北京大学学报(自然科学版) 第 58 卷 第 1 期 2022 年 1 月
Acta Scientiarum Naturalium Universitatis Pekinensis, Vol. 58, No. 1 (Jan. 2022)
doi: 10.13209/j.0479-8023.2021.101

# 结合自监督学习的多任务文本语义匹配方法

陈源[1] 丘心颖[1,2,†]

1. 广东外语外贸大学信息科学与技术学院, 广州 510006; 2. 广州市非通用语种智能处理实验室, 广东外语外贸大学, 广州 510006; † 通信作者, E-mail: xy.qiu@foxmail.com

**摘要** 基于文本交互信息对文本语义匹配模型的重要性, 提出一种结合序列生成任务的自监督学习方法。该方法利用自监督模型提取的文本数据对的交互信息, 以特征增强的方式辅助基于神经网络的语义匹配模型, 构建多任务的文本匹配模型。9 个模型的实验结果表明, 加入自监督学习模块后, 原始模型的效果都有不同程度的提升, 表明所提方法可以有效地改进深度文本语义匹配模型。

**关键词** 自监督学习; 文本语义匹配; 多任务学习

# Multi-task Semantic Matching with Self-supervised Learning

CHEN Yuan[1], QIU Xinying[1,2,†]

1. School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou 510006; 2. Guangzhou Key Laboratory of Multilingual Intelligent Processing, Guangdong University of Foreign Studies, Guangzhou 510006; † Corresponding author, E-mail: xy.qiu@foxmail.com

**Abstract** In semantic matching, the interaction information between pairs of texts is critical in predicting a matching score for the pairs. This paper proposes a multi-task learning framework with self-supervised learning for deep learning semantic matching problem. Specifically, a self-supervised model is designed for the paired sentences to regenerate each other with sequence-to-sequence generation method. Then a multi-task learning framework integrates the representation from the self-supervised generation with that of the deep matching model to predict the similarity score of the texts. Experimentations with 9 deep matching models prove that the proposed framework can improve the performances of the traditional deep matching models.

**Key words** self-supervised learning; semantic matching; multi-task learning

文本语义匹配研究两个文本之间语义等价的度量或语义相似匹配度问题, 是自然语言处理的基础任务之一。在基于深度神经网络信息检索(neural IR)的研究中, 文本语义匹配(semantic matching)旨在通过对两个文本进行分布式表示建模, 用更丰富的形式表示查询和文档的含义, 实现基于相似性学习的语义匹配[1−3]。语义匹配的计算方法可用于处理 Query-Doc 搜索[4]、Question-Answer 匹配[5]和自然语言推理[6]等任务。语义相似度的计算结果可以用于辅助自然语言处理领域的其他任务, 例如文本聚类[7]和机器翻译[8]。

文本语义匹配方法可以分为四大类: 基于字符串的方法、基于语料库的方法、基于世界知识的方法和其他方法[9]。基于字符串的方法只计算字符串的匹配程度, 不考虑语义信息; 基于语料库的方法通过构建词袋模型或者利用搜索引擎, 从现有的语料库中得到计算文本相似度的信息; 基于世界知识的方法从规范的知识库中提取信息计算相似度; 其他方法包括句法分析和多种方法互相结合的混合方法。近几年, 神经网络构建深度文本语义匹配模型的方法得到广泛的应用, 取得一些重要的进展。这些模型最初用于信息检索领域中查询项与文档之间

国家社会科学基金(17BGL068)和广东省自然科学基金(2018A030313777)资助
收稿日期: 2021−06−08; 修回日期: 2021−08−14

的相似性度量, 同样也可以解决文本语义匹配问题。Guo 等[10]将现有的深度文本语义匹配模型分为基于表示的模型[4,11−12]、基于交互的模型[11,13−16]以及混合模型[17], 基于表示的模型为每个文本分别构建固定维度的向量表示, 然后在潜在空间内执行匹配; 基于交互的模型计算两个文本词汇之间的交互(这种交互可以是标识值或者是句法或语义相似值), 然后从交互矩阵中整合得出匹配分数; 混合模型主要指多种机制混合的模型, 例如DUET[17]模型由两个子模型组成, 用于提取句子对的不同特征。这3类深度文本语义匹配模型各有优势。一个好的文本语义匹配模型不仅能够有效地学习文本的语义信息, 还能够捕获文本之间的交互信息。

近年来, 自监督学习(self-supervised learning)受到广泛关注。Liu 等[18]将自监督学习模型归纳为三大类, 分别是生成式模型(generative)、对比式模型(contrastive)和对抗式模型(adversarial)。生成式模型以自编码器(autoencoder)为代表; 对比式模型通过对比正负样本来学习表示; 对抗式模型保留由编码器和解码器组成的生成器结构, 生成器可为对抗式模型提供强大的学习表示能力。在自然语言处理领域中, 在自监督学习概念提出之前, 已经有语言模型体现自监督学习的思想。例如, Mikolov 等[19]提出的 Word2Vec 模型实现中心词预测(CBOW 模型)和邻近词预测(Skip-Gram 模型); Skip-Thought Vectors[20]、BERT[21]以及之后提出的诸多预训练语言模型的预训练任务中包含自监督学习的任务。自监督学习框架包含一个区别于下游核心任务的自动打标签的 Pretext(辅助)任务, 通过数据不同部分的交互, 实现数据的学习表示, 并将学习到的中间特征层表示或者模型权重用于下游的监督学习预测任务中, 从而降低对大量标记数据的需求, 并且充分利用每条数据可能关联的多种模式。

