·人工智能与模式识别·　　文章编号：1000-3428(2022)08-0129-07　　文献标志码：A　　中图分类号：TP18

融合多粒度信息与外部知识的短文本匹配模型

梁登玉，刘大明

（上海电力大学 计算机科学与技术学院，上海 200090）

摘　要：中文短文本通常使用单词序列而非字符序列进行语义匹配，以获得更好的语义匹配性能。然而，中文分词可能是错误或模糊的，容易引入噪声或者错误传播，从而损害模型的匹配性能。此外，多数中文词汇具有一词多义的特点，短文本由于缺少上下文环境，相比一词多义的长文本更难理解，这对于模型正确捕获语义信息是一个更大的挑战。提出一种短文本匹配模型，使用词格长短期记忆网络（Lattice LSTM）融合字符和字符序列的多粒度信息。引入外部知识HowNet解决多义词的问题，使用软注意力机制获取2个句子间的交互信息，并利用均值池化和最大池化算法进一步提取句子的特征信息，获取句子级语义编码表示。在数据集LCQMC和BQ上的实验结果表明，与ESIM、BIMPM和Lattice-CNN模型相比，该模型能有效提升中文短文本语义匹配的准确率。

关键词：短文本语义匹配；词格长短期记忆网络；多粒度信息；外部知识；软注意力机制

开放科学（资源服务）标志码（OSID）：

中文引用格式：梁登玉，刘大明.融合多粒度信息与外部知识的短文本匹配模型[J].计算机工程，2022，48(8)：129-135，143.

英文引用格式：LIANG D Y，LIU D M.Short text matching model combined with multi-granularity information and external knowledge[J].Computer Engineering，2022，48(8)：129-135，143.

Short Text Matching Model Combined with Multi-Granularity Information and External Knowledge

LIANG Dengyu，LIU Daming

（School of Computer Science and Technology，Shanghai University of Electric Power，Shanghai 200090，China）

【Abstract】Chinese short text semantic matching generally uses word sequences，rather than character sequences，to achieve higher semantic-matching performance.However，Chinese word segmentation may be inaccurate or ambiguous，and word segmentation errors can introduce noise and lead to error propagation，thus deteriorating the final matching performance.Meanwhile，most Chinese words are characterized by polysemy.Short texts are more difficult to understand than long texts due to lack of context，which presents a greater challenge for the model to correctly capture semantic information.To solve this problem，a short text matching model is proposed.This model uses the Lattice Long Short Term Memory（Lattice LSTM）to integrate the multi-granularity information of characters and character sequences，introduces external knowledge HowNet to solve the problem of polyonyms，and employs a soft-attention mechanism to capture the interaction between two sentences.Furthermore，mean and maximum pooling algorithms are used to extract sentence features and obtain semantic encoding representations at the sentence level.Experiments on LCQMC and BQ datasets show that the model effectively improves the performance of Chinese short text semantic matching compared with the ESIM，BIMPM，and Lattice-CNN models.

【Key words】short text semantic matching；Lattice Long Short Term Memory（Lattice LSTM）；multi-granularity information；external knowledge；soft-attention mechanism

DOI：10. 19678/j. issn. 1000-3428. 0062065

0　概述

传统的文本匹配方法主要从词汇层面衡量2个文本的匹配程度，即2个文本中出现相同词的个数越多，词序列的排序越接近，则相似度越高，典型的方法有TF-IDF、BM25等。由于词与词之间相互独立，没有考虑上下文语境，因此这种基于词汇重合度的匹配方法有很大的局限性，例如“苹果”在不同语

基金项目：甘肃省自然科学基金（SKLLDJ032016021）。

作者简介：梁登玉（1989—），女，硕士，主研方向为自然语言处理；刘大明（通信作者），副教授、博士。

收稿日期：2021-07-13　　**修回日期：**2021-09-20　**E-mail：**dyu_liang@mail.shiep.edu.cn

境下有不同的意义，可能表示水果，也可能表示公司。近年来，文本匹配的深度学习方法取得了一定进展[1-3]，使用深度学习方法的文本匹配模型首先对输入文本进行分词，然后使用词嵌入技术将分好的词转化为词向量，将待比较的2个句子向量通过同一个深度神经网络编码器映射到相同的向量空间中，最后使用分类技术计算2个句子的匹配程度。通过词嵌入技术训练词向量的常用方法是Word2Vec[4]和Glove[5]，其本质是基于共现信息训练词向量，这种词向量表示增强了词的上下文信息表示，但没有解决句子语义表示问题，因此需要将词向量输入到深度神经网络，即编码器中，获取句子级的上下文语义信息。例如，Text-CNN[6]使用卷积神经网络编码每个句子，基于双向的长短期记忆网络（Bi-directional Long Short Term Memory，BiLSTM）的文本匹配模型[7]使用双向长短期记忆神经网络实现对句子的编码表征。单纯的句子编码方法将每个句子编码成一个固定长度的向量，然后直接计算句子的相似度，模型设计简单、易于应用推广。但研究发现，通过增强句子间的交互能够提高文本匹配效果。句子对交互方法将单词对齐和句子对之间的交互考虑在内，并且通常在域内数据上训练时表现得更好。BiMPM[8]是一种双边多视角匹配模型，通过使用BiLSTM神经网络对每个句子进行编码，从多角度匹配2个句子。ESIM模型[9]采用2个BiLSTM神经网络分别对句子进行编码以及融合2个句子间的词对齐信息。ESIM模型在各匹配任务上取得了较好的效果。

