计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2019.20170917
56(4):854-865,2019

基于带权词格的循环神经网络句子语义表示建模

张祥文[1,2] 陆紫耀[1] 杨 静[1] 林 倩[1] 卢 宇[1] 王鸿吉[1] 苏劲松[1,2]
[1](厦门大学 福建厦门 361000)
[2](江苏省计算机信息处理技术重点实验室(苏州大学) 江苏苏州 215006)
(xwzhang@stu.xmu.edu.cn)

Weighted Lattice Based Recurrent Neural Networks for Sentence Semantic Representation Modeling

Zhang Xiangwen[1,2], Lu Ziyao[1], Yang Jing[1], Lin Qian[1], Lu Yu[1], Wang Hongji[1], and Su Jinsong[1,2]
[1](*Xiamen University*, *Xiamen*, *Fujian* 361000)
[2](*Jiangsu Provincial Key Laboratory for Computer Information Processing Technology* (*Soochow University*), *Suzhou*, *Jiangsu* 215006)

Abstract Currently, recurrent neural networks (RNNs) have been widely used in semantic representation modeling of text sequences in natural language processing. For those languages without natural word delimiters (e.g., Chinese), RNNs generally take the segmented word sequence as input. However, sub-optimal segmentation granularity and segmentation errors may affect sentence semantic modeling negatively, as well as subsequent natural language processing tasks. To address these issues, the proposed weighted word lattice based RNNs take the weighted word lattice as input and produce current state at each time step by integrating arbitrarily many input vectors and the corresponding previous hidden states. Weighted word lattice expresses a compressed data structure that contains exponential word segmentation results. To a certain extent, the weighted word lattice reflects the consistency of different word segmentation results. Specifically, lattice weights are further exploited as a supervised regularizer to refine weights modeling of the semantic composition operation in this model, leading to better sentence semantic representation learning. Compared with traditional RNNs, the proposed model not only alleviates the negative impact of segmentation errors but also is more expressive and flexible to sentence representation learning. Experimental results on sentiment classification and question classification tasks demonstrate the superiority of the proposed model.

Key words weighted word lattice; recurrent neural network; sentence semantics modeling; sentiment classification; question classification

摘 要 目前,循环神经网络(recurrent neural network, RNN)已经被广泛应用于自然语言处理的文本序列语义表示建模.对于没有词语分隔符的语言,例如中文,该网络以经过分词预处理的词序列作为标准输入.然而,非最优的分词粒度和分词错误会对句子语义表示建模产生负面作用,影响后续自然语言

收稿日期:2017-12-01;修回日期:2018-08-17
基金项目:国家自然科学基金项目(61672440);北京语言大学语言资源高精尖创新中心资助;国家语言文字工作委员会一般项目(YB135-49);中央高校基本科研业务费专项资金项目(ZK1024);苏州大学江苏省计算机信息处理技术重点实验室开放课题(KJS1520)
This work was supported by the National Natural Science Foundation of China (61672440), the Project of Beijing Advanced Innovation Center for Language Resources, the Scientific Research Project of National Language Committee of China (YB135-49), the Fundamental Research Funds for the Central Universities (ZK1024), and the Open Project of Jiangsu Provincial Key Laboratory for Computer Information Processing Technology (Soochow University) (KJS1520).
通信作者:苏劲松(jssu@xmu.edu.cn)

处理任务的进行.针对这些问题,提出基于带权词格的循环神经网络模型.该模型以带权词格作为输入,在每个时刻融合多个输入向量和对应的隐状态,融合生成新的隐状态.带权词格是一种包含指数级别分词结果的压缩数据结构,词格中的边权重在一定程度上体现了不同分词结果的一致性.特别地,利用词格权重作为融合函数中权重建模的监督信息,进一步提升了模型句子语义表示的学习效果.相比于传统循环神经网络,该模型不仅能够缓解分词错误对句子语义建模产生的负面影响,同时使得语义建模具有更强的灵活性.在情感分类和问句分类2个任务上的实验结果证明了该模型的有效性.

关键词 带权词格;循环神经网络;句子语义建模;情感分类;问句分类

中图法分类号 TP391

如何生成高质量的句子语义表示一直是自然语言处理的核心问题之一.由于现实中自然语言句子的数量是无限的,因此,我们训练好的模型往往需要处理从未在训练语料中出现过的句子.对此,传统方法通常以高频词或多元词串为基础来表示句子,然后在此基础上进行各种运算,以获得表示句子语义的向量.然而,这些方法往往需要人工事先定义特征,所以建模效率较为低下.近年来,随着深度学习研究及其应用的快速发展[1],学术界和产业界将目光转向了神经网络,通过构建深度神经网络来学习句子的语义表示[2],以应用到后续的自然语言处理任务中.

在基于深度学习的句子语义表示建模方面,循环神经网络(recurrent neural networks, RNNs)[3]得到了广泛应用.相比于传统的非神经网络模型,RNN能够保存序列的历史信息,因此对长序列文本具有更好的建模能力.特别地,RNN的一些变种,例如LSTM(long short term memory)[4]和GRU(gated recurrent unit)[5],进一步引入门机制(gating mechanism)来控制信息流动,提高捕获序列内部长距离依赖的能力.针对中文等没有天然词语分隔符的语言,神经网络模型有2种实现句子语义建模的方案:第1种方案直接建模字序列.该方法忽略词语的边界信息,不需要分词,而这一信息对于建模字、词之间的组合关系至关重要;第2种方案则先进行分词,然后以词为单位来建模.该方法同样存在缺陷:一方面,分词工具产生的错误分词对句子的结构造成破坏,并通过错误传播的形式对后续的表示建模产生负面影响;另一方面,使用单一的词序列来表示句子,使得文本表示建模缺乏灵活性.因此,对于中文等语言,如何利用RNN来提高句子语义表示建模的质量是一个有待深入研究的重要问题.

针对上述问题,本文提出基于带权词格的循环神经网络模型.词格是一个能够容纳多种分词结果的压缩数据结构,与单一分词结果相比,它具有丰富的表示能力.目前,词格已经广泛地应用于许多自然语言处理任务当中,并取得了很好的效果,例如机器翻译[6]和语音识别[7].通过基于带权词格进行句子语义表示建模,我们期望提出的模型可以减轻分词错误造成的错误传播,同时也能使句子语义表示建模具备更强的灵活性.在本文工作中,我们提出了2种基于带权词格的GRU神经网络模型:1)基于带权词格的浅层融合循环神经网络模型(shallow weighted word lattice RNN, SWWL-RNN),该模型直接对多个分词输入和相应的前隐状态进行融合,再输入到标准的RNN单元生成当前隐状态;2)基于带权词格的深层融合循环神经网络模型(deep weighted word lattice RNN, DWWL-RNN).不同于SWWL-RNN,该模型先根据每个分词输入和相应的前隐状态分别产生各自的当前隐状态,然后再对这些隐状态进行融合,生成最终的当前隐状态.显然,2种模型都以融合函数为核心.因此,针对隐状态的融合函数,本文尝试了4种不同的融合策略:

1) 池化(pooling)融合函数;

2) 门机制融合函数;

3) 基于词格边权重的融合函数;

4) 融入词格边权重的门机制融合函数.

最后,我们在情感分类和问句分类实验上,分析对比了2种模型、4种融合策略的效果.实验结果表明,基于带权词格的RNN模型的性能明显超过传统的RNN变体模型和现有的其他模型.

1 背 景

本节介绍本文工作的基础:带权词格[6]和GRU[5]循环神经网络.