多任务学习[22]旨在利用任务之间相互联系的信息来改进模型的泛化性能。Lee 等[23]使用一个与本文模型框架类似的多任务学习模型, 探讨将图片的旋转预测任务和常规的图片分类任务同时训练。其中旋转预测任务是一个自监督学习模型。其 Pretext 任务是将图片进行翻转, 通过预测翻转角度实现自监督学习模型的构建。根据深度神经网络中参数共享的方式, 多任务学习模型结构分为硬参数共享和软参数共享[24]。硬参数共享通常是多个任务共享隐藏层的参数, 输出时进行分支。软参数共享指每个任务都有自己的模型和参数, 然后使用一定的机制(如 L2 距离)在模型之间建立联系。

本文在现有的深度文本语义匹配模型基础上, 设计自监督学习来提取文本之间更深层次的交互信息, 提升基于深度神经网络的文本语义匹配的效果, 包含以下两个机制。

1) 本文设计的自监督 Pretext 任务通过句子对互换生成, 获取文本之间基于序列转换的深层交互信息, 以特征增强的方式来辅助现有的深度文本语义匹配模型。

2) 为了让自监督学习模型提取的交互信息能够与深度文本匹配模型建立动态关联, 本文使用多任务学习的方式, 将自监督学习模型训练过程中提取的交互信息作为动态特征, 参与文本语义匹配任务的调优, 以此改善两个模型的泛化性能, 使得提取的交互信息可以参与到深度文本语义匹配模型的学习中。

为了探讨所提出的自监督任务和多任务学习对基于表示、基于交互和混合模型的改进效果, 本文提出以下两个研究问题(research question, RQ)。

RQ1: 对于基于表示的文本语义匹配模型, 因其缺乏对交互信息的提取, 自监督学习模型提取的序列转换的交互信息能否成功地弥补这些模型的不足, 达到提升模型效果的目的?

RQ2: 对于基于交互的模型和混合模型, 因其已经提取文本之间的交互信息, 自监督学习模型提取的序列转换的交互信息是否会冗余?

本文选取 9 个基于神经网络的文本语义匹配模型, 在 5 份公开数据集上进行试验, 验证结合自监督学习的多任务文本语义匹配方法。

## 1 研究方法

### 1.1 模型框架

本文在现有深度文本语义匹配模型的基础上, 使用自监督学习模型提取句子对之间序列转换的交互信息, 并使用多任务学习的方式, 将提取的交互信息动态地参与深度文本语义匹配模型的训练。本文框架分为原始模型(original model, OM)和自监督模型(self-supervised Model, SSM)两部分(图 1)。整体框架采用多任务学习的硬参数共享, 为这两部分模型构建联系。