然而，以上几乎所有模型最初都是为英文文本匹配提出的。如果将其应用于中文文本匹配，通常有两种方法，一种是以汉字作为模型的输入，另一种是先把每个句子分割成单词，然后把这些单词作为输入。虽然基于字符的模型可以克服数据在一定程度上的稀疏[10]问题，但这些模型也存在一些缺点，比如没有充分利用词汇的语义信息，而这些信息可能对文本语义匹配有用。然而，基于词的模型经常遭受一些由词分割引起的潜在问题。例如，字符序列“南京市长江大桥”因为分词不同而产生2个不同意思，第1个“南京市\长江大桥”指的是一座桥，另一个“南京市长\江大桥”指的是一个人。这句话的模糊性可以通过使用更多的上下文信息来消除。此外，不同工具的细分粒度也不同。比如“长江大桥”可以分为“长江”和“大桥”2个词。鉴于这种情况，文献[11]使用词格长短期记忆网络（Lattice LSTM）来表示一个句子，能够在没有分词的情况下利用单词信息获取多粒度的句子表示。Lattice LSTM模型将输入的字符和所有能在词典匹配的单词一起编码输入到模型中，在词典中选出与字符最相关的单词，降低歧义发生的概率，同时考虑了字符和词2种粒度的输入，该模型在多个NLP任务中取得了显著提升效果。尤其是命名实体识别任务中，基于Lattice LSTM的模型[12]编码了一系列输入字符以及所有匹配词典的潜在单词，以获得更好的NER结果。文献[13-15]分别展示了Lattice LSTM模型在命名实体识别领域的应用。在神经机器翻译领域，文献[16]提出一种基于词格的递归神经网络编码器以压缩编码多个标记词格作为输入，不仅减轻了最佳标记方式标记错误的负面影响，而且具有更强的表达性和嵌入输入句子的灵活性。文献[17]使用Lattice LSTM模型获取多粒度信息用于中文分词任务，文献[18]使用Lattice LSTM模型融合外部知识，用于中文关系提取任务中。

受到Lattice结构在自然语言处理领域获得成功的启发，在文本匹配任务中，文献[19]引入Lattice结构，提出Lattice-CNN模型，利用基于Lattcie结构的CNN神经网络在词格上提取句子级特征。该模型不依赖于字符或单词级序列，而是将单词格作为输入，其中每个可能的单词和字符都将得到平等对待，并拥有自己的上下文信息，以便它们可以在每一层进行交互。对于每一层中的每个词，可以通过池方法以不同的粒度捕获不同的上下文词。虽然Lattice-CNN模型利用了词格，但它只关注了局部信息，缺乏全局的句子语义信息表示及句子之间的交互信息。文献[20]提出一种用于中文短文本匹配的神经图匹配方法，该方法以一对词格图作为输入，根据图匹配注意机制更新节点的表示。这种方法既处理了多粒度信息，又关注句子间的交互信息，该模型优于之前的模型效果。文献[19]和文献[20]证明了多粒度信息对文本匹配的重要性。

由于汉语词汇的多义性给语义理解带来了很大困难，因此短文本中的一词多义带来的问题比长文本中的一词多义问题更严重。在通常情况下，短文本的上下文信息较少，因此模型极难捕获正确的语义信息。例如“落后”，有落后和不如的意思，这2个句子“她的成绩落后于他”和“她的成绩不如他”应该是相似的，但如果不对“落后”这个词的多义性进行解释，模型将很难判定这2个句子是相似的。

本文提出一种融合多粒度信息和外部知识的短文本匹配模型，使用Lattice LSTM模型融合字符和单词级别的多粒度信息，降低因为分词错误导致的误差传播。引入知网HowNet[21]作为外部知识库，解决单词的多义性带来的语义干扰问题，并将HowNet与Lattice LSTM相结合，丰富Lattice LSTM词语级粒度的语义表示。此外，使用软注意力机制获取句子间的交互信息，利用BiLSTM融合2个句子的上下文信息，并通过最大池化和均值池化进一步提取特征信息，获取句子级语义表示，经过拼接之后输入预测层，使用softmax分类函数计算2个句子相似的概率。

1 本文模型

图1所示为本文模型的框架。其中，输入层输入2个句子A和Q，使用Word2Vec将原始字符序列转化为嵌入表示；融合外部知识的Lattice LSTM层对字符，单词序列以及使用HowNet释义后的词汇信

息进行编码表示；模型使用软注意力机制捕获句子间的交互信息；BiLSTM层用于综合全局上下文语义信息；池化层用于使用最大池化和均值池化进一步提取特征信息，获取句子级向量表示，并进行拼接；预测层使用前馈神经网络和softmax分类函数计算2个文本的相似度概率输出。

