doi:10.11835/j.issn.1000-582X.2020.09.005

基于句法依存卷积神经网络的句子相似度计算

铉 静[1],吴 琼[2],魏从悦[1],伍 星[1]
(1.重庆大学 计算机学院,重庆 400044;2.重庆工商大学 管理科学与工程学院,重庆 400067)

摘要:句子相似度计算是自然语言处理的一项基础任务,其准确性直接影响机器翻译、问题回答等下游任务的性能。传统机器学习方法主要依靠词形、词序及结构等浅层特征计算句子相似度,而深度学习方法能够融入深层语义特征,从而取得了更好效果。深度学习方法如卷积神经网络在提取文本特征时存在提取句子语义特征较浅、长距离依赖信息不足的缺点。因此设计了 DCNN(dependency convolutional neural network)模型,该模型利用词语之间的依存关系来解决该不足。DCNN 模型首先通过依存句法分析得到句子中词语之间的依存关系,而后根据与当前词存在一跳或者两跳关系的词语形成二元和三元的词语组合,再将这两部分信息作为原句信息的补充,输入到卷积神经网络中,以此来获取词语之间长距离依赖信息。实验结果表明,加入依存句法信息得到的长距离依赖能有效提升模型性能。在 MSRP(microsoft research paraphrase corpus)数据集上,模型准确度和 F1 值分别为 80.33%和 85.91,在 SICK(sentences involving compositional knowledge)数据集上模型的皮尔森相关系数能达到 87.5,在 MSRvid(microsoft video paraphrase corpus)数据集上模型的皮尔森相关系数能达到 92.2。

关键词:句子相似度;依存句法树;长距离依赖
中图分类号:TP391.1 **文献标志码**:A **文章编号**:1000-582X(2020)09-041-13

Sentence similarity computation based on syntactic dependency convolutional neural network

XUAN Jing[1], *WU Qiong*[2], *WEI Congyue*[1], *WU Xing*[1]
(1. College of Computer Science, Chongqing University, Chongqing 400044, P. R. China; 2. College of Management Science and Engineering, Chongqing Technology and Business University, Chongqing 400067, P. R. China)

Abstract: Sentence similarity computation is a basic task of many natural language processing, and its accuracy has a direct impact on the performance of language related systems, especially in machine translation, plagiarism detection, query ranking and question answering. Compared with the traditional methods that rely on shallow features like morphology, word sequence and grammar structure for sentence

收稿日期:2020-01-11
基金项目:重庆工商大学开放基金项目资助(KFJJ2019056,KFJJ2019050);重庆工商大学杰、优博士人才计划资助项目(2056001);重庆工商大学数据与信息管理方向学科建设资助项目(ZDPTTD201917)。
Supported by Open Funding of Chongqing Technology and Business University (KFJJ2019056, KFJJ2019050); Chongqing Technology and Business University Outstanding & Doctoral Program funding (2056001); Chongqing Technology and Business University data and Information Management discipline construction funding(ZDPTTD201917).
作者简介:铉静(1980—),女,重庆大学博士研究生,主要从事计算语言学方向研究,(E-mail)xuanjing8045@163.com。

similarity computation, deep learning methods can integrate the deep semantic features and achieve better results. However, deep learning methods using convolutional neural networks needs to overcome defects such as narrow receptive field and insufficient long-distance information dependence when extracting text features. In this paper, a DCNN(dependency convolutional neural network) model was established to carry out dependency-based syntactic analysis for information retrieval over longer distance. We made text parsing, employing Stanford NLP for syntactic analysis, and then retrieved mutual relationship between two words in a binary combination or triplet. As lexical supplement information embedded in these word combinations, the dependency information, in addition to that of the original sentence, was added up as Convolutional Neural Network input, thus constructing a Dependency CNN. Experiment results reveal that the long distance dependency information effectively improve the similarity computation performance in our proposed dependency model on MSRP(Microsoft research paraphrase corpus) dataset, and the accuracy and F1 value are 80.33% and 85.91 respectively. The Pearson correlation coefficient of the model reaches 87.5 on SICK (Sentences invloving compositional knowledge) dataset and 92.2 on MSRvid (Microsoft videl paraphrase corpus) dataset.

Keywords: sentence similarity computation; dependency parsing; long distance dependency

句子相似度计算作为机器翻译、剽窃检测、查询排序及问题回答等诸多自然语言处理任务的基础,其准确性很大程度上决定了这些应用系统的性能,因而如何提升句子相似度计算的准确性成为亟待解决的问题。

目前句子相似度计算已有大量研究,已有研究主要利用人工经验从文本中提取特征[1-4],再利用浅层统计模型进行分类,由于这种浅层统计模型对文本的表示能力较弱,所以系统性能的好坏依赖于人工提取特征的质量,但不同任务所需提取的特征往往不尽相同,致使人工提取特征费时费力,故而人们开始寻求新的特征提取方法。

深度学习在图像领域取得的优异成绩,人们将其引入到自然语言处理任务中,其优点在于无需人工干预就可从大规模的文本中学习词法、语法、语义等特征进而提升文本的表示能力。研究表明,深度学习在释义识别(PI,paraphrase identification)、答案选择(AS,answer selection)和文本蕴含(TE,textual entailment)任务上都取得了良好的效果。

