计算机科学
COMPUTER SCIENCE

融合义原相似度矩阵与字词向量双通道的短文本语义匹配策略

刘东旭，段利国，崔娟娟，常轩伟

引用本文

刘东旭，段利国，崔娟娟，常轩伟. 融合义原相似度矩阵与字词向量双通道的短文本语义匹配策略[J]. 计算机科学，2024，51(12)：250-258.

LIU Dongxu, DUAN Liguo, CUI Juanjuan, CHANG Xuanwei. Short Text Semantic Matching Strategy Fusing Sememe Similarity Matrix and Dual-channel of Char-Word Vectors [J]. Computer Science, 2024, 51(12): 250-258.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231100147

融合义原相似度矩阵与字词向量双通道的短文本语义匹配策略

刘东旭[1]　段利国[1,2]　崔娟娟[1]　常轩伟[1]
1 太原理工大学计算机科学与技术学院　山西 晋中 030600
2 山西电子科技学院　山西 临汾 041000
(2495628480@qq.com)

摘　要　短文本语义匹配任务的目的是判断两个短文本句子的语义是否一致。然而,现有的许多方法往往存在短文本语义信息不足、无法有效识别同义词等问题。针对这些不足,提出一种融合义原相似度矩阵与字词向量双通道的短文本语义匹配策略。首先,利用预训练模型 Bert 对输入的句子对进行编码;然后,对于句子中词级别的语义信息,利用 FastText 模型训练并获取文本的词向量,并加入 BiLSTM 模型进一步提取上下文语义信息。为了有效利用义原信息,在上述的双通道中分别加入多头注意力和用于对分离向量进行交互计算的协同注意力,并在注意力中分别融入对应的义原相似度矩阵,最后综合上述两部分向量推断出语义的一致性。在金融领域数据集 BQ 和开放域数据集 LCQMC 上的实验证明了所提算法的有效性。

关键词:自然语言处理;短文本;义原;协同注意力;字词向量
中图分类号　TP391

Short Text Semantic Matching Strategy Fusing Sememe Similarity Matrix and Dual-channel of Char-Word Vectors

LIU Dongxu[1],DUAN Liguo[1,2],CUI Juanjuan[1] and CHANG Xuanwei[1]
1 College of Computer Science and Technology,Taiyuan University of Technology,Jinzhong,Shanxi 030600,China
2 Shanxi University of Electronic Science and Technology,Linfen,Shanxi 041000,China

Abstract　The purpose of the short text semantic matching task is to judge whether the semantics of two short text sentences are consistent. However,many existing methods often have shortcomings such as insufficient semantic information of short text and inability to effectively identify synonyms. In response to these shortcomings,this paper proposes a short text semantic matching strategy that fuses sememe similarity matrix and dual-channel of char-word vectors. Firstly,the pre-trained model Bert is used to encode the input sentence pairs;for the word-level semantic information in the sentence,the FastText model is used to train and obtain the word vector of the text,and the BiLSTM model is added to further extract the contextual semantic information. Secondly,making effective use of the semantic information,multi-head attention and co-attention for interactive calculation of separation vectors are added to the above-mentioned dual-channel. And the semantic similarity matrix is integrated into the attentions respectively. Finally,infer the semantic consistency according to the above vectors. The effectiveness of the above algorithm is proved by experiments on the financial dataset BQ and the open domain dataset LCQMC.

Keywords　Natural language processing,Short text,Sememe,Co-attention,Char-Word vector

1　引言

短文本语义匹配(或语义一致性识别)任务是自然语言处理(Natural Language Processing,NLP)的重要任务之一,可广泛用于信息检索、自动问答和对话系统等下游任务。输入主要是短文本句子对,最终目的是判断短文本句子对的语义是否一致。

现有的文本匹配模型可分为传统匹配模型和深度学习匹配模型两大类。传统的文本匹配方法主要依靠人工定义特征来计算文本之间的相似度,这些方法难以提取深层语义信息。随着深度学习的快速发展以及相关大规模数据集的出现,基于深度学习的文本匹配方法受到越来越多的关注。其主要思想是通过深度学习方法将两个句子编码成向量,或者通过注意力机制将两个句子进行交互,然后进行语义一致性的判断。近年来,为了使模型能够基于更好的初始状态进行学习,取得更好的表现,预训练模型横空出世,且成为目前的主流方法。其中,Bert[1](Bidirectional Encoder Representation)模型的表现最为突出,该模型通过其多头注意力机制[2]学习语境化的

到稿日期:2023-11-22　返修日期:2024-05-08
基金项目:山西省自然科学基金(202203021221234,202303021211052)
This work was supported by the Natural Science Foundation of Shanxi Province,China(202203021221234,202303021211052).
通信作者:段利国(zhaixing202202@163.com)