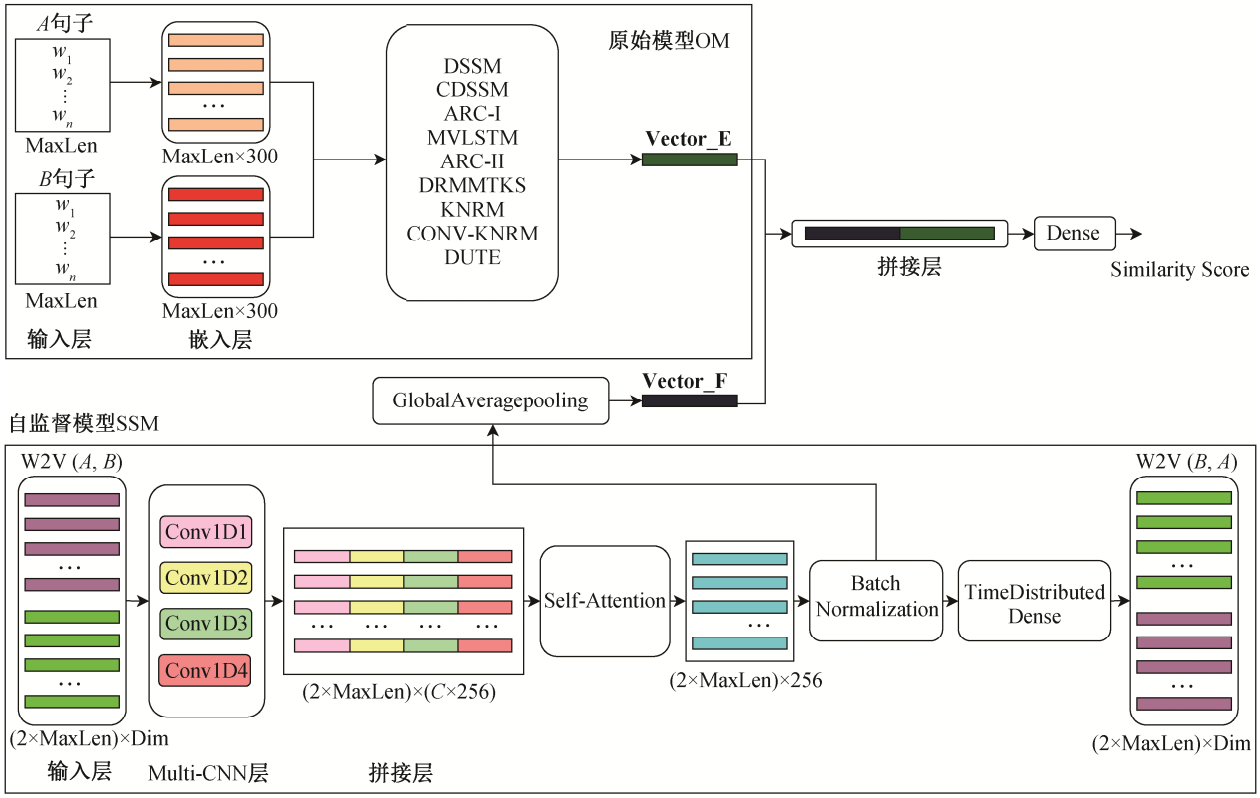


**图 1　模型框架**

Fig. 1　Model structure

## 1.2 基于深度神经网络的文本语义匹配原始模型(OM)

我们复现了 MatchZoo 工具库[25]中提供的 9 个基于深度神经网络的文本匹配模型，在本文中用 TSSM(text semantic matching model)表示。原始模型通过 TSSM，学习得到两个句子的特征交互向量 **Vector_E**。**Vector_E** 的计算方法如下。

1) 给定两个句子集合 $A=\{a_1, a_2, \ldots, a_n\}$ 和 $B=\{b_1, b_2, \ldots, b_n\}$，各有 $n$ 个句子，并组成 $n$ 个句对的数据集。将第 $i(i\in[1, n])$ 个句对中的 $A$ 句子表示为 $a_i=\{w_1^{a_i}, w_2^{a_i}, ..., w_m^{a_i}\}$，且 $w_x^{a_i}(x\in[1, m])$ 表示句子 $a_i$ 的第 $x$ 个字符/单词(中文文本为字符，英文文本为单词)。$m$ 表示图 1 中的 MaxLen，即句子序列的最大长度。同样，有 $b_i=\{w_1^{b_i}, w_2^{b_i}, ..., w_m^{b_i}\}$，$w_x^{b_i}(x\in[1, m])$。

2) 将句对 $i$ 的两个句子 $a_i$ 和 $b_i$ 通过 TSSM 的嵌入层得到嵌入表示，即矩阵 $\mathbf{Embed\_a}_i\in \mathrm{R}^{m\times \mathrm{Dim}}$ 和 $\mathbf{Embed\_b}_i\in R^{m\times \mathrm{Dim}}$，Dim 表示嵌入层维度，实验中设为 300。

3) 将句对 $i$ 的两个句子的嵌入表示输入 TSSM，得到 $\mathrm{Vector\_E}_i$:

$$\mathrm{Vector\_E}_i=\mathrm{TSSM}_i(\mathbf{Embed\_a}_i, \mathbf{Embed\_b}_i)。\quad (1)$$

4) 将 $\mathrm{Vector\_E}_i$ 输入以 Sigmoid 函数为激活函数的全连接层，得到两个句子的相似度分数 $\mathrm{Sim}_i$:

$$\mathrm{Sim}_i=\mathrm{Sigmoid}_i(W_o\mathrm{Vector\_E}_i+b_o),\quad (2)$$

其中，$W_o$ 和 $b_o$ 是可学习更新的参数。

将句子对的标签记为 $L=\{y_1, y_2, ..., y_n\}$，$y_i(i\in[1, n])$ 表示第 $i$ 个句子对的标签，用二分类交叉熵作为损失函数:

$$\mathrm{Loss_{OM}}=-(L\cdot\log(\mathrm{Sim}_i)+(1-L)\cdot\log(1-\mathrm{Sim}_i))。\quad (3)$$

## 1.3 自监督模型(SSM)

本文设计的自监督模型通过序列生成，用来提取句子对向量矩阵相互生成的交互信息，并将该交互信息用于辅助文本语义匹配任务。SSM 的 Pretext 任务是句子对的相互序列生成。具体算法如下。

1) SSM 的输入设计：将句对 $i$ 的两个句子 $a_i$ 和 $b_i$ 采用 Skip-gram 算法分别训练，得到 Word2Vec[21] 向量表示，即 $\boldsymbol{W}_{a_i}\in \mathrm{R}^{m\times \mathrm{Dim}}$，$\boldsymbol{W}_{b_i}\in \mathrm{R}^{m\times \mathrm{Dim}}$，$m$ 表示句子序列长度，Dim 表示向量维度，并拼接得到矩阵 $\mathbf{W2V\_AB}_i\in \mathrm{R}^{2m\times \mathrm{Dim}}$:

$$\mathbf{W2V\_AB}_i = \begin{bmatrix} \boldsymbol{W}_{a_i} \\ \boldsymbol{W}_{b_i} \end{bmatrix}。$$

