文章编号：1003-0077(2021)11-0034-09

面向问句复述识别的语义正交化匹配方法研究

朱朦朦，武恺莉，洪 宇，陈 鑫，张 民

（苏州大学 计算机科学与技术学院，江苏 苏州 215006）

摘 要：问句复述识别任务旨在判断两个自然问句的语义是否等价。问句的语义理解与交互是解决该任务的关键因素。现有工作通常基于问句的语义级编码，通过融合或交互的方式，抽取问句的浅层语义特征，以此支持复述问句之间的语义计算。但是如果能找到两个问句的相同点和不同点，就可以基于这些信息得到更为准确的判断结果。基于此想法，该文提出了语义正交化匹配方法，将语义正交化引入到问句复述识别任务中。通过语义正交化方法将每个问句拆分为与另一个问句的相似表示和差异表示，这不仅丰富了问句的语义表示，而且实现了问句的多粒度特征语义融合。该文在中文数据集 LCQMC 和英文数据集 Quora 上进行实验，证明了语义正交化匹配方法在问句复述识别任务中的有效性。

关键词：复述识别；正交化；多粒度

中图分类号：TP391 **文献标识码**：A

A Semantic Orthogonal Matching Method for Question Paraphrase Identification

ZHU Mengmeng, WU Kaili, HONG Yu, CHEN Xin, ZHANG Min

(School of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215006, China)

Abstract: Question paraphrase identification aims to identify whether two natural questions are semantically equivalence, with a core issue of semantic understanding. Current approach usually encoded the sentences into a vector representations and then the two representations are manipulated to give the proof to judge the equivalence. To further capture the the same and different points of the two questions, this paper propose a model to integrate the semantic orthogonal information. In this method, two questions are classified into similar and different representations, which enriches the representations of the questions and realizes the multi-granularity fusion. Experiments have been conducted on two real-world public datasets: LCQMC and Quora, and results demonstrate is the effectiveness of this method.

Keywords: paraphrase identification; orthogonal; multi-granularity

0 引言

问句复述识别，旨在判断两个自然问句的语义是否等价。该任务不仅要准确理解问句的语义，还要比较两个问句语义是否一致。比如问句“红米手机质量怎么样?”与“红米手机好不好”，两者都是询问红米手机好不好，为复述关系。作为一项基本任务，问句复述识别对其他自然语言处理任务的研究也存在重要价值。在机器翻译任务[1]中，问句复述识别可以更好地理解问句语义，得到更准确的翻译；在问答任务[2]中，通过找到相似语义的问句，获得正确答案。同时，问句复述识别还可以被应用在多个生活领域，如智能问答机器人[3]、搜索引擎等。

深度神经网络模型的出现推动了自然语言处理任务的研究，问句复述识别任务也不例外。卷积神经网络（Convolutional Neural Network, CNN）[4]具有处理速度快和特征提取能力强等优点，经常被应用在机器翻译和文本分类[5]任务中。循环神经网络（Recurrent Neural Network, RNN）、长短时记忆

收稿日期：2020-03-25 **定稿日期**：2020-04-11

基金项目：国家自然科学基金（61672367，61672368）；国家重点研发计划（2017YFB1002104）；江苏省研究生科研与实践创新计划（SJCX19_0926）

网络(Long Short Term Memory Network, LSTM)[6]，不仅可以处理串行化数据，还可以保留上下文信息，在文本处理任务中被广泛应用。前人在这些基本神经网络基础上，提出了大量优秀模型，用于问句复述识别任务的研究。问句复述识别通常被当作二分类任务，从问句语义理解和问句对的联合特征抽取两个方向进行研究。因此，模型可以被分为基于问句语义理解和基于融合语义抽取特征两类。Severyn 等人提出的 Convnet 模型[7]和 Wang 等人提出的 BiMPM 模型[8]都是为了获得更好的问句表示，属于基于问句语义理解一类。Gong 等人提出的 DIIN 模型[9]，则更关注问句对之间的交互和融合，属于基于融合语义抽取特征一类。以上方法都是将问句编码为统一向量表示，再对两个问句进行比较。然而，人们在日常生活中比较两个句子时，通常是根据两个问句的相同点和不同点进行判断的。因此，本文提出了语义正交化匹配方法(Semantic Orthogonal Matching Method, SOMM)，将语义正交化引入到该任务中，尝试找出两个问句的相似语义编码表示和差异语义编码表示。

通过使用 SOMM 方法，模型不仅可以获得一个问句相对于另一个问句的相似语义表示，而且可以获得一个问句相对于另一个问句的差异语义表示。比如输入问句 P 和问句 Q，该方法可以将问句 P 表示为 P_{diff} 和 P_{same} 两部分，其中，P_{diff} 是问句 P 相对于问句 Q 的差异语义表示，P_{same} 是问句 P 相对于问句 Q 的相似语义表示，同样地，对问句 Q 也会得到对应的 Q_{diff} 和 Q_{same}。每一个问句都会被拆分为差异表示和相似表示两部分。实验结果表明，应用该方法的模型获得了与前沿工作可比的性能。本文的主要贡献如下：

(1) 将语义正交化思想引入问句复述识别任务中，提出了语义正交化匹配方法，该方法获得的句子相似表示和差异表示，既增强了对问句语义的理解，又实现了多粒度的问句交互。