句子相似度计算常用的人工神经网络结构有循环神经网络(RNN,recurrent neural network)和卷积神经网络(CNN,convolutional neural networks)。RNN 模型擅长对整个句子进行建模,捕捉长距离依赖信息,但句子长度过长时,长距离依赖信息仍然会丢失。CNN 模型擅长抽取局部特征,但对长距离依赖信息的提取欠缺。为此,笔者提出了 DCNN 模型,通过句子中词语之间的依存关系,将长距离依赖信息融入 CNN 模型中,该模型既能捕捉长距离依赖信息,又可以很好地抽取关键短语信息。实验结果表明 DCNN 模型在 MSRP,SICK 等多个数据集上取得了良好效果。

主要贡献如下:

1)在句子中词语的词向量基础上,通过依存关系加入二元和三元关系词语的词向量增强句子的语法、语义表示,有效提升句子相似度计算的性能;

2)相对于注意力机制从语义方面获取长距离依赖信息,通过句子的依存句法结构信息同样可有效获取长距离依赖信息,并在句子相似度任务中取得更好效果。

3)注意力机制和依存句法结构分别从语义和结构 2 个维度获取长距离依赖信息同时融合使用,可进一步提升句子相似度计算的效果。

1 相关研究

1.1 基于传统机器学习的方法

传统机器学习方法计算句子相似度的方法主要包括:表层信息匹配计算法[5-6]、基于语义词典的计算

法[7]、基于句法分析的计算法[8]等,它们主要利用句子的浅层信息计算句子相似度。上述方法主要从单一方面来对句子进行表征,并进行相似度计算,这些方法优缺点各异,为了能够相互补充,人们开始尝试多特征融合的方法[9],这类方法可以综合多种传统句子相似度计算方法,再将这些方法的输出结果作为描述句子的特征值,综合考虑了句子表层信息、词义和句法特征等,从多个角度对句子进行特征表示,使其更加合理、科学;缺点在于依旧没有解决这些方法的不足之处。

1.2　基于深度学习的方法

深度学习无需人工构造特征就能自动抽取句子的语义信息。目前基于深度学习的句子相似度计算主要分为 RNN 和 CNN 两类,其中基于 CNN 的方法是主流方法。

1.2.1　基于 CNN 的句子相似度计算方法

Kim[10]等人提出了静态词向量和非静态词向量组合输入的 CNN 模型,该方法在反向传播过程中可以微调词向量,让词向量表示更贴近语料库,从而提升模型性能。Kalchbrenner's [11]等人在 CNN 中采用了动态池化的方法,该方法能减少句子的信息损失,且能保留部分位置信息。Yin[12]等人提出了基于 CNN 的注意力机制模型,该方法通过注意力机制来引导输入层的信息交互,首先通过相似度矩阵来赋予词向量不同权值,再对得到的词向量进行特征提取,实验表明该模型学习的句子表示在一系列任务中都有良好的效果。Yin[13]等人在 CNN 模型中融合了词、短 ngram、长 ngram 和句子粒度的特征,通过各个粒度的特征组合来提升对句子语义的理解。Wang [14]等人利用相似度矩阵将句子分解成相似相异信息来重构句子表示,再将新的句子表示输入 CNN 中提取特征,实验结果表明这种方式提取的特征对句子语义的相似度判断更加准确。He[15]等人在 CNN 的基础上采用 3 种池化方式、2 种卷积方式及 3 种滤波器窗口来充分挖掘文本特征,在所有基于神经网络的模型中,该方法在 MSRP 数据集上取得了最好的效果。

1.2.2　基于 RNN 的句子相似度计算方法

Socher[16]等人提出了一种用于释义检测的无监督递归自编码器(RAE,recursive auto encoder),该方法能够学习句法树中短语的特征表示,再将其用于句子相似度计算。Blunsom[17]等人提出了一种组合的自编码器模型来学习句子的高维特征表示,实验表明该模型学习的句子表示有效且具有通用性。Tai[18]等人提出了融入句法结构的长短时记忆网络,该方法在长短时记忆网络中加入句法结构信息,在语义关联预测和情感分类任务上均取得了良好的效果。杨萌[19]等人提出了一种基于浅层句法树和短语依赖树的长短时记忆网络,实验表明,结构化特征的加入会提升模型性能。

1.3　研究问题与动机

利用 CNN 进行句子表示的研究中利用卷积进行局部特征抽取,并利用多层卷积来捕获长期依赖关系。当网络层数过多时会导致模型难以训练和模型容易过拟合,因此既能抽取关键短语信息,同时又兼顾到长距离依赖信息,是神经网络模型改进的一个重点。常用方法有将 CNN 和 RNN 进行结合的方式,或是在 CNN 上进行改进。Lai[20]提出的 RCNN 模型用于句子分类任务,Emma 等人[21]提出了 IDCNN+CRF 模型用于命名实体识别,都是使用膨胀卷积来扩大感受。但是这 2 种方式是通过穷举组合和简单扩大视野的方式来提取长距离依赖信息,而非准确定位长距离依赖关系再提取句子的长距离信息。基于上述研究,提出了一个简化而有效的模型,该模型不需要多层 CNN 堆叠,而是将长距离依赖信息通过句子中词语之间的依存关系进行提取,再将这种信息作为原始输入信息的补充,以此解决长距离依赖信息不足的问题。

2　背景介绍

2.1　卷积神经网络

基于深度学习的自然语言处理中,卷积的输入通常为一句话所构成的词向量矩阵[22-23]。每一行代表一个词的词向量,所以在处理文本时,卷积核通常覆盖上下几个词,所以此时卷积核的宽度与输入的宽度相同,实现捕捉到多个连续词构成的局部特征,并且能够在同一类特征计算中共享权重。如图 1 所示,首先分别用了长度为 2 和 3 的卷积核进行局部特征提取,然后再利用池化操作进行显著特征的提取,最后进行特征汇集以形成句子表示。

