

基于字向量和 BiLSTM-CNN 的文本相似度计算方法

宋英, 张弛

(中国传媒大学计算机与网络空间安全学院)

摘要: 文本相似度在自然语言处理中有着重要的作用, 随着对深度学习的研究, 传统机器学习中出现的文本语义被忽略、人工获取文本特征时间长成本高等问题, 都能很好地加以处理, 然而在深层语义方面研究效果不是很理想。针对该问题, 本文提出了一种基于字向量和 BiLSTM-CNN 的文本相似度计算模型。首先利用 word2vec 模型对文本进行训练, 获得字向量集合; 其次将文本通过字向量集合进行向量化表示, 并输入到 BiLSTM-CNN 模型中, 通过 Attention 进行拼接获得文本语义向量; 最后采用 softmax 函数计算文本相似度。在 Chinese STS 数据集上进行验证和比较, 最终表明该方法比其他模型的准确率更高。

关键词: 文本相似度; 字向量; BiLSTM-CNN 模型

中图分类号: TP39

Text Similarity Calculation Method Based on Word Vector and BiLSTM-CNN

SONG Ying, ZHANG Chi

(School of Computer and Cyber Sciences, Communication University of China)

Abstract: Text similarity plays an important role in natural language processing. With the research of deep learning, the problems of text semantics being ignored in traditional machine learning, long time and high cost of manual acquisition of text features can be well dealt with. However, the effect of research on deep semantics is not very satisfactory. Aiming at this problem, this paper proposes a text similarity calculation model based on word vector and BiLSTM-CNN. First, the word2vec model is used to train the text to obtain the word vector set; secondly, the text is vectorized through the word vector set, and input into the BiLSTM-CNN model, and the text semantic vector is obtained by splicing through Attention; finally, the softmax function is used to calculate the text similarity. Validation and comparison on the Chinese STS dataset finally show that the method is more accurate than other models.

Key words: text similarity; word vector; BiLSTM-CNN model

0 引言

随着信息技术的发展, 文本相似度的用途越来越广泛, 例如查重系统^[1]、新闻推荐^[2]、网页去重^[3]、问答系统^[4]、商品召回^[5]等方面都有使用。通过对文本相似度的研究, 将其划分成了基于关键词匹配、向量空间和深度学习的三种计算方法。

基于关键词匹配的相似度计算方法仅仅适用于只考虑词形的文本相似度计算。该方法通常使用最长公共子串、N-gram 相似度、Jaccard 相似度以及编辑距离等算法实现文本相似度计算。俞婷婷等^[6]首先通过使用 N-gram 模型获取 k 个词语在文本中的数量和这些词的权重

作者简介: 宋英 (1995-), 女, 硕士研究生, 主要研究方向: 文本相似度

通信联系人: 张弛 (1978-), 男, 副教授, 主要研究方向: 智能媒体信息处理. E-mail: zhangchi@cuc.edu.cn

值，再根据 Jaccard 距离对两个文本进行相似度计算。李圣文等^[7]根据文中公共字符串的信息熵进行相似度计算。基于向量空间的相似度计算方法是先将文本映射添加到空间向量中，然后根据余弦相似度等一系列算法计算文本相似度。陈宏朝等^[8]使用《同义词词林》通过对路径和深度的计算从而获得词语的相似度。甘秋云^[9]通过研究向量空间模型（VSM）和 TF-IDF 词频统计算法提出了一种结合 TF-IDF 和向量空间模型的文本相似度算法。基于深度学习的的方法主要用于文本语义相似度计算，并且随着深度学习的发展，结合词向量的深度学习使用的较多。该方法使得语义相似度计算逐渐由人工设计特征向分布式表达和神经网络结构相结合的方式转换。如 DSSM^[10]、ConvNet^[11]、Tree-LSTM^[12]、Siamese LSTM^[13]等深度学习方法全部是利用词语或句子进行建模从而获得对应的向量，再通过适合该算法的距离公式计算文本的相似程度。

本文提出了一种基于字向量和 BiLSTM-CNN 的文本相似度计算模型。首先采用 word2vec 模型对两个文本进行字向量化，然后将结果作为 BiLSTM 和 CNN 的输入，分别提取特征值。最后将输出结果通过 Attention 进行拼接，采用 softmax 函数得到两个文本的相似度。

1 模型原理

1.1 字向量

由于神经网络计算过程为矩阵向量化运算，因此需要将语料进行向量化表示，即进行字或词嵌入（Embedding），该过程可以使用词向量，也可以使用字向量。词向量是 Bengio^[14]在 2003 年提出的，用来表达文本的深层含义。张春英等^[15]通过 word2vec 训练词向量，用于 CNN 相似度计算中。郭浩等^[16]在基于 CNN 和 BiLSTM 的短文本相似度计算中采用了词向量嵌入的方法，提高了相似度计算的准确性。相对于词向量，字作为文本的最小单位，更能表达文本语义，尤其是短文本，能够获得更全面的文本内容。何颖刚等^[17]提出了基于字向量和 LSTM 的句子相似度计算方法，并与词向量模型进行比较，实验结果表明字向量模型优于词向量模型，并且提高了相似度计算的准确率。

1.2 长短期记忆网络 LSTM

LSTM（Long-Short-Term Memory）是一种特殊的 RNN 模型，可以学习长期依赖信息，该模型主要包括输入门、遗忘门和输出门三个模块，LSTM 的总体框架如图 1 所示。