2) SSM的输出设计: 将句对 $i$ 的两个句子 $b_i$ 和 $a_i$ 的 Word2Vec 向量表示拼接, 得到矩阵 $\mathbf{W2V\_BA}_i \in \mathrm{R}^{2m\times \mathrm{Dim}}$:

$$\mathbf{W2V\_BA}_i = \begin{bmatrix} \boldsymbol{W}_{b_i} \\ \boldsymbol{W}_{a_i} \end{bmatrix}。$$

SSM的输入是将句子对 $AB$ 的矩阵 $\mathbf{W2V\_AB}_i \in \mathrm{R}^{2m\times \mathrm{Dim}}$ 作为输入, SSM 框架的标签是 $\mathbf{W2V\_BA}_i \in \mathrm{R}^{2m\times \mathrm{Dim}}$。SSM在训练过程中不改变序列长度, 使得每一个输入向量的输出与输出向量一一对应, 以 $\mathbf{W2V\_AB}_i$ 生成 $\mathbf{W2V\_BA}_i$ 的训练方式, 可以使自监督模型提取的交互信息不仅蕴含两个句子的上下文语义信息, 也蕴含序列转换的信息。

3) 卷积层特征提取: 使用 $C$ 层一维卷积层(Conv1D)构造多层卷积网络(multi-CNN), 提取 $\mathbf{W2V\_AB}_i$ 的 $N$ 元组特征, 并将这些特征进行拼接, 形成包含 $N$ 元组特征的矩阵, 记为 $\boldsymbol{N}_g \in \mathrm{R}^{2m\times 256C}$:

$$U_k = \mathrm{Conv1D}_k^{k+1}(\mathbf{W2V\_AB}_i), k \in [1, C], \tag{4}$$

$$\boldsymbol{N}_g = [U_1, U_2, ..., U_C], \tag{5}$$

其中, $\mathrm{Conv1D}_k^{k+1}$ 表示第 $k$ 层 Conv1D 且卷积核宽度为 $k$+1, $U_k$ 表示第 $k$ 层 Conv1D 的输出。

就 Multi-CNN 层数的设置而言, 对于中文文本, 考虑到其存在多字词语, 将 $C$ 设置为 4, 卷积核大小分别为 2, 3, 4 和 5, 可以同时提取字向量矩阵的二元、三元、四元和五元特征; 对于英文文本, 将 $C$ 设置为 3, 卷积核大小分别为 2, 3 和 4, 用于同时提取字向量矩阵的二元、三元和四元特征。

4) 序列特征提取和模型输出: 将步骤 3 中的多层卷积网络的输出作为自注意力机制层(Self-attention)[26]的输入, 以此提取 $N$ 元组的序列特征, 同时 Self-attention 输出的每个节点都包含整个序列的信息。自注意力机制的输出经过标准化后, 使用以 Softmax 为激活函数的 Time Distributed 全连接网络, 得到 SSM 的输出, 记为 $\mathbf{W2V\_}\widehat{\mathbf{BA}}_i$:

$$\mathrm{BN} = \mathrm{Batch\_Normalization}(\mathrm{Self\_Attention}(\boldsymbol{N}_g)), \tag{6}$$

$$\mathbf{W2V\_}\widehat{\mathbf{BA}}_i = \mathrm{Softmax}_i(\boldsymbol{W}_s \cdot \mathrm{BN} + \boldsymbol{b}_s), \tag{7}$$

其中, $\boldsymbol{W}_s$ 和 $\boldsymbol{b}_s$ 是可学习更新的参数。

余弦相似度考虑的是向量夹角的大小, 适用于判断生成的向量与真实向量的相似性。相比之下, MSE(均方误差)和 MAE(平均绝对值误差)更多地考虑预测值与真实值之间的距离, 未考虑相似性。SSM 以余弦相似度作为损失函数:

$$\mathrm{Loss}_{\mathrm{SSM}} = -\mathrm{Cosine}_i(\mathbf{W2V\_BA}_i, \mathbf{W2V\_}\widehat{\mathbf{BA}}_i)。 \tag{8}$$

### 1.4 多任务学习(OM+SSM)

本文提出的多任务学习框架首先需要将自监督学习过程习得的文本互换生成时的交互关系信息, 提供给下游核心任务(即原始模型)。具体地, 本文将 SSM 提取的经过归一化的交互信息(BN), 经过池化层求和平均, 得到向量 **Vector_F**:

$$\mathrm{Vector\_F}_i = \mathrm{GlobalAveragePooling}_i(\mathrm{BN})。 \tag{9}$$

然后, 将原始模型的 $\mathrm{Vector\_E}_i$ 和交互信息 $\mathrm{Vector\_F}_i$ 拼接后, 输入以 Sigmoid 函数为激活函数的全连接层, 得到相似度分数 $\mathrm{Sim\_Score}_i$:

$$\mathrm{Sim\_Score}_i = \mathrm{Sigmoid}(W_m[\mathrm{Vector\_E}_i, \mathrm{Vector\_F}_i] + b_m), \tag{10}$$

