,使得

中文反讽数据集最终由反讽和非反讽句子各为 2 398 条构成。部分反讽语料的链接如下: <https://pan.baidu.com/s/1ueThtdKHvtNCdhR3v0OVBQ>。

4.2 模型参数

数据集中最长的微博包含 132 个词, Word Embedding 设为 300 维, 即 $n=132, v=300$ 。实验采用五次交叉验证, 即从数据集中随机抽取 1/5 作为测试集, 剩下的 4/5 作为训练集。实验中模型卷积窗口大小为 3, 4, 5, 每种窗口 128 个, 学习率 0.001, 正则化系数 λ 为 10^{-8} , 训练迭代次数为 50, dropout 率为 0.5。

4.3 实验结果及分析

为了验证本文方法的有效性, 对中文数据集设置了三组实验。

(1) 语言特征有效性验证

本实验是对比词袋模型(BOW)与结合语言特征的词袋模型(BOW+TZ), 分类器采用支持向量机(SVM)、朴素贝叶斯(NB)和随机森林(RF)三种方法, 实验设置如表 2 所示, 实验结果如表 3 所示。

表 2 第一组实验设置说明

方法	方法说明
BOW	单独使用词袋模型
BOW+TZ	将人工抽取的反讽特征与词袋模型结合。按照 χ^2 统计量大小选取了前 3 000 维, 特征权重采用 TF-IDF 方法

表 3 第一组实验结果比较

方法	分类器	准确率	召回率	F 值
BOW	SVM	0.6531	0.4860	0.5574
	NB	0.6703	0.4391	0.5306
	RF	0.6856	0.4065	0.5104
BOW+TZ	SVM	0.7106	0.6269	0.6661
	NB	0.7057	0.5052	0.5889
	RF	0.7107	0.4871	0.5780

由表 3 可以看出, 结合了反讽特征的词袋模型(BOW+TZ)的反讽识别准确率和召回率优于单独使用词袋模型(BOW)的机器学习方法, 而且召回率提升较为明显。说明仅通过传统的词袋模型不能反映句子中上下文的语义信息。另外, 微博中反讽表达具有复杂性, 单纯通过词语难以识别反讽, 而选取

反讽的语言特征可以有效地缓解这一问题。为了具体分析其性能, 以五次交叉验证的一次实验结果为例。词袋模型与结合语言特征的词袋模型在分类器 SVM 下的混淆矩阵对比, 如表 4 所示。

表 4 在 SVM 分类器下混淆矩阵对比

BOW+SVM		预测		BOW+TZ+SVM		预测	
		1	0			1	0
实际	1	233	247	实际	1	301	179
	0	124	356		0	123	357

从表 4 的实验结果可以看出, 加入反讽语言特征后的 SVM(BOW+TZ+SVM)能够多识别出 68 条反讽句, 说明了人工选取的语言特征是有效的。例如可以识别出“感谢铁道部, 居然足足省了五毛!”。

(2) Word Embedding 词向量有效性验证

本实验是将整个句子进行向量化表示, 对比随机初始化词向量(R-Word2Vec-S+CNN)和使用特定领域数据预训练词向量(S-Word2Vec-S+CNN)的性能。其中, 预训练词向量是在 20 万条微博数据的语料上, 采用 3.1 节介绍的词嵌入方法所得。实验的分类器都是在 CNN 的深度学习框架下进行, 实验结果如表 5 所示。

表 5 第二组实验结果比较

方法	准确率	召回率	F 值
R-Word2Vec-S+CNN	0.735 4	0.704 4	0.719 6
S-Word2Vec-S+CNN	0.810 4	0.726 4	0.766 1

由表 5 可以看出, 通过微博特定领域预训练词向量能够捕捉词语间的关联关系, 较好地刻画微博中词语分布, 使得实验性能有较大幅度的提升。同时与表 3 中最好的传统机器学习方法(BOW+TZ+SVM)相比, 性能也有较大的提升, 说明深度学习能够通过低层特征组合形成更加抽象的高层特征, 用于挖掘出句子的深层语义信息, 进而学习到有用的隐式特征, 提高了分类的准确性。因此, 后续第三组实验中的词向量均采用本节的预训练方式。

(3) 融合语言特征的卷积神经网络模型的有效性验证

为了验证融合语言特征的卷积神经网络模型的有效性, 本实验与仅输入语言特征的卷积神经网络进行对比, 且与文献[12]采用的 CRF 模型也进行了比较。实验设置如表 6 所示, 实验结果如表 7 所示。