词语表示，并在大型语料库上进行无监督预训练，在多个任务领域中都表现卓越。两种方法在文本匹配方面都取得了较为不错的效果，但是在数据规模稀疏或者缺少外部知识的情况下，上述方法的表现并不尽如人意。外部知识在短文本匹配任务中发挥着很大的作用，添加外部知识可以弥补数据稀疏导致的信息量不足的缺陷，使得模型更好地获取文本语义信息，有助于模型推断文本之间的关系。实验发现，在测试某些句子对时常常出现对同义词识别不精确的情况。对此，某些外部知识的引入可以帮助模型更好地识别同义词等某些特殊词语对。总的来说，现有模型在短文本语义匹配任务上有着以下不足：

1）由于 Bert 等模型的初始训练基本单位为字级别，因此模型对词级别的语义信息捕捉得不够充分。

2）语义信息不足且未加入特定的外部知识。对于短文本语义匹配而言，句子长度较短，包含的语义信息不够充足，且模型对常识知识的判断也不够理想。

3）同义词识别不精确。经常出现不同词语在相同的语境下表达出相同语义的情况，很多模型无法有效对其进行识别。

鉴于上述问题，本文引入义原知识进行改进。其中，义原是语言学中提出的一种独立语义单元，每个字词都对应着多个义原。经过义原识别，模型能更加有效地识别文本中的同义词。本文引入 OpenHowNet[3] 作为义原知识库工具。OpenHowNet 中所使用的数据构建秉承还原论思想，即所有词语的含义可以由更小的语义单位构成，而这种语义单位被称为“义原”。对此，本文提出了融合义原相似度矩阵与字词向量双通道协同优化的短文本语义匹配策略。首先，利用 Bert 和 FastText[4] 模型组成字、词双通道分别对文本进行编码，获取文本的字、词级别的语义信息。此外，词级别通道加入 BiLSTM[5] 模型进一步提取上下文语义信息。为了解决同义词以及未加入特定外部知识的问题，在双通道中分别加入多头注意力和用于对分离向量进行交互计算的协同注意力[6]，同时引入开源工具 OpenHowNet 进行相似度计算，构建义原相似度矩阵，并将其注入注意力机制中。最后，将得到的句子向量拼接并输入到分类器中进行文本语义一致性的判断。实验表明，与现有模型相比，融合义原相似度矩阵与字词向量双通道协同优化的短文本语义匹配策略可以在一定程度上提高文本语义匹配的准确率。

针对目前模型在短文本语义匹配任务上的不足，本文做出以下 3 点贡献：

1）为了解决模型对词级别语义信息捕捉不充分这一问题，本文提出利用字、词双通道对文本进行语义提取。

2）提出一种将义原知识融入短文本语义匹配模型的方法。通过义原知识计算句子对文本中字或词的相似度分数并构建相似度矩阵，将其融入注意力机制的计算分数模块中。

3）在 BQ 和 LCQMC 数据集上进行实验。在上述两个数据集上，与众多模型相比，本文模型都取得了最好效果，足以证明其优异性。

2 相关工作

近年来，随着深度学习的快速发展，人们提出了大量的方法来完成文本语义匹配这一任务。在语义信息获取方面，经典短文本匹配模型 DSSM[7] 解决了 LSA（Latent Semantic Analysis）和 LDA（Latent Dirichlet Analysis）等方法中字典爆炸的问题，但其因为使用了词袋模型而丢失了上下文信息。ESIM[8] 模型结合 BiLSTM 和注意力机制，首次提出在局部推理中进行句子对之间的交互。DIIN[9] 模型使用 CNN 和 LSTM 进行特征提取，在模型的输入层同时使用了词向量和局部向量，输入一些额外的句法特征，并使用 DenseNet 进行特征提取。MwAN[10] 模型使用多种注意机制（剪接、双线性、点乘、减法）来充分捕捉句子对之间的关系，最后通过 GRU 和全连接层对多个结果进行加权组合后输出最终概率。

随着 Bert 模型的提出，各种预训练模型接踵而至，并在 NLP 领域的众多任务中都取得了不俗的效果。与以往模型不同，Bert 模型的基本组成是注意力机制，因此在解决文本序列的长距离依赖问题方面取得了重大成效。后续出现的 Bert 模型的变种，如 ALBert[11]，RoBERTa[12]，SemBERT[13]，ERNIE[14]，K-BERT[15] 和 DeBERTa[16] 等，同样表现优异。其中，SemBERT，ERNIE 和 K-BERT 在 BERT 模型的基础上添加了知识，但方式不同。SemBERT 和 K-BERT 是一种无需预训练的微调方法，能够从预训练的 BERT 中加载模型参数。SemBERT 从预先训练的语义角色标注中引入明确的上下文语义。K-BERT 是一种知识增强语言模型，其将知识三元组作为领域知识注入句子。ENRIE 同时利用词汇、句法和语义知识信息在 BERT 的基础上进行改进。虽然上述几种 BERT 变种被证明可以提高各种语言理解任务的性能，但其结果在很大程度上取决于特定训练数据的大小。