其中, $W_m$ 和 $b_m$ 是可学习更新的参数。

在训练过程中, 多任务学习总体损失函数为

$$\mathrm{Loss}_{\mathrm{ML}} = \mathrm{Loss}_{\mathrm{OM}} + \lambda \mathrm{Loss}_{\mathrm{SSM}}, \tag{11}$$

其中, $\lambda \in (0, 1)$ 是自监督模型损失函数的权重系数。在本文实验中, $\lambda$ 取值为 0.5。

## 2 实验设置

### 2.1 数据集介绍

本文选用的数据集是文本语义匹配二分类任务的公开数据集, 包括微软研究释义语料库(MSRP)①、2018 年微众银行智能客服问句匹配大赛数据集(CCKS18-T3)②、天池“公益 AI 之星”挑战赛数据集(TCAI20)③和首届全球人工智能技术创新大赛赛道三数据集(GAIIC21-T3 和 GAIIC21-T3M)④。各数据集的分布情况见表 1。

① https://www.microsoft.com/en-us/download/details.aspx?id=52398
② https://www.biendata.xyz/competition/CCKS2018_3
③ https://tianchi.aliyun.com/competition/entrance/231776/introduction
④ https://tianchi.aliyun.com/competition/entrance/531851/introduction

**表 1　数据集分布情况**

Table 1　Data set statistics

| 数据集 | 训练集/对 | 验证集/对 | 测试集/对 | 平均长度 | 评价指标 | 语种 |
|---|---|---|---|---|---|---|
| MSRP | 4076 | – | 1725 | 18.92 | F1-Score | 英文 |
| CCKS18-T3 | 100000 | 10000 | 10000 | 11.37 | Accuracy | 中文 |
| TCAI20 | 8747 | 2002 | 601 | 13.68 | Accuracy | 中文 |
| GAIIC21-T3 | 70000 | – | 30000 | 6.47 | AUC | 中文 |
| GAIIC21-T3M | 100000 | – | – | 6.47 | AUC | 中文 |

## 2.2 深度文本语义匹配模型和评价方法

如表 2 所示，本文选取 9 个模型进行实验。OM 的参数是 MatchZoo 工具库[25]的默认参数。SSM 的参数如表 3 所示。Multi-CNN 每一层的以及 Self-Attention 神经元数量固定为 256，激活函数都是 Relu。实验中，对 MSRP，CCKS18-T3，TCAI20 和 GAIIC21-T3 这 4 个数据集的训练集采用 10 折交叉验证，选取效果最好的一折模型用于测试集进行测试。GAIIC21-T3M 的训练集是 GAIIC21-T3 的训练集和测试集的总和。GAIIC21-T3M 数据采用整个训练集 10 折交叉验证的方法，对比 GAIIC21-T3 的测试，前者评估数据标签分布较相似时模型的表现，后者评估模型在独立测试数据上的表现。采用的评价指标包括 F1-score (MSRP)、Accuracy (CCKS18-T3 和 TCAI20)和 AUC (GAIIC21-T3 和 GAIIC21-T3M)。

本文设计了分解模型以及分解模型和 SSM 结合的多任务模型。分解模型(Self-attention，SA)指将自监督学习过程中习得的文本交互信息 Vector_F$_i$ 输入以 Sigmoid 函数为激活函数的全连接层，得到句子对的相似度分数。此模型评估自监督学习中的 Pretext 任务学习的文本交互信息 Vector_F$_i$ 是否可以独立用于文本相似度计算。基于此，本文还设计了多任务模型 SA+SSM，即将 SA 分解模型的损失函数与 SSM 模型的损失函数加权求和，作为多任务 SA+SSM 模型预测句子对的相似度，加权方式与 OM+SSM 多任务学习的损失函数相同，$\lambda$ 取值为 0.5。