(2) 分别在中文复述识别语料 LCQMC[10]和英文复述识别语料 Quora[11]上进行实验，实验结果证明了语义正交化匹配方法的有效性，且发现该方法对编辑距离小的问句判别更为准确。

本文的组织结构如下，第 1 节主要回顾前人在问句复述识别任务上的相关工作；第 2 节具体介绍本文提出的 SOMM 方法；第 3 节给出实验部分；第 4 节分析和对比实验结果；第 5 节为总结与展望。

1　相关工作

问句复述识别是文本匹配的一个分支，早期的问句复述识别通过字符串相似度或词频等特征判断两个句子语义是否相同。随着对自然语言理解研究的深入，人们希望机器不仅可以理解问句语义，还可以识别问句语义是否一致。对深度神经网络建立二元分类模型，即可实现对问句的复述识别。根据处理问题的侧重点不同，可以将现有模型架构分为以下两类。

1.1　基于句子语义表示的模型架构

基于句子语义表示的模型架构分为单句语义表示模型和跨句语义表示模型，下面分别对两种模型的具体做法进行介绍。

1.1.1　单句语义表示模型

单句语义表示模型一般使用孪生神经网络(Siamese Network)或伪孪生神经网络(Pseudo-Siamese Network)。孪生神经网络由共享权重值的两个神经网络组成，而伪孪生神经网络的两个神经网络不共享权重。在问句复述识别任务中，首先将两个问句分别输入到神经网络中，得到问句的句子表示，再计算这两个句子表示的距离，作为两个问句的相似程度，最后根据相似度进行分类，获得复述识别结果。Convnet 模型是典型的基于单句语义表示的模型架构，该模型使用字符级输入，用 LSTM 等网络分别对问句进行句子编码，再计算两个句子表示的相似度，将得出的结果作为分类器的输入，进而判别两个句子是否互为复述关系。单句语义表示的模型架构简单，训练速度快，比较容易实现，但其缺点是：语义表示较为单一，没有考虑另外一个句子的语义影响，并且两个问句之间没有交互。

1.1.2　跨句语义表示模型

跨句语义表示模型是指在对句子进行编码表示的同时，结合另一个句子的语义信息，得到更丰富的语义表示。该类模型一般利用注意力机制[12]。在对句子编码的过程中，将两个句子中的注意力相关的语义编码到彼此的句子表示中，增强句子原有的语义表示。BiMPM 模型是跨句语义表示的代表，该模型首先使用双向长短时记忆网络(Bidirectional Long Short Term Memory Network, BiLSTM)[13]分别对问句进行编码，得到每个句子的前向语义表示和后向语义表示，再利用 4 种不同的注意力机制，将两个句子的前

向语义表示和后向语义表示分别进行交互，最后通过聚合交互的输出得到每个句子的语义表示。此种基于跨句语义的表示模型在一定程度上弥补了单句语义表示模型的缺点，但交互过程较为复杂。

1.2 基于句子融合语义抽取特征的模型架构

基于句子融合语义抽取特征的模型，主要是解决如何从融合语义中抽取出联合特征的问题。模型首先对两个句子的向量表示进行融合，再对融合向量进行特征提取。该类模型更多地关注句子表示的融合以及联合特征的抽取。在DIIN模型中，作者首先分别对两个句子进行编码，得到对应的向量表示，再将两个句子的表示按位融合，利用DenseNet[14]作为卷积特征提取器提取联合特征。这类模型架构可以更好地将句子特征融合，并抽取出高维的联合特征，但其无法获得更丰富的句子语义表示。

2 语义正交化匹配方法

2.1 模型概览

针对问句复述识别任务，本文从问句语义理解和问句对之间的交互关系两个方向进行探索，提出了SOMM方法。问句复述识别任务，即对给定的两个自然问句 P 和 Q，识别两者是否互为复述关系。其中句子 $P_w=(p_{w1},\cdots,p_{wi},\cdots,p_{wm})$，句子 $Q_w=(q_{w1},\cdots,q_{wj},\cdots,q_{wn})$，$m$、$n$ 分别为问句 P 和问句 Q 的长度，p_{wi}、q_{wj} 分别为问句 P 的第 i 个单词和问句 Q 的第 j 个单词。

本文提出的SOMM方法，可以被运用到现有的模型架构中获得更好的性能。SOMM方法的整体架构如图1所示，共有4层，分别为词向量层、语义正交化编码层、语义交互层和输出层。

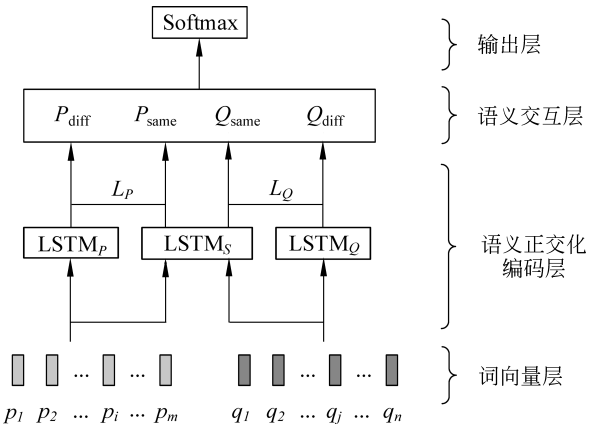

图1 使用SOMM方法的模型架构图

首先，模型通过词向量层，对问句中的每个单词进行向量表示，再将其输入到语义正交化编码层，获得句子的表示，两个句子表示在语义交互层进行融合，最后经过输出层获得判别结果。下面将通过一个例子具体介绍本文提出的模型框架。比如问句 P 为“港囧什么时候上映”，问句 Q 为“芈月传什么时候上映”。首先将分词后的两个问句分别经过词向量层进行句子编码，再将句子编码输入到语义正交化层。

