

基于 E-CNN 神经网络的情绪原因识别方法

慕永利¹, 李旻¹, 王素格^{1,2}

(1 山西大学计算机与信息技术学院, 山西太原 030006

2 山西大学计算机智能与中文信息处理教育部重点实验室, 山西太原 030006)

摘要: 文本情绪原因识别作为一个新型的研究方向在文本情绪分析领域占据重要地位。本文结合卷积神经网络, 提出了一种基于集成卷积神经网络的情绪原因识别方法。该方法通过词向量、卷积、池化等操作充分融合了句子的语义信息, 利用多个 CNN 集成降低数据不平衡性对情绪原因识别的影响, 克服了传统情绪原因识别方法的繁琐规则制定、特征抽取、特征空间降维等过程。实验结果表明, 本文的方法在情绪原因识别方面取得了较好的效果, 对于情绪归因的方法研究具有一定的指导作用。

关键字: 情绪原因识别; E-CNN; 卷积; 池化

中图分类号: TP391

文献标识码: A

Emotion Cause Detection Method Based on Ensemble Neural Network

Mu Yongli¹, Li Yang¹, Wang Suge^{1,2}

(1 School of Computer & Information Technology, Shanxi University, Taiyuan, Shanxi 030006, China;

2 Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education Shanxi University, Taiyuan, Shanxi 030006, China)

Abstract: Emotion cause detection of text, as a new and difficult research direction, plays an important role in the field of text emotion analysis. In this paper, combining with a convolution neural network for modeling sentences, this paper puts forward a new emotion cause detection method based on ensemble convolution neural network. This method fully combines the semantic information by operations such as word embedding, convolution, pooling, and it uses multiple CNN integrations to reduce the impact of data imbalance on emotional reasons detection, and it gets rid of the tedious processes, such as rule-making, feature extraction, feature dimension reduction and so on. The experimental results show the proposed method has a good effect, and it also has a directive function in terms of emotion cause detection.

Key words: Emotion Cause Detection; E-CNN; Convolution; Pooling

1 引言

文本情绪分析是自然语言处理领域非常重要的研究方向之一, 它主要研究文本所蕴含的情绪及与情绪相关的深层信息。目前, 文本情绪分析大都研究聚焦在情绪分类方法, 王蓉等人^[1]采用 bag-of-words 表示文本, 通过联合学习对文本的情感和情绪进行分类。Shoushan Li 等人^[2]同样采用 bag-of-words 表示文本, 然后利用依赖因子图模型预测句子级别的情绪类型。伴随着深度学习技术的成熟, 基于深度学习的情绪分类方法也呈现出比传统机器学习方法更为有效的结果。张志华^[3]提出基于卷积神经网络对长文本进行情感语义建模, 解决了长文本(句子)的情感分类问题。Yaqi Wang 等人^[4]利用卷积神经网络对句子建模, 实现了微博文本数据的多标签情绪分类。对于情绪分析问题, 若忽略了文本情绪相关的深层次信息的挖掘,

收稿日期:

定稿日期:

基金项目: 国家自然科学基金资助项目(61632011, 61573231, 61672331);

作者简介: 慕永利 (1990—), 男, 硕士研究生, 主要研究方向为情感分析; 李旻 (1988—), 女, 博士研究生, 主要研究方向为文本挖掘; 王素格 (1964—), 女, 教授, 主要研究方向为自然语言处理、情感分析。

将影响其分析的效果。例如，2016年的罗尔诈捐门事件，公众情绪从最初的同情演变为后面的愤怒情绪，我们关注的重点可能不是公众情绪是什么，而是公众为什么会从同情演变为愤怒的情绪，由此可见，事件中的情绪原因的识别是非常重要的和必需的。所谓情绪原因识别就是针对文本中出现的被描述者的情绪，抽取出触发被描述者情绪产生的原因信息。

目前，情绪归因一般研究人员大都采用基于规则和统计的方法^[10-16]。随着深度学习在自然语言处理领域中的广泛应用，人们开始利用神经网络模型，通过对句子进行建模用于处理文本分类任务。XinChi Chen^[5]采用 GRNN(Gated Recursive Neural Network)、Nal Kalchbrenner^[6]和 Yoon Kim^[7]通过不同卷积和池化的 CNN 神经网络对句子建模，最后他们都在 SST^[8]、TREC^[9]数据集上分别进行情感分类和问题分类，最后实验验证了采用 CNN 的句子建模可以获得最好的分类性能。由此可见，神经网络不仅可以应用到文本情感分类任务(SST^[8]数据集上的情感分类)，还可以应用到文本问答类型的分类任务(TREC^[9]数据集上的问题分类)。

本文针对文本情绪原因识别任务，提出了一种基于卷积神经网络的集成(E-CNN 模型)的情绪原因识别方法，首先利用卷积神经网络对句子进行建模，在词向量的训练表示时，融合上下文语义信息，然后通过多个 CNN 的集成，用于文本情绪原因的识别。实验结果表明，本文提出的 E-CNN(Ensemble Convolution Neural Network)方法，在情绪原因识别方面获得了较好的实验性能。

2 相关工作

针对情绪原因的识别，Ying Chen^[10]等人通过对标注语料的分析，发现超过 80%的情绪原因信息位于出现情绪的核心子句的前后两个子句中，Sophia Yat Mei Lee^[5]同样根据对标注的语料分析，获得情绪原因信息大多位于出现情绪的核心句的上下文信息中。为了获得这些情绪原因，研究人员主要采用基于规则和基于统计的方法。