1.1 带权词格

带权词格[6]是一种包含指数级别分词结果的

压缩数据结构. 图 1 所示为 1 个句子根据 3 种不同的分词标准进行分词的结果及相对应的词格结构. 3 种分词标准,分别来自北京大学(Peking University, PKU)、中文树库(Chinese treebank, CTB)和微软研究院(Microsoft Research, MSR)公开的分词语料训练的分词模型. 如图 1(d)所示,给定由 N 个字组成的 1 个序列 $c_{1:N}=c_1c_2\cdots c_N$,带权词格在形式上表现为 1 个带权重的有向图 $G=\langle V,E\rangle$. 这里,V 表示结点的集合,其中结点 $v_i\in V(i=1,2,\cdots,N-1)$ 表示 c_i 和 c_{i+1} 之间的位置. 此外,词格还包含 2 个特殊的结点:1)v_0,该结点在 c_1 之前,表示字序列的开始位置;2)v_N,该结点在 c_N 之后,表示字序列的结束位置. E 表示边的集合,以边 $e_{i:j}$ 为例,它以 v_i 为起点,并指向 v_j,同时覆盖了字序列 $c_{i:j}$,$c_{i:j}$ 对应潜在的一个候选分词. 而 $e_{i:j}$ 对应的权重 $weight_{e_{i:j}}$,则代表 $c_{i:j}$ 被作为候选分词的可能性.

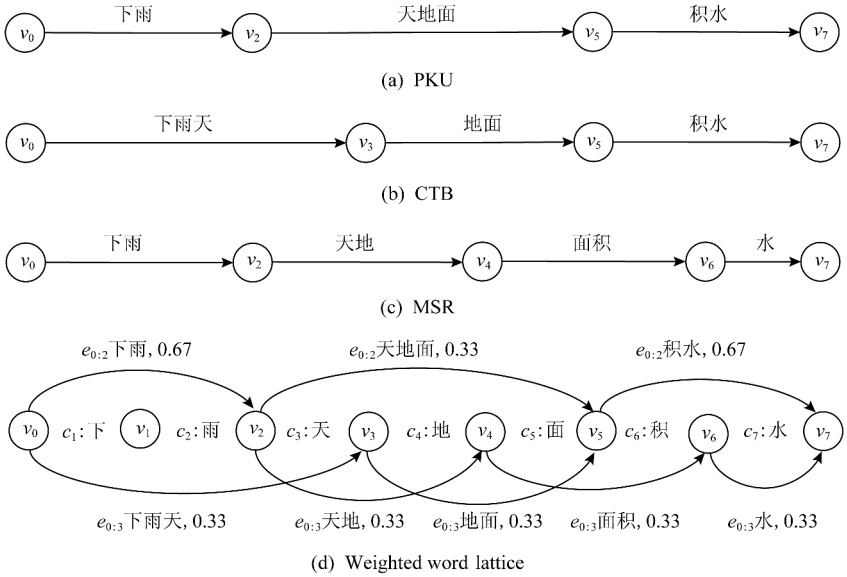

Fig. 1 A weighted word lattice

图 1 一个句子的带权词格

词格中的边权重可以使用前后向算法[8-9]来计算. 具体而言,对于结点 v_i,我们首先递归遍历它左边的前序结点,以迭代方式累加计算出从 v_0 到 v_i 的路径数目 α_{v_i},即:

$$\alpha_{v_i}=\sum_{v_{i_k}}\alpha_{v_{i_k}},\tag{1}$$

其中,v_{i_k} 是结点 v_i 的第 k 个前序结点. 然后,对于结点 v_j,我们递归地遍历它右边的后序结点,同样以迭代累加的方式计算出从 v_N 到 v_j 的路径数目 β_{v_j},即:

$$\beta_{v_j}=\sum_{v_{j_k}}\beta_{v_{j_k}},\tag{2}$$

其中 v_{j_k} 是结点 v_j 的第 k 个后序结点. 最后,$weight_{e_{i:j}}$ 可定义为

$$weight_{e_{i:j}}=\frac{\alpha_{v_i}\times\beta_{v_j}}{\alpha_{v_n}}.\tag{3}$$

如图 1(d)所示,从 v_0 指向 v_3 的边 $e_{0:3}$,覆盖了 c_1 到 c_3 的字序列,表示一个候选词“下雨天”,其权重为 0.33. 边权重在一定程度上体现了不同分词标准的一致性. 权重越大,边覆盖的字序列被切分为词的可能性就越高. 同时,边权重也增强了词格的容错性,使词格结构的信息表示更加丰富,从而得以有效应用于各种自然语言处理任务中.

1.2 GRU 循环神经网络模型

RNN[1]虽然具有较好的文本序列建模能力,但仍然面临着模型参数梯度消失和爆炸的难题[10-12]. 对此,研究者引入了带有门机制的 LSTM[4] 和 GRU[5] 来控制网络信息流动,以提高 RNN 在长序列文本上的建模能力. 由于 GRU 与 LSTM 性能相同,同时所需参数更少. 因此,本文选择 GRU 作为循环神经网络单元进行文本建模. 需要说明的是,本文方法同样适用于 LSTM 等其他 RNN 的变种模型.

如图 2 所示,与 RNN 相同,GRU 在每个输入

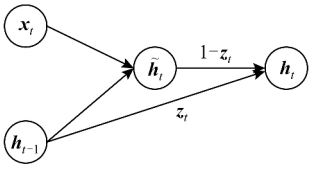

Fig. 2 A GRU unit

图 2 GRU 单元

单元循环地应用 1 个转移函数,以生成当前时刻的隐状态表示.

具体来说,时刻 t 的隐状态向量 $\boldsymbol{h}_t\in\mathrm{R}^d$,由当前输入向量 $\boldsymbol{x}_t\in\mathrm{R}^d$ 和前一时刻的隐状态向量 $\boldsymbol{h}_{t-1}$ 生成:

$$\boldsymbol{h}_t=f(\boldsymbol{x}_t,\boldsymbol{h}_{t-1}),\tag{4}$$

其中,$f(*)$ 通常定义为一个仿射变换及双曲正切函数 tanh. 对于文本序列而言,$\boldsymbol{x}_t$ 是句子中第 t 个词的向量表示,$\boldsymbol{h}_t$ 则代表到时刻 t 为止的词序列向量.

正如本节第 1 段所述,GRU 在 RNN 的基础上,进一步引入了重置门和更新门来控制信息流动. 图 2 所示为一个时刻 t 的 GRU 单元,其转移函数定义为

$$\boldsymbol{r}_t=\sigma(\boldsymbol{W}^{(\mathrm{r})}\boldsymbol{x}_t+\boldsymbol{U}^{(\mathrm{r})}\boldsymbol{h}_{t-1}+\boldsymbol{b}^{(\mathrm{r})}),\tag{5}$$

$$\boldsymbol{z}_t=\sigma(\boldsymbol{W}^{(\mathrm{z})}\boldsymbol{x}_t+\boldsymbol{U}^{(\mathrm{z})}\boldsymbol{h}_{t-1}+\boldsymbol{b}^{(\mathrm{z})}),\tag{6}$$

$$\tilde{\boldsymbol{h}}_t=\sigma(\boldsymbol{W}^{(\mathrm{c})}\boldsymbol{x}_t+\boldsymbol{U}^{(\mathrm{c})}(\boldsymbol{r}_t\odot\boldsymbol{h}_{t-1})+\boldsymbol{b}^{(\mathrm{c})}),\tag{7}$$

$$\boldsymbol{h}_t=\boldsymbol{z}_t\odot\boldsymbol{h}_{t-1}+(1-\boldsymbol{z}_t)\odot\tilde{\boldsymbol{h}}_t,\tag{8}$$

其中,$\boldsymbol{r}_t$ 是重置门,用于控制前状态的信息流动:当它的值接近 0 时,将使得 GRU 单元忽略前状态信息并使用当前输入进行重置,这使得 GRU 单元丢弃被认为在未来无用的信息,从而得到一个更加紧凑的隐状态表示;$\boldsymbol{z}_t$ 是更新门,用于控制在当前状态有多少前状态信息被保留;$\tilde{\boldsymbol{h}}_t$ 是利用重置门对历史信息进行过滤后的候选隐状态向量;σ 是一个逻辑斯蒂函数,$\odot$ 表示逐元素乘法. 式(5)~(7)中 $\boldsymbol{W}$ 和 $\boldsymbol{U}$ 是参数矩阵,用于对输入和隐状态向量进行线性变换;$\boldsymbol{b}$ 是一个偏置项向量;$\boldsymbol{W}$,$\boldsymbol{U}$,$\boldsymbol{b}$ 的上标 r,z,c 分别表示该参数对应的是重置门、更新门、候选隐状态.