表 6 第三组实验设置说明

方法	方法说明
S-Word2Vec-S+CRF	将反讽的自动识别问题转化为序列标注问题,采用 IBO2 的标注集合来标记功能块,每个标记由两部分组成,第一部分表示功能块的位置,第二部分表示功能块的类型,两部分之间通过“-”连接,实验中采用的是 CRF++
S-Word2Vec-S+CNN	仅将整个句子的预训练词向量输入 CNN 中
S-Word2Vec-TZ+CNN	仅将预训练的特征词向量输入 CNN 中
S-Word2Vec-S+S-Word2Vec-TZ+CNN	融合预训练特征词向量和句子词向量,输入 CNN 中

表 7 第三组实验结果比较

模型	准确率	召回率	F 值
S-Word2Vec-S+CRF	0.8029	0.6075	0.6917
S-Word2Vec-S+CNN	0.8104	0.7264	0.7661
S-Word2Vec-TZ+CNN	0.5322	0.4333	0.4778
S-Word2Vec-S+S-Word2Vec-TZ+CNN	0.8208	0.8166	0.8187

实验结果表明,仅采用特征词向量(S-Word2Vec-TZ+CNN)的结果最差,说明反讽的结构比较复杂,单独使用反讽特征构成的特征矩阵难以刻画句子序列的深层语义信息,进而无法体现深度学习的优势。融合语言特征的卷积神经网络(S-Word2Vec-S+S-Word2Vec-TZ+CNN)实验结果最好,说明句子的矩阵表示 S 经过池化产生的表示向量 p 和特征的矩阵表示 T 经过池化产生的表示向量 p ,在连接融合层进行拼接后,能够增大某种隐式反讽特征的权重,同时结合 CNN 能够获取深层语义的优点,从而提高分类的准确率。以五次交叉验证中的一次实验结果为例,对比是否融合语言特征的卷积神经网络实验结构的混淆矩阵如表 8 所示。

表 8 融合语言特征的卷积神经网络混淆矩阵对比

S-Word2Vec-S+CNN		预测		S-Word2Vec-S+S-Word2Vec-TZ+CNN		预测	
		1	0			1	0
实际	1	348	132	实际	1	392	88
	0	82	398		0	86	394

由表 8 可以看出,利用 S-Word2Vec-S+CNN 比表 4 中的采用传统机器学习方法(BOW+SVM)多识别出 115 条反讽,同时非反讽也多识别出 42

条,说明神经网络能够自动地获取有效特征的反讽,例如,“这些年,有冯巩在,我对春晚能否无聊一直比较放心”。融合语言特征的卷积神经网络(S-Word2Vec-S+S-Word2Vec-TZ+CNN)比没有融合语言特征的卷积神经网络(S-Word2Vec-S+CNN)多识别出 44 条反讽,说明语言特征是有效的,例如,“矮油你是摄影记者里文字最好的,文字里面摄影最好的”。

4.4 错误分析

我们从实验结果发现,虽然本文反讽识别的性能还是不错的,但也存在一类需要常识背景才能识别的反讽句,例如:“在春晚现场热烈的气氛下,这位大姐终于睡着了。”在正常情况下,在“春晚现场热烈的气氛下”是不可能睡着的,这个句子和常识相悖,是反讽句,但是我们选取的语言特征仅仅是句子中的一些词、短语和标点符号,加上训练的数据集较小,深度学习没有学到相应的特征,导致这类反讽句识别错误。同时另一类需要微博上下文语境才能识别的反讽句。例如:“希望春晚能够看到这个节目,我要上人大……”。这句话利用本文的方法识别为非反讽,其原因是该微博的上文为:“新闻说,人民大学在自主招生中勇于突破,大胆捞钱,11 岁的富二代,7 岁开奥迪,家庭有背景,人大就让娃娃读了本科!”如果将该微博的上文和该微博一起作为测试句,就能识别出该句应该为反讽。另外,实验结果还发现某些人工选取的特征会对非反讽的识别造成干扰,例如:“刷微博,看春晚,微博神吐槽让春晚精彩不断!”本身应为非反讽句,在融合反讽的语言特征后(S-Word2Vec-S+S-Word2Vec-TZ+CNN)被识别为反讽句,从而降低了非反讽识别的精度。