**表 2　深度文本语义匹配模型**

Table 2　Neural semantic matching models

| 模型 | | 模型描述 |
|---|---|---|
| 基于表示的模型 | ARC-I[11] | 采用重复堆叠的一维卷积神经网络和一维最大池化层，分别提取两个句子的 N-gram 特征，没有对交互信息进行提取 |
| | DSSM[4] | 是第一个提出深度语义匹配的模型，虽然可以提取句子的语义信息，但缺少对句子对之间的交互信息的提取 |
| | CDSSM[12] | 是在 DSSM 的基础上，通过使用使用卷积神经网络捕获句子的 N-gram 信息，缺少对句子对之间的交互信息的提取 |
| 基于交互的模型 | ARC-II[11] | 弥补了 ARC-I 缺少对交互信息提取的缺点。本文实验使用相加得到两个 N-gram 信息的交互矩阵 |
| | DRMMTKS[13] | 在查询项与文档之间建立交互，为每个查询项创建一个固定长度的匹配直方图，输入全连接神经网络来得到相似度 |
| | K-NRM[14] | 使用高斯核函数(RBF kernel)来捕获词汇之间的软匹配(soft-match)信号特征，将特征组合在一起，经过全连接网络得到相似度 |
| | CONV-KNRM[15] | 是 K-NRM 的一个变体，在形成 Translation 矩阵之前使用多个卷积神经网络来提取两个句子词汇之间的 N-gram 信息 |
| | MV-LSTM[16] | 基于 Bi-LSTM 的语义模型，两个句子的向量矩阵进行交互得到新的向量表示，然后使用 Top-$K$ 最大池化层和全连接神经网络进行降维，得到两个句子的相似度 |
| 混合模型 | DUET[17] | 混合模型中的 Local 模型用于捕获词汇与词汇的匹配信息，得到 $L$ score；Distributed 模型学习两个句子的语义向量的交互信息，得到 $D$ score；将 $L$ score 和 $D$ score 求和得到相似度 |

**表 3　神经网络参数设置**

Table 3　Neural network parameters

| 数据集 | 序列长度 | BatchSize | OM Embedding 层输出维度 | SSM 输入维度 | SSM 输出维度 | Multi-CNN 每层卷积核大小 |
|---|---|---|---|---|---|---|
| MSRP | 34 | 64 | (34, 300) | (34, 300) | (68, 300) | 2, 3, 4 |
| CCKS18-T3 | 40 | 64 | (40, 300) | (40, 100) | (80, 100) | 2, 3, 4, 5 |
| TCAI20 | 20 | 64 | (20, 300) | (20, 300) | (40, 300) | 2, 3, 4, 5 |
| GAIIC21-T3 | 37 | 64 | (37, 300) | (37, 100) | (74, 100) | 2, 3, 4, 5 |
| GAIIC21-T3M | 30 | 64 | (30, 300) | (30, 100) | (60, 100) | 2, 3, 4, 5 |

## 3 实验结果与评价

我们通过表4的实验结果讨论本文提出的两个研究问题。F1-Score, Accuracy和AUC取值均在0~1范围内。

首先讨论RQ1。通过表4基于表示的模型的实验结果可以看到，加入SSM后，ARC-I模型在5个数据集分别提升2.8%, 2.7%, 2.9%, 3.9%和1.4%; DSSM模型分别提升了0.5%, 5.1%, 21.1%, 16.2%和12.4%; CDSSM的提升分别为1.3%, 8.8%, 18.6%, 12.0%和31.7%。由此可见，基于表示的模型在加入SSM后，在5个数据集上的效果都得到提升。自监督学习提取的交互信息能够弥补这些模型的不足。

针对RQ2，由表4看到，加入SSM后，ARC-II模型在5个数据集上的性能分别提升1.9%, 0.1%, 1.5%, 6.0%和7.5%; DRMMTKS模型分别提升2.9%, 2.7%, 1.1%, 28.6%和36.0%; K-NRM加入SSM分别提升2.1%, 4.4%, 11.9%, 33.9%和47.4%; CONV-KNRM分别提升3.1%, 2.1%, 16.5%, 4.4%和10.2%; MV-LSTM分别提升5.0%, 3.0%, 4.6%, 8.9%和11.6%; 对于混合模型DUET的提升分别是2.4%, 2.0%, 2.6%, 8.4%和9.3%。实验结果表明，在基于交互的模型和混合模型的实验中，加入SSM后效果均有提升，说明SSM基于句子序列转换提取的交互信息与这些模型提取的交互信息并不冲突，也不冗余，并且能够有效地增强原始深度匹配模型提取的交互信息，提升模型的整体效果。

通过比较分解模型SA与原始模型OM在5个数据集上的实验结果，发现SA模型在其中3个数据集上的效果优于所有OM模型。这说明，本文设计的自监督辅助任务能够学习到有效的、可用于文本相似度计算的文本交互信息。同时，结合SA的多任务模型SA+SSM也在一个数据集(GAIIC21-T3)上取得最优结果。在其他4个数据集上取得最佳结果的是本文提出的结合自监督学习的多任务模型(OM+SSM)，说明其对下游任务是有效的。