2 基于带权词格的 GRU 循环神经网络

受现有工作[6,13-14]的启发,本节对基于词格的循环神经网络[15]进行扩展,提出了基于带权词格的 GRU 循环神经网络,以学习句子语义表示,用于后续的自然语言处理任务. 显然,与词序列相比,带权词格具有更为丰富的信息和更为复杂的网络拓扑结构. 以它为基础来进行神经网络建模将面临着 2 个难题:1)在带权词格中,一个句子通常会存在许多分词结果,这意味着当前单元可能会同时存在多个输入和多个前隐状态,传统循环神经网络[4-5]无法建模这样的结构;2)带权词格的边权重能够较好地区别不同分词结果的可能性. 如何在本文所提出的模型中体现出不同分词结果在句子建模过程中作用的差异,是本文研究工作的一个关键问题.

在建模过程中,我们的模型以句子的字序列为输入,逐字地读取句子. 在时刻 t,对于当前结点 v_t,我们首先确定以字 c_t 为结尾的一个入度边集合,即 $\{e_{t_k,t}=(\boldsymbol{x}_{t_k},\boldsymbol{h}_{t_k})\mid 0\leqslant t_k<t,0\leqslant k<K\}$,这里 K 表示入度边数,即以字 c_t 为结尾的不同候选分词的数量,$\boldsymbol{x}_{t_k}$ 和 $\boldsymbol{h}_{t_k}$ 分别为第 k 个候选分词的词向量表示和相应的前隐状态. 本文提出的 2 种基于带权词格的循环神经网络模型,分别是:基于带权词格的浅层融合 GRU 模型和基于带权词格的深层融合 GRU 模型. 这 2 个模型均以融合函数为核心,分别通过浅层、深层融合产生时刻 t 的隐状态. 针对融合函数,我们将在 2.3 节中详细介绍 4 种不同的融合策略.

2.1 浅层带权词格

浅层带权词格 GRU 模型的单元结构如图 3 所示:

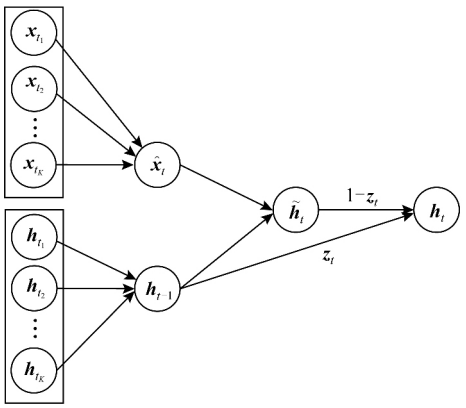

Fig. 3 A SWWL-GRU unit

图 3 浅层带权词格 GRU 单元

针对 GRU[3],我们使用词向量和相应的前隐状态构成的集合 $\{(\boldsymbol{x}_{t_k},\boldsymbol{h}_{t_k})\}$ 来表示词格中结点 v_t 的入度边集合. 接着,分别融合词向量 $\{\boldsymbol{x}_{t_k}\}$ 和前隐状态 $\{\boldsymbol{h}_{t_k}\}$ 的 2 个集合,生成 $\hat{\boldsymbol{x}}_t$ 和 $\hat{\boldsymbol{h}}_{t-1}$,并作为当前时刻唯一的输入词向量和前隐状态,传递给循环单元,生成时刻 t 的隐状态. 特别地,对于 LSTM[2] 等包含额外记忆单元的 RNN 变种,入度边集合则表示为 $\{(\boldsymbol{x}_{t_k},\boldsymbol{h}_{t_k},\boldsymbol{m}_{t_k})\}$,其中 $\boldsymbol{m}_{t_k}$ 表示对应的前一个记忆单元. 因此,通过合并入度边,新的当前隐状态得以容纳多种潜在的候选分词. 不难看出,这一模型主要关注如何对循环单元的输入,也就是 $\{\boldsymbol{x}_{t_k}\}$ 和 $\{\boldsymbol{h}_{t_k}\}$ 分别进行融合,而并不需要修改循环单元内部结构,因此适用于任何基于 RNN 的变种模型[2-3].

形式上,该单元的建模函数定义为

$$\hat{\boldsymbol{x}}_t=g(\boldsymbol{x}_{t_1},\boldsymbol{x}_{t_2},\cdots,\boldsymbol{x}_{t_K}),\tag{9}$$

$$\hat{\boldsymbol{h}}_{t-1}=g(\boldsymbol{h}_{t_1},\boldsymbol{h}_{t_2},\cdots,\boldsymbol{h}_{t_K}),\tag{10}$$

$$\boldsymbol{r}_t=\sigma(\boldsymbol{W}^{(\mathrm{r})}\hat{\boldsymbol{x}}_t+\boldsymbol{U}^{(\mathrm{r})}\hat{\boldsymbol{h}}_{t-1}+\boldsymbol{b}^{(\mathrm{r})}),\tag{11}$$

$$\boldsymbol{z}_t=\sigma(\boldsymbol{W}^{(z)}\hat{\boldsymbol{x}}_t+\boldsymbol{U}^{(z)}\hat{\boldsymbol{h}}_{t-1}+\boldsymbol{b}^{(z)}),\quad(12)$$

$$\tilde{\boldsymbol{h}}_t=\sigma(\boldsymbol{W}^{(c)}\hat{\boldsymbol{x}}_t+\boldsymbol{U}^{(c)}(\boldsymbol{r}_t\odot\hat{\boldsymbol{h}}_{t-1})+\boldsymbol{b}^{(c)}),\quad(13)$$

$$\boldsymbol{h}_t=\boldsymbol{z}_t\odot\hat{\boldsymbol{h}}_{t-1}+(1-\boldsymbol{z}_t)\odot\tilde{\boldsymbol{h}}_t,\quad(14)$$

其中,$\hat{x}_t$ 和 $\hat{h}_{t-1}$ 是经过语义融合操作后得到的输入词向量和前隐状态. 式(9)(10)分别用于融合 K 个输入词向量和前隐状态,而式(11)~(13)则与标准 GRU 完全一致. $g(*)$ 是融合函数,其具体定义在 2.3 节详细介绍.