在语义正交化层中，编码后的问句 P 和 Q 被依次输入到LSTM_S中，由于参数共享，编码时两个问句中的相似部分“什么时候上映”被突出，获得对应的P_{same}与Q_{same}。编码后的问句 P 同时被输入到LSTM_P中，在正交化损失L_P的限制下，问句 P 相对于问句 Q 的差异部分“港囧”被突出，获得P_{diff}，同理，问句 Q 相对于问句 P 的差异部分“芈月传”也被突出，获得Q_{diff}。在语义交互层，将两个问句的相似表示与差异表示分别进行比较，由于“港囧”和“芈月传”两个差异表示区别较大，最后在输出层就可以判定问句 P 和问句 Q 是不是复述关系。

2.1.1 词向量层

该层的主要功能是将句子中每个单词或短语编码为向量表示，由这些向量表示拼接得到向量矩阵，用于表示整个句子。单词的向量表示称为词向量，获得词向量通常有两种方法：一种是使用预训练词向量，如GloVe词向量[15]，该类词向量一般由语言模型在特定语料上训练得到，可以被直接使用；另一种是随机初始化的词向量，该类词向量在模型训练过程中会被不断更新。针对中文数据集，本文使用的是Shen等人[16]提供的预训练词向量，该词向量是由Shen等人在百度百科语料上使用Word2Vec模型训练得到的。针对英文数据集，本文使用的是GloVe词向量，该词向量由Pennington等人在维基百科语料库上使用词共现矩阵和GloVe模型学习得到。词向量层不仅能够将单词编码为词向量，还可以使用词性标注、句法树分析等预处理工具，将词法和句法信息也编码到句子向量表示中。词向量层将得到两个句子的向量表示，分别为$P_E:[p_{E1},\cdots p_{Ei},\cdots,p_{Em}]$和$Q_E:[q_{E1},\cdots q_{Ej},\cdots,q_{En}]$，其中$p_{Ei}$为句子 P 的第 i 个单词的向量表示，q_{Ej}为句子 Q 的第 j 个单词的表示。

2.1.2 语义正交化编码层

该层是对一般句子编码层的改进，将语义正交化的思想引入到句子编码中，使得问句在语义编码

时，既可以获得与另外一个问句的相似表示，也可以获得与另外一个问句的差异表示，从而丰富句子的向量表示。该层利用 BiLSTM 作为基础的句子编码器，将句子的每一个词向量依次输入到编码器中，就可以获得句子的隐状态表示。对于问句 P 或 Q 的第 t 个时刻，前后向 LSTM 输出的隐状态分别为 $\overrightarrow{h_t}$、$\overleftarrow{h_t}$，将两个方向的隐状态拼接得到联合隐状态 h_t，具体计算如式(1)～式(3)所示。

$$\overrightarrow{h_t}=\overrightarrow{\mathrm{LSTM}}(\overrightarrow{h_{t-1}},X)\quad t=1,\cdots,n \tag{1}$$

$$\overleftarrow{h_t}=\overleftarrow{\mathrm{LSTM}}(\overleftarrow{h_{t+1}},X)\quad t=n,\cdots,1 \tag{2}$$

$$\boldsymbol{h}_t=[\overrightarrow{h_t},\overleftarrow{h_t}]\quad t=1,\cdots,n \tag{3}$$

其中，X 为输入句子第 t 个时刻的词向量，既可以表示问句 P 的词向量又可以表示问句 Q 的词向量；n 为总步数；$[\cdot,\cdot]$ 为拼接操作。

在本层中，共使用 3 个不同的 BiLSTM，两个私有，一个公有。两个问句都分别被输入到私有 BiLSTM 和公有 BiLSTM 中，由于公有的 BiLSTM 为两个问句的共享网络，可以获得每个问句相对于另一个问句的相同部分的表示，即 P_{same} 和 Q_{same}。当加上正交限制后，私有的 BiLSTM 就可以获得每个问句相对于另一个问句的不同部分的特有表示，即 P_{diff} 和 Q_{diff}。这样就达到了将每个句子都拆分成相似表示和差异表示两个部分的效果。本文将在 2.2 节具体介绍语义正交化方法。

2.1.3　语义交互层

该层主要是对语义向量的交互和融合。在第 1 节的相关工作中，本文介绍了三类现有的模型架构。按照模型架构的不同，语义交互层也存在三种不同的交互方式。

对于基于单句语义表示的模型架构，语义交互层如图 2 所示。

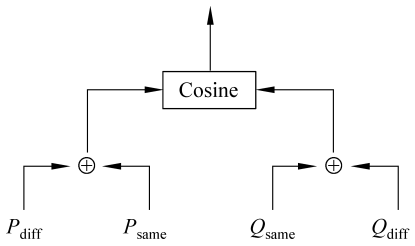

图 2　基于单句语义表示的语义交互层

将问句 P 表示为 P_{same} 与 P_{diff} 的拼接，问句 Q 表示为 Q_{same} 与 Q_{diff} 的拼接，对这两个问句拼接后表示计算余弦相似度，作为分类器的输入，该过程计算如式(4)～式(6)所示。