进一步分析同一个数据集下不同模型的提升情

**表 4 实验结果(%)**

Table 4 Comparing model performances (%)

| | 模型 | MSRP(F1-Score) | CCKS18-T3(Accuracy) | TCAI20(Accuracy) | GAIIC21-T3(AUC) | GAIIC21-T3M(AUC) |
|---|---|---|---|---|---|---|
| 基于表示的OM模型与OM+SSM多任务模型 | ARC-I | 75.51 | 69.53 | 74.88 | 85.34 | 93.08 |
| | **ARC-I+SSM** | ***77.64*** | ***71.40*** | ***77.04*** | ***88.69*** | ***94.35**** |
| | DSSM | 80.24 | 73.22 | 68.72 | 76.86 | 81.78 |
| | **DSSM+SSM** | ***80.64*** | ***76.93*** | ***83.19*** | ***89.29*** | ***91.96*** |
| | CDSSM | 79.75 | 70.47 | 70.88 | 79.83 | 69.51 |
| | **CDSSM+SSM** | ***80.75*** | ***76.67*** | ***84.03*** | ***89.42*** | ***91.56*** |
| 基于交互的OM模型和混合OM模型与OM+SSM多任务模型 | ARC-II | 77.41 | 71.23 | 75.21 | 80.53 | 83.87 |
| | **ARC-II+SSM** | ***78.85*** | ***71.32*** | ***76.37*** | ***85.37*** | ***90.18*** |
| | DRMMTKS | 78.81 | 75.73 | 87.02 | 68.51 | 67.03 |
| | **DRMMTKS+SSM** | ***81.09**** | ***77.74*** | ***88.02**** | ***88.11*** | ***91.16*** |
| | K-NRM | 78.13 | 74.01 | 73.88 | 67.01 | 63.10 |
| | **K-NRM+SSM** | ***79.77*** | ***77.25*** | ***82.70*** | ***89.75*** | ***93.04*** |
| | CONV-KNRM | 77.73 | 76.42 | 74.54 | 83.61 | 81.95 |
| | **CONV-KNRM+SSM** | ***80.17*** | ***78.03**** | ***86.86*** | ***87.31*** | ***90.28*** |
| | MV-LSTM | 76.04 | 75.17 | 79.67 | 81.18 | 83.63 |
| | **MV-LSTM+SSM** | ***79.87*** | ***77.43*** | ***83.36*** | ***88.43*** | ***93.30*** |
| | DUET | 76.69 | 74.78 | 83.53 | 82.34 | 85.39 |
| | **DUET+SSM** | ***78.56*** | ***76.25*** | ***85.69*** | ***89.29*** | ***93.37*** |
| 分解模型与SA+SSM多任务 | Self-attention (SA) | 80.50 | 75.57 | 81.03 | 89.75 | 93.13 |
| | SA+SSM | 80.76 | 75.79 | 82.69 | **89.88*** | 94.12 |

说明：加粗斜体数字表示OM+SSM模型优于OM模型的结果；粗体星号表示各数据集的最佳结果；分解模型SA优于OM模型的结果用下划线表示。

**表 5　SSM 对不同数据集的提升效果(%)**

Table 5　Self-supervised model improvement by data set (%)

| 数据集 | ARC-I +SSM | DSSM +SSM | CDSSM +SSM | ARC-II +SSM | DRMMTKS +SSM | K-NRM +SSM | CONV-KNRM +SSM | MVLSTM +SSM | DUET +SSM | 平均 |
|---|---|---|---|---|---|---|---|---|---|---|
| MSRP | 2.8 | 0.5 | 1.3 | 1.9 | 2.9 | 2.1 | 3.1 | 5.0 | 2.4 | 2.44 |
| CCKS18-T3 | 2.7 | 5.1 | 8.8 | 0.1 | 2.7 | 4.4 | 2.1 | 3.0 | 2.0 | 3.43 |
| TCAI20 | 2.9 | 21.1 | 18.6 | 1.5 | 1.1 | 11.9 | 16.5 | 4.6 | 2.6 | 8.98 |
| GAIIC21-T3 | 3.9 | 16.2 | 12.0 | 6.0 | 28.6 | 33.9 | 4.4 | 8.9 | 8.4 | 13.59 |
| GAIIC21-T3M | 1.4 | 12.4 | 31.7 | 7.5 | 36.0 | 47.4 | 10.2 | 11.6 | 9.3 | 18.61 |

况。表 5 展示结合自监督模型后各个数据集效果的提升效果。可以看到，对于 MSRP 数据集，9 个模型的提升效果都不太明显，平均提升 2.44%。该数据集的句子提取自多个新闻网站，每个句子都来自不同的新闻文章，很好地消除了句子之间可能存在的语义相似性，也可能导致句子之间主题的共性较少且主题复杂，显示 SSM 在应对不同主题的句子对相互生成的鲁棒性较弱。对于 CCKS18-T3 数据集，所有模型的提升效果也不够好，平均提升 3.42%。该数据集来自微众银行智能客服问句匹配，其核心是句子对之间的意图匹配，而 SSM 缺少对语句意图特征的提取和表示，显示基于序列生成的自监督模型缺乏深层语义特征提取的能力。对于其他 3 个数据集，增加自监督学习带来的模型提升都较明显。TCAI20 是新冠疫情相似句判断，GAIIC21-T3(M)是人工智能助手对话短文本匹配。这些数据集的语句之间主题较为相近，SSM 可以提取出质量较高的交互信息。