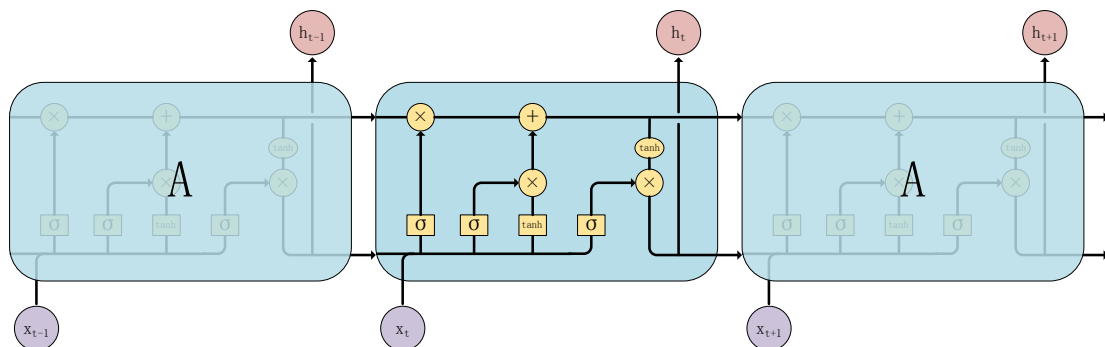


图 1 LSTM 总体框架

Fig. 1 LSTM overall framework

(1) 细胞状态：细胞状态可以认为是信息传送带，只有极少的线性交互，用以保证信息不变。如下图中 C_{t-1} - C_t 为细胞状态。

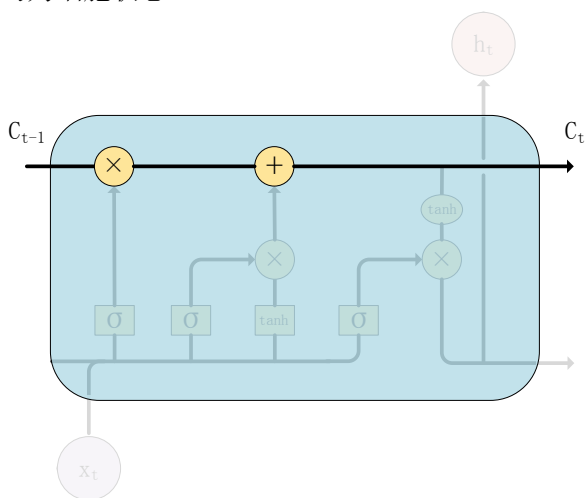


图 2 细胞状态

Fig. 2 Cell state

(2) 输入门：输入门用来判断是否将信息输入到细胞状态，主要包含两部分：sigmoid 神经网络层和 \tanh 激活函数。sigmoid 确定输入的信号控制， \tanh 激活函数确定输入的文本内容。如图 3 所示，上一层输出的 h_{t-1} 连接该层的 x_t ，经过 sigmoid 输入门生成一个数值 i_t (0-1 之间)，同时，在 \tanh 上进行非线性转换获得 \tilde{C}_t ，最后将数值 i_t 和 \tilde{C}_t 做乘法运算，决定输入到细胞状态的信息。以下是 i_t 和 \tilde{C}_t 的计算公式：

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

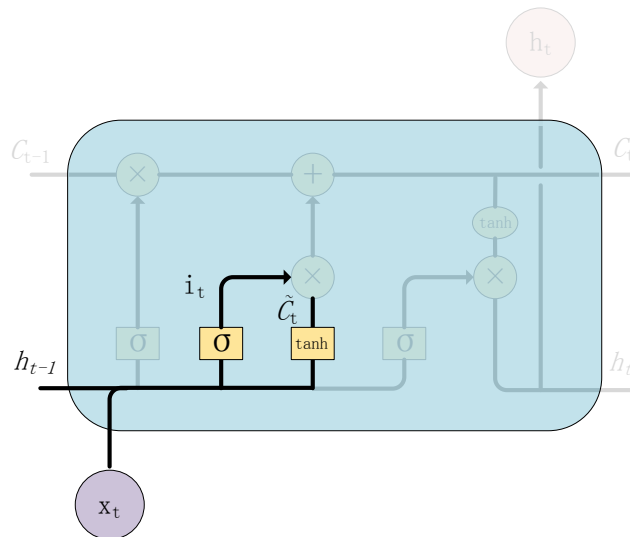


图3 输入门

Fig. 3 Input gate

(3) 遗忘门：遗忘门是由 Gers 提出的，用来决定细胞状态中信息是否可以传送。如图 4 所示，上一层输出的 h_{t-1} 连接该层的 x_t ，经过 sigmoid 遗忘门生成数值 f_t (0-1 之间)，最后将数值 f_t 和细胞状态 C_{t-1} 做乘法运算，决定细胞状态是否继续传送。以下是 i_t 和 C_{t-1} 的计算公式：