$$P=[P_{\mathrm{same}},P_{\mathrm{diff}}] \tag{4}$$

$$Q=[Q_{\mathrm{same}},Q_{\mathrm{diff}}] \tag{5}$$

$$h_{\mathrm{last}}=\mathrm{Cosine}(P,Q) \tag{6}$$

其中，Cosine()为余弦相似度函数，h_{last} 为该层的语义交互结果。

对于基于跨句语义表示的模型架构，语义交互层如图 3 所示。将 P_{same} 与 Q_{same} 使用双向注意力机制进行交互，得到 P'_{same} 和 Q'_{same}，将 P_{diff} 与 Q_{diff} 也使用双向注意力机制进行交互，得到 P'_{diff} 和 Q'_{diff}。问句 P 表示为 P'_{same} 和 P'_{diff} 拼接；问句 Q 表示为 Q'_{same} 和 Q'_{diff} 拼接，最后将两个问句的表示拼接，作为分类器的输入，该过程计算如式(7)～式(9)所示。

$$P'_{\mathrm{same}},Q'_{\mathrm{same}}=\mathrm{Attention}(P_{\mathrm{same}},Q_{\mathrm{same}}) \tag{7}$$

$$P'_{\mathrm{diff}},Q'_{\mathrm{diff}}=\mathrm{Attention}(P_{\mathrm{diff}},Q_{\mathrm{diff}}) \tag{8}$$

$$\boldsymbol{h}_{\mathrm{last}}=[[P_{\mathrm{same}},P_{\mathrm{diff}}],[Q_{\mathrm{same}},Q_{\mathrm{diff}}]] \tag{9}$$

其中，Attention 为双向注意力机制。

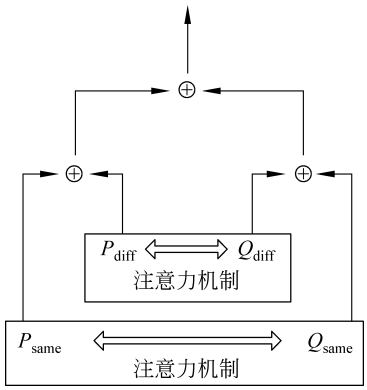

图 3　基于跨句语义表示的语义交互层

针对基于句子融合语义抽取特征的模型架构，语义交互层如图 4 所示。首先将问句 P、Q 分别表示为 P_{same} 与 P_{diff}、Q_{same} 与 Q_{diff} 的拼接，如式(4)、式(5)所示。再将两个句子表示按位融合，利用 DenseNet 作为卷积特征提取器提取联合特征，并将这联合特征作为分类依据，$\boldsymbol{h}_{\mathrm{last}}$ 的计算如式(10)所示。

$$\boldsymbol{h}_{\mathrm{last}}=\mathrm{DenseNet}(P\odot Q) \tag{10}$$

其中，$\odot$ 表示按位相乘运算。

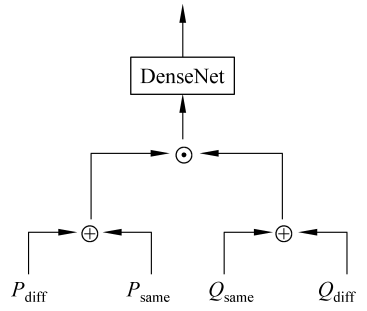

图 4　基于融合语义抽取特征的语义交互层

2.1.4 输出层

该层是对已经获得的特征向量 $\boldsymbol{h}_{\text{last}}$ 进行解码，一般使用全连接网络和 softmax 函数进行二分类，预测出每个分类的概率，预测如式(11)所示。

$$\hat{y}=\text{softmax}(\boldsymbol{W}\boldsymbol{h}_{\text{last}}+b) \tag{11}$$

其中，$\hat{y}$ 为预测类别的概率，W 为需要训练的权重，b 为偏置项。

2.2 正交化匹配方法

前人在解决复述识别问题时，通常是将每个句子进行编码，得到一个统一的句子表示。这个统一的句子表示既包含了和另外一个句子相关的部分，也包含了和另外一个句子不相关的部分。复述识别任务的核心问题是比较两个问句的相同点和不同点。

这种统一的句子表示，使得模型不仅要提取出两个问句的不同点和相同点，还要对其进行比较，增加了模型的复杂度。常见的区分两个句子的方法有两种：第一种方法是利用注意力机制，根据相关性系数，使得每个问句都可以注意到另一个问句的信息。这种方法只能在一定程度上反映每个语义片段和另外一个问句中的相关性，但不能很好地区分相关与不相关部分；第二种方法是将句子表示进行融合，在融合的语义表示中抽取联合特征，用来当作复述关系的判别依据。这一种方法利用联合特征作为判断依据，只利用了两个问句的相关部分的语义信息，而没有利用不相关部分的语义信息，这就无法做到两个句子语义信息的充分比较与交互。

对于给定的两个问句，若能抽取出两个问句的相关语义信息和不相关语义信息，就可以实现多粒度的语义信息比较和交互，使得复述关系的识别更为准确。因此，本文提出 SOMM 方法，使句子语义正交化，实现对句子信息的拆分，利用拆分后的句子表示进行句子之间的交互。在第 4 节将会给出具体的对比结果。如图 1 语义正交化编码层所示，对于给定的问句 P 和问句 Q，模型利用 3 个不同的 BiLSTM 对其进行编码，分别为私有特征语义编码器LSTM_P、LSTM_Q和公有特征语义编码器LSTM_S。