基于规则的情绪原因识别方法方面，Ying Chen^[10]和 Sophia Yat Mei Lee^[11-12]等人建立了一个情绪归因的语料库，并且根据标注的语料库建立了相应的规则，用于情绪原因句子的识别。Kai Gao^[13]等人提出一种基于规则的情绪原因识别模型，然后对相应的微博数据进行情绪原因的抽取，然而，制定的规则并不能完全覆盖所有的语言现象，而且同一个子句可能同时匹配多个规则，很容易造成规则冲突。再者不同领域语料的语言结构有一定的区别，针对特定领域的文本制定的规则并不能很好的适用于其它领域的文本，需要耗费巨大的人力物力重新分析语言结构用于添加新的规则。Alena Neviarouskaya^[14]通过句法、语法和规则相结合的方法，分析“乐”的 8 种情绪原因的语言现象，以此来推测一段文本的情绪类别和情绪原因。Weiyuan Li^[15]等人通过抽取情绪原因特征进行情绪分类，其特征抽取采用基于规则的方法。

在基于统计方法的情绪原因识别方面，袁丽^[16]提取语言学线索词的特征、句子距离特征、候选词词法特征等信息，然后得到特征向量空间，最后应用 SVM 分类器和条件随机场对文本情绪归因进行了判别。Lin Gui 等人^[17]既通过建立 25 条规则来进行文本情绪原因的预测，还运用分类方法来预测文本情绪原因。李逸薇^[18]将情绪归因识别任务看成序列标注，并建立了相应的模型，将所有候选原因子句当成一个序列，从而标记出哪些属于原因子句，她特别指出在利用序列标注模型进行情绪原因识别的过程中，上下文特征尤为重要。Ghazi D 等人^[19]利用 FrameNet 建立了情绪和情绪原因相关联的数据集，然后利用 CRF 来预测文本句子级别的情绪。Gaoyan Ou 等人^[20]建立文本公众情绪和情绪事件之间关系，利用文本情绪来进行文本中事件的预测。Lin Gui 等人^[21]首先构建了一个情绪归因语料库，标注了表达情绪的核心子句，在核心句的前后子句中标注情绪原因子句，然后从核心句的前后每个子句中抽取候选原因事件，通过训练分类器，最后判定抽取的候选原因事件是否是情绪核心子句

的原因事件。

综上所述，无论是基于规则的情绪原因识别方法，还是基于统计的情绪原因识别方法，二者都表明情绪原因的相关信息一般都位于文本中情绪核心句的上下文信息中，针对这一重要的语言现象，本文提出了一种基于 E-CNN 模型的情绪原因识别方法，该方法有别于目前的情绪原因识别方法，而是利用了词向量和卷积神经网络，同时融合了句子的语义信息进行情绪原因识别。

3 E-CNN 情绪原因识别方法

3.1 情绪原因问题描述

为了分析情绪和情绪原因在句子中出现的情况，我们将情绪出现的文本子句称为情绪核心子句，被描述者情绪原因信息所在的子句称为情绪原因子句，这里所描述的子句可以是逗号、问号、感叹号、句号为结尾的句子，子句示例信息见表 1。通过在数据集中的统计，获得情绪子句在数据集中的统计信息见表 2 所示。

表 1 情绪原因识别示例表

子句位置	子句信息	标签
S^{-3}	夜里到达商城县上石桥镇派出所	0
S^{-2}	王从贤老人已早早地来到派出所等候	0
S^{-1}	一见女儿安然无恙地下车	1
S^0	王从贤老泪纵横	0
S^{+1}	拉着雷成林的手久久不松	0
S^{+2}	雷成林随后不要任何回报	0
S^{+3}	连夜返回金寨	0

表 2 数据集相关统计信息示例表

相关信息	百分比
情绪原因子句占比	18.366%
非情绪原因子句占比	81.634%
情绪原因信息位于核心子句上下文 7 个子句中的占比	98.06%
情绪原因位于其它位置的占比	1.94%
包含一个原因信息子句的占比	98.06%
包含多个原因信息子句的占比	1.94%

其中，表 1 中的 S^0 表示情绪核心子句所在位置， S^{-3} 、 S^{-2} 、 S^{-1} 表示紧邻情绪核心子句前的三个子句， S^{+1} 、 S^{+2} 、 S^{+3} 表示紧邻情绪核心子句的后三个子句。标签信息为 1 表示为情绪原因子句，标签信息为 0 表示非情绪原因子句。

(1) 从表 1 可知，示例中的上下文句子中情绪原因子句仅有 1 个，非情绪原因子句有 6 个，就单个示例来看，情绪原因信息在整个示例的上下文信息所占比例较小，导致在对情绪核心子句上下文中的子句进行情绪原因识别时，可以利用的信息也较少，从而使非情绪原因信息所占比重较大。因此，在整个上下文信息中进行情绪原因信息的识别存在非平衡问题。