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

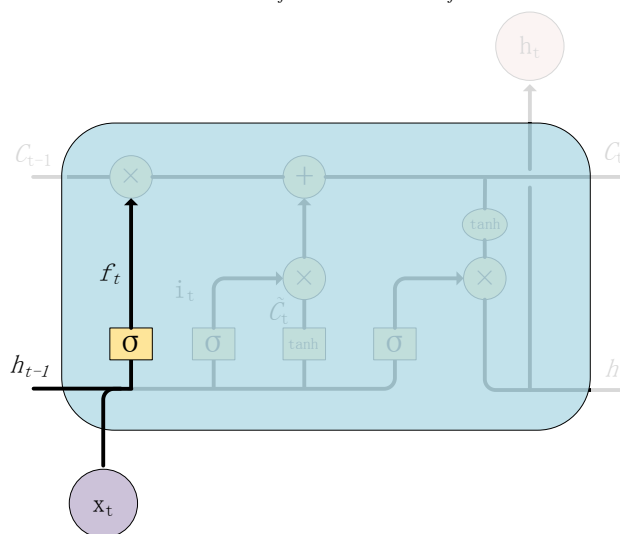


图4 遗忘门

Fig. 4 Forgotten gate

(4) 输出门：输出门是将信息从细胞状态中输出，同样包含 sigmoid 神经网络层和 tanh 激活函数。sigmoid 神经网络层决定输出的信号控制，tanh 激活函数决定输出的文本内容。上一层输出的 h_{t-1} 连接该层的 x_t ，经过 sigmoid 输出门生成数值 o_t (0-1 之间)，同时，在 tanh 上进行非线性转换，最后将转换结果和数值 o_t 做乘法运算，决定可以从细胞状态输出的信息，并连接到下一个细胞。以下是 o_t 和 h_t 的计算公式：

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

100

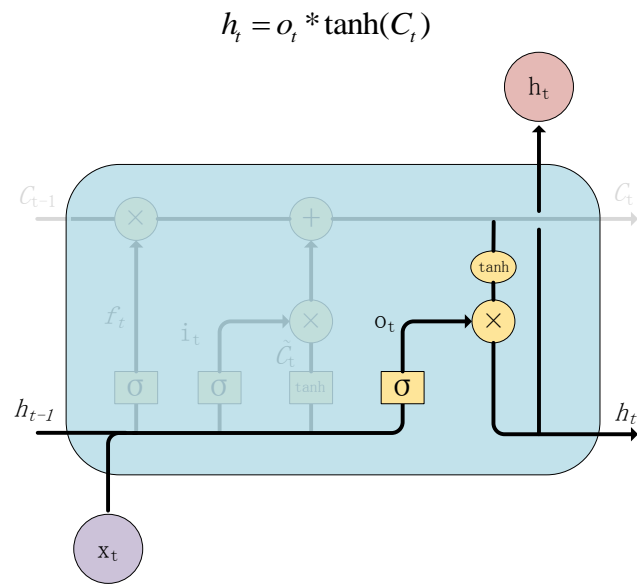


图 5 输出门
Fig. 5 Output gate

1.3 卷积神经网络 CNN

105

卷积神经网络（CNN）类似于其他神经网络，它们都拥有若干个神经元。其中包括能学习的权重和偏置值，而且任意一个神经元都能收到输入值并进行卷积运算，经过神经网络后输出则是所有分类的分值。与其他神经网络不同的是卷积神经网络通常是很多层构成的，可以进行特征提取，这些层中的输入输出都是三维的数据，而且其中有的需要加入参数进行训练，有的则不用。卷积神经网络一般包括卷积层、Dropout 层以及池化层。CNN 模型如下图 6 所示。

110

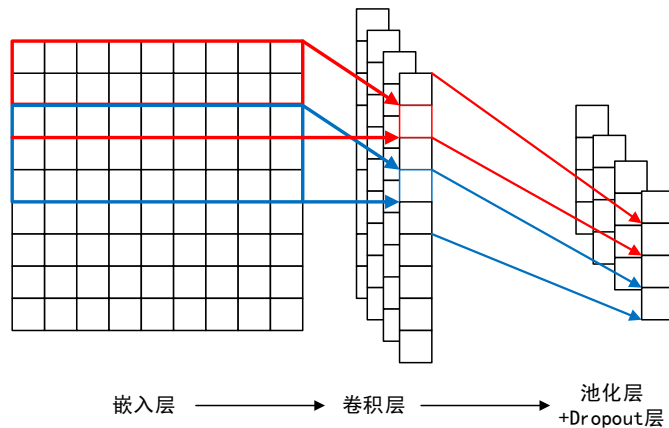


图 6 CNN 模型
Fig. 6 CNN model

115

在 CNN 模型中卷积层的重心是特征提取，池化层和卷积层一般情况下是相连接的，这两层的主要任务是信息聚合及特征降维。通常池化层使用的方法有三个：最大值池化、平均池化以及概率池化。最大值池化是将特征值最大的当成输出值，而平均池化则是将所有特征值的平均数当成输出值。以上这两个池化方法都有弊端：最大值池化方法通常会导致过拟合，

而平均池化则会导致信息削弱。第三个方法概率池化介于以上两个方法之间，对全部数据根据数的大小分别加上概率，然后根据概率进行亚采样操作，获取窗口的特征值。

120 深度学习模型有很多优点，然而如果实验数据太少，就会导致过拟合的情况发生。为有效的解决这一问题就要通过 Dropout 层，该层的主要作用是利用随机删除输入层神经元的方法进行神经元连接更新。