公有特征语义编码器LSTM_S被两个问句共享使用，问句 P 和问句 Q 被分别输入到该网络中，利用网络权重值共享，得到问句 P 相对于问句 Q 的相似编码表示 P_{same} 以及问句 Q 相对于问句 P 的相似编码表示 Q_{same}。P_{same} 和 Q_{same} 的计算如式(12)、式(13)所示。

$$P_{\text{same}}=\text{LSTM}_S(P_E) \tag{12}$$

$$Q_{\text{same}}=\text{LSTM}_S(Q_E) \tag{13}$$

其中，P_E、Q_E 分别为句子 P 和 Q 的词向量表示。

私有特征语义编码器LSTM_P用于对问句 P 进行编码，在加入正交限制后，就获得了问句 P 相对于问句 Q 不同的编码表示 P_{diff}；私有特征语义编码器 LSTM_Q 用于对问句 Q 进行编码，同样加入正交限制，获得问句 Q 相对于问句 P 不同的编码表示 Q_{diff}。P_{diff}、Q_{diff} 的计算如式(14)、式(15)所示。

$$P_{\text{diff}}=\text{LSTM}_P(P_E) \tag{14}$$

$$Q_{\text{diff}}=\text{LSTM}_Q(Q_E) \tag{15}$$

其中，P_E、Q_E 分别为句子 P 和 Q 的词向量表示。

句子表示的正交化通过损失函数实现，本文使用的是 Bousmails 等人[17]提出的损失函数，其计算如式(16)所示。

$$L_{\text{orth}}(S,H)=\|S^TH\|_F^2 \tag{16}$$

其中，$\|\cdot\|_F^2$ 为弗罗贝尼乌斯范数，S 和 H 为两个需要被正交化限制的矩阵。本文还使用了交叉熵损失函数计算模型真实值与预测值的损失，具体如式(17)所示。

$$L(\hat{y},y)=\sum_{i=1}^{N}\sum_{j=1}^{C}y_i^j\log(\hat{y}_i^j) \tag{17}$$

其中，y_i^j 表示真实类别标签；$\hat{y}_i^j$ 表示模型预测的概率值；N 表示训练样本的总个数；C 表示类别的总个数。因此，模型最终的损失函数如式(18)所示。

$$L_{\text{all}}=L+L_P+L_Q \tag{18}$$

其中，L_{all} 为总的损失，L 为交叉熵损失，可由式(17)计算得出，L_P、L_Q 分别为句子 P 和 Q 的正交化损失，其计算如式(19)、式(20)所示。

$$L_P=L_{\text{orth}}(P_{\text{same}},P_{\text{diff}}) \tag{19}$$

$$L_Q=L_{\text{orth}}(Q_{\text{same}},Q_{\text{diff}}) \tag{20}$$

3 实验配置

本节主要介绍实验的相关配置，包括使用的数据集、评价指标以及实验参数设置。

3.1 数据集与评价指标

本文分别在两个数据集上进行实验，验证 SOMM 方法在问句复述识别任务中的有效性。

3.1.1 LCQMC 数据集与评价指标

Liu 等人[10]提出了大规模中文数据集 LCQMC，该数据集包含大量的中文问句对，每一个问句对都有一个类别标签，表示该问句对是否互为复述，其正负样本分布如表 1 所示。在本文中，LCQMC 使用的数

据集划分与原作者发布的一致。

表 1　LCQMC 数据集的样本分布

数据集	总共	正样本	负样本
训练集	238 766	138 574	100 192
开发集	8 802	4 402	4 400
测试集	12 500	6 250	6 250

在 LCQMC 数据集上，本文采用 4 个指标对其性能进行评估，分别为精确率 P，召回率 R，调和平均值 F_1 值 、准确率 Acc。

3.1.2　Quora 数据集与评价指标

Quora 数据集为问句复述识别任务中常用的英文数据集，该数据集包含大量英文问句对及其对应的类别标签，其数据量的大小以及对应的正负样本分布如表 2 所示。在本文的实验中，使用的 Quora 数据集划分与 BiMPM 论文中的数据集划分一致。

表 2　Quora 数据集的样本分布

数据集	总共	正样本	负样本
训练集	384 348	139 306	245 042
开发集	10 000	5 000	5 000
测试集	10 000	5 000	5 000

在 Quora 数据集上，本文采用准确率 Acc 评估模型性能。

3.2　参数设置

对于中文数据集 LCQMC，本文使用结巴分词对问句进行处理，得到每个问句对应的单词，再使用 Shen 等人提出的中文预训练词向量，将单词映射成向量表示。对于英文数据集 Quora，直接使用 300 维的 GloVe 词向量对问句中的单词进行向量映射。在训练阶段，batch_size 为 64，LSTM 的隐藏单元为 200，优化函数为 Adamax，学习率为 0.002。

本文实验平台配置如下：操作系统为 CentOS 7.5，显卡型号为 GTX 1080 Ti，显存大小为 12GB。本文使用 Pytorch 深度学习框架进行实验，Python 版本为 3.6.3。

4　实验结果分析

本节主要介绍 SOMM 方法和业内其他方法在 LCQMC 及 Quora 数据集上的实验结果对比，针对结果进行相关分析；并探索了不同数据分布对 SOMM 方法的影响。