2021 年，Lyu 提出 LET[17] 模型该模型利用 HowNet 进行文本蕴涵识别。首先对每个词的义位的初始向量进行基于图注意力的变换，然后对每个词进行注意力融合，最后通过 GRU 和 BERT 词向量的整合得到最终的词向量。Bai 等提出的 Syntax-BERT[18] 模型，通过即插即用的方式在预训练模型中加入句法树结构，且无需从头预训练。

Li 等[19] 提出的基于孪生网络和字词向量相结合的文本相似度匹配方法，从字词编码的角度对短文本匹配任务进行优化，并取得了一定成效。Lyu 等[20] 提出基于异质信息网的短文本特征扩充方法，文中表示可以利用知识库等进行信息扩充，以解决信息稀疏等问题。该方法为本文方法提供了一定的参考性。Yu 等提出一种用于语义文本相似度的已知主题的离散潜在变量模型[21]，该模型通过矢量量化学习用于句子对表示的共享潜在空间。与之前仅限于局部语义上下文的模型相比，该模型可以通过主题建模探索更丰富的语义信息。为了提升模型对细微差异的捕捉能力，Wang 等提出 DABERT[22] 模型。该模型通过自适应融合模块，利用注意力学习差异和相似特征的聚合生成描述句子对匹配细节的向量。为了在不同粒度上进行匹配判断，Zou 等提出 DC-Match[23] 模型。该模型将关键词和意图分离，对两部分的匹配进行联合训练，并以分治的方式进行文本语义匹配。为了增加关键线索的匹配语义信息，Chen 等在 2022 年提出一种增加外部知识的模型[24]，该模型通过搜索引擎抽取相关匹配

句子的上下文，并将该上下文加入原始文本匹配句子对，丰富了原始匹配语义。

2023 年，Zhang 等[25]提出的多任务联合训练的长文本多实体情感分析方法中去除冗余信息的方法也给予了本文一定的参考性。为了增强模型理解和推理能力，Jiang 等提出 KETM[26]模型，该模型选择利用维基词典检索文本单词定义作为外部知识加入模型中。

上述模型虽然从不同角度提升了文本匹配的效果，但从同义词识别的角度来看，效果都不够理想，所以本文提出的融合义原相似度矩阵与字词向量双通道的短文本语义匹配策略主要针对该问题进行优化。

3 模型

短文本语义匹配任务中，给定两个句子 $T_a=\{t_a^1,\cdots,t_a^i,\cdots,t_a^m\}$ 和 $T_b=\{t_b^1,\cdots,t_b^j,\cdots,t_b^n\}$，目的是判断两个句子的语义是否相同。其中，$t_a^i$ 和 t_b^j 分别代表两个句子的第 i 或 j 个字符，m 和 n 代表句子长度。本章将介绍一种融合义原相似度矩阵与字词向量双通道协同优化的短文本语义匹配模型，并详细介绍模型的各个模块与对应的功能。本文模型结构如图 1 所示，共分为 3 层，分别为编码层、交互层和预测层。