2 模型结构

125 本文提出的相似度计算模型主要通过字向量进行文本向量化，利用两个神经网络模型进行训练，该模型的结构如图 7 所示，其中包括以下几个步骤：

(1) 输入两个文本，通过字向量将文本转换成向量矩阵。

(2) 将两个文本的向量矩阵分别输入 BiLSTM 和 CNN 模型中进行特征提取，获得文本的深层语义。

130 (3) 通过 Attention 注意力机制，将 BiLSTM 和 CNN 模型训练的结果进行拼接，以此获得两个文本的语义向量。

(4) 采用 softmax 函数进行计算，从而获得两个文本的相似度。

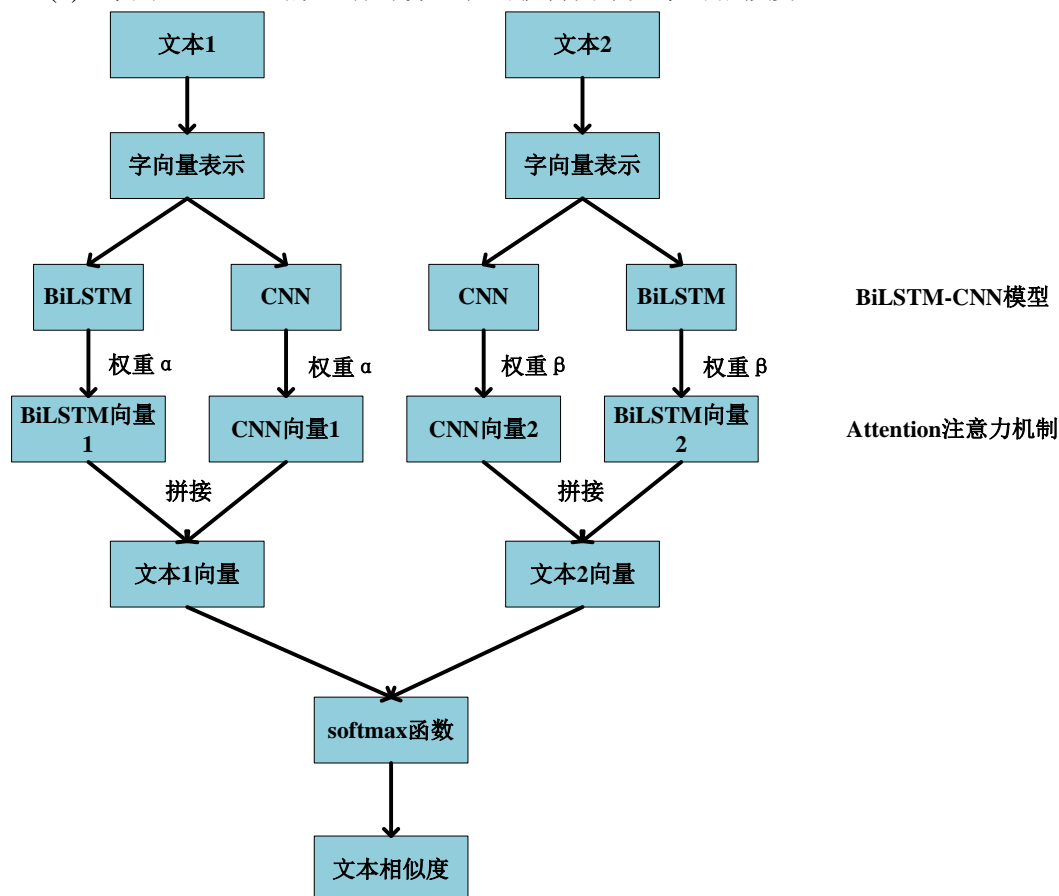


图 7 基于字向量和 BiLSTM-CNN 的短文本相似度计算架构图

Fig. 7 Architecture diagram of short text similarity calculation based on word vector and BiLSTM-CNN

135

2.1 字向量表示

字向量表示是用数字表示文本的处理过程，该过程主要进行文本数据清洗、字编码、向量化三个步骤，过程如图 8 所示。

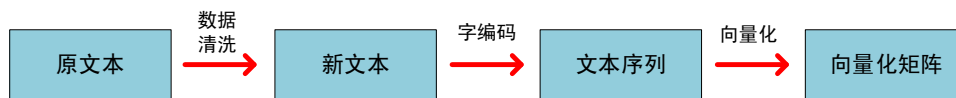


图 8 文本向量化

Fig. 8 Text vectorization

140

(1) 一般文本数据通常是冗余的、不完整、不一致、甚至有很多“噪音”，因此进行数据清洗是不可缺少的步骤。数据清洗主要是去除与文本内容无关的乱码和停用词，否则会影响实验效果。本文采用停用词表去停用词，之后利用 Beautiful Soup 库进行数据清洗。

(2) 将清洗后的数据进行字编码，每个字都有唯一的编码。根据字编码生成文本向量，该向量为固定维度，不足的地方进行补零操作。

145

(3) 对上一步中的文本向量利用 word2vec 模型进行向量化表示，使得文本转化为向量化矩阵，以此作为神经网络的输入层（Embedding）。

2.2 BiLSTM-CNN 特征提取

本文中提到的神经网络模型 BiLSTM-CNN 融合了双向长短期记忆网络（BiLSTM）和卷积神经网络（CNN），将文本字向量化后分别输入到这两个网络中进行特征提取。BiLSTM 模型对输入的文本数据进行学习，其中正向的 LSTM 学习文本数据的历史信息，反向的 LSTM 学习文本数据的未来信息，然后将正反方向学到的信息进行拼接作为模型的输出。CNN 模型通过卷积层和池化层对文本数据进行特征提取，获得文本语义信息。