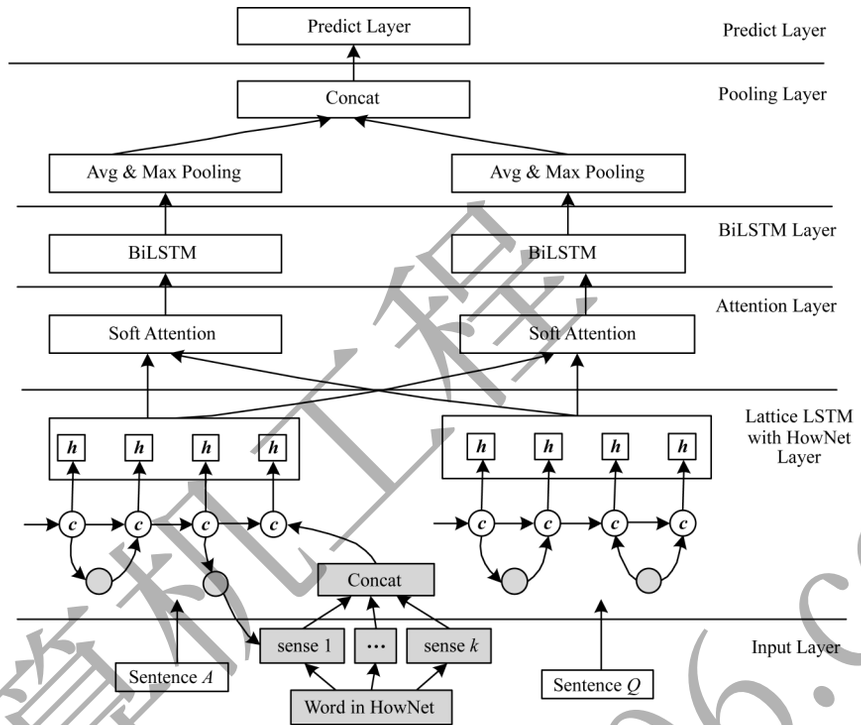

图1　本文模型框架

Fig.1　Framework of the model in this paper

本文并未使用基于Bert等有更强语义能力的预训练语言模型来设计模型，主要考虑到以下几点：

1）在词向量表示方面，本文使用了Word2Vec来训练词向量，因为Bert等预训练语言模型虽然具有更好的语义表示能力，但模型参数多，加载速度慢，内存、时间等消耗较大，而Word2Vec简单高效，特别适合从大规模、超大规模的语料中获取高精度的词向量表示。

2）在与词格和外部知识结合方面，本文使用Lattice LSTM模型。因为类似于Bert这样的大型预训练语言模型通常采用一系列细粒度单元（汉字）作为输入，并且是按位置排序的序列，这使得利用单词格和保留位置关系变得困难。同时，传统的掩蔽语言模型（Masked Language Model，MLM）可能会使基于单词格的预训练语言模型学习到错误的语义表示。原因是这样的词格可能会引入冗余，即一个字符可以包含在多个文本单元中。在掩蔽语言模型中，模型可能引用与随机屏蔽的文本单元重叠的其他文本单元，而不是真实的上下文，导致信息泄漏。而Lattice LSTM模型结构相对简单，且不会造成信息泄露。

因此，综合考虑到实验效果、内存消耗等原因，本文没有使用基于Bert等有更强语义能力的预训练语言模型。

1.1　输入层

本文模型的输入为基于字符的2个句子，输入层采用训练好的Word2Vec模型将每个字符转换为低维实数向量，这个过程通过查找字符嵌入矩阵，即可对字符进行编码。

1.2　Lattice LSTM with HowNet层

虽然模型将字符序列作为直接输入，但为了充分捕捉单词级特征，还需要输入句子中所有潜在单词的信息。一个潜在的单词是任何字符子序列，它与词典D中的一个单词相匹配，词典D建立在分割的大原始文本之上。1.2.1节将介绍如何使用Lattice LSTM融合字符和单词两个粒度的信息。

为更好地理解Lattice LSTM模型的结构，首先给出每个LSTM单元的计算公式，如式（1）所示：

$$\begin{aligned}
&\boldsymbol{f}_t=\sigma(\boldsymbol{W}_f\times[\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{B}_f)\\
&\boldsymbol{i}_t=\sigma(\boldsymbol{W}_i\times[\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{B}_i)\\
&\tilde{\boldsymbol{C}}_t=\tanh(\boldsymbol{W}_C\times[\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{B}_C)\\
&\boldsymbol{c}_t=\boldsymbol{f}_t\times\boldsymbol{c}_{t-1}+\boldsymbol{i}_t\times\tilde{\boldsymbol{C}}_t\\
&\boldsymbol{o}_t=\sigma(\boldsymbol{W}_o\times[\boldsymbol{h}_{t-1},\boldsymbol{x}_t]+\boldsymbol{B}_o)\\
&\boldsymbol{h}_t=\boldsymbol{o}_t\times\tanh(\boldsymbol{c}_t)
\end{aligned}\tag{1}$$

其中：$\boldsymbol{f}_t$表示遗忘门限；$\boldsymbol{i}_t$表示输入门限；$\tilde{\boldsymbol{C}}_t$表示t时刻细胞状态的候选值；$\boldsymbol{c}_t$表示cell状态（这里是循环发生的地方）；$\boldsymbol{c}_{t-1}$表示$t-1$时刻的细胞状态，$\boldsymbol{o}_t$表示输出门限；$\boldsymbol{h}_t$表示当前单元的输出；$\boldsymbol{h}_{t-1}$表示前一时刻单元的输出；$\boldsymbol{x}_t$表示当前时刻记忆单元的输入；tanh为双曲正切函数；带角标的$\boldsymbol{W}$和$\boldsymbol{B}$为模型参数；σ为sigmoid激活函数。

1.2.1　Lattice LSTM模型

Lattice LSTM模型的结构如图2所示。

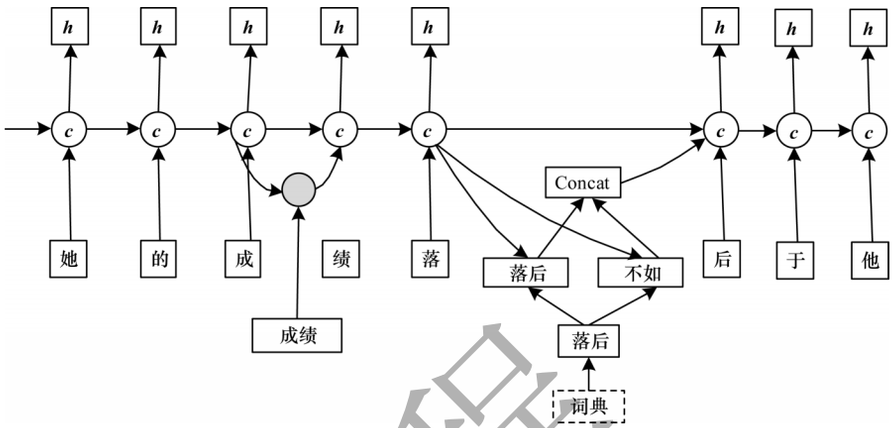

图2 Lattice LSTM模型的结构

Fig.2 Structure of Lattice LSTM model

和基于字向量的LSTM模型相比，字符单元向量 $\boldsymbol{c}_i^c$ 的计算不仅使用了字向量，还运用了子序列 $\boldsymbol{w}_{b,e}^d$。用 $\boldsymbol{x}_{b,e}^w$ 表示每个词向量，$\boldsymbol{x}_{b,e}^w$ 的计算方法如式(2)所示：