(2) 由表 2 可知，整个数据集中情绪原因子句约占整个句子的 18.366%，原因信息位

于包含情绪核心子句在内的上下文 7 个子句中约占 98.06%，原因信息位于其它位置约占 1.94%；包含一个原因子句信息的示例约占 98.06%，包含多个原因子句信息的示例占 1.94%。通过我们的考察以及对 Lin Gui 标注语料的统计，发现几乎所有的情绪原因信息都位于紧邻情绪核心子句的上下文信息中。这与文献[10]得到的结论相同。因此，从紧邻情绪核心子句的上下文中识别情绪原因信息将成为情绪原因识别问题的关键。

3.2 情绪原因识别模型整体框架

由 3.1 节对情绪原因识别问题的分析，可将情绪原因识别任务看成一个分类问题，主要判别一个核心子句的上下文子句是否包含情绪原因信息。传统的分类方法普遍采用词袋模型对句子向量表示，再采用 SVM 等传统机器学习方法进行分类。其不足是不能很好地充分融合句子中词与词之间的语义信息。另外，由于非情绪原因信息所占比重较大，存在情绪原因信息与非情绪原因信息之间的比例存在非平衡问题。因此，本文结合 CNN 卷积神经网络提出了一种集成的卷积神经网络（E-CNN）情绪归因模型，对情绪原因的进行分类识别。模型整体框架如图 1 所示。

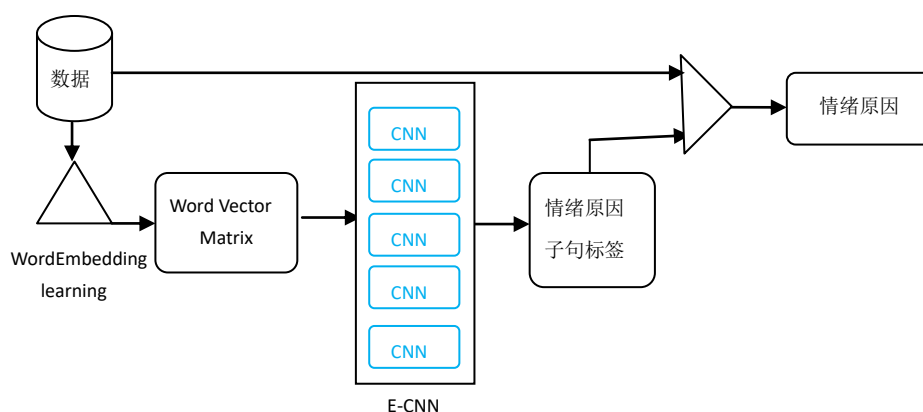


图 1 E-CNN 情绪原因识别框架图

如表 1 所示，对于一组文本，通过 E-CNN 模型得到每个子句的一个概率值，数据集中多原因子句的示例占比很小，因此，将概率最大的一个候选原因子句作为每组示例的情绪原因子句。

3.3 E-CNN 模型

对于情绪核心子句的上下文子句进行分类用于识别情绪原因子句时，由于原因子句与非原因子句所占比例的不平衡性，我们将每个示例的原因子句和非原因子句分别构成多个较平衡的数据集，分别在各个新数据集上进行相对应的 CNN 卷积神经网络的训练；然后将训练好的多个 CNN 集成组合构成 E-CNN 模型；最后在 E-CNN 上进行情绪原因子句的分类用于识别包含情绪原因信息的子句。

基于 E-CNN 进行情绪原因识别的过程如图 2 所示。

由图 2 可知，E-CNN 是由多个训练好的 CNN 集成。

在 E-CNN 模型中，（1）将句子中的词语利用 Word2Vec 工具进行词向量表示，然后通过词向量矩阵化操作将句子中的词向量组合成为一个词向量矩阵；（2）通过 CNN 模型中的卷积操作融合词与词之间的语义信息；（3）再通过 CNN 模型中 Max-pooling 池化操作、连接操作和 attention 操作将候选原因子句表示为一个向量；（4）通过 sigmoid 函数和集成操作得到候选原因子句的概率值，在核心子句的上下文的 7 个候选子句中，概率最大的子句即为情绪原因子句。