150

BiLSTM 模型即双向 LSTM，它是由两个单独的 LSTM 组合构成，如图 9 是一层的 BiLSTM 模型。

155

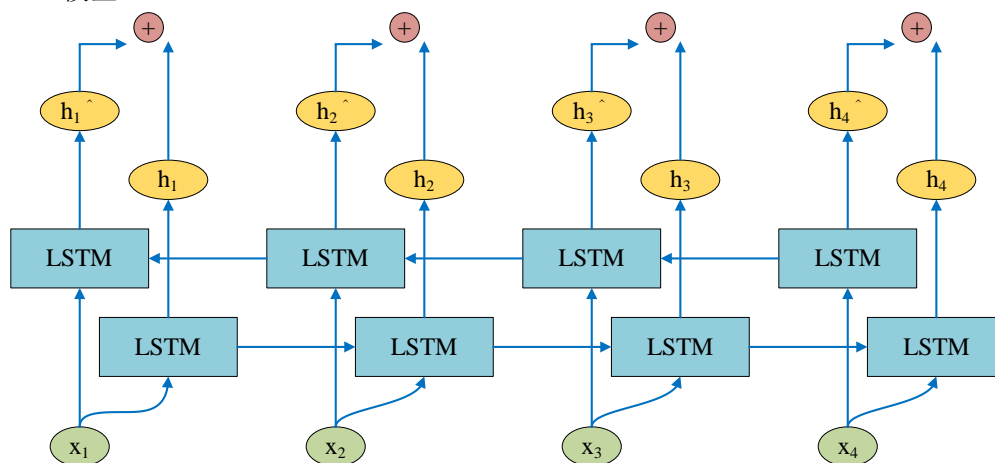


图 9 一层的 BiLSTM 模型

Fig. 9 One-layer BiLSTM model

本文采用 BiLSTM 对文本进行特征提取能够学习文本历史和未来的信息，从而获得的

上下文信息比较全面，BiLSTM 的输出公式为：

$$h(t)=[o_{pre},o_{post}]$$

其中 $h(t)$ 为正反方向学习到的信息进行拼接的向量； o_{pre} 为正向的 LSTM 学习文本数据的历史信息向量； o_{post} 为反向的 LSTM 学习文本数据的历史信息向量。

在 CNN 模型中，一般卷积层和池化层是相连的，卷积层通过不断改变卷积核矩阵，获取不同的特征值。也就是初始化卷积核的矩阵值以后，通过梯度下降来降低损失值，从而获得更优的权重参数，该过程是自动进行调整的。CNN 引入激活函数，用于卷积层和全连接层中，使得网络表达能力更强。池化层与卷积层类似，同样有一个固定窗口，通过改变窗口大小进行输出，不同的是池化层没有参数。该层的主要作用是降低卷积层对位置信息的敏感度，避免因位置改变导致特征提取受到较大影响，通过对数据降维，加快运算速度，防止出现过拟合现象。

2.3 相似度计算

在 LSTM 和 CNN 隐藏层中引入加权的注意力机制 Attention，进行加权平均计算，输入到 softmax 函数中获得文本相似度。BiLSTM 网络模型输出向量设为 $h[h_1, h_2, \dots, h_n]$ ，其加权 Attention 为 $\alpha[\alpha_1, \alpha_2, \dots, \alpha_n]$ ，对输出向量进行加权获得文本语义向量 S_1 ，同样 CNN 网络模型输出向量设为 $c[c_1, c_2, \dots, c_n]$ ，其加权 Attention 为 $\beta[\beta_1, \beta_2, \dots, \beta_n]$ ，对输出向量进行加权获得文本语义向量 S_2 。将加权后的向量进行拼接，获得向量 $[S_1, S_2]$ ，输入到 softmax 函数中获得相似程度，具体公式如下：

$$S = \text{softmax}\left(\frac{S_1 S_2}{\|S_1\| \|S_2\|}\right)$$

3 实验

3.1 数据集与评价指标

3.1.1 数据集

Chinese STS 数据集：Chinese STS 数据集是中文文本，该数据集格式是两个文本对应一个标签，标签值由 0 到 5 的整数，数字越大，文本的相似程度就越高。该数据集有 100000 个句子对，本文实验中将 80% 的数据集作为训练集，20% 作为测试集。

3.1.2 评价指标

通常使用评价标准来评估一个算法的性能，其中普遍被使用的评价标准包括以下几个：精确度（Precision）、召回率（Recall）以及 F1 值（F1-Measure）。针对样本真正的类别和

算法预测的类别，有下面几个定义：

TP (True Positive)：预测是正，实际也是正的样本。

190 TN (True Negative)：预测是负，实际也是负的样本。

FP (False Positive)：预测是正，实际是负的样本。

FN (False Negative)：预测是负，实际是正的样本。

可以用如下表 1 混淆矩阵来描述：

表 1 混淆矩阵

195

Tab. 1 Confusion matrix

	实际是正	实际是负
预测是正	TP	FP
预测是负	FN	TN

精确度 (Precision) 指的是算法预测是正例的样本中实际是正例的概率，其公式如下：

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

召回率 (Recall) 代表的是算法预测结果正确的正例样本在所有正例样本中的概率，其公式如下：

200

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

一般情况下，精确度和召回率之间是成反比的，也就是其中一个评价标准有所提高就会引起另一个评价标准的降低。F 值 (F-Measure) 是将精确度 P 和召回率 R 进行了结合，通过两者的调和平均进行算法评价，其公式如下：