$$\boldsymbol{x}_{b,e}^w=\boldsymbol{e}^w(\boldsymbol{w}_{b,e}^d) \tag{2}$$

其中：$\boldsymbol{e}^w$ 为词向量矩阵。定义词单元 $\boldsymbol{c}_{b,e}^w$ 记录 $\boldsymbol{x}_{b,e}^w$ 的递归状态。$\boldsymbol{c}_{b,e}^w$ 的计算方法如式(3)和式(4)所示：

$$\begin{bmatrix}\boldsymbol{i}_{b,e}^w\\ \boldsymbol{f}_{b,e}^w\\ \tilde{\boldsymbol{c}}_{b,e}^w\end{bmatrix}=\begin{bmatrix}\sigma\\ \sigma\\ \tanh\end{bmatrix}\left(\boldsymbol{W}^{w^{\mathrm{T}}}\begin{bmatrix}\boldsymbol{x}_{b,e}^w\\ \boldsymbol{h}_b^c\end{bmatrix}+\boldsymbol{B}^w\right) \tag{3}$$

$$\boldsymbol{c}_{b,e}^w=\boldsymbol{f}_{b,e}^w\odot\boldsymbol{c}_b^c+\boldsymbol{i}_{b,e}^w\odot\tilde{\boldsymbol{c}}_{b,e}^w \tag{4}$$

其中：$\boldsymbol{i}_{b,e}^w$ 代表输入门；$\boldsymbol{f}_{b,e}^w$ 代表遗忘门；$\boldsymbol{W}^{w^{\mathrm{T}}}$ 和 $\boldsymbol{B}^w$ 为可训练的模型参数；σ 表示sigmoid函数；$\odot$ 为Hadamard Product，即操作矩阵中对应的元素相乘。词序列单元没有输出门，只在字符级执行。$\boldsymbol{c}_{b,e}^w$ 的存在使隐藏层中的 $\boldsymbol{c}_j^c$ 值被更多不同的信息流影响。例如图2中，$\boldsymbol{c}_6^c$ 的输入包括 $\boldsymbol{x}_6^c$(后)、$\boldsymbol{c}_{5,6}^w$(落后)。通过构建额外的门 $\boldsymbol{i}_{b,e}^c$ 实现对 $\boldsymbol{c}_{b,e}^w$ 到 $\boldsymbol{c}_{b,e}^c$ 之间的信息流控制。计算方法如式(5)所示：

$$\boldsymbol{i}_{b,e}^c=\sigma(\boldsymbol{W}^{l^{\mathrm{T}}}\times\begin{bmatrix}\boldsymbol{x}_e^c\\ \boldsymbol{c}_{b,e}^w\end{bmatrix}+\boldsymbol{B}^l) \tag{5}$$

其中：$\boldsymbol{W}^{l^{\mathrm{T}}}$ 和 $\boldsymbol{b}^l$ 为模型参数。利用 $\boldsymbol{c}_{b,e}^w$ 和 $\tilde{\boldsymbol{c}}_j^c$ 的值计算 $\boldsymbol{c}_j^c$ 的值，计算公式如式(6)所示：

$$\boldsymbol{c}_j^c=\sum_{b\in\{b'|w_{b',e}^d\in D\}}\boldsymbol{\alpha}_{b,j}^c\odot\boldsymbol{c}_{b,j}^w+\boldsymbol{\alpha}_j^c\odot\tilde{\boldsymbol{c}}_j^c \tag{6}$$

其中：$\boldsymbol{\alpha}_{b,j}^c$ 和 $\boldsymbol{\alpha}_j^c$ 分别为 $\boldsymbol{i}_{b,j}^c$ 和 $\boldsymbol{i}_j^c$ 归一化后的值，它们的和为1。$\boldsymbol{\alpha}_{b,j}^c$ 和 $\boldsymbol{\alpha}_j^c$ 的计算公式如式(7)和式(8)所示：

$$\boldsymbol{\alpha}_{b,j}^c=\frac{\exp(\boldsymbol{i}_{b,j}^c)}{\exp(\boldsymbol{i}_j^c)+\sum\limits_{b'\in\{b'|w_{b',e}^d\in D\}}\exp(\boldsymbol{i}_{b',j}^c)} \tag{7}$$

$$\boldsymbol{\alpha}_j^c=\frac{\exp(\boldsymbol{i}_j^c)}{\exp(\boldsymbol{i}_j^c)+\sum\limits_{b'\in\{b'|w_{b',e}^d\in D\}}\exp(\boldsymbol{i}_{b',j}^c)} \tag{8}$$

最终隐藏层的计算公式如式(9)所示：

$$\boldsymbol{h}_j^c=\boldsymbol{o}_j^c\odot\tanh(\boldsymbol{c}_j^c) \tag{9}$$

其中：$\odot$ 为Hadamard Product；o_j^c 的计算参考式(1)。

1.2.2 融合外部知识的Lattice LSTM编码器

虽然基于Lattice LSTM的编码器可以利用字符和单词信息，但不能充分考虑中文的歧义性。例如图2中，单词“落后”有“落后”和“不如”2种含义，但在基本的Lattice LSTM编码器中，$\boldsymbol{w}_{5,6}^d$ 只有一种表示。因此，通过引入外部知识改进Lattice LSTM模型，可以构建一个更全面的词汇。