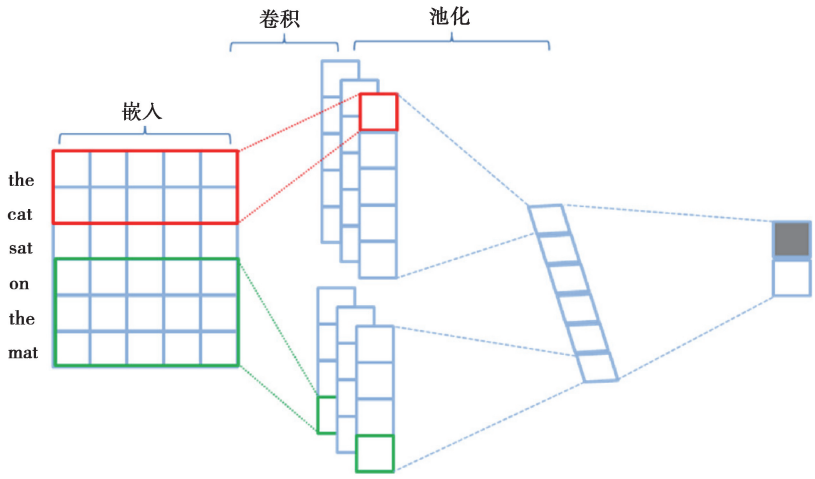

图1 基于CNN的句子表示

Fig. 1 Sentence representation based on CNN

2.2 依存句法

依存句法是通过依存关系将2个词语紧密连接起来，用以表示两者之间的依存关系，最终表示为整个句子句法关系的结构。研究表明在深度学习中通过依存句法树引入更多信息，可以有效提升自然语言处理任务的性能。

2.3 网络结构

句子相似度计算常用的网络结构有单输入网络和双输入网络。单输入网络是指首先将2个句子进行信息融合，融合方式主要有将2个句子进行拼接或者是将2个句子作为双通道形式，如图2的(a)和(b)所示，而后再输入到网络结构中，这种网络的缺点是过早的丢失了句子单独的特征信息，因而对2个句子的相似性判断会产生不利的影响。双输入网络可以避免单输入网络过早丢失句子单独信息的不足，常用的双输入网络又可以细分为伪孪生网络和孪生网络，如图3的(a)和(b)所示。其中孪生网络共享训练权重，这样不仅可以对句子的内部信息进行一定的交互，又可以加快模型训练速度，因而模型使用孪生网络结构。

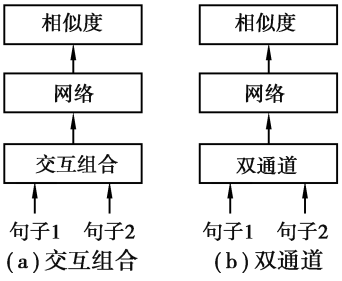

图2 单输入网络

Fig. 2 Single input network

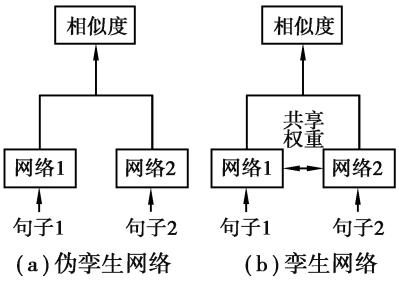

图3 双输入网

Fig. 3 Two-input network

3 模型

提出的DCNN模型整体框架如图4所示，主要包含2个模块，一是句子表示模块：主要是对句子进行建模，提取出与任务有关句子的词法、语法、语义等特征，二是相似度计算模块：主要是用来对2个句子的特征表示进行相似度计算。

3.1 句子表示

句子表示模块由输入层、卷积层、池化层组成。输入层为三类信息词向量信息的组合，输入形式为词向量矩阵。卷积层主要通过一定数目和不同大小的卷积核来保证特征提取的丰富性，池化层通过不同的池化

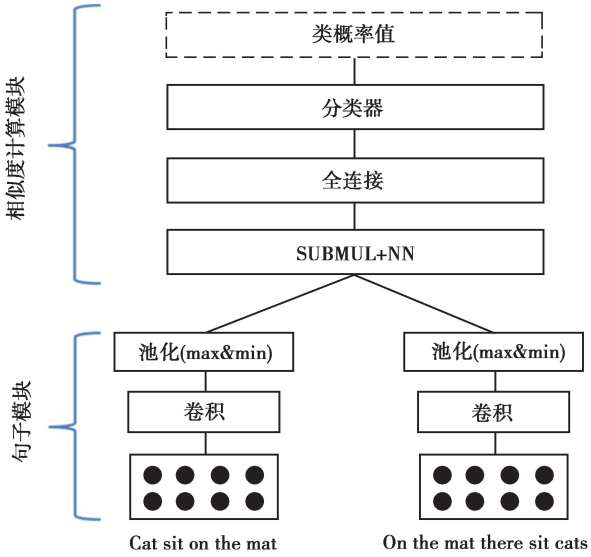

图 4　相似度计算模型

Fig. 4　Similarity calculation model

方式来保留更多的特征信息。

1)输入层

输入信息分别是原句、通过依存句法的一跳关系得到的二元组和两跳关系得到的 3 元组。①原句部分

模型输入为句子的词向量矩阵,首先通过查表将其表示成词向量形式,将无相互关系的稀疏向量转化为相互联系且有固定大小的稠密向量。模型使用 3 种公开发布的词向量,分别是文献[24]提出的 Paragram 词向量和文献[25]提出的 Paragram-Phrase-XXL 词向量、以及 Glove 词向量,3 种词向量采用连接方式进行组合,如公式 1 所示,其中⊕代表串联算子。

$$\text{Embedding} = \text{Embedding}_{\text{Paragram}} \oplus \text{Embedding}_{\text{Paragram-Phrase-XXL}} \oplus \text{Embedding}_{\text{Glove}} 。 \tag{1}$$