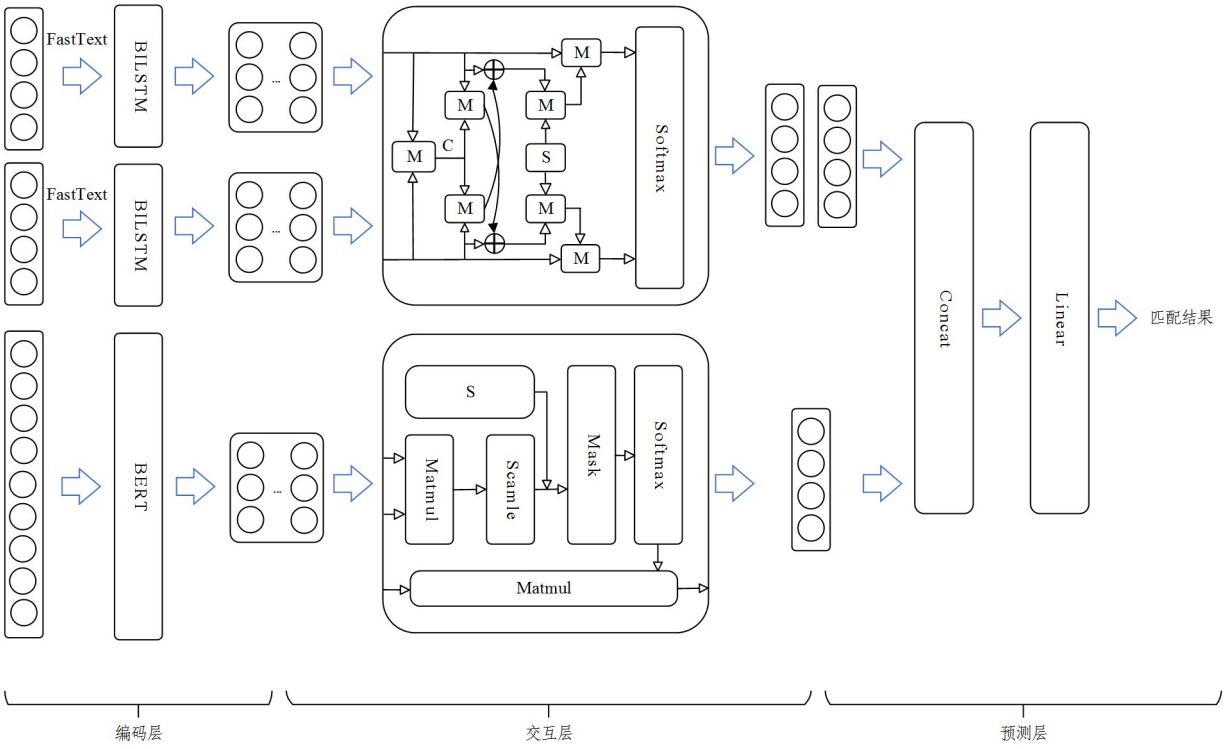

图 1 策略模型图

Fig. 1 Strategy model diagram

3.1 编码层

3.1.1 字级别

编码层的作用是对文本进行建模，通过神经网络获取深层语义信息。在本文中，字级别通道使用 BERT 模型。与以往的 Word2Vec 和 FastText 等静态词向量模型相比，BERT 在训练时使用了大量语料，可以获得更加丰富的语义表示，且可以随着训练更新上下文表示。该模型的输入部分表示如下：

$$\boldsymbol{bert}_{\text{in}}=\{[CLS],t_a^1,\cdots,t_a^m,[SEP],t_b^1,\cdots,t_b^n,[SEP]\} \quad (1)$$

其中，t_a^i 代表第一个句子的第 i 个字符，t_b^j 代表第二个句子的第 j 个字符。

将 $\boldsymbol{bert}_{\text{in}}$ 送入 Bert 模型中，得到如下输出：

$$\boldsymbol{bert}_{\text{out}}=\{[CLS],x_a^1,\cdots,x_a^m,[SEP],x_b^1,\cdots,x_b^n,[SEP]\} \quad (2)$$

其中，$\boldsymbol{x}_a^i$ 代表第一个句子的第 i 个隐藏层向量，$\boldsymbol{x}_b^j$ 代表第二个句子的第 j 个隐藏层向量。

3.1.2 词级别

由于以字为单位的建模方法在处理中文数据集时存在语义确定性不高的问题，模型往往难以提取词语级别的真实语义特征，因此本文引入 FastText 模型来提取词级别语义。

FastText 通过学习词向量来捕捉文本中的语义特征，从而实现文本的语义表征。具体步骤如下：

1）文本预处理。FastText 在开始时对原始文本进行预处理，包括分词等操作，以便后续处理。

2）构建词表。FastText 在预处理文本的基础上构建一个词表，词表中包含文本中出现的所有词汇，每个词汇都有一个唯一的 ID。

3）训练词向量。FastText 使用连续词袋模型（Continuous Bag of Words，CBOW）学习到每个词汇的词向量。连续词袋模型通过上下文单词的周围词汇来预测当前词汇，从而学习词汇之间的语义关系。

4）文本向量表示。对于一个给定的文本，FastText 将其中的词向量进行平均或求和，从而得到整个文本的向量表示。

FastText 的训练过程使用随机梯度下降算法，从大规模文本数据中迭代地更新词向量和模型参数。通过训练，FastText 能够捕捉到词汇之间的语义关系，从而更好地理解和表示文本。

本模块词向量通道的输入如下：

$$\boldsymbol{out}_{\text{in\_fasttext1}}=\{w_a^1,\cdots,w_a^i,\cdots,w_a^m\} \tag{3}$$

$$\boldsymbol{out}_{\text{in\_fasttext2}}=\{w_b^1,\cdots,w_b^j,\cdots,w_b^n\} \tag{4}$$

其中，w_a^i代表第一个句子的第 i 个词，w_b^j代表第二个句子的第 j 个词。

将上述的$\boldsymbol{out}_{\text{in_fasttext1}}$和$\boldsymbol{out}_{\text{in_fasttext2}}$输入到 FastText 中可以得到输出：

$$\boldsymbol{out}_{\text{fasttext1}}=\{\boldsymbol{v}_a^1,\cdots,\boldsymbol{v}_a^m\} \tag{5}$$

$$\boldsymbol{out}_{\text{fasttext2}}=\{\boldsymbol{v}_b^1,\cdots,\boldsymbol{v}_b^n\} \tag{6}$$

其中，$\boldsymbol{v}_a^i$代表 FastText 模型输出的第一个句子的第 i 个隐藏层向量，$\boldsymbol{v}_b^j$代表 FastText 模型输出的第二个句子的第 j 个隐藏层向量。