给定一个词 $w_{b,e}$，首先通过检索知网HowNet获得它的所有 k 个义项。利用Sense($w_{b,e}$)表示 $w_{b,e}$ 的义集，然后通过SAT模型[22]将每个 $\mathrm{sen}_k^{(w_{b,e})}\in\mathrm{Sense}(w_{b,e})$ 转换为实值向量 $\boldsymbol{x}_{b,e,k}^{\mathrm{sen}}$。SAT模型基于Skip-gram可以联合学习单词和意义表示。如此，$\boldsymbol{w}_{b,e}$ 表示的是一个向量集，如式(10)所示：

$$\boldsymbol{x}_{b,e}^{\mathrm{sen}}=\{\boldsymbol{x}_{b,e,1}^{\mathrm{sen}},\boldsymbol{x}_{b,e,2}^{\mathrm{sen}},\cdots,\boldsymbol{x}_{b,e,k}^{\mathrm{sen}}\} \tag{10}$$

单词 $w_{b,e}$ 的第 k 个意义表示是 $\boldsymbol{x}_{b,e,k}^{\mathrm{sen}}$。对于与词典 D 匹配的每个单词 $w_{b,e}$，把它的所有意义表示都考虑到计算中。单词 $w_{b,e}$ 的第 k 个意义的计算类似于式(3)，具体如式(11)所示：

$$\begin{bmatrix}\boldsymbol{i}_{b,e,k}^{\mathrm{sen}}\\ \boldsymbol{f}_{b,e,k}^{\mathrm{sen}}\\ \tilde{\boldsymbol{c}}_{b,e,k}^{\mathrm{sen}}\end{bmatrix}=\begin{bmatrix}\sigma\\ \sigma\\ \tanh\end{bmatrix}\left(\boldsymbol{W}^{w^{\mathrm{T}}}\begin{bmatrix}\boldsymbol{x}_{b,e,k}^{\mathrm{sen}}\\ \boldsymbol{h}_b^c\end{bmatrix}+\boldsymbol{B}^w\right) \tag{11}$$

其中：$\boldsymbol{c}_{b,e,k}^{\mathrm{sen}}$ 表示单词 $w_{b,e}$ 的第 k 个意义的存储单元，其计算公式与式(4)类似，具体见式(12)：

$$\boldsymbol{c}_{b,e,k}^{\mathrm{sen}}=\boldsymbol{f}_{b,e,k}^{\mathrm{sen}}\odot\boldsymbol{c}_b^c+\boldsymbol{i}_{b,e,k}^{\mathrm{sen}}\odot\tilde{\boldsymbol{c}}_{b,e,k}^{\mathrm{sen}} \tag{12}$$

将所有的意义表示合并成一个综合表示 $\boldsymbol{c}_{b,e}^{\mathrm{sen}}$，计算公式如式(13)所示：

$$\boldsymbol{c}_{b,e}^{\mathrm{sen}}=\sum_k\boldsymbol{\alpha}_{b,e,k}^{\mathrm{sen}}\odot\boldsymbol{c}_{b,e,k}^{\mathrm{sen}},\quad\boldsymbol{\alpha}_{b,e,k}^{\mathrm{sen}}=\frac{\exp(\boldsymbol{i}_{b,e,k}^{\mathrm{sen}})}{\sum\limits_{k'}^{K}\exp(\boldsymbol{i}_{b,e,k'}^{\mathrm{sen}})} \tag{13}$$

将多义词的所有意义合并到单词表示 $\boldsymbol{c}_{b,e}^{\mathrm{sen}}$ 中，可以更好地表示多义词。然后，类似于式(5)~式(8)，所有以索引e结尾的单词循环路径均将流入细胞单元 $\boldsymbol{c}_e^c$，计算公式如式(14)所示：

$$\boldsymbol{c}_e^c=\sum_{b\in\{b'|w_{b',e}^d\in D\}}\boldsymbol{\alpha}_{b,e}^{\mathrm{sen}}\odot\boldsymbol{c}_{b,e}^{\mathrm{sen}}+\boldsymbol{\alpha}_e^c\odot\tilde{\boldsymbol{c}}_e^c \tag{14}$$

最终隐藏层的计算公式与式(9)相似。

通过计算得到隐藏层的所有输出向量($\boldsymbol{h}_1$, $\boldsymbol{h}_2$, …, $\boldsymbol{h}_l$)，其中l为句子长度，即字向量的个数。

1.3 Attention层

本文文本匹配模型的输入为2个句子，设为$\boldsymbol{S}_m=\{\boldsymbol{c}_1^m, \boldsymbol{c}_2^m, \cdots, \boldsymbol{c}_{l_m}^m\}$和$\boldsymbol{S}_n=\{\boldsymbol{c}_1^n, \boldsymbol{c}_2^n, \cdots, \boldsymbol{c}_{l_n}^n\}$，将2个句子分别输入到融入外部知识的Lattice LSTM模型中，得到输出，如式(15)所示：

$$\boldsymbol{p}_m=\{\boldsymbol{h}_1^m, \boldsymbol{h}_2^m, \cdots, \boldsymbol{h}_{l_m}^m\}, \boldsymbol{p}_n=\{\boldsymbol{h}_1^n, \boldsymbol{h}_2^n, \cdots, \boldsymbol{h}_{l_n}^n\} \tag{15}$$

其中：l_m和l_n分别为2个句子的长度。

在Attention层比较2个句子的软注意力权重，也就是对于句子$\boldsymbol{S}_m$和$\boldsymbol{S}_n$序列，分别计算$\boldsymbol{S}_m$相对于$\boldsymbol{S}_n$以及$\boldsymbol{S}_n$相对于$\boldsymbol{S}_m$的注意力权重，从而得到2个不同的权重分布，通过这种方式捕获2个句子之间的交互信息，计算公式如式(16)所示：

$$\begin{aligned}&\boldsymbol{e}_{ij}=\boldsymbol{p}_{m_i}^{\mathrm{T}}\boldsymbol{p}_{n_j},\\&\hat{\boldsymbol{p}}_{m_i}=\sum_{j=1}^{l_n}\frac{\exp(\boldsymbol{e}_{ij})}{\sum_{k=1}^{l_n}\exp(\boldsymbol{e}_{ik})}\boldsymbol{p}_{n_j}, \forall j\in[1,2,\cdots,l_m]\\&\hat{\boldsymbol{p}}_{n_j}=\sum_{i=1}^{l_m}\frac{\exp(\boldsymbol{e}_{ij})}{\sum_{k=1}^{l_m}\exp(\boldsymbol{e}_{kj})}\boldsymbol{p}_{m_i}, \forall j\in[1,2,\cdots,l_n]\end{aligned} \tag{16}$$