2.2 深层带权词格

深层带权词格 GRU 循环神经网络的单元结构如图 4 所示:

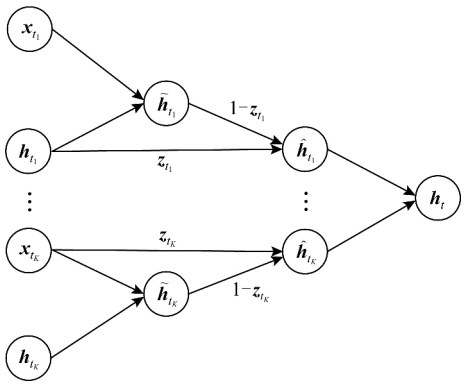

Fig. 4 A DWWL-GRU unit

图 4 深层带权词格 GRU 单元

以入度边集合$\{(\boldsymbol{x}_{t_k},\boldsymbol{h}_{t_k})\}$为基础,我们将每条边$(\boldsymbol{x}_{t_k},\boldsymbol{h}_{t_k})$独立地输入到循环单元中,生成隐状态集合$\{\hat{\boldsymbol{h}}_{t_k}\}$. $\hat{\boldsymbol{h}}_{t_k}$ 表示第 k 种潜在候选分词对应的隐状态,接着融合所有的候选隐状态,得到新的当前隐状态. 与浅层带权词格 GRU 相似,这一模型并不对循环单元结构造成影响,因此仍然适用于任何 RNN 的变种模型[2-3].

与浅层带权词格 GRU 单元相比,深层模型在更细粒度的语义表示层次上,对多种分词结果进行分词状态的融合生成. 浅层模型选择融合循环单元的输入,而深层模型采取对循环单元的输出进行融合的方式. 简单来说,两者的具体区别在于选取融合操作的时机不同. 从时间复杂度来考虑,深层模型的时间复杂度为 $O(KN)$,即关于句子的字数和边的最大个数成正比;而浅层模型的时间复杂度为 $O(N)$,与基于字的普通 RNN 模型相等. 这 2 个模型涉及到融合函数的部分,时间代价可以忽略不计.

形式上,该单元的建模函数定义为

$$\boldsymbol{r}_{t_k}=\sigma(\boldsymbol{W}^{(r)}\boldsymbol{x}_{t_k}+\boldsymbol{U}^{(r)}\boldsymbol{h}_{t_k}+\boldsymbol{b}^{(r)}),\quad(15)$$

$$\boldsymbol{z}_{t_k}=\sigma(\boldsymbol{W}^{(z)}\boldsymbol{x}_{t_k}+\boldsymbol{U}^{(z)}\boldsymbol{h}_{t_k}+\boldsymbol{b}^{(z)}),\quad(16)$$

$$\tilde{\boldsymbol{h}}_{t_k}=\sigma(\boldsymbol{W}^{(c)}\boldsymbol{x}_{t_k}+\boldsymbol{U}^{(c)}(\boldsymbol{r}_{t_k}\odot\boldsymbol{h}_{t_k})+\boldsymbol{b}^{(c)}),\quad(17)$$

$$\hat{\boldsymbol{h}}_{t_k}=\boldsymbol{z}_{t_k}\odot\boldsymbol{h}_{t_k}+(1-\boldsymbol{z}_{t_k})\odot\tilde{\boldsymbol{h}}_{t_k},\quad(18)$$

$$\boldsymbol{h}_t=g(\hat{\boldsymbol{h}}_{t_0},\hat{\boldsymbol{h}}_{t_1},\cdots,\hat{\boldsymbol{h}}_{t_{K-1}}),\quad(19)$$

其中,$\boldsymbol{x}_{t_k}$,$\boldsymbol{h}_{t_k}$ 与 2.1 节公式符号的含义相同. 式(15)~(18)用于生成第 k 个分词对应的隐状态,式(19)采用语义融合函数 $g(*)$ 生成 $\boldsymbol{h}_t$.

2.3 融合函数

在常见的基于字或词的模型中,句子可以被视为一个特殊的有向无环图,其中每个结点的入度和出度均为 1. 然而,对于词格,每个结点的入度和出度则至少为 1,因此基于 RNN 的序列建模模型[4-5]无法处理词格结构的输入数据[6-7].

在浅层、深层带权词格模型的基础上,我们进一步提出了使用融合函数来融合循环单元的输入或输出,生成单一的压缩表示,以转换成标准循环单元能够接受的输入形式. 这里,本文在文献[15]中 2 种融合函数的基础上,进一步提出 2 种基于词格边权重的融合函数. 为了不失一般性,本文以深层带权词格中的 $\boldsymbol{h}_t$ 为例,描述在带权词格 GRU 单元中如何使用这些融合函数. 需要注意的是,这些定义同样适用于生成其他向量,例如 $\boldsymbol{x}_t$.

首先介绍文献[15]中 2 种基础的融合函数:池化融合函数与门机制融合函数;接着,介绍本文提出的以门机制为基础的 2 种新融合函数.

1) 池化融合函数

与文献[15-16]中的做法类似,我们使用一个最大池化(max pooling)运算来融合$\{\hat{\boldsymbol{h}}_{t_k}\}$. 池化运算能够自动捕捉最重要的分词状态信息用于句子建模. 形式上,基于池化运算的融合函数定义为

$$\boldsymbol{h}_t=\max(\hat{\boldsymbol{h}}_{t_1},\hat{\boldsymbol{h}}_{t_2},\cdots,\hat{\boldsymbol{h}}_{t_K}),\quad(20)$$

其中,$\max(*)$是一个逐元素最大值函数.

池化融合函数忽略了词格的边权重信息,直接通过聚集入度边集合对应的隐状态来获取最重要的特征.

2) 门机制融合函数

目前,门机制已经大量应用于神经网络中,用以自动学习不同输入信息的权重. 与文献[16]相似,该融合函数在形式上定义为

$$\boldsymbol{h}_t=\sum_{k=1}^{K}\frac{\sigma(\hat{\boldsymbol{h}}_{t_k}\boldsymbol{u}^{(g)}+b^{(g)})}{\sum_{k'=1}^{K}\sigma(\hat{\boldsymbol{h}}_{t_{k'}}\boldsymbol{u}^{(g)}+b^{(g)})}\hat{\boldsymbol{h}}_{t_k},\quad(21)$$

其中,$\boldsymbol{u}^{(g)}$ 和 $b^{(g)}$ 分别是门机制融合函数的参数向量和偏置项标量,上标 g 表示门.

门机制融合函数则计算每个隐状态的归一化分数,作为边的权重,对隐状态进行加权平均. 这个分

数可以视为动态生成的边权重,表示模型将该边作为候选分词的置信度.

3) 基于词格边权重的融合函数

带权词格的一大特点是边权重可以有效区分不同分词结果的可能性.基于词格边权重,我们将 $\boldsymbol{h}_t$ 定义为不同分词结果的隐状态的加权和,即:

$$\boldsymbol{h}_t = \sum_{k=1}^{K} weight_{e_{t_k:t}} \hat{\boldsymbol{h}}_{t_k}, \tag{22}$$

其中,$weight_{e_{t_k:t}}$ 是根据式(3)计算出的边权重.显然,在这种融合方式中,融合权重主要取决于词格本身,而独立于网络模型.

与门机制融合函数不同,基于词格边权重的融合函数使用 1.1 节所述算法计算的词格边权重,对隐状态进行加权平均.同门机制生成的动态权重相比,词格边权重是静态的,可以直接表示边上的词作为候选分词的可能性.