②二元组和三元组

二元组是指依存句法树中由一跳依存关系得到的 2 个依赖词的组合,三元组是指依存句法树中由两跳依存关系得到的 3 个依赖词的组合,下面以“My dog also likes eating sausage.”为例进行说明,经过斯坦福句法分析后得到图 5 的形式

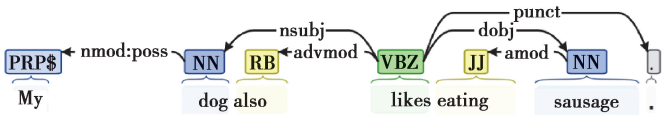

图 5　依存句法分析示例

Fig. 5　Examples of dependency syntax analysis

其中箭头所指方向为被依赖词,且箭头内容为依存关系,每个单词上方表示其词性,如图 5 所单词 My 的依赖词为 dog,而 dog 的依赖词为 likes,所以“My dog”和“dog likes”为二元组组合,而“My dog likes”就是一个三元组组合。需要注意的是提取后的二元组和三元组是作为原句信息的补充,为了将长距离依赖信息添加到原句中,所以需要将提取后的词语组合转换为和原句中词语顺序相同,以避免和原句中提取的词语组合信息产生冲突。按照这种多跳依存关系也可以提取到句子 4 个词语以及更多词语的组合,但是实验发现当词语组合超过 3 个词时,会导致数据稀疏问题,因而模型只使用提取的二元组和三元组。图 6 为提取的二元组和三元组信息。

③输入信息的组合

模型的最终输入为原句、二元组和三元组的组合,二元组和三元组是为了增加句子的长距离依赖信息,

三类信息都以词向量矩阵的形式输入模型中，如图 6 下部所示。

二元组	三元组
My dog	My dog likes
dog likes	dog likes sausage
also likes	dog also likes
likes sausage	also likes sausage
eating sausage	likes eating sausage

图 6 二元组和三元组

Fig. 6 Binary tuples and triple tuples

2)卷积层

模型卷积层结构示意图如图 6 的中上部所示，以下对三类输入的卷积过程分别介绍：

①原句

原句部分使用宽为 1、2、3(W_s 代表卷积核的宽度)的卷积核，且卷积步长设置为 1，分别由图 7 中左半部分红色、绿色和黄色框所示，主要用来提取句子短 N－gram 特征。

② 二元组和三元组

二元组是将原句中具有两两依赖关系的词语提取出来，所以这一部分只使用宽度为 2 的卷积核，如图 7 中间部分红色框所示，且卷积步长设置为 2(每 2 个二元组都是独立)，三元组只使用宽度为 3 的卷积核，如图 7 右部分黄色框所示，卷积步长设置为 3。

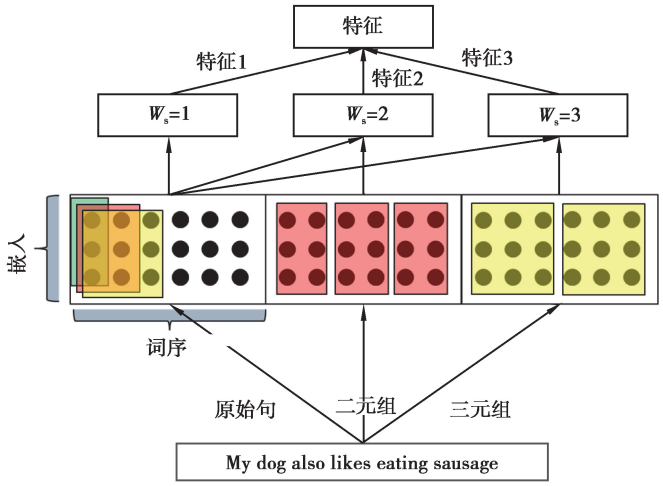

图 7 输入层及卷积层结构示意图

Fig. 7 Schematic diagram of input layer and convolution layer

通过实验发现，当原句中宽度为 2 和 3 的卷积核分别与二元组和三元组的卷积核进行参数共享得到的模型性能更佳，故如图 7 上半部分所示的共享参数模型，其中箭头所指是所使用卷积核的种类。

③特征组合

原句部分通过卷积核提取特征得到 Feature1，二元组通过卷积核提取特征得到 Feature2，三元组通过卷积核提取特征得到 Feature3，最终将得到的 3 部分特征进行组合得到最后的特征。

2)池化层

常用池化方式有最大池化、最小池化和平均池化。最大池化得到的数据主要用来判断 2 个句子的相似性，最小池化得到的数据主要用来判断 2 个句子的相异性，这两类信息组合起来更有利于判断 2 个句子之间