1.4 BiLSTM层

BiLSTM层用于将2个匹配向量序列聚合成固定长度的匹配向量。由于单向LSTM接收某个序列输入的信息，其网络中单个单元仅参考了该序列的上文信息，而没有考虑下文信息。为了弥补这一缺点，充分利用上下文信息，采用BiLSTM网络。BiLSTM的网络输出层能够完整地捕获输入序列中每一个点关于过去和未来的上下文信息。

根据LSTM网络的结构，每个LSTM单元根据式(1)进行计算。

使用BiLSTM神经网络结构，对来自注意力层的输出进一步编码，如式(17)所示：

$$\begin{aligned}&\boldsymbol{u}_m=\mathrm{BiLSTM}(\hat{\boldsymbol{p}}_{m_i}, i), \forall i\in[1,2,\cdots,l_m],\\&\boldsymbol{u}_n=\mathrm{BiLSTM}(\hat{\boldsymbol{p}}_{n_j}, j), \forall j\in[1,2,\cdots,l_n]\end{aligned} \tag{17}$$

1.5 池化层

在池化层中，本文使用最大池化和均值池化进一步捕捉文本的特征信息，计算公式如式(18)所示：

$$\begin{aligned}&\boldsymbol{u}_{m,\mathrm{avg}}=\sum_{i=1}^{l_m}\frac{\boldsymbol{u}_{m_i}}{l_m}, \boldsymbol{u}_{m,\max}=\max_{i=1}^{l_m}\boldsymbol{u}_{m_i},\\&\boldsymbol{u}_{n,\mathrm{avg}}=\sum_{j=1}^{l_n}\frac{\boldsymbol{u}_{n_j}}{l_n}, \boldsymbol{u}_{n,\max}=\max_{j=1}^{l_n}\boldsymbol{u}_{n_j},\\&\boldsymbol{v}_m=[\boldsymbol{u}_{m,\mathrm{avg}}; \boldsymbol{u}_{m,\max}], \boldsymbol{v}_n=[\boldsymbol{u}_{n,\mathrm{avg}}; \boldsymbol{u}_{n,\max}]\end{aligned} \tag{18}$$

最后将2个向量拼接得到输出向量，如式(19)所示：

$$\boldsymbol{o}=[\boldsymbol{v}_m, \boldsymbol{v}_n] \tag{19}$$

1.6 预测层

预测层用于评估概率分布$p(y|\boldsymbol{S}_m, \boldsymbol{S}_n)$，衡量2个句子的相似度。将上一层的输出向量$\boldsymbol{o}_{\mathrm{out}}$输入到一个具有2层结构的前馈神经网络中，使用softmax激活函数计算2个文本的相似度概率值，计算公式如式(20)所示：

$$p(y|\boldsymbol{S}_m, \boldsymbol{S}_n)=F(\boldsymbol{o}_{\mathrm{out}}) \tag{20}$$

其中：$F(\cdot)$代表1个具有两层的前馈神经网络和1个softmax激活后输出层。

最后，使用交叉熵Cross-Entropy函数作为损失函数，表达式如式(21)所示：

$$L_{\mathrm{loss}}=-[y\times\log_a(p)+(1-y)\times\log_a(1-p)] \tag{21}$$

其中：$\boldsymbol{p}$为预测概率值；y为真实值。

2 实验结果与分析

2.1 数据集

本文在LCQMC和BQ中文数据集上进行实验。LCQMC是一个用于问题匹配的大规模开放领域语料库，该数据集中的样本包含一对句子和一个二进制标签，该标签指示这2个句子是否具有相同的含义或具有相同的意图，若具有相同的含义或意图，则标签为1，反之标签为0。LCQMC数据集包含238 766条训练集，8 802条验证集，12 500条测试集。LCQMC数据集示例如表1所示。

表1　LCQMC数据集示例

Table1　Example of LCQMC dataset

问题1	问题2	含义是否相同	标签
货到付款的网站是哪个？	什么购物网站是货到付款的？	相同	1
这种图片是用什么软件制作的？	这种图片制作是用什么软件呢？	相同	1
蛋黄吃多了有什么坏处？	吃鸡蛋白过多有什么坏处？	不同	0

BQ数据集是一个针对特定领域的大规模银行问题匹配语料库，由120 000个中文句子对组成，包括100 000个训练样本、10 000个开发样本和10 000个测试样本。每一对还与一个二进制标签相关联，该标签指示两个句子是否具有相同的含义。

本文的词典词汇来自数据集LCQMC和BQ，对数据集进行清洗处理并去除停用词之后，使用Jieba分词工具进行分词，同时使用知网HowNet和百度百科词汇信息加入较新颖的词汇，例如“去哪儿网”、“余额宝”、“悦诗风吟”等，这些词即使使用Jieba分词工具也不能得到准确分词。最后，使用Word2vec进行词汇训练，获得词向量。