本文建立的 CNN 卷积神经网络模型如下图 3 所示, CNN 包含输入层, 卷积层, 池化层, Attention 层, 连接层。

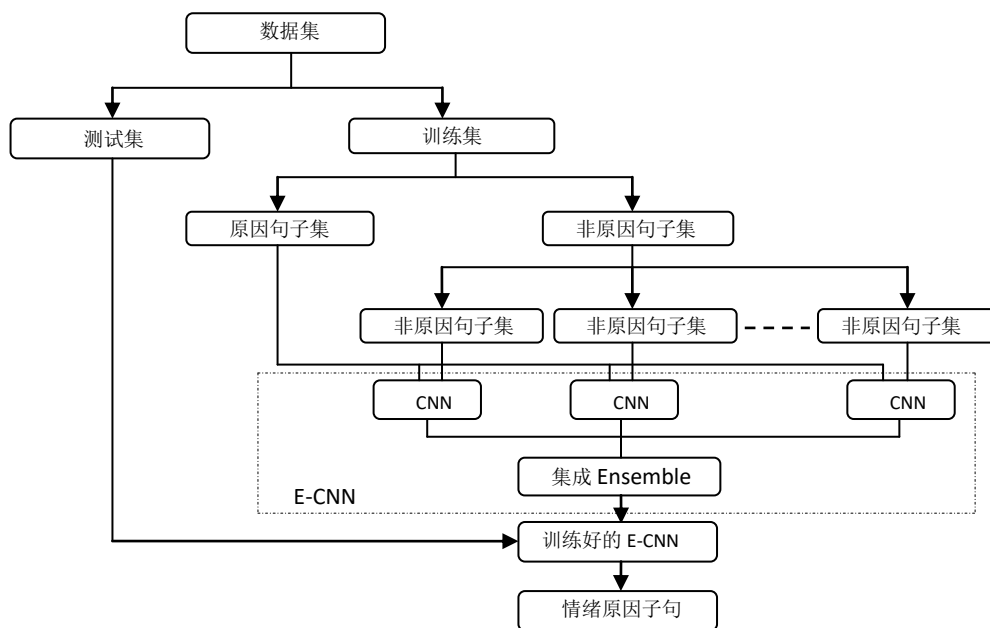


图 2 E-CNN 情绪原因识别过程图

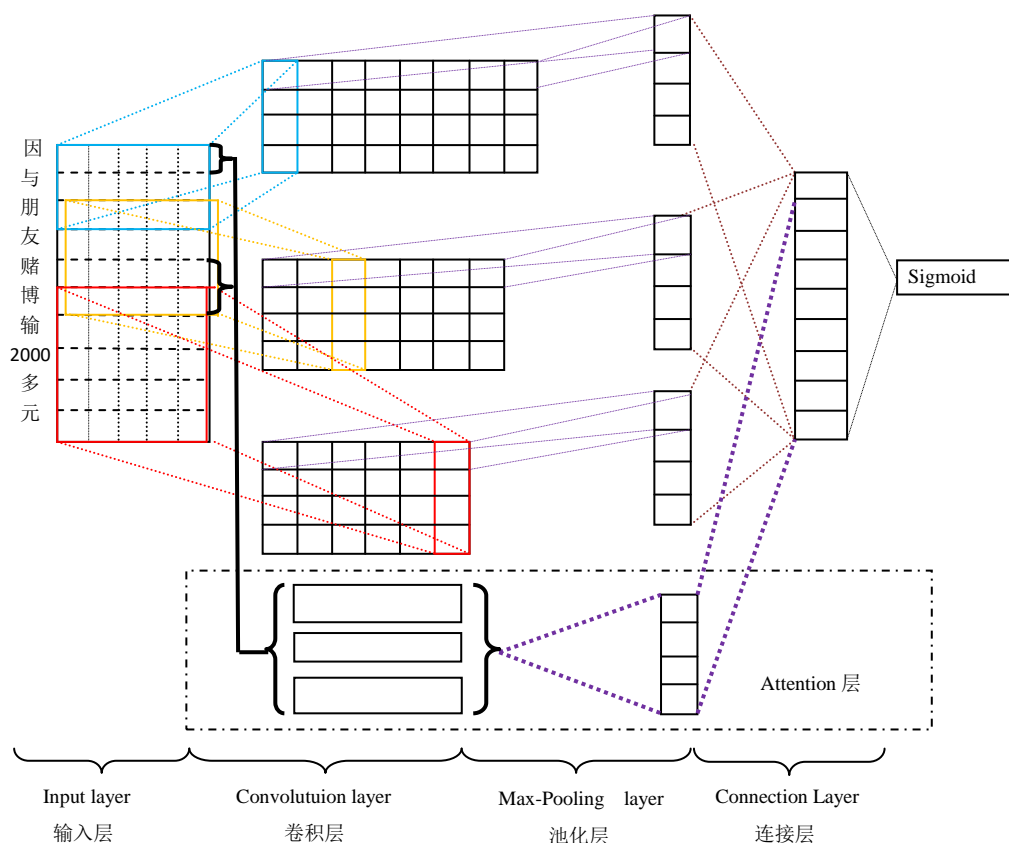


图 3 CNN 模型图

(1) **输入层:** 该层主要是初始化输入数据, 通过 Word2Vec 将句子中的词语表示成为词向量, $x_i \in R^k$ 表示 k 维词向量, X 表示 $n \times k$ 维的词向量矩阵。采用 \oplus 运算操作将词向

量连接表示为一个词向量矩阵 X ，见公式（1）所示。

$$X = x_1 \oplus x_2 \oplus x_3 \mathbf{L} \mathbf{L} \oplus x_i \mathbf{L} \mathbf{L} \oplus x_n, \quad x_i \in R^k \quad (1)$$

这里的 x_i 是一维行向量， x_i^T 是 x_i 的转置， \oplus 表示连接操作， \oplus 的具体数学运算含义如下示例所示。

$$Z = z_1 \oplus z_2 = [1 \quad 2] \oplus [3 \quad 4] = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad z_i \text{ 表示行向量}$$

$$Z = z_1^T \oplus z_2^T = \begin{bmatrix} 1 \\ 2 \end{bmatrix} \oplus \begin{bmatrix} 3 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}, \quad z_i^T \text{ 表示列向量}$$