针对 FastText 提取上下文语义不够充分的问题，本文模型加入了 BiLSTM 模型。BiLSTM 由前向 LSTM 和后向 LSTM 组合而成，主要特点是能够在输入和输出序列之间的映射过程中提取前文信息。而双向结构会为输出序列提供每一个时间节点完整的过去和未来的上下文信息，从而使输出序列可以提取到句子的全局语义特征。将 FastText 模块输出的$\boldsymbol{out}_{\text{fasttext1}}$和$\boldsymbol{out}_{\text{fasttext2}}$输入 BiLSTM，得到输出$\boldsymbol{out}_{\text{bilstm1}}$和$\boldsymbol{out}_{\text{bilstm2}}$：

$$\boldsymbol{out}_{\text{bilstm1}}=BiLSTM(\boldsymbol{out}_{\text{fasttext1}}) \tag{7}$$

$$\boldsymbol{out}_{\text{bilstm2}}=BiLSTM(\boldsymbol{out}_{\text{fasttext2}}) \tag{8}$$

3.2 交互层

为了有效利用义原知识，该模块采用两个注意力机制，分别为多头注意力（Multi-Attention）和协同注意力（Co-Attention）。多头注意力形式如下：

$$\boldsymbol{MultiHead}(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=Concat(\boldsymbol{h}_1,\cdots,\boldsymbol{h}_n)\boldsymbol{W}^O \tag{9}$$

$$\boldsymbol{h}_i=Attention(\boldsymbol{Q}\boldsymbol{W}_i^Q,\boldsymbol{K}\boldsymbol{W}_i^K,\boldsymbol{V}\boldsymbol{W}_i^V) \tag{10}$$

其中，$\boldsymbol{h}_i$代表注意力的第 i 个头，$\boldsymbol{W}_i^Q,\boldsymbol{W}_i^K,\boldsymbol{W}_i^V,\boldsymbol{W}_i^O\in(R^{d_{\text{model}}\times d_q},R^{d_{\text{model}}\times d_k},R^{d_{\text{model}}\times d_v},R^{nd_v\times d_{\text{model}}})$是可训练的权重矩阵，$d_{\text{model}}$是模型的隐藏层大小，$n$ 是多头注意力的头数。注意力计算方式为点积：

$$scores=\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}+\boldsymbol{MASK} \tag{11}$$

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{softmax}\left(\frac{scores}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{12}$$

协同注意力最早用于视觉问答领域，后来用于机器阅读领域。以机器阅读理解为例，普通注意力机制通过结合问题和文本段落二者的信息，生成一个关于文本段落各部分的注意力权重，对文本信息进行加权。而协同注意力（Parallel）则是一种双向的注意力，不仅要给阅读的文本段落生成一个注意力权重，还要给问题也生成一个注意力权重。对于文本匹配任务来说，协同注意力机制可以帮助我们更好地捕捉两个句子相关的信息。

协同注意力有两种形式，分别为 Parallel 和 Alternating。由于短文本语义匹配任务的训练数据为同等权重的句子对，因此本文采用 Parallel 范式。

Parallel 范式的协同注意力会同时关注两个句子，通过计算两个句子特征的相似度来协同交互两个句子。具体来说，给定第一个句子 $\boldsymbol{V}\in\boldsymbol{R}^{d\times N}$、第二个句子 $\boldsymbol{Q}\in\boldsymbol{R}^{d\times T}$ 和亲和力矩阵 $\boldsymbol{C}\in\boldsymbol{R}^{T\times N}$，则：

$$\boldsymbol{C}=\tanh(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{W}_b\boldsymbol{V}) \tag{13}$$

$$\boldsymbol{H}^v=\tanh(\boldsymbol{W}_v\boldsymbol{V}+(\boldsymbol{W}_q\boldsymbol{Q})\boldsymbol{C}) \tag{14}$$

$$\boldsymbol{a}^v=\text{softmax}(\boldsymbol{w}_{hv}^{\mathrm{T}}\boldsymbol{H}^v) \tag{15}$$

$$\boldsymbol{H}^q=\tanh(\boldsymbol{W}_q\boldsymbol{Q}+(\boldsymbol{W}_v\boldsymbol{V})\boldsymbol{C}^{\mathrm{T}}) \tag{16}$$

$$\boldsymbol{a}^q=\text{softmax}(\boldsymbol{w}_{hq}^{\mathrm{T}}\boldsymbol{H}^q) \tag{17}$$

其中，$\boldsymbol{W}_b\in R^{d\times d}$；$\boldsymbol{W}_v,\boldsymbol{W}_q\in R^{d\times d}$，$\boldsymbol{\omega}_{hv},\boldsymbol{\omega}_{hq}\in R^k$为可训练的权重矩阵；$a^v\in R^N$和$a^q\in R^T$分别为两个句子的每个 *Token* 对应的注意力分数。