4) 融入词格边权重的门机制融合函数

该融合函数与 2)基于门机制的融合函数相类似.不同的地方在于,门机制融合函数是无监督的,直接受模型训练目标影响,而相比之下,基于词格边权重的门机制融合函数则利用词格边权重作为外部监督信息来改进门机制学习到的融合权重.具体而言,我们要求门机制学习到的融合权重与词格边权分布尽量接近.为此,本文进一步引入门机制权重与词格边权重的欧式距离来作为惩罚项:

$$R_{\text{gate}} = \frac{1}{|D|} \sum_{s \in D} \sum_{t=1}^{N_s} \| \boldsymbol{g}_t^s - \hat{\boldsymbol{w}}_t^s \|_2, \tag{23}$$

其中,D 表示训练语料,N_s 是句子 s 的字个数,$\boldsymbol{g}_t^s$ 是门机制融合函数中的权重分布向量.特别地,为了保证 2 个分布具有可比性,本文将词格的边权重集合 $\{weight_{e_{t_k:t}} | 0 \leqslant k < K\}$ 进行归一化,生成向量 $\hat{\boldsymbol{w}}_t^s$.

融入词格边权重的门机制融合函数进一步使用静态边权重作为正则化项,指导动态边权重的生成,这一方法可以视为门机制融合函数与基于词格边权重的融合函数的结合.

上述 4 种融合函数,各自以递进的方式,从静态和动态到动静态结合地利用词格边权重,从而充分发挥模型的运算能力和利用词格结构提供的监督信息.

3 模型目标和训练

基于带权词格的 GRU 模型的训练过程与标准 RNN 相同.模型目标函数与后续所应用任务紧密相关.对于分类任务,本文模型首先建模学习句子语义表示,然后通过一个 softmax 层来预测句子的标签分布:

$$\hat{\boldsymbol{p}}(s;\boldsymbol{\theta}) = softmax(\boldsymbol{W}^{(y)} \boldsymbol{h}_{N_s} + \boldsymbol{b}^{(y)}), \tag{24}$$

其中,$\boldsymbol{\theta}$ 代表模型参数;$\boldsymbol{h}_{N_s} \in \mathrm{R}^d$ 是句子 s 在时刻 t 的隐状态,作为句子的向量表示;$\boldsymbol{W}^{(y)}$ 和 $\boldsymbol{b}^{(y)}$ 分别是 softmax 层的参数矩阵和偏置项向量,上标 y 表示该层的输出用于预测标签.设数据中共有 L 个候选标签,$\hat{\boldsymbol{p}}(s;\boldsymbol{\theta}) \in \mathrm{R}^L$ 为模型建模的概率分布,并且满足 $\sum_{l=1}^{L} \hat{p}^l(s;\boldsymbol{\theta}) = 1$.给定训练数据 D,模型的目标函数最终定义为

$$J(\boldsymbol{\theta}) = -\frac{1}{|D|} \sum_{s \in D} \sum_{l=1}^{L} p^l(s) \times \ln \hat{p}^l(s;\boldsymbol{\theta}) + \lambda \times R_{\text{gate}}, \tag{25}$$

其中,$p^l(s)$ 是句子真实标签的 one-hot 向量的第 l 个分量,R_{gate} 是根据式(23)定义的惩罚项.当本文模型使用前 3 种融合函数时,$\lambda = 0$;反之,当使用第 4 种融合函数时,λ 为一个大于 0 的常数.

本文采用基于 Adadelta[17] 的随机梯度下降算法来优化模型.此外,本文在训练过程中使用 dropout[18] 和最大范数正则化[19] 来防止模型训练过拟合.

4 实验与分析

为了验证本文模型的有效性,我们将 2 种基于带权词格的 GRU 循环神经网络和 4 种融合策略,分别应用于情感分类和问句分类任务,与传统 GRU 及现有的其他模型进行比较.

4.1 任务和数据集

本文将在情感分类和问句分类 2 个数据集上测试我们提出的方法.下面从数据集大小和数据特点等方面分别介绍这 2 个数据集.

1) 情感分类

数据集来自于新浪微博,为了保证数据信息的充分性,我们删除长度不足 6 个字的句子,然后安排 2 名标注人员对句子按照不同的情感(消极、中性和积极)倾向进行独立标注,最后保留标注结果完全一致的数据作为实验数据.按照上述方式,本任务实验数据集共包含消极情感句子 4 454 条、中性情感句子 5 100 条和积极情感句子 5 594 条.然后,本文采取分层抽样的方式,按照 7∶1∶2的比例从每个类别随机抽取样本,将数据划分为训练集(10 603 条实例)、验证集(1 514 条实例)和测试集(3 031 条实例).句子的平均长度 17.19 个词或 25.69 个字.

2) 问句分类

数据来自 FudanQuestionBank① 提供的中文问句分类数据集. 为了降低数据类别不均衡问题的影响,本文只选取数据量最大的 5 个分类. 该数据包含 1 517,4 987,1 101,3 185,2 174 条文本,对应的类别分别为枚举、事实、评价、推荐和需求. 同样,本文对该数据集按照 7∶1∶2的比例划分为训练集(9 075 条实例)、验证集(1 297 条实例)和测试集(2 592 条实例). 平均长度为 9.33 个词或 14.60 个字.

3) 带权词格生成

本文使用北京大学(PKU)、宾州大学中文树库(CTB)以及微软研究院(MSR)的分词语料分别训练 3 个分词模型,然后按照 1.1 节中所述方法生成每个句子的带权词格.

4.2 实验设置

本文所考察的对比模型包括:

1) GRU

GRU[5] 的最后一个序列状态作为句子语义表示用于预测句子标签. 另外,除了最简单的单层单向 GRU 模型之外,本文还同时比较了 3 个 GRU 的简单变种模型:双层单向(2 layer GRU, 2L-GRU)、单层双向(bidirectional GRU, BiGRU)和双层双向(2 layer bidirectional GRU, 2L-BiGRU)模型.

2) LSTM

LSTM[4] 的实验设置与 GRU 模型相同. 这一对比实验的目的是验证 GRU 与 LSTM 的性能,证明 2 个 RNN 的变种模型在本文 2 个任务上的效果相近.

3) CNN

卷积神经网络(convolutional neural network, CNN)[20] 使用不同大小的窗口处理输入序列,能够获得句子在不同粒度,包括字、词语甚至短语级别的语义信息. 本文参考 Kim[20] 的实验设置,使用单层卷积神经网络模型.

4) DCNN

动态卷积神经网络(dynamic convolutional neural network, DCNN)[21] 通过利用动态 k 最大池化操作,具有与 RNN 相似的处理变长序列,以及捕捉句子内部长短距离依赖关系的能力. DCNN 使用 2 个卷积层,k 最大池化操作的 $k=4$.

5) RAE

RAE(recursive autoencoder)[22] 通过贪婪方式构造文本序列的树结构,并将树的根结点作为该句子的向量表示. RAE 能够建模序列中词与词之间的组合顺序关系,学习句子内部成分的结构特征. 模型参数参考 Socher 等人[22] 的实验设置.

6) MulSrc

MulSrc(multiple source)独立地建模句子的字序列及词序列,最终通过 2.3 节所述的融合函数将句子表示进行一次融合,生成句子的语义表示. MulSrc 可以同时基于字和词建模,是本文模型的简化版本. 与词格不同,由于不存在句子级别的权重,我们简单地使用平均分布作为加权系数(Avg),模拟 2.3 节中基于边权重的融合函数,与本文所提出的带权词格 GRU 模型进行对比.

7) SWWL-GRU 和 DWWL-GRU

本文提出的基于带权词格的 GRU 循环神经网络模型在 4 种融合函数上进行了实验,相应的模型分别记为 SWWL(Pool), SWWL(Gate), SWWL(Weight), SWWL(wGate), DWWL(Pool), DWWL(Gate), DWWL(Weight), DWWL(wGate).