2.2 评价指标

本文分别采用准确率(Accuracy，Acc)和F1值作为测评指标，计算公式如式(22)所示：

$$\begin{aligned}&A_{\mathrm{Acc}}=\frac{T_{\mathrm{TP}}+T_{\mathrm{TN}}}{T_{\mathrm{TP}}+F_{\mathrm{FN}}+F_{FP}+T_{\mathrm{TN}}}\\&P=\frac{T_{\mathrm{TP}}}{T_{\mathrm{TP}}+F_{FP}}\end{aligned}$$

$$R=\frac{T_{\mathrm{TP}}}{T_{\mathrm{TP}}+F_{FN}}$$

$$\mathrm{F1}=2\frac{P\times R}{P+R} \tag{22}$$

其中：T_{TP}为真正例；T_{TN}真负例；F_{FP}为假正例；F_{FN}为假负例。

2.3 实验环境

本文实验采用型号为RTX 2080S的GPU服务器，在Python3.6和Pytorch1.5的环境下运行。由于本文采用了基于改进的Lattice LSTM模型，因此需要设定各个部分的参数。对于字符嵌入和词格嵌入，嵌入尺寸分别设置为100维和200维，2个LSTM隐藏层的大小设置为200维，dropout设置为0.5，学习率设置为0.001。在训练模型时，损失函数使用交叉熵损失函数，使用Adam优化器更新参数。使用训练集训练模型，并使用测试集评估模型效果。模型参数设定如表2所示。

表2 模型参数

Table 2 Model parameter

参数	取值
字符嵌入尺寸/维度	100
Lattice嵌入尺寸/维度	200
LSTM 隐藏层/维度	200
dropout	0.5
学习率	0.001

2.4 实验分析

本文设计了3组实验，分别是不同模型的对比实验、探究Jieba分词和Lattice词格对实验结果的影响实验、消融实验。

2.4.1 不同模型的对比实验

将BiMPM[8]、ESIM[9]、Lattice-CNN[19]与本文模型进行对比，实验结果如表3和表4所示。由表3和表4可知，本文模型在2个数据集上的准确率和F1值均高于其他3个模型。虽然ESIM模型综合BiLSTM模型和注意力机制，BiMPM模型利用了更多角度的信息，从多个视角提取句子的特征，但可能因为分词错误影响了模型效果。而Lattice-CNN模型则因为缺乏全局的句子语义表示和多角度的交互信息，而导致效果更差。

表3 不同模型在数据集LCQMC下的对比实验结果

Table 3 Comparative experimental results of different models under LCQMC dataset %

模型	准确率	F1值
Lattice-CNN模型	80.23	81.51
ESIM模型	81.34	82.53
BIMPM模型	82.25	83.67
本文模型	83.54	84.32

表4 不同模型在数据集BQ下的对比实验结果

Table 4 Comparative experimental results of different models under BQ dataset %

模型	准确率	F1值
Lattice-CNN模型	77.31	77.63
ESIM模型	80.62	80.74
BIMPM模型	80.53	80.44
本文模型	82.65	82.43

2.4.2 Jieba分词和Lattice词格对实验结果的影响

使用Jieba分词容易出现错误，Jieba分词和Lattice词格对句子相似性的预测效果如表5所示。其中，标签为1代表两个句子语义相似，标签为0代表两个句子语义不相似。从表5可以看出，分词对句子语义的影响比较大，例如“去\哪儿”、“去\哪儿\网”和“去哪儿网”，“余额\宝”和“余额宝”，它们只有被正确分词，才能提供正确的词向量信息。综上所述，分词的正确与否，对文本语义影响很大，本文融合了Lattice结构的文本匹配模型，可以获得比ESIM和BIMPM模型更好的性能。

表5 Jieba分词和Lattice词格对句子相似性的预测效果

Table 5 Prediction results of sentence similarity based on Jieba and Lattice segmentation

问题序号	Jieba分词	Lattice词格	标签	
			Jieba分词	Lattice词格
问题1 分词结果	去哪儿订特价机票 去\哪儿\订\特价机票	去哪儿订特价机票 去\哪儿\订\特价机票	1	0
问题2 分词结果	去哪儿网特价机票怎样订 去\哪儿\网\特价机票\怎样\订	去哪儿网特价机票怎样订 去哪儿网\特价机票\怎样\订		
问题3 分词结果	余额不足可以分期付款吗 余额\不足\可以\分期付款吗	余额不足可以分期付款吗 余额\不足\可以\分期付款\吗	1	0
问题4 分词结果	余额宝可以分期付款吗 余额\宝\可以\分期付款\吗	余额宝可以分期付款吗 余额宝\可以\分期付款\吗		

2.4.3 消融实验

对本文模型的不同方面进行消融研究，评估不同的池化策略（均值、最大值）、注意力层和融合外部知识对实验结果的影响。实验结果如表6所示。

由表6可知，使用均值池化和最大池化对模型的影响相对较小，使用均值池化比使用最大池化的准确率高0.14个百分点，因此本文综合使用了最大池化和均值池化策略。在对注意力机制的消融研究发现，有无注意力层对模型的性能影响很大，因为注

意力层提供了2个句子的交互信息，对计算2个句子的相似度影响很大。由表6还可知，从HowNet上引入外部知识丰富词汇的语义信息，可以提升模型性能。可以肯定的是，一旦数据集中的句子包含更多具有多义性的词汇，例如“水分”，具有“水汽”和“夸耀”的意思，“落后”除了有落在后面的意思，还具有“不如”的意思，引入外部知识对模型的提升效果更加明显。

表6　在LCQMC数据集下的消融实验结果

Table 6　Result of the ablation experiment under LCQMC dataset　%

策略	准确率	F1值
使用最大池化	83.35	83.41
使用均值池化	83.49	83.61
使用注意力层	83.54	84.32
不使用注意力层	78.04	78.33
引入外部知识	83.54	84.32
不引入外部知识	83.01	83.31

3　结束语

本文提出融合多粒度信息和外部知识的短文本匹配模型，使用由多个分词假设形成的成对单词格和外部词汇知识作为模型输入，并结合软注意力机制获取2个句子的交互信息。借鉴BiMPM模型和ESIM模型的交互机制，充分融合Lattice-CNN模型的优点，同时引入外部知识获取更丰富的词汇表示信息。在短文本问题匹配数据集LCQMC和BQ上的实验结果表明，本文模型能有效提升文本匹配的准确率。后续将考虑使用Bert等具有更强语义的预训练模型与词格结构构建文本匹配模型，在提升模型语义表示能力的同时，进一步降低模型复杂度。

参考文献

[1] GONG Y C，LUO H，ZHANG J. Natural language inference over interaction space[EB/OL]. [2021-06-10]. https://www. researchgate. net/publication/319700831_Natural_Language_Inference_over_Interaction_Space.