最后，将注意力分数应用到每个 *Token*。

$$\hat{\boldsymbol{v}}=\sum_{n=1}^{N}\boldsymbol{a}_n^v\boldsymbol{v}_n \tag{18}$$

$$\hat{\boldsymbol{q}}=\sum_{t=1}^{T}\boldsymbol{a}_t^q\boldsymbol{q}_t \tag{19}$$

其中，v_n和q_t分别代表两个句子的每一个 Token。

目前，普通注意力机制在捕获语义信息时，无法精确计算同义词的注意力权重，所以本模块对上述注意力机制进行改进。首先，构建相似度矩阵 SimMatrix，其中的每一项代表注意力机制中每次计算时两个 Token 对应字或词语的相似度分数，之后将相似度矩阵 $\boldsymbol{S}$ 与原本注意力机制的 scores 部分进行结合，使模型更加关注两句中相似度较高的字词对，且降低对非同义词字词对的关注。多头注意力部分的改进如下：

$$scores=\boldsymbol{Q}\boldsymbol{K}^{\mathrm{T}}\odot S+MASK \tag{20}$$

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V})=\text{softmax}\left(\frac{scores}{\sqrt{d_k}}\right)\boldsymbol{V} \tag{21}$$

协同注意力部分的改进如下：

$$C=\tanh(\boldsymbol{Q}^{\mathrm{T}}\boldsymbol{W}_b\boldsymbol{V}) \tag{22}$$

$$\boldsymbol{H}^v=\tanh(\boldsymbol{W}_v\boldsymbol{V}+(\boldsymbol{W}_q\boldsymbol{Q})\boldsymbol{C}) \tag{23}$$

$$\boldsymbol{a}^v=\text{softmax}(\boldsymbol{w}_{hv}^{\mathrm{T}}\boldsymbol{H}^v\odot\boldsymbol{S}) \tag{24}$$

$$\boldsymbol{H}^q=\tanh(\boldsymbol{W}_q\boldsymbol{Q}+(\boldsymbol{W}_v\boldsymbol{V})\boldsymbol{C}^{\mathrm{T}}) \tag{25}$$

$$\boldsymbol{a}^q=\text{softmax}(\boldsymbol{w}_{hq}^{\mathrm{T}}\boldsymbol{H}^q\odot\boldsymbol{S}) \tag{26}$$

其中，$\boldsymbol{S}$ 代表相似度矩阵 SimMatrix。

改进后的多头注意力的示意图如图 2 所示。

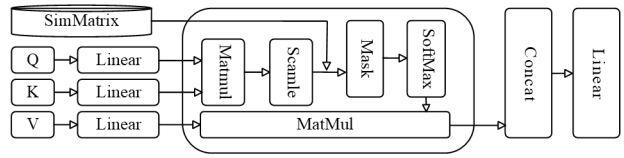

图 2 注入相似度矩阵的注意力机制

Fig. 2 Attention mechanism injected with similarity matrix

对于矩阵中相似度的计算，本文使用的是 OpenHowNet 开源工具的接口。构建 SimMatrix 的方法如下：设两个句子 $a=\{w_a^1,\cdots,w_a^i,\cdots,w_a^m\}$和 $b=\{w_b^1,\cdots,w_b^j,\cdots,w_b^n\}$，构造的相似度矩阵大小为 $m\times n$，其中w_a^i和w_b^j分别代表两个句子的第 i 和 j 个字符。本文根据 OpenHowNet 中基于义原的语义关系计算 $\boldsymbol{S}$ 中每个单元格的值。如果字词对(w_a^i,w_b^j)中的内容在 HowNet 中是同义词（基于义原），则单元格$S_{ab}=1$。如果两者不是同义词，则根据 Wu-Palmer[27] 提出的基于 HowNet 中拓扑距离计算词相似度的方法，将相似度分数设置为 0～1 之间的值。此外，如果两个词中的一个或两个在 HowNet 中