此外,字序列与词序列相比,是更简单的一种句子表示形式. 为了研究这种表示是否有助于文本语义建模,本文同样引入字序列到 SWWL-GRU 和 DWWL-GRU 的词格中,并与之进行对比实验与分析.

在实验参数方面,本文统一使用 dropout[18] 防止模型训练过拟合,并根据验证集结果对 dropout 值进行选择. 我们根据验证集挑选式(22)中调节惩罚项的 λ 值,将其设为 1.0. 词表由语料中出现次数在 2 次及以上的高频词构成. 词向量和隐状态的维度分别为 50 和 300 维. 所有模型均使用基于随机梯度下降的 Adadelta 算法[17] 实现优化,批梯度更新的大小为 1. 每个模型分别训练 5 次,根据开发集的效果选择最优模型,并取测试集上的平均准确率作为最终结果.

4.3 实验结果分析

表 1 和表 2 分别给出了基线模型与本文模型在情感分类和问句分类任务上的实验结果. 从表 1 和表 2 中数据可以看出,本文模型的分类效果要显著高于单一字序列或词序列的模型.

从表 1 和表 2 可得出 5 条结论:

1) GRU 和 LSTM 的性能相近

GRU[5] 与 LSTM[4] 模型,是针对梯度消失和梯

① https://code.google.com/archive/p/fudannlp/

度爆炸问题[10-12]所提出的2个RNN[3]变种模型.在本文实验中,GRU和LSTM在2个数据集上的性能没有表现出显著差异,然而GRU模型具有更少的参数,因此在某种程度上降低了过拟合的风险.

Table 1 Results of Baseline Models on Sentiment Classification and Question Classification

表1 基线模型的情感分类和问句分类实验结果 %

Model	Sentiment				Question			
	Char	CTB	MSR	PKU	Char	CTB	MSR	PKU
GRU[5]	70.8	73.3	**73.7**	72.8	86.4	86.2	86.1	84.8
LSTM[4]	69.2	73.2	73.3	72.6	**86.6**	**86.3**	86.0	84.8
2L-GRU	69.5	**73.7**	73.1	72.6	86.5	86.2	**86.4**	84.5
BiGRU	**71.4**	**73.7**	73.5	**73.1**	**86.6**	86.0	85.9	**86.2**
2L-BiGRU	70.7	72.6	72.6	72.4	86.2	86.2	86.0	85.0
CNN[20]	65.1	69.0	68.0	68.1	82.4	81.5	80.1	79.9
DCNN[21]	66.4	64.8	62.2	65.2	84.2	80.6	80.5	71.9
RAE[22]	59.9	68.6	68.8	68.4	72.1	79.1	79.2	78.0

Notes: The values in boldface indicate the best accuracy in that experimental group.

Table 2 Results of Our Work on Sentiment Classification and Question Classification

表2 本文模型的情感分类和问句分类实验结果

Model	Gating	Sentiment		Question	
		Words	Char+Words	Words	Char+Words
MulSrc	Pool	72.1	71.7	85.5	85.9
	Avg	71.7	71.9	**85.7**	86.1
	Gate	**72.5**	**72.6**	85.5	**86.2**
Ours	SWWL(Pool)	73.7	72.8	86.3	87.2
	SWWL(Gate)	73.8	73.1	86.2	87.3
	SWWL(Weight)	74.0	74.0	86.5	87.1
	SWWL(wGate)	73.7	73.3	86.3	87.4
	DWWL(Pool)	74.3	73.9	**87.0**	87.1
	DWWL(Gate)	72.3	71.6	84.8	85.5
	DWWL(Weight)	74.1	**74.2**	86.8	87.2
	DWWL(wGate)	**74.6**	73.6	86.8	**87.7**

Notes: The values in boldface indicate the best accuracy in that experimental group.

2) 基于词序列的GRU及其变种模型一致地优于基于字序列的模型

分词在中文等没有词语分隔符的自然语言处理任务中具有非常重要的作用,这是因为分词可以在一定程度上消除纯字序列存在的语义歧义现象.相比之下,基于字序列的模型忽略了句子中的词语边界信息,从而无法消除句子中存在的语义歧义,导致模型学习到的句子语义表示不能很好地服务于分类任务.然而,模型在问题分类任务上的结果并没有明显地反映出这一趋势.据统计结果显示,GRU具备强大的捕捉长短期依赖的能力,但对于短序列而言,短期依赖则占据了主要地位.这使得循环神经网络无法发挥其建模长期依赖关系的优势,从而对以短句为主的问句分类任务,弱化了基于词序列与基于字序列的模型在结果上的差异.

3) 基于CNN的模型效果弱于基于RNN的模型

正如文献[23]所示,相比于CNN,RNN模型对长序列文本的建模优势较为明显.尽管CNN在速度上具有明显优势,但在性能表现上却难以取代RNN.就表1中实验结果来说,CNN由于同时使用多个卷积核,使其在某种程度上能够捕捉所有的多元词串,从而与本文所提出的模型一样具备建模不同分词结果的能力.然而,并非所有多元词串都能表示一般意义上的有效词语,因此CNN也同时引入

了更多的错误分词,导致基于 CNN 的模型在 2 个任务上的表现均明显不如 RNN 模型

4) 基于带权词格的模型优于基于字、词序列以及 MulSrc 的模型

相比于对比模型,本文提出的 2 个模型在情感分类和问句分类任务上均一致取得了更高的准确率. 首先,只基于字和词建模的模型,缺乏表达分词多样性的能力;其次,同时基于字和词建模的 MulSrc 模型,由于仅在句级别融合句子语义表示,使得句子的最小单元无法在字、词序列间进行交互. 此外,在大部分情况下,DWWL-GRU 性能超过 SWWL-GRU,取得了 2 个任务上的最好结果,这证明深层次的语义融合比浅层次的语义融合效果更好. 此外,本文提出的 2 个模型在使用融入词格边权重的门机制融合函数上均取得最好结果,其次分别是门机制融合函数,以及基于词格边权重的融合函数. 这一实验结果与我们的直觉相符. 首先,门机制是无监督的权重学习,而基于词格边权重的融合函数则直接根据词格边权重来进行加权融合. 相比之下,融入词格边权重的门机制融合函数,有效结合了上述 2 种融合机制的特点,进一步提高了所生成权重的质量,从而得到了更好的融合文本语义表示.

5) 融合函数的建模能力影响模型性能

引入字信息后,基于带权词格的模型在问句分类任务上的效果得到进一步提升,但情感分类任务的效果却降低. 直观上看,字信息的引入能够有效扩充词格的信息量. 但实际而言,情感分类的词表大小为 42 685,问句分类则只有 11 634,因此情感分类任务的词表更大,更难学习到有效的句子表示. 在情感分类上引入字后,词格模型所要建模的分词组合数量进一步增加. 我们的融合函数无法充分建模所有相应的分词情况,从而加剧了数据稀疏问题的影响. 问句分类任务则恰恰相反,单纯就词表大小而言,即使引入字,词格模型中可能的分词组合数量也远远低于情感分类. 因此与 Words 相比,我们的模型对于问句分类任务可以在 Char+Words 上更有效地建模,充分利用引入字后的词格信息增益,进而提升模型效果. 实际上,本文提出的 4 种融合函数中最复杂的 wGate 融合函数,依然只包含一个与隐状态同等维度的参数向量,所以建模能力有限. 因此,一个更复杂的融合函数应当能够在情感分类的 Char+Words 上进一步改进模型的性能. 但为了证明基于带权词格的循环神经网络模型相对于传统基于词序列模型的有效性,我们在尽量不引入额外参数的前提下,保证融合函数足够简单. 本文的讨论范围限于验证基于带权词格模型的有效性,因此我们将对具有更强学习能力的融合函数的研究放到未来工作中深入探讨.