## 4 结语

本文提出自监督模型的辅助任务，以句子对中两个句子相互生成的方式，获取文本之间基于序列转换的深层交互信息，以特征增强的方式辅助下游模型。同时，为了让自监督模型学习到的交互信息动态地参与文本匹配任务，本文提出采用硬参数共享的多任务学习方式，将文本匹配模型与自监督模型相结合。结果表明，加入自监督学习框架后，所有模型的效果均得到提升，证明本文用的自监督学习模型构建多任务学习，以特征增强的方式辅助文本语义匹配任务的设计是有效的。

本文的模型和方法存在一定的局限性。当数据集中的句子对存在比较复杂多样的主题，或者句子对中之间存在深层次(比如意图)匹配时，我们提出的自监督模型的提升效果不够显著。未来的研究中，需要探讨在设计自监督模型时，如何有效地应对文本对之间主题的复杂性，如何提取深层意图语义的交互特征以及相似文本的语法结构交互特征。

## 参考文献


[1] Li H, Xu J. Semantic matching in search. Foundations and Trends in Information Retrieval, 2014, 7(5): 343–469

[2] Rao J, Liu L, Tay Y, et al. Bridging the gap between relevance matching and semantic matching for short text similarity modeling // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMN LP-IJCNLP). Hong Kong, 2019: 5370–5381

[3] Chandrasekaran D, Mago V. Evolution of semantic similarity — a survey. ACM Computing Surveys (CSUR), 2021, 54(2): 1–37

[4] Huang P S, He X, Gao J, et al. Learning deep structured semantic models for web search using clickthrough data // Proceedings of the 22nd ACM international conference on Information & Knowledge Management. San Francisco, 2013: 2333–2338

[5] Chen D, Fisch A, Weston J, et al. Reading Wikipedia to answer open-domain questions // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vancouver, 2017: 1870–1879

[6] Parikh A, Täckström O, Das D, et al. A decomposable attention model for natural language inference // Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin, 2016: 2249–2255

[7] 杨开平. 基于语义相似度的中文文本聚类算法研究[D]. 成都: 电子科技大学, 2018

[8] Wieting J, Berg-Kirkpatrick T, Gimpel K, et al. Beyond BLEU: training neural machine translation with semantic similarity // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, 2019: 4344–4355

[9] 陈二静, 姜恩波. 文本相似度计算方法研究综述. 数据分析与知识发现, 2017, 1(6): 1–11

[10] Guo J, Fan Y, Ai Q, et al. A deep relevance matching model for ad-hoc retrieval // Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. Indianapolis, 2016: 55–64

[11] Hu B, Lu Z, Li H, et al. Convolutional neural network architectures for matching natural language sentences // Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2. Montreal, 2014: 2042–2050

[12] Shen Y, He X, Gao J, et al. Learning semantic representations using convolutional neural networks for web search // Proceedings of the 23rd International Conference on World Wide Web. Seoul, 2014: 373–374

[13] Guo J, Fan Y, Ji X, et al. DRMMTKS [EB/OL]. (2019–10–05) [2021–08–14]. https://github.com/NTMC-Community/MatchZoo

[14] Xiong C, Dai Z, Callan J, et al. End-to-end neural adhoc ranking with kernel pooling // Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval. Tokyo, 2017: 55–64

[15] Dai Z, Xiong C, Callan J, et al. Convolutional neural networks for soft-matching N-grams in ad-hoc search // 11th ACM International Conference. Los Angeles, 2018: 126–134

[16] Wan S, Lan Y, Guo J, et al. A deep architecture for semantic matching with multiple positional sentence representations // Proceedings of the AAAI Conference on Artificial Intelligence. Phoenix, 2016: 2835–2841

[17] Mitra B, Diaz F, Craswell N. Learning to match using local and distributed representations of text for web search // Proceedings of the 26th International Conference on World Wide Web. Perth, 2017: 1291–1299

[18] Liu X, Zhang F, Hou Z, et al. Self-supervised learning: generative or contrastive [EB/OL]. (2021–03–20) [2021–08–14]. https://arxiv.org/abs/2006.08218

[19] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [EB/OL]. (2013–09–07) [2021–08–14]. https://arxiv.org/abs/1301.3781

[20] Kiros R, Zhu Y, Salakhutdinov R R, et al. Skip-thought vectors // Advances in Neural Information Processing Systems. Montreal, 2015: 3294–3302

[21] Devlin J, Chang M W, Lee K, et al. Bert: pre-training of deep bidirectional transformers for language understanding // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis, 2019: 4171–4186

[22] Caruana R. Multitask learning. Machine Learning, 1997, 28(1): 41–75

[23] Lee H, Hwang S J, Shin J. Rethinking data augmentation: self-supervision and self-distillation [EB/OL]. (2019–12–24) [2021–08–14]. https://openreview.net/forum?id=SkliR1SKDS

[24] Ruder S. An overview of multi-task learning in deep neural networks [EB/OL]. (2017–06–15) [2021–08–14]. https://arxiv.org/abs/1706.05098

[25] Guo J, Fan Y, Ji X, et al. Matchzoo: a learning, practicing, and developing system for neural text matching // Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval. Paris, 2019: 1297–1300

[26] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need // Advances in Neural Information Processing Systems. California, 2017: 5998–6008