4.1　LCQMC 实验结果

本节主要介绍 SOMM 方法在 LCQMC 数据集上的实验结果。为了充分验证该方法的有效性，本文在现有的两类模型架构上使用 SOMM 方法进行实验，即基于句子语义表示的模型架构和基于句子融合语义抽取特征的模型架构，其中，基于句子语义表示的模型架构又分为单句语义表示和跨句语义表示。对于每一种模型架构，本文都选取了一个具有代表性的模型作为基础模型。三个基础模型的实现方式如下。

对于基于单句语义表示的模型架构，本文使用的基础模型架构如下：将两个问句输入到词向量层和 BiLSTM 中，得到每个问句的向量表示，再根据两个向量表示的余弦相似度进行分类，判断两个问句是否互为复述。

对于基于跨句语义表示的模型架构，本文使用 BiMPM 模型架构作为基础架构。为了获得句子的跨句语义表示，该模型首先使用 BiLSTM 获得句子的前向表示和后向表示，然后使用 4 种不同的注意力机制分别对两个句子的前向表示和后向表示进行交互融合，获得每个句子的跨句语义表示，最后将两个问句的跨句语义表示进行拼接融合，输入分类器。

针对基于句子融合语义抽取特征的模型架构，本文将 Gong 等人提出的 IIN 作为基础架构，首先将两个句子进行编码得到对应的向量表示，再将两个句子的表示按位融合，利用 DenseNet 作为卷积特征提取器提取联合特征，并将联合特征作为分类依据。

本文分别在三种基础架构上使用 SOMM 方法并进行对比实验。首先，将基于单句语义表示的交互层与 SOMM 集成，形成第一套实验系统；其次，将上述系统中的单句语义表示替换为跨句语义表示，从而形成第二套实验系统；最后，将融合语义特征的交互层与 SOMM 进行集成，形成第三套实验系统。本文 2.1 节对上述架构给出了详细解释，并分别对应于图 2～图 4。在数据集 LCQMC 上的实验结果如表 3 所示。

表 3　LCQMC 语义正交化实验结果

模型	P	R	F_1	Acc
BiLSTM	77.02	91.84	83.78	82.22
BiLSTM＋SOMM	80.66	88.91	84.59	83.80
BiMPM	78.55	91.76	84.64	83.35
BiMPM＋SOMM	81.96	91.31	86.38	85.61
IIN	80.84	88.82	82.87	82.43
INN＋SOMM	81.49	90.46	85.74	84.96

从表 3 的实验结果可以看出，无论在基于语义表示的模型架构，还是在基于句子融合语义抽取特征的模型架构，使用本文提出的 SOMM 方法，在调和平均值 F_1 值与准确率 Acc 上都得到了显著的提升，充分证明了 SOMM 方法的有效性。对于单语义模型 BiLSTM，在使用 SOMM 方法后，F_1 值提升了 0.81%，Acc 提升了 1.58%；对于跨句语义模型 BiMPM，在使用 SOMM 方法后，F_1 值提升了 1.74%，Acc 提升了 2.26%；对于语义融合抽取模型 IIN，在使用 SOMM 方法后，F_1 值提升了 2.87%，Acc 提升了 2.53%。

为了与现有模型的性能进行比较，表 4 列举了现有模型在 LCQMC 上的实验结果。结果表明，本文在 BiMPM 模型上使用 SOMM 方法后，无论词向量层使用字符级还是单词级，调和平均值 F_1 与准确率 Acc 都超过了现有模型的性能，证明了 SOMM 方法的有效性。

表 4 LCQMC 实验结果

模型	P	R	F_1	Acc
CNN_{char}	67.1	85.6	75.2	71.8
CNN_{word}	68.4	84.6	75.7	72.8
$BiLSTM_{char}$	67.4	91.0	77.5	73.5
$BiLSTM_{word}$	70.6	89.3	78.92	76.1
$BiMPM_{char}$	77.6	93.9	85.0	83.4
$BiMPM_{word}$	77.7	93.5	84.9	83.3
DFF_{char}	78.58	93.88	85.51	84.15
DFF_{word}	77.69	94.08	85.06	83.53
$Ours_{char}$	**79.14**	**93.70**	**85.80**	**84.50**
$Ours_{word}$	**81.96**	**91.31**	**86.38**	**85.61**

当模型按字符输入时，F_1 值分别比 CNN、BiLSTM、BiMPM、DFF 高了 10.6%、8.3%、0.8%、0.29%，Acc 值分别比 CNN、BiLSTM、BiMPM、DFF 高了 12.7%、11%、1.1%、0.35%。当模型按单词输入时，F_1 值分别比 CNN、BiLSTM、BiMPM、DFF 高了 10.68%、7.46%、1.48%、1.32%，Acc 值分别比 CNN、BiLSTM、BiMPM、DFF 高了 12.81%、9.51%、2.31%、2.08%。

4.2 Quora 实验结果

在 Quora 语料上，为了验证语义正交化方法的有效性，本文使用 BiMPM 模型作为基础架构进行实验。实验结果如表 5 所示。

表 5 Quora 语义正交化实验结果

模型	开发集	测试集
BiMPM	86.20	86.33
BiMPM+SOMM	87.97	88.32

由于实验环境或其他因素限制，本文未重现出 BiMPM 模型原作者在 Quora 语料上的实验结果，即在开发集上准确率为 88.69%，测试集上准确率为 88.17%，但本实验的核心是验证语义正交化方法的有效性。实验结果表明，BiMPM 在加上语义正交化方法后，准确率在 Quora 语料的开发集上提升了 1.77%，在测试集上提升了 1.99%。表 6 为现有模型在 Quora 数据集上的结果。