的相似性。平均池化则是将句子的信息进行了平滑操作,故而使得句子之间的差异性降低,不利于句子相似度的判断,所以只使用最大池化和最小池化 2 种方式。

3.2　相似度计算模块

相似度计算层使用了元素级别的特征比较方式,对网络提取的特征进行细粒度对应比较。

3.2.1　SUBMNL+NN

模型中句子表示的每一部分意义各不相同,简单使用诸如余弦相似度等距离公式来计算势必会影响效果。wang[26]等人提出了 6 种基于元素级别的特征比较框架,运算迅速便捷,通过在多个任务上测试,发现 SUBMULT+NN 的计算方法要优于其余 5 种,故而模型使用最优的 SUBMULT+NN 的计算方法来作为相似度比较的公式

$$t_j = f(\bar{a}_j, h_j) = ReLU\left(W\begin{bmatrix}(\bar{a}_j - h_j)\odot(\bar{a}_j - h_j)\\(\bar{a}_j \odot h_j)\end{bmatrix} + b\right), \tag{2}$$

其中:⊙表示元素点乘;$\bar{a}_j$表示第一个句子经过网络得到的特征表示;h_j表示表示第二个句子经过网络得到的特征表示;W 是全连接层的权重,其计算是通过卷积层来实现;b 为该层的偏置,上述特征比较框架使用了 ReLU 激活函数,还尝试了 Tanh 及 Sigmoid 等激活函数,在 MSRP 数据及上 Tanh 效果要优于 ReLU,因而最终使用如下的相似度计算公式

$$t_j = f(\bar{a}_j, h_j) = \mathrm{Tan}h\left(W\begin{bmatrix}(\bar{a}_j - h_j)\odot(\bar{a}_j - h_j)\\(\bar{a}_j \odot h_j)\end{bmatrix} + b\right)。 \tag{3}$$

3.2.2　全连接层和 Softmax 层

全连接层在模型中主要起到分类作用,实现将特征域转化到标签域的过程。Softmax 层将特征值转化为概率分布从而得到句子相似度的分类结果。

4　实验结果与数据分析

4.1　实验设计

4.1.1　数据集

数据集选用句子相似度计算常用的 3 个经典数据集:MSRP,MSRVID 和 SICK。表 1 为 3 个数据集的基本统计情况,包括数据集的构成,句子长度统计和正负样例的分布等信息,3 个数据集中 MSRP 数据集平均长度最长,其余 2 个数据集平均长度较短,数据集的平均句长影响模型加入句法信息的多少。

MSRP 数据集没有设立验证集,因此在训练数据集上采用 5 折交叉验证的方式。对于 MSRvid 数据集采用 MSRP 数据集上调优化的超参数,对于 SICK 数据集,模型使用全部的训练数据,在迭代训练过程中对验证数据集进行评测。

表 1　数据集介绍

Table 1　Introduction to dataset

数据集	MSRP			MSRvid			SICK			
	训练	测试	总体	训练	测试	总体	训练	验证	测试	总体
数据量	4 076	1 725	5 801	750	750	1 500	4 500	500	4 927	9 927
平均长度	20.95	20.78	20.90	6.70	6.63	6.67	9.68	9.89	9.66	9.68
最大长度	39	37	39	24	18	24	36	30	30	36
最小长度	6	7	6	3	3	3	3	4	3	3
正负样例比值	2.08	1.98	2.05	0.78	0.92	0.85	2.74	3.03	2.87	2.82

4.1.2 评价指标

MSRP数据集是来以0(不相似)和1(相似)表示2个句子是否相似的二分类任务,因此采用了ACC和F1作为评价指标。SICK数据集和MSRvid数据集分别是以[0,5]和[1,5]多级标签的多分类任务,因此采用了皮尔逊系数作为评价指标。如表2所示。

表2 评价指标

Table 2 Evaluating indicator

数据集	任务	标签级别	评价指标
MSRP	二分类	(0,1)	Acc/F1
MSRvid	多分类	[0,5]	Pearson
SICK	多分类	[1,5]	Pearson

4.1.3 超参数设置

超参数设置如表3所示。

表3 超参数设置

Table 3 Super parameter Settings

超参数	值
Learning rate	0.0001
Number_filters	500
Filter_size	1,2,3
Embedding_size	525
Dropout	0.5
L2	10.0
Hidden_numbers	250

4.1.4 损失函数设计

二分类问题使用的损失函数为交叉熵,多分类任务使用的损失函数为KL散度。对于样本(x,label)来讲,x为句子对$\{S_1,S_2\}$的集合,lable为样本x对应的标签集合,w为模型需要训练的参数,假设某个样本的真实标签为y_t,该样本$y_{\text{gold}}=1$的概率为y_p,则该样本的损失函数为如下

$$\text{Loss}_{\text{CrossEntropy}}=-\log(y_t*\log(y_p)+(1-y_t)\log(1-y_t))。\tag{4}$$

为了进一步防止模型过拟合,对模型中的参数进行L_2正则化,其中λ为正则化参数,正则化主要通过降低模型的复杂度来防止过拟合。故而模型最后的损失函数表达式为

$$\text{Loss}=-\log(y_t*\log(y_p)+(1-y_t)\log(1-y_t))+\frac{\lambda}{2}\|w\|_2^2。\tag{5}$$

SICK和MSRvid使用的损失函数为KL散度,其中λ为正则化参数,m为每次训练样例的个数,y_t是样本的真实标签,y_t预测结果,w为模型的权重。

$$\text{Loss}_{KL}=\frac{1}{m}\sum_{k=1}^{m}KL(y_t\parallel y_p)+\frac{\lambda}{2}\|w\|_2^2。\tag{6}$$

4.2 实验及结果分析

设计了4个实验以验证所提出DCNN模型的有效性。实验1和实验2对贡献1进行了证明,是实验3对贡献2和3进行了证明,实验4对分析了模型中参数的影响。

4.2.1　实验 1

本实验目的是对提出的 DCNN 模型与该领域中模型进行对比,表 4 显示了一些 CNN 用于相似度计算的模型在 3 个数据及上的实验结果。结果表明模型在句子相似度计算上能达到良好的效果。