(2) 卷积层：该层采用不同大小的卷积窗口进行卷积操作，将句子中的局部语义信息融合。通过卷积窗口选取 c 个句子中的局部词向量首尾相连， g_j 被表示为一个具有局部语义信息的新向量，见公式（2）所示。 $G_{(n-c+1) \times (c*k)}$ 表示将 g_j 通过连接运算操作表示为一个具有句子全局语义信息的新矩阵，见公式（3）所示。 $M_{d \times (n-c+1)}$ 表示 $G_{(n-c+1) \times (c*k)}$ 通过 σ 函数表示为最终的卷积层矩阵，见公式（4）所示。

$$g_j = (x_j, x_{j+1} \mathbf{L}, x_{j+c-1}), 1 \leq j \leq n-c+1 \quad (2)$$

$$G_{(n-c+1) \times (c*k)} = g_1 \oplus g_j \oplus \mathbf{L} \oplus g_{n-c+1} \quad (3)$$

$$M_{d \times (n-c+1)} = \sigma \left(W_{d \times (c*k)} \cdot G_{(n-c+1) \times (c*k)}^T + B_{d \times (n-c+1)} \right), \sigma = \frac{e^{-b}}{1 + e^{-b}} \quad (4)$$

σ 表示 *sigmoid* 函数， c 表示卷积窗口大小， $W_{d \times (c*k)}$ 是卷积核系数矩阵， $B_{d \times (n-c+1)}$ 是卷积核偏置矩阵。

(3) 池化层：该层通过 Max-pooling 操作^[6]从一个卷积层矩阵中选取代表句子全局语义信息中的有效特征。 Q 表示池化后代表句子语义信息的池化层向量。 $M_{d \times (n-c+1)}$ 表示卷积层矩阵，见公式（5）和公式（6）所示。

$$M_{d \times (n-c+1)} = \begin{bmatrix} \mathbf{r} \\ \mathbf{m}_1 \\ \mathbf{M} \\ \mathbf{r} \\ \mathbf{m}_k \\ \mathbf{M} \\ \mathbf{r} \\ \mathbf{m}_d \end{bmatrix}, d \in R \quad (5)$$

$$\mathbf{r} Q = \begin{bmatrix} \max(\mathbf{r} \mathbf{m}_1) \\ \mathbf{M} \\ \max(\mathbf{r} \mathbf{m}_k) \\ \mathbf{M} \\ \max(\mathbf{r} \mathbf{m}_d) \end{bmatrix} \quad (6)$$

这里的 $\max(\mathbf{r} \mathbf{m}_k)$ 表示从向量 \mathbf{m}_k 的所有元素中取一个最大值元素。

(5) attention 层：根据对所有情绪原因子句的句子结构的分析可以得出，超过 90% 的原因子句含有动词和动词短语，而且很多情绪原因子句还含有代表原因信息的连词，如“由于”、“因为”等连词。针对这些情绪原因子句特殊的语言结构，我们通过在 CNN 中添加 attention 层来增强情绪原因子句语言现象对情绪原因子句识别的影响。在模型中通过句法分

析和词性标注选取动词和原因连词。如下图 4 的例子所示：

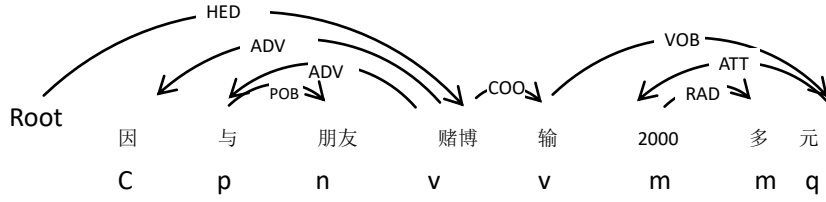


图 4 句法分析示例图

如图 4 所示，如果一个句子中至少含有一个动词，那么该句子的核心词(HED 表示核心关系)就是动词。通过词性标注可知，图 4 中的“因”(词性标注中的 c 表示连词)为连词。综上所述，从候选原因子句中选取核心词(通过 HED 关系选取句子的核心词)和原因连词(词性标注中的 c 表示连词，如果没有连词则不选)，选取核心词和连词构成新的词向量矩阵 E ，然后通过全连接操作得到一个 attention 向量，见公式 (7) 所示。

$$A_{1 \times k} = W_{1 \times a} \cdot E_{a \times k} \quad (7)$$

如图 3 中的 attention 层所示， a 为被选中的词的总个数， k 表示词向量的维度。

(6) 连接层：在这一层，将经过不同卷积窗口大小卷积操作和池化操作后的所有向量表示为一个代表句子语义信息的向量。将池化后所有代表句子语义信息通过连接层操作表示为一个最终代表句子全局语义信息的向量 \vec{S} ，然后通过 σ 函数得出一个概率值 P ，见公式 (8) 和 (9) 所示。 P 表示通过一个 CNN 模型得到的候选原因子句的概率值。

$$\vec{S} = W' \cdot A^T + \sum_{i=1}^s (W_i \cdot \vec{Q}_i + \vec{b}_i), s \in R \quad (8)$$

$$P = \sigma(V \cdot \vec{S} + b) \quad (9)$$

\vec{Q}_i 是通过第 i 个卷积层操作和池化层操作得到相对应的向量。 W' 表示 $m \times k$ 的系数矩阵。 s 表示卷积窗口的个数， W_i 表示 $m \times d$ 维的系数矩阵。 V 表示 $1 \times m$ 维的系数矩阵。 \vec{b}_i 表示偏置。