不存在，或者对该单词不存在有效的 Wu-Palmer 相似度值（例如停止词等），则直接将相似度分数设置为 0。字级别相似度矩阵如图 3 所示。

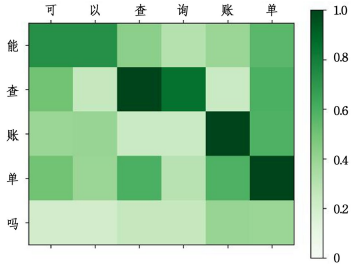

图 3 字级别相似度矩阵

Fig. 3 Char-level similarity matrix

FastText 对应的词向量部分与上述的 SimMatrix 相似度矩阵略有不同，如图 4 所示。

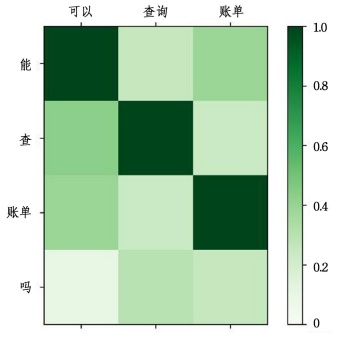

图 4 词级别相似度矩阵

Fig. 4 Word-level similarity matrix

最后，*Co-Attention* 的输出为两个 $1\times hidden\_size$ 维度的向量，分别为 $\boldsymbol{h}_a$ 和 $\boldsymbol{h}_b$；*MultiAttention* 的输出为维度为（*batch_size*，*seq_size*，*hidden_size*）的向量。提取多头注意力中的 [*CLS*] 部分后，得到向量 $\boldsymbol{h}_m\in(batch\_size,1,hidden\_size)$。

3.3 预测层

经过交互层后，可以得到 $\boldsymbol{h}_a$，$\boldsymbol{h}_b$ 和 $\boldsymbol{h}_m$ 3 个向量，将三者进行拼接，并输入到全连接层中。

$$\boldsymbol{h}_{\text{out}}=[\boldsymbol{h}_a;\boldsymbol{h}_b;\boldsymbol{h}_m] \tag{27}$$

$$\boldsymbol{h}_{\text{final}}=Linear(\boldsymbol{h}_{\text{out}}) \tag{28}$$

其中，$\boldsymbol{h}_{\text{final}}$ 中较大概率对应的标签为句子对的匹配结果。本文使用的损失函数如下：

$$L=-\frac{1}{N}\sum_{i=1}^{N}(y_i\log(p_i)+(1-y_i)\log(1-p_i)) \tag{29}$$

其中，y_i 表示第 i 个样本的标签，阳性样本的标签为 1，阴性样本的标签为 0；p_i 表示该样本被预测为阳性样本的概率。

4 实验

4.1 数据集

本文使用两个数据集进行实验。

第一个数据集为 BQ，该数据集是关于金融领域问题匹配的，由 12 万对句子组成，包括 10 万个训练样本，1 万个验证样本，1 万个测试样本。每一对句子关联着一个 0-1 标签，0 表示不匹配，1 表示匹配。其中，正样本和负样本的数量是相同的。数据集中样本分布的统计如表 1 所列。

表 1 BQ 数据集统计

Table 1 BQ dataset statistics

集合	总计	正样本	负样本
训练集	100000	50000	50000
验证集	10000	5000	5000
测试集	10000	5000	5000

第二个数据集为 LCQMC，该数据集是一个大规模开放域的问题匹配语料库，由 260068 对汉语句子组成，其中训练样本 238766 个，验证样本为 8802 个，测试样本为 12500 个。每一对句子与一个 0-1 标签相关联，0 表示不匹配，1 表示匹配。在该数据集中，正样本比负样本多 30%。划分后的数据集统计如表 2 所列，样本举例如表 3 所列，两个数据集句子长度统计如图 5 和图 6 所示。

表 2 LCQMC 数据集统计

Table 2 LCQMC dataset statistics

集合	总计	正样本	负样本
训练集	238766	138574	100192
验证集	8802	4401	4401
测试集	12500	6250	6250

表 3 LCQMC 数据集样本举例

Table 3 Sample example of LCQMC dataset

句子对	标签	标签含义
大家觉得她好看吗 大家觉得跑男好看吗	0	不匹配
求秋色之空漫画全集 求秋色之空全集漫画	1	匹配
石家庄天气如何 石家庄天气怎样	1	匹配

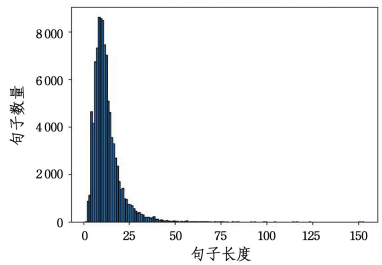

图 5 BQ 数据集句子长度统计

Fig. 5 Sentence length statistics of BQ dataset

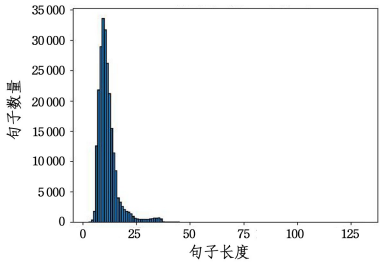

图 6 LCQMC 数据集句子长度统计

Fig. 6 Sentence length statistics of LCQMC dataset

从图 5 和图 6 中可以看出，两个数据集的句子长度大多集中在 0～50 个字符之间，所以 Bert 模型对应的字级别通道中，输入的文本结构和长度为：1（[*CLS*]）+50（*sen*1）+1（[*SEP*]）+50（*sen*2）+1（[*SEP*]），总长度为 103 个字符；