5 结束语

本文研究了微博这一特定领域中反讽的识别,简要介绍了当前中英文反讽的研究现状,根据中文特点总结了七种反讽的语言特征,结合卷积神经网络的优势提出了融合语言特征的卷积神经网络的反讽识别方法。实验结果表明,深度学习框架能够大幅度地提升反讽识别的精度,融合语言特征后实验效果能获得进一步的提升。然而,本文还存在着一些不足,对于需要常识背景的反讽句以及需要微博上下文语境的反讽句实验的效果都不理想,因此这两方面是我们下一步努力的方向,同时构建更丰富

的反讽语料库也是我们下一步重点研究的工作。

参考文献

- [1] 涂靖. 反讽的语用特征和限制条件[J]. 外语学刊, 2002 (1): 77-81.
- [2] 刘正光. 反语理论综述[J]. 解放军外国语学院学报, 2002(4): 16-21.
- [3] Xiang Li. Irony illustrated: A cross-culture exploration of situational irony in China and the United States [M]. New York: Sino-platonic Papers, 2008.
- [4] Gonzalez-Ibanez R, Muresan S, Wacholder N. Identifying sarcasm in Twitter: A closer look[C]//Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, 2011: 581-586.
- [5] Reyes A, Posso P, Veale T. A multidimensional approach for detecting irony in Twitter[J]. Language Resources & Evaluation, 2013, 47(10): 239-268.
- [6] Konstantin Buschmeier, Philipp Cimiano, Roman Klinger. An impact analysis of features in a classification approach to irony detection in product reviews [C]//Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2014: 42-49.
- [7] Edwin Lu-nando, Ayu Purwarianti. Indonesian social media sentiment analysis with sarcasm detection[C]//Proceedings of the Advanced Computer Science and International Systems (ICACSIS), 2013 International Conference on IEEE, 2013: 195-198.
- [8] David Bamman, Noah A Smith. Contextualized sarcasm detection on Twitter[C]//Proceedings of the Ninth International AAAI Conference on Web and Social Media, 2015: 574-577.
- [9] Aditya Joshi, Pushpak Bhattacharyya, Mark James Carman. Automatic sarcasm detection: A survey[J]. ACM Computing Surveys (CSUR), 2017, 50(5): 73-95.
- [10] Aniruddha Ghosh, Tony Veale. Fracking sarcasm using neural network[C]//Proceedings of NAACL-HLT, 2016: 161-169.
- [11] Yu-Hsiang Huang, Hen-Hsen Huang, Hsin-Hsi Chen. Irony detection with attentive recurrent neural networks [C]//Proceedings of the ECIR, 2017: 534-540.
- [12] Abhijit Mishra, Kuntal Dey, Pushpak Bhattacharyya. Learning cognitive features from gaze data for sentiment and sarcasm classification using convolutional neural network[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017: 377-387.
- [13] Lotem Peled, Roi Reichart. Sarcasm SIGN: Interpreting sarcasm with sentiment based monolingual machine translation[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017: 1690-1700.
- [14] Tang Y J, Chen H. Chinese irony corpus construction and ironic corpus construction and ironic structure analysis [C]//Proceedings of the 25th International Conference on Computational Linguistics, 2014: 1269-1278.
- [15] 邓钊, 贾修一, 陈家骏. 面向微博的中文反语识别研究[J]. 计算机工程与科学, 2015, 37(12): 2312-2317.
- [16] 罗婷. 社交网络评论中的反语识别研究[D]. 昆明: 云南财经大学硕士学位论文, 2017.
- [17] 孙晓, 何家劲, 任福继. 基于多特征融合的混合神经网络模型讽刺语用判别[J]. 中文信息学报, 2016, 30 (6): 215-233.
- [18] 石铁兰. 汉语谐音及其文化内涵研究[J]. 赤峰学院学报(汉文哲学社会科学版), 2014, 35(5): 138-140.
- [19] Hinton G E. Learning distributed representations of concepts[C]//Proceedings of the 8th Annual Conference of the Cognitive Science Society, 1986: 1-12.
- [20] Mikolov T, Sutskever I, Chen Kai, et al. Distributed representations of words and phrases and their compositionality[EB/OL]. [2013-10-16]. <http://arxiv.org/pdf/1310.4546>.



卢欣(1990—), 硕士研究生, 主要研究领域为自然语言处理。
E-mail: 731971961@qq.com



王素格(1964—), 博士, 教授, 主要研究领域为自然语言处理与情感分析。
E-mail: wsg@sxu.edu.cn



李旻(1988—), 博士研究生, 主要研究领域为情感分析。
E-mail: 770624917@qq.com