(7) CNN 集成(E-CNN)：候选原因子句通过每个 CNN 都可以得到一个概率值。将所有 CNN 得到概率值相加求和取平均值，最终得到表示候选原因子句是否为情绪原因子句的概率值 \mathbf{P} ，见公式 (10) 所示。

$$\mathbf{P} = \frac{1}{h} \sum_{i=1}^h P_i \quad (10)$$

这里 h 表示 CNN 的个数。 \mathbf{P} 表示当前子句是不是情绪原因子句的概率。

4 实验与分析

4.1 实验数据与评价

本文提出的 E-CNN 模型是在 Lin Gui^[21]标注的情绪原因数据集(11799 个子句)上进行实验的。实验评估方式和 Lin Gui^[21]在其数据集上实验的评估方式一样，采用精确率、召回率、F 值。实验过程中，我们随机将数据集的 90% 作为训练集，10% 作为测试集，反复实验 10

次，求取各个衡量指标的平均值。采用 Wrod2Vec 进行句子词向量的训练表示。训练过程采用反向传播方法进行参数调整。损失函数采用交叉熵损失函数， y_i 表示标注数据中的原因子句 i 的标签值， \hat{y}_i 表示通过 E-CNN 模型预测的原因子句的概率值， L_2 表示将 L_2 范式作为一个惩罚项。

$$loss(\text{损失函数}) = \frac{1}{n} \sum y_i \ln \hat{y}_i + (1 - y_i) \ln (1 - \hat{y}_i) + \frac{\lambda}{2} L_2 \quad (11)$$

4.2 参数设置

通过在 4.1 节数据集上进行了大量实验，并进行了参数调整，获得的超参数设置如下，词向量维数 k 设为 50，CNN 的个数 h 设为 5 个，卷积窗口 s 的个数为 3 个，3 个卷积窗口 c 的大小分别为设为 3、4、5，卷积核的大小 d 设为 300，学习率 α 设为 0.01，批大小 $batchsize$ 设为 50， λ 设为 10^{-8} 。

为了验证在一个 CNN 模型的迭代次数对情绪原因发现的影响，本文随机选取原因子句和非原因子句为 1:1 的数据集（原因子句 2167，非原因子句 2167）上进行迭代实验。实验结果如图 5 所示。

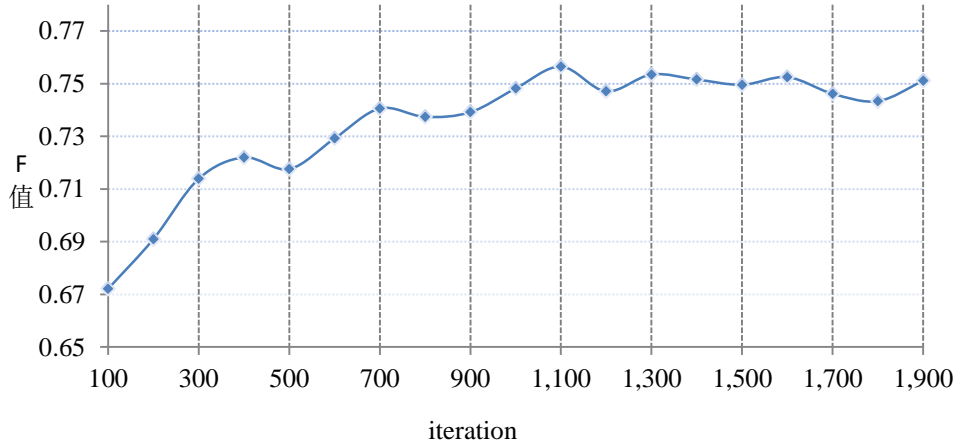


图 5 测试集上的 F 值随迭代次数变化图

由图 5 可知，随着 iteration 的增大，在测试集上的情绪原因发现的 F 值也逐渐增大。当 iteration 超过 100 时，F 值就快速达到 0.6721，说明在前 100 词迭代过程中，模型快速收敛靠近最优解。当 iteration 超过 1100 以后，情绪原因发现的 F 值基本保持不变甚至略有下降，说明当 iteration 达到 1100 时，模型基本达到了稳定状态，iteration 超过 1100 以后，可能出现了过拟合的现象。因此，为了使模型达到情绪原因识别的最优状态，也为了防止过拟合的出现，在实验过程中我们将 iteration 设为 1100。

4.3 实验结果与分析

为了验证本文所提方法的有效性，设置了方法间的比较实验和不同评价方法的实验。

(1) 方法间的对比实验。

选用了如下的方法进行对比实验。

①基于规则

RB(Rule based method): 基于规则的方法^[11];

CB(Commonsense based method): 基于常识的方法^{[22][23]};

ML(Rule base features for machine learning): 利用基于规则的方法进行特征抽取，然后

利用机器学习方法进行分类^[24];

②基于 SVM 分类器:

$K_{ET-O}^{[21]}$ 、 $K_{ET-M}^{[21]}$: 均是利用词袋模型进行句子中候选原因事件的向量表示, 然后利用树核 SVM^[21]和修改后的树核 SVM 进行分类^[21], 如果子句中包好了原因事件, 则该子句为原因子句;

$K_{new+O}^{[21]}$ 、 $K_{new*O}^{[21]}$ 、 $K_{new+M}^{[21]}$ 、 $K_{new*M}^{[21]}$: 均是利用词袋模型进行句子中候选原因事件的向量表示, 然后利用多种不用核函数结合的 SVM 进行分类^[21], 如果子句中包好了原因事件, 则该子句为原因子句;