$$F_{\beta} = \frac{(\beta^2 + 1)P * R}{\beta^2 * (P + R)}$$

205

其中， β 是用来调节查准率和查全率权重的参数，一般情况下 β 值取 1，取 1 时的 F1 值计算公式如下所示：

$$F_1 = \frac{2PR}{P + R}$$

3.2 参数设置

210

设置句子长度，神经网络模型对输入的矩阵的长度有规定，但是实际情况中的文本长度各不相同，所以必须要规定句子的长度。如果输入的句子太长，那么就要删掉句子中大于规定长度的内容；相反，如果输入句子太短，就要对句子缺少的部分补 0，从而达到规定的长度。训练数据经过预处理后，选定句子长度为 180。

初始化 Embedding 层，Embedding 层在初始化时使用的是字向量来提高实验结果的精确度，该实验使用的字向量是通过 word2vec 模型训练得到的。

215

设置参数，神经网络模型对数据进行训练时，需要设置激活函数、优化函数等，并对卷

积核、池化窗口等参数设定初始值。对于该实验中的卷积神经网络，通过多次的数据训练，从而得到了最优参数值。表 2 为该模型的参数设置。

表 2 参数设置

Tab. 2 Parameter settings

模型参数	数值
字向量维度	300
BiLSTM 状态维度	200
学习速率	0.001
卷积窗口大小	3
步长	1
池化窗口	3
dropout	0.2
batch_size	128
epoch	50
优化函数	Adam
激活函数	Relu

3.3 实验结果及分析

为了验证字向量比词向量更能表示文本语义，设计了针对字向量和词向量的对比实验。选取了 word2vec、CNN、LSTM 和本文所提出的模型在相同数据集上进行实验，这里为了方便对比，将字向量用 CV 表示，词向量用 WV 表示，并采用精确度作为评价标准，实验结果如表 3 所示。

表 3 不同向量表示模型的实验结果

Tab. 3 Experimental results of different vector representation models

模型	Precision
WV-word2vec	0.513
CV-word2vec	0.557
WV-CNN	0.785
CV-CNN	0.891
WV-LSTM	0.816
CV-LSTM	0.905
WV-BiLSTM-CNN	0.923
CV-BiLSTM-CNN	0.987

实验对比结果如图 10 所示，从图中可以看出字向量比词向量在模型训练中效果更好，说明对文本分词会丢失部分语义信息或分词不正确，都会导致相似度计算结果不准确。字向量虽然在不同文本中语义不同，影响计算结果，但比词向量造成的影响要小，说明字向量更加准确。同时，对比模型可以发现本文提取的模型要优于其他模型，表明该方法的有效性。

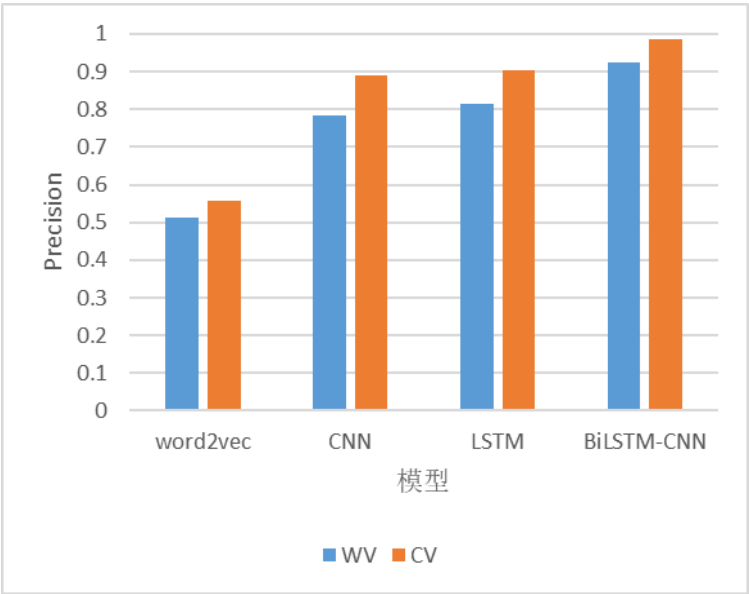


图 10 不同向量表示模型对比结果

Fig. 10 Different vector representation model comparison results

针对 CNN 和 LSTM 网络的结合情况，采用文献^[18]中所提到的模型进行对比，将字向量
235 作为所有待比对网络模型的输入，在相同的数据集上进行训练，利用精确度、召回率和 F1
进行模型评价。以下是所要对比的模型。

- (1) LSTM-CNN-serial: 将字向量输入到 LSTM 网络中，进行特征提取，将输出结果作为 CNN 网络的输入，再次进行特征提取并对输出结果进行相似度计算。
- (2) BiLSTM-CNN-serial: 将字向量输入到双向 LSTM 网络中，进行特征提取，将输出
240 结果作为 CNN 网络的输入，再次进行特征提取并对输出结果进行相似度计算。
- (3) LSTM-CNN-parallel: 将字向量分别输入到 LSTM 和 CNN 网络模型中进行特征提取，将两者的输出结果进行拼接后计算相似度。
- (4) BiLSTM-CNN-parallel: 将字向量分别输入到双向 LSTM 和 CNN 网络模型中进行
245 特征提取，将两者的输出结果进行拼接后计算相似度。