表 6 Quora 实验结果

模型	开发集	测试集
BiMPM	88.69	88.17
$FFNN_{word}$	85.07	84.35
$FFNN_{char}$	86.01	85.06
$DECATT_{paralex\text{-}char}$	87.8	87.77
pt-DECAT T_{word}	88.44	87.57
pt-DECAT T_{char}	88.89	88.4
DIIN	89.44	89.06
DFF	88.61	88.83
Ours	**87.97**	**88.32**

4.3 测试结果分析

为了进一步探索数据分布对 SOMM 方法的影响，本节针对测试集做了补充实验。LCQMC 的测试集共有 12 500 条，Quora 的测试集共有 10 000 条。本文为了观测模型在测试集上的性能，将两个测试集按照莱文斯坦距离进行划分。莱文斯坦距离又称 Levenshtein 距离，是编辑距离的一种，常用来比较两个字符串之间的相似度[18]。下文的编辑距离都是指莱文斯坦距离。该距离是指对于两个字符串，一个字符串转换为另外一个字符串所需要的最少编辑次数，其中，编辑操作分为插入、删除和替换。

本文使用编辑距离衡量两个问句的差异程度。编辑距离越大，两个字符串差距越大，两个问句的差异程度也越大。在中文数据集 LCQMC 中，两个问句之间的编辑距离是指一个问句转换为另一个问句所需的最少编辑汉字的次数。在英文数据集 Quora 中，两个问句之间的编辑距离是指一个问句转换为另一个问句所需的最少编辑单词的次数。图 5 是两个测试集按编辑距离划分的数据分布图，横坐标为编辑距离的范围，纵坐标为具体的数据量。

图 5 表明，LCQMC 测试集整体的编辑距离较小，10 以内的编辑距离占据了总测试集数量的 92% 以上，而 Quora 测试集的编辑距离分布较为均衡，

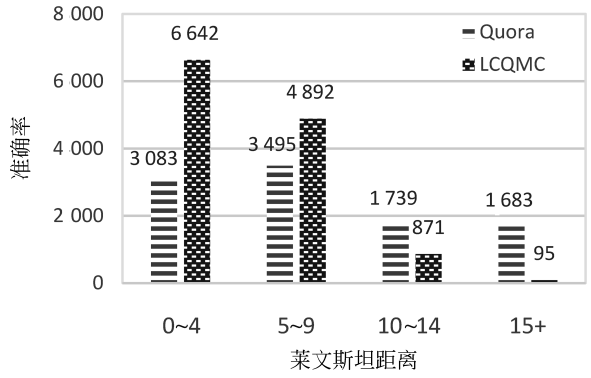

图 5　测试集数据分布图

但编辑距离在 10 以内的也占据了 65%以上。由此可以看出，两个测试集的问句对大部分都很相似。依据图 5 的数据划分，在基础模型 BiMPM 上使用 2.1 节中介绍的 SOMM 方法进行对比实验，并比较这两个模型在不同数据分布上的性能表现。

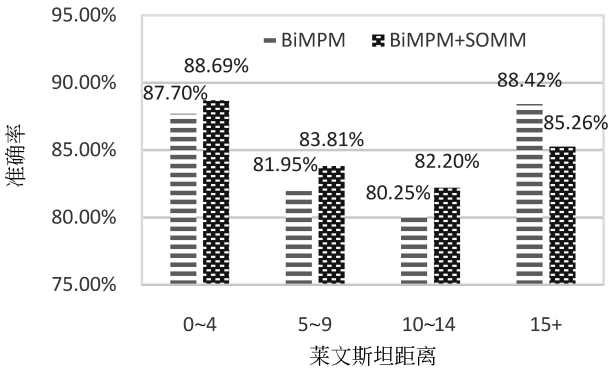

图 6　准确率分布图——LCQMC

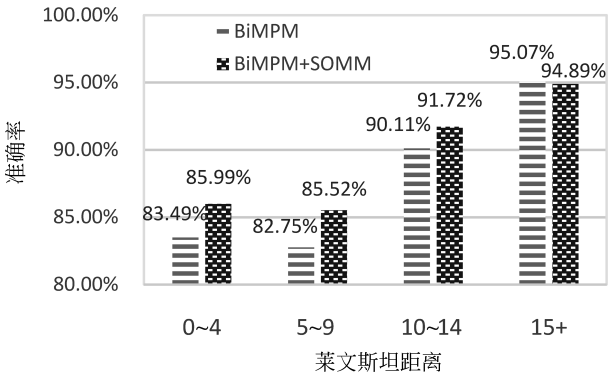

图 7　准确率分布图—Quora

图 6 为模型在 LCQMC 测试集上的实验结果，图 7 为模型在 Quora 测试集上的实验结果。其中，横坐标为编辑距离的范围，纵坐标为模型的准确率。实验结果表明，无论是 LCQMC 数据集，还是 Quora 数据集，在编辑距离为 15 以内的测试数据上，使用了 SOMM 方法的 BiMPM 模型在评价指标准确率上均有所提升。这表明 SOMM 方法在编辑距离小于 15 的数据上有着更好的表现。对于编辑距离较小的问句对，SOMM 方法可以更准确地区分出两个问句的相同点和不同点，再将两个问句的相同点和不同点分别进行比较，就得出了较为准确的判断依据。