[2] LIU X，CHEN Q C，CHONG D，et al. LCQMC：a large-scale Chinese question matching corpus[C]//Proceedings of the 27th International Conference on Computational Linguistics. Washington D. C.，USA：IEEE Press，2018：1952-1962.

[3] LAN W W，XU W. Neural network models for paraphrase identification，semantic textual similarity，natural language inference，and question answering[EB/OL]. [2021-06-10]. https://arxiv. org/abs/1806. 04330.

[4] MIKOLOV T，CHEN K，CORRADO G，et al. Efficient estimation of word representations in vector space[EB/OL]. [2021-06-10]. https://arxiv. org/abs/1301. 3781.

[5] PENNINGTON J，SOCHER R，MANNING C. Glove：global vectors for word representation[C]//Proceedings of 2014 Conference on Empirical Methods in Natural Language Processing. Stroudsburg，USA：Association for Computational Linguistics，2014：1532-1534.

[6] HE T，HUANG W L，QIAO Y，et al. Text-attentional convolutional neural network for scene text detection[J]. IEEE Transactions on Image Processing，2016，25(6)：2529-2541.

[7] MUELLER J，THYAGARAJAN A. Siamese recurrent architectures for learning sentence similarity [C]// Proceedings of the 30th AAAI Conference on Artificial Intelligence. San Francisco，USA：AAAI Press，2016：2786-2792.

[8] WANG Z G，HAMZA W，FLORIAN R. Bilateral multi-perspective matching for natural language sentences[C]// Proceedings of the 26th International Joint Conference on Artificial Intelligence. Melbourne，Australia：International Joint Conferences on Artificial Intelligence Organization，2017：4144-4150.

[9] CHEN Q，ZHU X D，LING Z H，et al. Enhanced LSTM for natural language inference[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Stroudsburg，USA：Association for Computational Linguistics，2017：1657-1668.

[10] LI X Y，MENG Y X，SUN X F，et al. Is word segmentation necessary for deep learning of Chinese representations？[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Stroudsburg，USA：Association for Computational Linguistics，2019：3242-3252.

[11] ZHANG Y，WANG Y L，YANG J. Lattice LSTM for Chinese sentence representation [J]. IEEE/ACM Transactions on Audio，Speech，and Language Processing，2020，28：1506-1519.

[12] ZHANG Y，YANG J. Chinese NER using lattice LSTM [C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Stroudsburg，USA：Association for Computational Linguistics，2018：1554-1564.

[13] LIU W，XU T G，XU Q H，et al. An encoding strategy based word-character[C]//Proceedings of the 2019 Conference of the North. Stroudsburg，USA：Association for Computational Linguistics，2019：2379-2389.

[14] 崔丹丹，刘秀磊，陈若愚，等. 基于Lattice LSTM的古汉语命名实体识别[J]. 计算机科学，2020，47(S2)：18-22.
CUI D D，LIU X L，CHEN R Y，et al. Named entity recognition in field of ancient Chinese based on lattice LSTM[J]. Computer Science，2020，47(S2)：18-22. (in Chinese)

[15] 赵耀全，车超，张强. 基于新词发现和Lattice-LSTM的中文医疗命名实体识别[J]. 计算机应用与软件，2021，38(1)：161-165，249.
ZHAO Y Q，CHE C，ZHANG Q. Chinese medical named entity recognition based on new word discovery and lattice-LSTM[J]. Computer Applications and Software，2021，38(1)：161-165，249. (in Chinese)

（下转第143页）

（上接第135页）

[16] SU J S, TAN Z X, XIONG D Y, et al. Lattice-based recurrent neural network encoders for neural machine translation[C]//Proceedings of the 31st AAAI Conference on Artificial Intelligence. San Francisco, USA: AAAI Press, 2017: 3302-3308.

[17] 张文静,张惠蒙,杨麟儿,等. 基于Lattice-LSTM的多粒度中文分词[J]. 中文信息学报,2019,33(1):18-24.
ZHANG W J, ZHANG H M, YANG L E, et al. Multi-grained Chinese word segmentation with Lattice-LSTM[J]. Journal of Chinese Information Processing, 2019, 33(1): 18-24. (in Chinese)

[18] LI Z R, DING N, LIU Z Y, et al. Chinese relation extraction with multi-grained information and external linguistic knowledge[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, USA: Association for Computational Linguistics, 2019: 4377-4386.

[19] LAI Y X, FENG Y S, YU X H, et al. Lattice CNNs for matching based Chinese question answering[C]//Proceedings of the 33rd AAAI Conference on Artificial Intelligence. San Francisco, USA: AAAI Press, 2019: 6634-6641.

[20] CHEN L, ZHAO Y B, LYU B E, et al. Neural graph matching networks for Chinese short text matching[C]//Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online. Stroudsburg, USA: Association for Computational Linguistics, 2020: 6152-6158.

[21] DONG Z D, DONG Q. HowNet: a hybrid language and knowledge resource [C]//Proceedings of International Conference on Natural Language Processing and Knowledge Engineering. Washington D. C., USA: IEEE Press, 2003: 820-824.

[22] NIU Y L, XIE R B, LIU Z Y, et al. Improved word representation learning with sememes[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, USA: Association for Computational Linguistics, 2017: 2049-2058.

编辑　赖玉玲