根据表 1 可知,MSRP 数据集平均长度较长,需加入二元组和三元组合信息,而提取 SICK 和 MSRvid 数据集的三元组会导致数据稀疏,故而其只使用二元组信息。

表 4　实验结果

Table 4　Experimental results

Modle	MSRP (acc/F1)	SICK (Pearson)	MSRvid (Pearson)
Hu et al. (2014) ARC-I	69.6/80.3	—	—
Hu et al. (2014) ARC-II	69.9/80.9	—	—
Yin and Schütze.(2015)(without pretraining)	72.5/81.4	—	—
Yin and Schütze.(2015)(with pretraining)	78.1/84.4		
Yin and Schütze.(2015)(ABCNN)	78.9/84.8		
Wang et al.(2015)	78.4/84.7		
He et al.(2015)(MPCNN)	78.6/84.7	0.869	0.909
Conneau et al.(2018)(untrained)	73.2/81.6	0.860	—
This work	80.33/85.91	0.875	0.922

由表 4 可以看出,模型在加入了长距离依赖信息后,有效提升了模型计算相似度的准确性,且准确度比性能最好的 ABCNN 模型高出 1.43%,F1 值高出 1.11%。

- Hu 等人[27]提出的模型通常认为是卷积神经网络用于句子相似度任务的基线,包括 ARC-I 模型和 ARC-II 模型,Yin 等人[13]提出的 BCNN 模型和 ABCNN 模型[12]在 CNN 的基础上比较了不同粒度的句子特征,He 等人[15]提出的 MPCNN 模型使用了基于词语粒度和基于词向量粒度的两种卷积方式,从多个角度来提取句子的特征表示,但是以上方法都没有考虑句子的长距离依赖信息,且相似度计算单元都缺乏基于元素级别的比较。
- Wang 等人[14]提出的网络模型通过将句子对分解为相似相异信息来判断 2 个句子的语义相似程度。缺点还是没有考虑到句子的长距离依赖信息。
- Conneau[28]等人提出的模型是多任务学习的一种代表方式,通过在多个数据及上进行交叉训练来得到符合多个子任务的模型,这种模型的优点是通用性强,但是相比于针对特定任务指定的特定网络来说,这种多任务学习的模型还是逊色一些。

4.2.2　实验 2

本实验为消融实验,进一步明确二元组和三元组在解决长距离依赖中的作用,其中去除句法信息的模型为模型的基线模型,具体实验结果见表 5。

表 5　消融实验结果

Table 5　Experimental results of ablation components

模型	精确度	F1 值
模型(CNN+句法信息)	80.33	85.91
CNN (去除句法信息 base line)	77.04	84.09
去除二元组	78.03	84.37
去除三元组	78.51	84.65

- 由表5的实验结果可知，加入句法信息能有效提升模型对长距离依赖信息的提取，从而提升模型性能；
- 其中二元组对模型性能的贡献要大于三元组，这主要是因为二元组的词语组合相比三元组来说要多；

4.2.3 实验3

本实验目的是对利用依存句法和注意力机制解决长距离依赖的在句子相似度任务中的效果进行直接对比。在原句部分加入自注意力机制，主要对词语级别的特征起作用，其算法过程如表6。

表6 自注意力机制算法流程

Table 6 Algorithm flow of self-attention mechanism

输入：句子 S

①其中词向量维度为 d

②Ms、Mws 中下表为句子长度

输出：赋予注意力机制权重的结果

1：将句子 S 转化为词向量矩阵 $\boldsymbol{M}_s$

2：For i in (length(w_s))：

For j in (length(w_s))：

dis = euclidean_distance($w_s[i]$,$w_s[j]$)

3：End

4：Attention matrix $\boldsymbol{A}=1.0/(1.0+\text{dis})$

5：$\text{Weight\_row}=\sum_{i=1}^{\text{row-wise}}\text{Attentionmatrix}\ \boldsymbol{A}$

6：将权重 Weight_row 赋予矩阵 $\boldsymbol{M}w_s$

7：得到赋予注意力权重的矩阵 $\boldsymbol{A}_\boldsymbol{M}$ws

8：将 $\boldsymbol{A}_\boldsymbol{M}$ws 作为网络新的原句输入

- 由表7的可知，CNN模型中通过自注意力机制和依存句法结构来解决长距离依赖信息均能有效提升句子相似的计算性能；
- 注意力机制是通过直接计算句子中词语之间的语义相似度来获得长距离依赖信息，但句子中语义相近的词在结构上不一定具有依赖关系，因此通过依存句法可以从句法结构上更加准确地获取长距离依赖关系，实验结果表明基于依存句法在句子相似度计算任务中更优于基于注意力机制的方法。
- 模型中同时加入自注意力机制和句法信息得到的实验效果最好，说明两者是从不同的方面来获取的长距离依赖信息。即融合句法信息和自注意力机制能使模型性能更佳。

表7 基于句法结构和基于注意力机制对比实验

Table 7 Self-attention mechanism comparison experiment

模型	精确度	F1值
CNN(base line)	77.04	84.09
CNN+attention	78.86	85.02
CNN+依存句法	80.33	85.91
CNN+attention+依存句法	80.61	86.04

4.2.4 实验4

为了进一步分析模型中各个参数的影响，模型在MSRP数据集的基础上对实验模型中的所有参数进行控制变量分析，研究模型中单一变量对模型性能的影响。

- Glove.6B词向量有50维到300维4种词向量，由图8可知，在词向量维度为200时候实验结果表现最好，当词向量由50维增加到100维再到200维时，模型性能逐步提升，而当词向量由100维提升到200维时，模型的性能只有微弱的提升，维度增加到300维时，模型性能反而下降，因为词向量的维度