③基于深度学习

CNN: 采用单个 CNN 在数据集上进行实验; E-CNN: 采用 E-CNN 在数据集上进行实验。最后 CNN 和 E-CNN 均选择选取概率最大候选原因子句作为该组示例的情绪原因子句;

上述方法数据集与参数均在第 4.1 和 4.2 节进行了介绍。规则、SVM 和 CNN 三大类方法间的比较实验结果见表 3 所示。

表 3 各种方法的实验结果比较

Method		Precision	Recall	F-measure
规则	RB	0.6747	0.4287	0.5243
	CB	0.2672	0.7130	0.3887
	RB+CB	0.5435	0.5307	0.5370
	RB+CB+ML	0.5921	0.5307	0.5597
SVM	K_{ET-O}	0.3982	0.4134	0.4057
	K_{ET-M}	0.4583	0.4745	0.4662
	K_{new+O}	0.6446	0.6779	0.6608
	K_{new*O}	0.6492	0.6701	0.6595
	K_{new+M}	0.6588	0.6927	0.6752
	K_{new*M}	0.6673	0.6841	0.6756
CNN	CNN	0.3453	0.3576	0.3513
	E-CNN	0.5714	0.5437	0.5569

由上表 3 可知:

①与基于规则的情绪归因方法比较,E-CNN 在准确率、召回率和 F 值上均高于 RB+CB, 准确率、召回率、F 值分别达到 0.5714、0.5437、0.5569。同时 E-CNN 的召回率高于 RB+CB+ML。由上述可知, E-CNN 比上述基于规则的情绪原因识别方法效果更好。

②在基于 SVM 分类器的情绪归因方法中, 本文和 Lin Gui^[18]提出的基于核函数的 SVM 情绪原因子句分类方法进行比较, E-CNN 方法在精确率、召回率、F 值都明显高于 K_{ET-M} 。而 K_{new*M} 中由于加入了 NER 以及 Parsering 结果, 所以 E-CNN 方法不及 K_{new*M} 。同时 K_{new*M} 在候选原因子句抽取原因事件时, 将会丢失不包含情绪原因事件的句子, 主要原因在事件抽取时需要每个子句必须包含动词或者动词短语, 而有些原因子句中并不一定包含动词或动词短语。另一方面, 虽然 E-CNN 实验效果低于 K_{new*M} 方法, 但是 E-CNN 方法可以识别不包含情绪原因事件的子句, 以及各种类型语言结构的情绪原因子句。

③在和单个 CNN 方法比较中, E-CNN 方法的精确率、召回率、F 值都明显高于本文单个 CNN 方法。主要原因使用 E-CNN 方法, 可以分别将每组示例(表 1 所示)中的各个非

情绪原因子句和同一个情绪原因子句组成新的数据集，分别训练 CNN 会充分融合情绪原因子句和各个子句的句间关系。因此，E-CNN 的实验效果更好。

(2) 三种评价方法

由于情绪原因识别问题本身存在不平衡问题，有时抽取概率最大的一个候选原因子句并不能很好的帮助研究人员进行情绪原因识别问题的研究，因此，在 E-CNN 模型进行情绪原因子句的识别后，对每组示例预测的情绪原因子句进行按概率从大到小排序，分别选取概率较大的一个和多个子句同时作为情绪原因子句。这里选取概率最大的一个候选原因子句作为该组示例的情绪原因子句位 E-CNN，选取概率最大的两个候选原因子句作为原因子句，且只要两个子句至少包含一个原因子句即视为正确 E-CNN-2，选取概率最大的三个候选原因子句作为原因子句，且只要三个子句中至少包含一个原因子句即视为正确；三种方式的实验结果如下表 4 所示

表 4 E-CNN 实验结果对比表

Method	Precision	Recall	F-measure
E-CNN	0.5714	0.5437	0.5569
E-CNN-2	0.8252	0.7586	0.7904
E-CNN-3	0.9593	0.9367	0.9479

由表 4 可知，利用 **E-CNN-2** 识别两个候选原因子句中至少包含一个原因子句的精确率达到 0.8252，召回率达到 0.7586。在 **E-CNN-3** 识别三个候选原因子句中至少包含一个原因子句的精确率高达 0.9593，召回率高达 0.9367。由此可知，E-CNN 模型可以帮助情绪原因识别研究方法进行情绪原因的前期识别过滤，同时，也可以用于有关情绪原因识别的问答类的问题的候选子句的选取。

5 结束语

情绪原因识别方法需要对上下文信息进行很好的融合和处理，而 CNN 卷积神经网络可以在句子建模过程中充分考虑句子中所有的词义信息的融合，并且还可以从所有语义信息中选取了有效的全局语义信息。另一方面，情绪原因子句与非情绪原因子句在实际语料中存在极不平衡情况，基于两种因素，本文提出的 E-CNN 模型，充分利用卷积神经网络方法的融合句子中词与词之间上的语义信息和非平衡数据。为了增强因果句子的显式特征，在 E-CNN 模型中还加入了 attention 机制，以增强对情绪原因子句的识别。