为了探究所提出模型的工作机制,以性能最好的 DWWL(wGate)为例,我们在图 5 中展示了一个句子的文本建模结果. 在所示词格中,每条边标注有一个分数,该分数为模型动态生成的权重,表示该边所对应的词,在特定上下文中被作为一个候选分词的可能性,该权重直接影响模型的文本语义表示建模质量.

图 5 中所示为句子:"不然肯定是纳税人白花冤枉钱."的建模结果. 句中存在歧义的部分集中在 $v_8 \sim v_{13}$ 部分,即"白花冤枉钱"这一片段,根据上下文,我们判断其正确的分词结果应当为"白花/冤枉钱"或"白花/冤枉/钱". 图 5 中粗边表示错误的候选分词,实边表示正确的候选分词. 可以观察到,词格中存在来自不同分词模型产生的错误分词,如"白花冤"和"枉钱". 结点 v_{13} 有 3 条入度边,分别对应:"钱"、"枉钱"、"冤枉钱"3 个候选分词. 其中,正确的分词"钱"和"冤枉钱"被作为候选词的置信度 p 为 0.36 和 0.35;而错误分词"枉钱"的置信度 p 只有 0.29. 尽管"白花冤"在结点 v_{11} 的置信度为 1.00,但由于错误分词"枉钱"存在于"白花冤"的分词路径中,因此该路径依然得到了更低的分数. 我们可以将模型建模的边置信度视为概率,通过路径的概率来

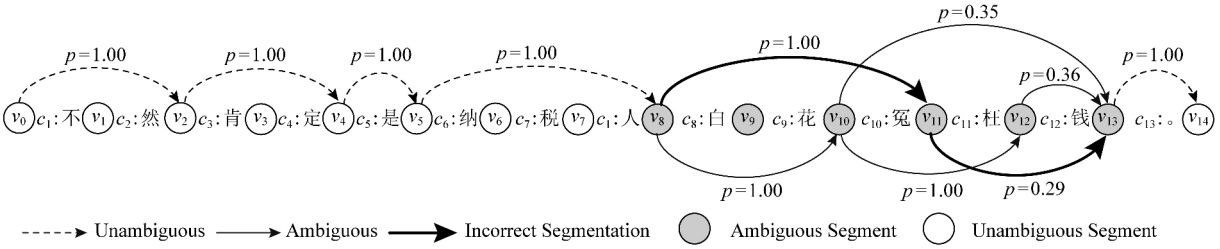

Fig. 5 The semantic modeling of an example sentence

图 5 一个例句的语义建模

更好地理解这一示例.图5中包含错误分词"白花冕/枉钱"的路径,其概率 $p(\text{false})=1.00\times0.29=0.29$.而包含正确分词"白花/冕枉/钱"和"白花/冕枉钱"的路径,通过将其概率相加,可知正确路径的总概率为 $p(\text{true})=1.00\times1.00\times0.36+1.00\times0.35=0.71$.因而在示例中正确路径的置信度是远高于错误路径的.不难看出,词格模型具有容错的能力,当错误的候选分词被赋予低权重后,错误路径的权重被降低,而正确路径所产生的影响通过高权重放大,从而减轻纯词序列中分词错误传播的问题.另一方面,单纯基于字和词序列的建模方法,则易受到错误分词的影响,而基于带权词格的模型则能够利用其容错能力来保证即使存在错误分词,模型仍然能够学习到高质量的句子语义表示.

5 相关工作

目前,基于深度神经网络的文本语义表示学习已经成为自然语言处理的热门研究方向.其中,神经词袋(bag-of-words)模型是最为简单的一个模型,它对句子中所有词的词向量取平均直接得到句子的语义表示向量.显然,这种建模方式忽略了对文本语义表示极为重要的词序信息.因而,许多研究者转向研究考虑词序信息的模型,包括序列神经网络模型和拓扑神经网络模型等.典型的序列神经网络模型包括RNN[3],LSTM[4,24-30],以及带门机制的其他变形[31-33].而与序列神经网络模型不同,拓扑神经网络模型依赖给定的词间拓扑结构来建模生成文本语义表示[22,34-36].例如句子的依存和组合范畴语法可被作为骨架用于学习句子语义表示[28,37-39].进一步,一些研究者提出多维度的神经网络模型,该类模型将文本组织成一个多维网格而非序列作为输入[40-41].此外,除了上述模型,卷积神经网络也被用于句子建模[20-21].该类网络也是以词向量序列作为输入,建模过程中通过多层的卷积和池化操作来得到句子语义表示.

在上述工作中,与本文较为相关的工作主要有文献[15,27-28]中所提出的模型.文献[27-28]在本质上属于拓扑神经网络模型,分别将序列LSTM扩展到树结构和森林结构的网络.文献[40]提出了基于网格的LSTM,把LSTM单元按照多维网格的方式排列,以应用到一维、二维甚至更多维度的序列数据的语义建模学习.此外,文献[42]提出在生成当前隐状态时,对RNN中多个前隐状态使用与本文门机制相似的方式分别计算权重,然后将多个前隐状态加权输入到RNN单元.文献[15]提出基于词格的循环神经网络,通过Pooling运算和门机制来融合生成词格单元的输入.不同于这些网络,本文工作在文献[15]的基础上进行扩展,引入了带权词格来提高句子建模的能力,更重要的是本文模型引入词格权重来指导融合函数的建模学习,进一步提高词格循环神经网络语义表示的学习效果.

6 总 结

文本提出了2种基于带权词格的GRU循环神经网络模型,用于句子的语义表示建模.2种模型均以带权词格为基础,利用任意数量的输入词和前隐状态信息来融合生成当前隐状态,最终得到句子语义表示.在以句子语义表示为基础的情感分类和问句分类2个任务上的实验结果证明了本文模型的有效性.

未来,我们将在下面3个研究方向展开工作:

1)研究如何把带权词格集成到其他神经网络中,例如卷积神经网络等;

2)融入词格边权重的门机制融合函数虽然取得最好效果,但与其他融合函数相比优势有限,如何设计其他更加有效融合函数也是下一步工作的重点之一;

3)本文所使用构造词格的方法较为简单,因此,我们将尝试使用其他的语言学信息构造词格,以进一步提升模型性能.

参 考 文 献

[1] Yu Kai, Jia Lei, Chen Yuqiang, et al. Deep learning: Yesterday, today, and tommorrow [J]. Journal of Computer Research and Development, 2013, 50(9): 1799-1804 (in Chinese)
(余凯, 贾磊, 陈雨强, 等. 深度学习的昨天、今天和明天[J]. 计算机研究与发展, 2013, 50(9): 1799-1804)

[2] Chen Ke, Liang Bin, Ke Wende, et al. Chinese micro-blog sentiment analysis based on multi-channels convolutional neural networks [J]. Journal of Computer Research and Development, 2018, 55(5): 945-957 (in Chinese)
(陈珂, 梁斌, 柯文德, 等. 基于多通道卷积神经网络的中文微博情感分析[J]. 计算机研究与发展, 2018, 55(5): 945-957)

[3] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model [C] //Proc of the 11th Annual Conf of the Int Speech Communication Association. Phoenix, Arizona: ISCA, 2010: 1045-1048

[4] Hochreiter S, Schmidhuber J. Long short-term memory [J]. Neural Computation, 1997, 9(8): 1735-1780

[5] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [C] //Proc of the 19th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2014: 1724-1734