代表了词语的特征,特征越多越能准确的将词与词进行区分,维度过大反而会导致词与词之间关系弱化,故而词向量的维度选择很依赖应用场景。

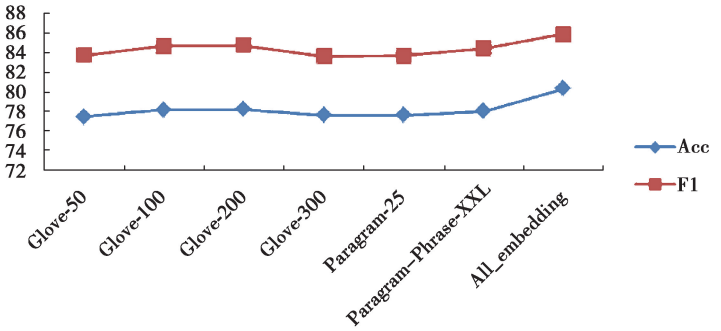

图 8　单一词向量对比实验

Fig. 8　Single word vector contrast experiment

- 模型训练中使用 Dropout 技术是非常必要的,使目前防止模型过拟合的主流方式,模型使用 tensorflow 框架,其中 Dropout 的比率为模型中保留节点的百分比,由图 9 可知将 Dropout 的比率设置为 0.5 时是模型效果最好的,这是因为 Dropout 为 0.5 时产生的模型种类是最多的。由实验数据可以看出,Dropout 的值对于模型性能的影响也并非以 0.5 为中心呈对称状,还应当与隐含层节点个数等其他参数有关系。

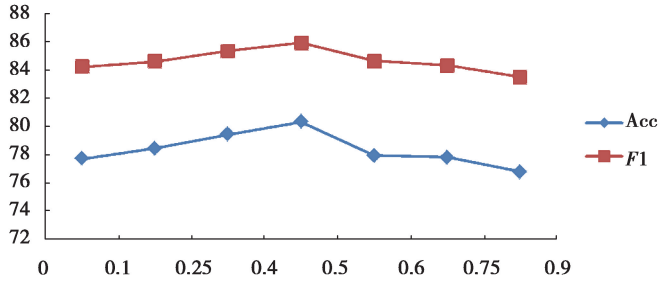

图 9　Dropout 对比实验

Fig. 9　Dropout comparison experiment

图 10 的实验结果表明,随着卷积核数量的增加,模型性能会逐步提升,最后逐渐趋于平稳,但在增大卷积核数目的同时也会导致模型的训练时间和内存开销变大,故而卷积窗口数量的设置要综合考虑性能和开销的,将卷积核窗口设置为 500。

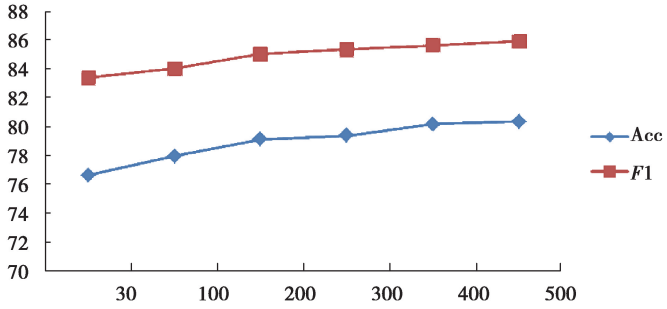

图 10　卷积核数目对比实验

Fig. 10　Convolution kernel number comparison experiment

5 结 语

卷积神经网络在提取句子特征时通常是将固定大小的卷积窗口作用于词向量矩阵,以此来提取顺序的局部特征,而后通过后续的池化操作整合成全句特征,这个过程存在的问题是卷积核只是对顺序且相邻的词语组合进行了特征提取,而忽略了互不相邻但是有强语义关联的词语组合,因而研究提出了在卷积神经网络中加入依存句法信息的方式来解决上述问题,实验结果表明,加入结构化的特征能让句子相似度的计算更加准确,改进的方法在一定程度上提升了相似度计算的准确度,如何在特征中加入句子的结构信息是下一步研究的重点。

参考文献:

[1] Wan S, Dras M, Dale R, et al. Using dependency-based features to take the "Para-farce" out of paraphrase[J]. Proceedings of the Australasian Language Technology Workshop (ALTW 2006), 2006(2005): 131-138.

[2] Hassan S. Measuring semantic relatedness using salient encyclopedic concepts[M]. Denton, USA: University of North Texas, 2011.

[3] Dipanjan D, Noah A S. Paraphrase identification as probabilistic quasi-synchronous recognition[C]// ACL '09: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. New York, USA: ACM Press, 2019(1): 468-476.

[4] 赵谦, 荆琪, 李爱萍, 等. 一种基于语义与句法结构的短文本相似度计算方法[J]. 计算机工程与科学, 2018, 40(7): 1287-1294.
ZHAO Qian, JING Qi, LI Aiping, et al. A short text similarity calculation method based on semantics and syntax structure[J]. Computer Engineering & Science, 2018, 40(7): 1287-1294. (in Chinese)

[5] 李妍, 刘茂福, 姬东鸿. 基于支持向量机的中文文本蕴涵识别研究[J].计算机应用与软件, 2014, 31(4): 51-55.
LI Yan, LIU Maofu, JI Donghong. On SVM-based Chinese textual entailment recognition[J]. Computer Applications and Software, 2014, 31(4): 51-55. (in Chinese)

[6] Cranias L, Papageorgiou H, Piperidis S. A matching technique in example-based machine translation[C]//COLING '94: Proceedings of the 15th conference on Computational linguistics. Morristown, NJ, USA: Association for Computational Linguistics, 1994(1): 100-104.