对以上四个模型进行实验对比，实验结果如表 4 所示。

表 4 不同模型的对比结果

Tab. 4 Comparison results of different models

模型	Precision	Recall	F1
LSTM-CNN-serial	0.845	0.765	0.827
BiLSTM-CNN-serial	0.896	0.749	0.885
LSTM-CNN-parallel	0.927	0.876	0.932
BiLSTM-CNN-parallel	0.985	0.892	0.978

上表中的实验结果可以看出，CNN 和 LSTM 网络模型的 serial 结合方式优于 parallel 结合方式，这表明两个网络模型分别进行特征提取要比一个网络提取特征后输入到另一个网络
250 要更加精确。同时双层的 LSTM 比单层的 LSTM 获得的特征值更加丰富。

为了更加清晰地分析这四个模型，本文根据网络的迭代次数进行实验，在相同数据集上进行模型评价，进一步证明本文所提方法的有效性。实验结果如图 11-图 13。

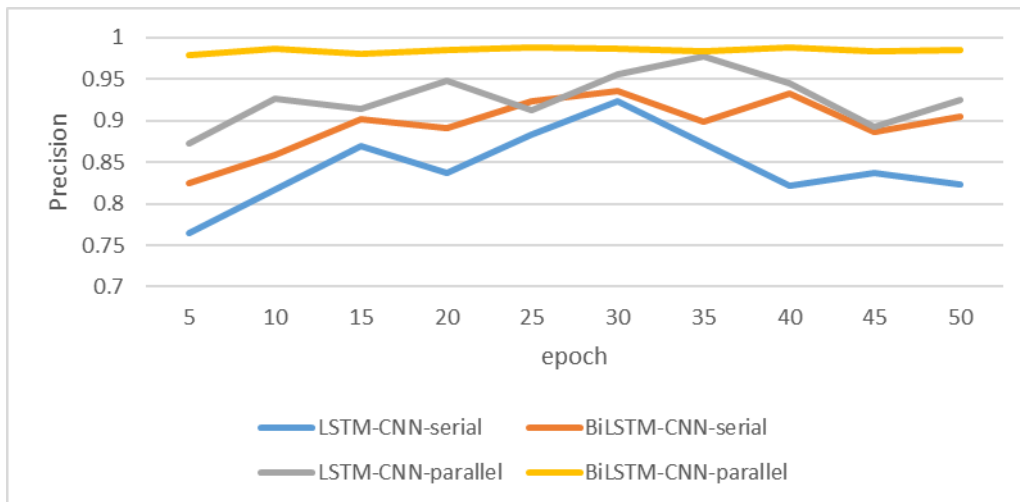


图 11 不同模型的精确度对比结果

Fig. 11 Accuracy comparison results of different models

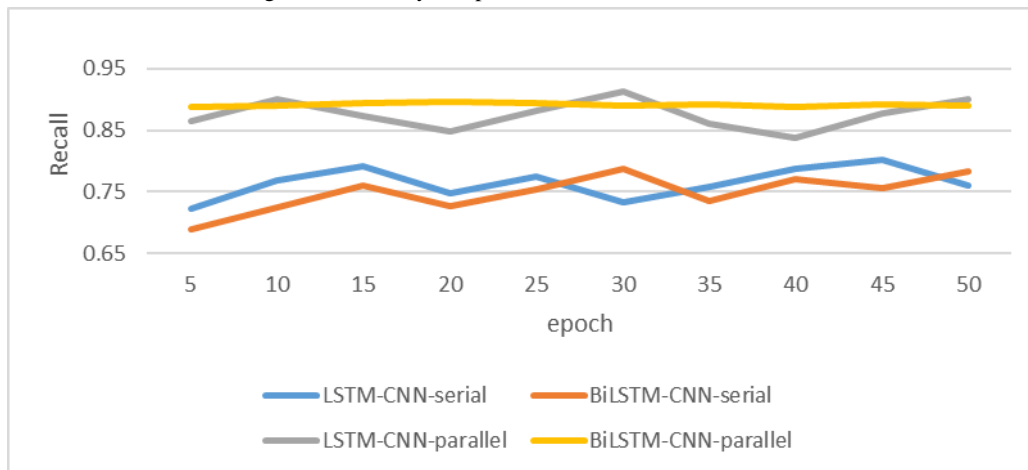


图 12 不同模型的召回率对比结果

Fig. 12 Comparison of recall rates of different models

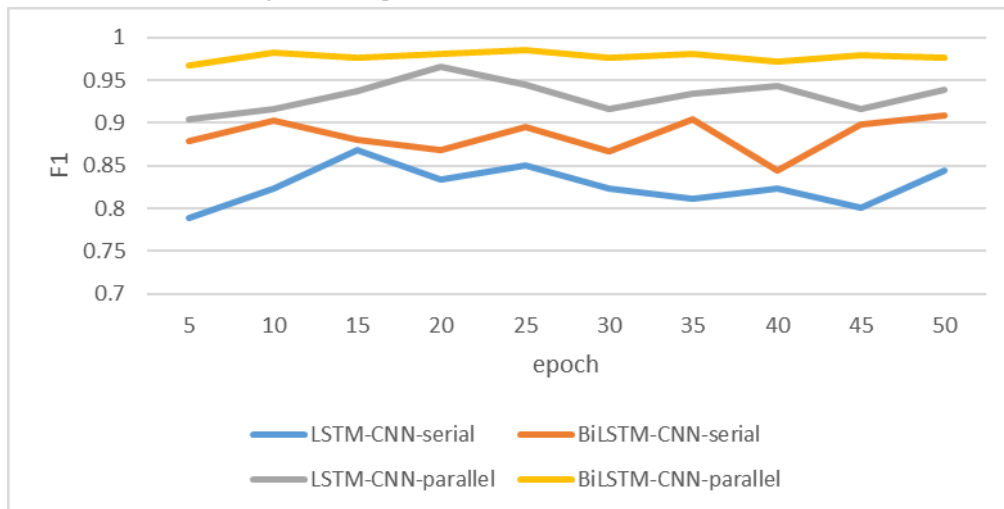


图 13 不同模型的 F1 对比结果

Fig. 13 F1 comparison results of different models

从以上三个图中可以看出, 本文所提方法在精确度、召回率和 F1 的评价中大多要高于其他模型, 其准确率上下浮动很小, F1 也比较高。由于本文结合了字向量和 CNN、LSTM 两个网络模型, 从而提高了文本相似度计算的准确度, 表明了该方法的有效性。

4 结论

本文给出了一种基于字向量和两个网络模型的文本相似度方法。利用 word2vec 模型获得字向量, 将文本通过字向量进行向量化, 分别输入 CNN 和 LSTM 网络中进行训练, 将输出进行拼接并计算相似度, 最后在中文数据集上进行实验比较。通过对比实验表明字向量比词向量更能表达文本的语义, 并且由于同时使用两个网络进行训练, 提取的特征值更丰富, 使得相似度计算结果更准确。

[参考文献] (References)

- [1] 张鑫琪. 基于 LSTM-DSSM 的论文查重系统研究与实现 [D]. 辽宁大学, 2021. DOI:10.27209/d.cnki.glniu.2021.000904.
- [2] 张庆颖. 基于 Simhash 和 CNN 的相似新闻推荐 [D]. 电子科技大学, 2020. DOI:10.27005/d.cnki.gdzku.2020.002798.
- [3] 王宇成. 基于 Simhash 和层次聚类算法的网页去重技术研究 [D]. 南京邮电大学, 2019. DOI:10.27251/d.cnki.gnjdc.2019.000544.
- [4] 王兵, 郑亚梅, 陈茂柯, 高凌云. 基于 Tri-BiLSTM-CNN 的钻井安全问答系统 [J]. 西南石油大学学报(自然科学版), 2020, 42(06): 157-164.
- [5] 马震, 戚成亮. 基于改进的深度语义匹配模型在商品召回方面的应用 [J]. 电子元器件与信息技术, 2020, 4(12): 126-127. DOI:10.19772/j.cnki.2096-4455.2020.12.062.