[6] Dyer C, Muresan S, Resnik P. Generalizing word lattice translation [C] //Proc of the 46th Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA: ACL, 2008: 1012-1020

[7] Ladhak F, Gandhe A, Dreyer M, et al. LatticeRNN: Recurrent neural networks over lattices [C] //Proc of the 17th Annual Conf of the Int Speech Communication Association. Phoenix, Arizona: ISCA, 2016: 695-699

[8] Charniak E, Johnson M. Coarse-to-fine *n*-best parsing and MaxEnt discriminative reranking [C] //Proc of the 43rd Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA: ACL, 2005: 173-180

[9] Huang Liang. Forest reranking: Discriminative parsing with non-local features [C] //Proc of the 46th Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA: ACL, 2008: 586-594

[10] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult [J]. IEEE Transactions on Neural Networks, 1994, 5(2): 157-166

[11] Bengio Y, Frasconi P, Simard P. The problem of learning long-term dependencies in recurrent networks [C] //Proc of IEEE Int Conf on Neural Networks. Piscataway, NJ: IEEE, 1993: 1183-1188

[12] Erhan D, Manzagol P A, Bengio Y, et al. The difficulty of training deep architectures and the effect of unsupervised pre-training [C] //Proc of the 12th Int Conf on Artificial Intelligence and Statistics. Cambridge, MA: MIT Press, 2009: 153-160

[13] Jiang Wenbin, Mi Haitao, Liu Qun. Word lattice reranking for Chinese word segmentation and part-of-speech tagging [C] //Proc of the 22nd Int Conf on Computational Linguistics. Stroudsburg, PA: ACL, 2008: 385-392

[14] Wang Zhiguo, Zong Chengqing, Xue Nianwen. A lattice-based framework for joint Chinese word segmentation, pos tagging and parsing [C] //Proc of the 51st Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2013: 623-627

[15] Su Jinsong, Tan Zhixing, Xiong Deyi, et al. Lattice-based recurrent neural network encoders for neural machine translation [C] //Proc of the 31st AAAI Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2017: 3302-3308

[16] Le P, Zuidema W. The forest convolutional network: Compositional distributional semantics with a neural chart and without binarization [C] //Proc of the 20th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2015: 1155-1164

[17] Zeiler M D. ADADELTA: An adaptive learning rate method [OL]. 2012 [2017-10-01]. https://www.arxiv.org/abs/1212.5701

[18] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: A simple way to prevent neural networks from overfitting [J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958

[19] Srebro N, Shraibman A. Rank, trace-norm and max-norm [C] //Proc of the 18th Int Conf on Computational Learning Theory. Berlin: Springer, 2005: 545-560

[20] Kim Y. Convolutional neural networks for sentence classification [C] //Proc of the 19th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2014: 1746-1751

[21] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences [C] //Proc of the 52nd Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2014: 655-665

[22] Socher R, Pennington J, Huang E H, et al. Semi-supervised recursive autoencoders for predicting sentiment distributions [C] //Proc of the 16th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2011: 151-161

[23] Yin Wenpeng, Kann K, Yu Mo, et al. Comparative study of CNN and RNN for natural language processing [OL]. 2017 [2017-10-01]. https://www.arxiv.org/abs/1702.01923

[24] Graves A, Jaitly N, Mohamed A. Hybrid speech recognition with deep bidirectional LSTM [C] //Proc of the 8th IEEE Workshop on Automatic Speech Recognition and Understanding. Piscataway, NJ: IEEE, 2013: 273-278

[25] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks [C] //Proc of Int Conf on Acoustics, Speech and Signal Processing. Piscataway, NJ: IEEE, 2013: 6645-6649

[26] Vinyals O, Toshev A, Bengio S, et al. Show and tell: A neural image caption generator [C] //Proc of the 28th IEEE Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2015: 3156-3164

[27] Tai Kaisheng, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks [C] //Proc of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Int Joint Conf on Natural Language Processing. Stroudsburg, PA: ACL, 2015: 1556-1566

[28] Le P, Zuidema W. Compositional distributional semantics with long short term memory [C] //Proc of the 4th Joint Conf on Lexical and Computational Semantics. Stroudsburg, PA: ACL, 2015: 10-19

[29] Zhu Xiaodan, Sobihani P, Guo Hongyu. Long short-term memory over recursive structures [C] //Proc of the 32nd Int Conf on Machine Learning. Lille, France: PMLR, 2015: 1604-1612

[30] Liu Pengfei, Qiu Xipeng, Chen Xinchi, et al. Multi-timescale long short-term memory neural network for modelling sentences and documents [C] //Proc of the 20th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2015: 2326-2335

[31] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling [OL]. 2014 [2017-10-01]. https://www.arxiv.org/abs/1412.3555

[32] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks [C] //Proc of the 28th Conf on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2014: 3104-3112

[33] Chen Xinchi, Qiu Xipeng, Zhu Chenxi, et al. Sentence modeling with gated recursive neural network [C] //Proc of the 20th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2015: 793-798

[34] Socher R, Lin C C, Ng A Y, et al. Parsing natural scenes and natural language with recursive neural networks [C] //Proc of the 28th Int Conf on Machine Learning. New York: ACM, 2011: 129-136

[35] Socher R, Huval B, Manning C D, et al. Semantic compositionality through recursive matrix-vector spaces [C] //Proc of the 17th Joint Conf on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. Stroudsburg, PA: ACL, 2012: 1201-1211

[36] Socher R, Perelygin A, Wu J, et al. Recursive deep models for semantic compositionality over a sentiment treebank [C] //Proc of the 18th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2013: 1631-1642

[37] Hermann K M, Blunsom P. The role of syntax in vector space models of compositional semantics [C] //Proc of the 51st Annual Meeting on Association for Computational Linguistics. Stroudsburg, PA: ACL, 2013: 894-904

[38] Iyyer M, Boyd-Graber J, Claudino L, et al. A neural network for factoid question answering over paragraphs [C] //Proc of the 19th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2014: 633-644

[39] Mou Lili, Peng Hao, Li Ge, et al. Discriminative neural sentence modeling by tree-based convolution [C] //Proc of the 20th Conf on Empirical Methods in Natural Language Processing. Stroudsburg, PA: ACL, 2015: 2315-2325

[40] Kalchbrenner N, Danihelka I, Graves A. Grid long short-term memory [OL]. 2015 [2017-10-01]. https://www.arxiv.org/abs/1507.01526

[41] Graves A, Fernández S, Schmidhuber J. Multi-dimensional recurrent neural networks [C] //Proc of the 17th Int Conf on Artificial Neural Networks. Berlin: Springer, 2007: 549-558

[42] Soltani R, Jiang Hui. Higher order recurrent neural networks [OL]. 2016 [2017-10-01]. https://www.arxiv.org/abs/1605.00064

Zhang Xiangwen, born in 1994. Master candidate. His main research interests include natural language processing and neural machine translation.

Lu Ziyao, born in 1996. Master candidate. His main research interests include natural language processing and neural machine translation.

Yang Jing, born in 1994. Master candidate. Her main research interests include natural language processing and neural machine translation.

Lin Qian, born in 1995. Master candidate. Her main research interests include natural language processing and neural machine translation.

Lu Yu, born in 1996. Graduate student. Her main research interests include natural language processing and neural machine translation.

Wang Hongji, born in 1968. PhD. Associate professor in Xiamen University. His main research interests include automata theory, cryptography and information security.

Su Jinsong, born in 1982. Received his PhD degree from Chinese Academy of Sciences. Associate professor in Xiamen University. His main research interests include natural language processing and machine translation.