在 E-CNN 模型中，虽然 E-CNN-2 和 E-CNN-3 两种实验的结果最佳的，但 E-CNN 的识别结果不及 Knew*M，下一步，需要将事件的主体信息考虑到模型中，同时引入句法信息和命名体信息。另一方面，需要给出一个更为合理的度量，用于判断候选情绪原因子句中真正的情绪原因子句。

参考文献

- [1]Wang R, Li S, Zhou G, et al. Joint sentiment and emotion classification with integer linear programming[C]. In Proceedings of International Conference on Database Systems for Advanced Applications: Springer International Publishing, 2015, 259-265.
- [2]Shoushan Li, Lei Huang, Rong Wang. et al. Sentence-level emotion classification with label and context dependence[C]. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language

- Processing, 2015, 1045–1053.
- [3]张志华.基于深度学习的情感词向量及文本情感分析的研究[D].华东师范大学, 2016.
- [4]Wang Y, Feng S, Wang D, et al. Multi-label Chinese microblog emotion classification via convolutional neural network[C]. In Proceedings of Asia-Pacific Web Conference: Springer International Publishing, 2016,567-580.
- [5]Xinchi Chen, XipengQiu, ChenxiZhu,et al. Sentence modeling with gated recursive neural network[C]. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015,793–798.
- [6]Nal Kalchbrenner, Edward Grefenstette, Phil Blunsom. A convolutional neural network for modeling sentences[C]. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, 655–665.
- [7]Yoon Kim. Convolutional neural networks for sentence classification[C]. In Proceedings of Conference on Empirical Methods in Natural Language Processing, 2014,1746-1751.
- [8]Richard Socher, Alex Perelygin, Jean Y. Wu, et al. Recursive deep models for semantic compositionality over a sentiment treebank[C]. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013, 1631–1642.
- [9]Xin Li, Dan Roth. Learning question classifiers[C]. In Proceedings of the International Conference on Computational Linguistics, 2002,12(24):556—562.
- [10]Ying Chen, Sophia Yat Mei Lee, Shoushan Li, et al. Emotion cause detection with linguistic constructions[C].In Proceedings of the 23rd International Conference on Computational Linguistics,2010, 179–187.
- [11]Sophia Yat Mei Lee, Ying Chen, et al. A text-driven rule-based system for emotion cause detection[C]. In Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Tex,2010, 45–53.
- [12]Lee S Y M, Chen Ying, Li Shoushan, et al. Emotion cause events: corpus construction and analysis[C]. In Proceedings of the International Conference on Language Resources and Evaluation, Computational Intelligence, 2010, 1121–1128.
- [13]Kai Gao, HuaXu, Jiushuo Wang. A rule-based approach to emotion cause detection for chinese micro-blogs[J]. Expert Systems with Applications,2015, 42(9):4517-4528.
- [14]Neviarouskaya A, Aono M. Extracting causes of emotions from text[C]. In Proceedings of International Joint Conference on Natural Language Processing, 2013, 932–936.
- [15]Weiyuan Li, Hua Xu. Text-based emotion classification using emotion cause extraction[J]. Expert Systems with Applications, 2014, 41(4):1742–1749.
- [16]袁丽. 基于文本的情绪自动归因方法研究[D]. 哈尔滨工业大学, 2014.
- [17]Lin Gui, Li Yuan, RuifengXu, Bin Liu,et al. Emotion cause detection with linguistic construction in chinese weibo text[J]. Communications in Computer & Information Science, 2014, 496(s 3–4):457-464.
- [18]李逸薇,李寿山,黄居仁,高伟.基于序列标注模型的情绪原因识别方法[J].中文信息学报,2013, 27(5):93–99.
- [19]Ghazi D, Inkpen D, Szpakowicz S. Detecting emotion stimuli in emotion-bearing

- Sentences[M]. Computational Linguistics and Intelligent Text Processing: Springer International Publishing, 2015, 152-165.
- [20]GaoyanOu, Wei Chen, Tengjiao Wang, et al. Exploiting community emotion for microblog event detection[C]. In Proceedings of Conference on Empirical Methods in Natural Language Processing, 2014, 1159-1168.
- [21]Lin Gui, Dongyin Wu, RuifengXu,et al. Event-driven emotion cause extraction with corpus construction[C].In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, 1639–1649.
- [22]Irene Russo, TommasoCaselli, Francesco Rubino,et al. EMO-cause: an easy-adaptable approach to emotion cause contexts[C]. In Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis, 2011, 153–160.
- [23]RuifengXu, ChengtianZou, YanzhenZheng,et al. A new emotion dictionary based on the distinguish of emotion expression and emotion cognition[J]. Journal of Chinese Information Processing, 2013,27(6): 82–90.
- [24]Y. Chen, S. Y. M. Lee, S. Li, et al. Emotion cause detection with linguistic constructions[C]. In Proceedings of International Conference on Computational Linguistics, 2010, 179-187.

通信作者：慕永利

联系电话：18734894348

地址：山西省太原市小店区坞城路92号山西大学计算机与信息技术学院

邮编：030006

E-mail: 1203962063@qq.com

作者简介：

慕永利(1990-)，男，硕士研究生，主要研究方向为情感分析。

E-mail: 1203962063@qq.com

王素格(1964-)，女，博士，教授，主要研究方向为自然语言处理。

E-mail: wsg@sxu.edu.cn

李晞(1990-)，女，博士研究生，主要研究方向为文本挖掘。

E-mail: 770624917@qq.com