[7] 葛斌, 李芳芳, 郭丝路, 等. 基于知网的词汇语义相似度计算方法研究[J]. 计算机应用研究, 2010, 27(9): 3329-3333.
GE bin, LI Fangfang, GUO Silu, et al. Word's semantic similarity computation method based on Hownet[J]. Application Research of Computers, 2010, 27(9): 3329-3333. (in Chinese)

[8] 李彬, 刘挺, 秦兵, 等. 基于语义依存的汉语句子相似度计算[J]. 计算机应用研究, 2003, 20(12): 15-17.
LI Bin, LIU Ting, QIN Bin, et al. Chinese sentence similarity computing based on semantic dependency relationship analysis[J]. Application Research of Computers, 2003, 20(12): 15-17. (in Chinese)

[9] 吴全娥, 熊海灵. 一种综合多特征的句子相似度计算方法[J]. 计算机系统应用, 2010, 19(11): 110-114.
WU Quane, XIONG Hailing. Method for sentence similarity computation by integrating multi-features[J]. Application Research of Computers, 2010, 19(11): 110-114. (in Chinese)

[10] Kim Y. Convolutional neural networks for sentence classification[EB/OL]. 2014: arXiv:1408.5882[cs.CL]. https://arxiv.org/abs/1408.5882

[11] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences[EB/OL]. 2014: arXiv:1404.2188[cs.CL]. https://arxiv.org/abs/1404.2188

[12] Yin W P, Schütze H, Xiang B, et al. ABCNN: attention-based convolutional neural network for modeling sentence pairs[J]. Transactions of the Association for Computational Linguistics, 2016, 4: 259-272.

[13] Yin W P, Schütze H. Convolutional neural network for paraphrase identification[C]// Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Stroudsburg, PA, USA: Association for Computational Linguistics, 2015: 901-911.

[14] Zhiguo W, Mi H, Ittycheriah A.Sentence similarity learning by lexical decomposition and composition[EB/OL]. 2016: arXi v:1602.07019 [cs.CL]. https://arxiv.org/abs/1602.07019

[15] He H, Gimpel K, Lin J. Multi-perspective sentence similarity modeling with convolutional neural networks[C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA:

Association for Computational Linguistics, 2015: 1576-1586.

[16] Socher R, Huang E H, Pennington J, et al. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection[J]. Advances in Neural Information Processing Systems, 2011, 24: 801-809.

[17] Hermann K M, Blunsom P. The role of syntax in vector space models of compositional semantics[C]// Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. Sofia, Bulgaria, USA: Association for Computational Linguistics, 2013: 894-904.

[18] Tai K S, Socher R, Manning C. Improved semantic representations from tree-structured long short-term memory networks[EB/OL]. 2015: arXiv:1503.00075[cs.CL]. https://arxiv.org/abs/1503.00075

[19] 杨萌, 李培峰, 朱巧明. 一种基于 Tree-LSTM 的句子相似度计算方法[J]. 北京大学学报(自然科学版), 2018, 54(3): 481-486.
YANG Meng, LI Peifeng, ZHU Qiaoming. An approach of sentence similarity on tree-LSTM[J]. Acta Scientiarum Naturalium Universitatis Pekinensis,2018, 54(3): 481-486. (in Chinese)

[20] Lai S, Xu L, Liu K, et al. Recurrent convolutional neural networks for text classification[C/OL]. 2019 International Joint Conference on Neural Networks (IJCNN). Piscataway, NJ: IEEE, 2019(2019-09-30)[2020-05-25]. https://doi.org/10.1109/IJCNN.2019.8852406

[21] Strubell E, Verga P, Belanger D, et al. Fast and accurate entity recognition with iterated dilated convolutions[EB/OL]. 2017: arXiv:1702.02098[cs.CL]. https://arxiv.org/abs/1702.02098

[22] Mou L L, Li G, Jin Z, et al. TBCNN: a tree-based convolutional neural network for programming language processing [EB/OL]. 2014: arXiv:1409.5718[cs.LG]. https://arxiv.org/abs/1409.5718

[23] Ma M B, Huang L, Xiang B, et al. Dependency-based convolutional neural networks for sentence embedding[EB/OL]. 2015: arXiv:1507.01839[cs.CL]. https://arxiv.org/abs/1507.01839

[24] Wieting J, Bansal M, Gimpel K, et al. From paraphrase database to compositional paraphrase model and back[J]. Transactions of the Association for Computational Linguistics, 2015, 3: 345-358.

[25] Wieting J, Bansal M, Gimpel K, et al. Towards universal paraphrastic sentence embeddings[EB/OL]. 2015: arXiv:1511.08198[cs.CL]. https://arxiv.org/abs/1511.08198

[26] Wang S H, Jiang J. A compare-aggregate model for matching text sequences[EB/OL]. 2016: arXiv:1611.01747[cs.CL]. https://arxiv.org/abs/1611.01747

[27] Hu B T, Lu Z D, Li H, et al. Convolutional neural network architectures for matching natural language sentences[J]. Neural Information Processing Systems,2014, 3: 2042-2050.

[28] Zhang Y, Yang Q. An overview of multi-task learning[J]. National Science Review, 2018, 5(1): 30-43.

(编辑　侯　湘)