例 1 是 LCQMC 测试集上的两个问句对：

问句 1：木瓜牛奶怎么煮？

问句 2：怎么做木瓜炖牛奶？

标签：1

问句 3：欣赏是什么意思？

问句 4：欣赏的赏是什么意思？

标签：0　　（例 1）

基础模型 BiMPM 对这两个问句对都预测错误，但在加上 SOMM 方法后都预测正确。例 1 中的问句 1 与问句 2 互为复述，问句 3 与问句 4 不是复述关系。虽然两个问句对的编辑距离都为 2，但是对应的标签却不相同。正是因为基础模型 BiMPM 使用了 SOMM 方法，获得了更好的性能。

SOMM 方法既区分出了两个问句的不同点和相同点，又使得每个问句的相似语义表示与差异语义表示相互关联，不仅丰富了问句的表示，又使得两个问句之间充分交互，才得到了更为准确的预测结果。

5　总结与展望

本文针对问句复述识别任务的特点，将 SOMM 方法应用到该任务中，实现了对问句语义的拆分。通过 SOMM 方法，每个问句都被拆分为两个部分，即与另一个问句的相关部分和不相关部分，得到一种新型的问句表示。这种新型的问句表示，可以直接被应用在基于单句语义模型架构或基于融合语义抽取特征的模型架构上，也可以在基于跨句语义模型的架构上实现句子之间的多粒度语义交互。本文在中文数据集 LCQMC 和英文数据集 Quora 上进行实验，验证了 SOMM 方法在问句复述识别任务上的有效性。为了探索 SOMM 方法在不同数据分布上的性能表现，本文还将两个数据集的测试集按照编辑距离划分，观测模型性能，证明了使用 SOMM 方法的模型更适用于编辑距离小于 15 以内的数据。

在未来工作中，将进一步完善 SOMM 方法在问句复述识别任务中的应用，将基于跨句语义的模型架构与基于融合语义抽取特征的模型架构相结合，实现更深层次的语义理解和交互。同时，也会尝试将

SOMM方法应用到其他自然语言理解任务中。目前,问句复述识别模型主要是基于语义相似度,还未实现基于意图相似度。大规模的数据集和意图相似的准确定义都将成为未来的重要研究方向。

参考文献

[1] Wu Y, Schuster M, Chen Z, et al. Google′s neural machine translation system: Bridging the gap between human and machine translation[J]. arXiv preprint arXiv:1609.08144, 2016.

[2] Lan W, Xu W. Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering[C]//Proceedings of the 27th International Conference on Computational Linguistics, 2018: 3890-3902.

[3] 文博. 面向智能客服机器人的交互式问句理解研究[D]. 哈尔滨: 哈尔滨工业大学硕士学位论文, 2014.

[4] Kalchbrenner N, Grefenstette E, Blunsom P. A convolutional neural network for modelling sentences[J]. arXiv preprint arXiv:1404.2188, 2017.

[5] Conneau A, Schwenk H, Barrault L, et al. Very deep convolutional networks for text classification[J]. arXiv preprint arXiv:1606.01781, 2016.

[6] Chen Q, Zhu X, Ling Z, et al. Enhanced LSTM for natural language inference[J]. arXiv preprint arXiv: 1609.06038, 2016.

[7] Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks[C]// Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015: 373-382.

[8] Wang Z, Hamza W, Florian R. Bilateral multi-perspective matching for natural language sentences[J]. arXiv preprint arXiv:1702.03814, 2017.

[9] Gong Y, Luo H, Zhang J. Natural language inference over interaction space[J]. arXiv preprint arXiv:1709.04348, 2017.

[10] Liu X, Chen Q, Deng C, et al. LCQMC: A large-scale Chinese question matching corpus[C]//Proceedings of the 27th International Conference on Computational Linguistics, 2018: 1952-1962.

[11] Chen Z, Zhang H, Zhang X, et al. Quora question pairs[DS/OL]. [2020-2-28]. https://www.kaggle.com/c/quora-question-pairs.

[12] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017: 6000-6010.

[13] Zhou P, Shi W, Tian J, et al. Attention-based bidirectional long short-term memory networks for relation classification[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016: 207-212.

[14] Iandola F, Moskewicz M, Karayev S, et al. DenseNet: Implementing efficient convnet descriptor pyramids[J]. arXiv preprint arXiv:1404.1869, 2014.

[15] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014: 1532-1543.

[16] Shen L, Zhe Z, Renfen H, et al. Analogical reasoning on Chinese morphological and semantic relations[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018: 138-143.

[17] Bousmalis K, Trigeorgis G, Silberman N, et al. Domain separation networks[C]//Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016: 343-351.

[18] 刁兴春, 谭明超, 曹建军. 一种融合多种编辑距离的字符串相似度计算方法[J]. 计算器应用研究, 2010, 27(12): 4523-4525.

朱朦朦(1994—),硕士研究生,主要研究领域为文本匹配、自动问答。
E-mail: mmzhu01@gmail.com

洪宇(1978—),通信作者,博士,教授,主要研究领域为自动问答、问题生成。
E-mail: tianxianer@gmail.com

武恺莉(1996—),硕士研究生,主要研究领域为问题生成、自动问答。
E-mail: wukaili0112@gmail.com