- [6] 俞婷婷, 徐彭娜, 江育娥等. 基于改进的 Jaccard 系数文档相似度计算方法 [J]. 计算机系统应用, 2017, 26(12): 137-142. DOI:10.15888/j.cnki.csa.006123.
- [7] 李圣文, 凌微, 龚君芳, 周长征. 一种基于熵的文本相似性计算方法 [J]. 计算机应用研究, 2016, 33(03): 665-668.
- [8] 陈宏朝, 李飞, 朱新华, 马润聪. 基于路径与深度的同义词词林词语相似度计算 [J]. 中文信息学报, 2016, 30(05): 80-88.
- [9] 甘秋云. 基于 TF-IDF 向量空间模型文本相似度算法的分析 [J]. 池州学院学报, 2018, 32(03): 41-43. DOI:10.13420/j.cnki.jczu.2018.03.009.
- [10] He H, Gimpel K, Lin J. Multi-perspective sentence similarity modeling with convolutional neural networks [C]//Conference on Empirical Methods in Natural Language Processing. 2015: 1576-1586.
- [11] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks [J]. Computer Science, 2015, 5(1): 36.
- [12] Mueller J, Thyagarajan A. Siamese recurrent architectures for learning sentence similarity [C]//Thirtieth AAAI Conference on Artificial Intelligence. AAAI Press, 2016: 2786-2792.
- [13] Neculoiu P, Versteegh M, Rotaru M. Learning text similarity with siamese recurrent networks [C]//Proceedings of the 1st Workshop on Representation Learning for NLP. 2016: 148-157.
- [14] Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model [J]. Journal of Machine Learning Research, 2003, 3(6): 1137-1155.
- [15] 张春英, 李春虎, 付其峰. 基于 WV-CNN 的中文文本语义相似度计算方法 [J]. 华北理工大学学报(自然科学版), 2019, 41(01): 123-132.
- [16] 郭浩, 许伟, 卢凯, 唐球. 基于 CNN 和 BiLSTM 的短文本相似度计算方法 [J]. 信息技术与网络安全, 2019, 38(06): 61-64+68. DOI:10.19358/j.issn.2096-5133.2019.06.012.
- [17] 何颖刚, 王宇. 一种基于字向量和 LSTM 的句子相似度计算方法 [J]. 长江大学学报(自然科学版), 2019, 16(01): 88-94+9. DOI:10.16772/j.cnki.1673-1409.2019.01.017.
- [18] 谭咏梅, 刘姝雯, 吕学强. 基于 CNN 与双向 LSTM 的中文文本蕴含识别方法 [J]. 中文信息学报, 2018, 32(07): 11-19.