FastText 模型对应的词级别通道中，句子长度也设置为 50 个字符。

对于长度超过 50 的句子，先采取删除停用词或虚词的方法缩短句子长度，如果仍不满足要求，则直接将超出部分截断。

4.2 实验参数设置

实验参数如表 4 所列。

表 4 实验参数展示

Table 4 Display of experimental parameters

参数	值
操作系统	Linux 3.10
GPU	V100 32GB
框架	Pytorch
Batch Size	64
Learn Rate	2×10^{-5}
Epoch	10
损失函数	交叉熵损失函数
优化器	AdamW

4.3 实验评估方法

模型判定句子对是否匹配的评估指标是准确率(ACC)和 F1 值(F1)。具体计算式如下：

$$ACC=\frac{TP+TN}{TP+TN+FP+FN} \tag{30}$$

$$P=\frac{TP}{TP+FP} \tag{31}$$

$$R=\frac{TP}{TP+FN} \tag{32}$$

$$F1=\frac{2\times P\times R}{P+R} \tag{33}$$

其中，TP(True Positive)为真正例，FP(False Positive)为假正例，FN(False Negative)为假负例，TN(True Negative)为真负例。

4.4 实验结果与分析

为了验证本文模型的有效性，本节将其与如下模型进行对比。

BERT-Base-Chinese：基于谷歌 BERT 模型的中文语言模型。它是由清华大学自然语言处理实验室(THUNLP)开发的，使用了大规模的中文语料库进行训练，可以用于各种自然语言处理任务。

BERT-wwm：谷歌发布的 BERT 的升级版本。原有基于 WordPiece 的分词方式会把一个完整的词切分成若干个子词，在生成训练样本时，这些子词会被随机 mask。但在 wwm 中，一个完整词的所有子词会被同时 mask。

BERT-wwm-ext：BERT-wwm 的升级版，其增加了训练数据集和训练步数。

ERNIE：相比于 BERT，ERNIE 改进了两种 mask 策略。在 ERNIE 中，将由多个字组成的 phrase 或者 entity 当成一个统一单元，在训练时，这个单元中的所有部分统一被 mask。

MacBERT-Base[28]：使用全词 masked 以及 N-gram masked 策略来选择候选 Token 进行 masked，单词级别的 unigram 到 4-gram 的比例为 40%，30%，20%，10%，且论文建议在 mask 阶段使用类似的单词进行 masking。

K-BERT：为了在特定领域发挥更好的性能，K-BERT 将知识图谱(Knowledge Graph，KG)集成到预训练模型中，在知识查询后将知识注入到预训练模型的输入部分形成句子树。

上述模型和本文模型的对比实验结果如表 5 所列。

表 5 对比实验

Table 5 Contrast experiment

(%)

Model	BQ		LCQMC	
	ACC	*F1*	*ACC*	*F1*
BERT-Base-Chinese	84.50	84.00	85.73	86.86
BERT-wwm	84.89	84.29	86.80	87.78
BERT-wwm-ext	84.71	83.94	86.68	87.71
ERNIE	84.67	84.20	87.00	**88.06**
MacBERT-Base	85.20	—	87.00	—
K-BERT	—	—	87.10	—
Ours	**85.42**	**85.40**	**87.34**	87.28

从表 5 可以看出，本文提出的方法在 BQ 数据集的 ACC 和 F1 指标以及 LCQMC 数据集的 ACC 指标上都取得了最好的效果，证明了本文模型的有效性。在对于同义词识别不精确和知识过少的问题上，BERT-Base-Chinese，BERT-wwm 和 BERT-wwm-ext 等模型并未采取额外的措施，所以模型表现较差。K-BERT 模型虽融入知识图谱，丰富了字词与句子语义，但并未显式地将同义词额外处理，模型效果相较于其他模型虽然有所提升，但并不明显。而本文模型选择加入义原相似度矩阵，显式地将义原知识注入到模型中，针对同义词识别不精确的问题进行优化，并进一步提取词级别语义信息构成多粒度信息。相较于上述模型，本文模型表现突出，在 BQ 和 LCQMC 数据集的多个指标上取得了最好效果。

4.5 消融实验

为了验证本文模型中各个模块的有效性，在 BQ 数据集上进行消融实验，具体实验结果如表 6 所列。表中，SimMatrix 代表义原相似度矩阵。

表 6 消融实验

Table 6 Ablation experiment

ACC/%	84.62	84.71	84.67	84.88	85.02	**85.42**
F1/%	84.41	84.69	84.64	84.87	84.98	**85.40**
FastText			√	√	√	√
BiLSTM					√	√
SimMatrix		√		√		√
Attention	√	√	√	√	√	√

从表 6 中的前两列和后两列可以看到，本文提出的融入相似度矩阵的注意力机制相较于普通注意力机制的方法，在 ACC 指标上和 F1 指标上都有所提升。由于在注意力机制中注入了相似度矩阵，模型更加关注句子的相似部分且降低了无关词语的注意分数，故而提高了模型性能。从第二列与第四列的对比可以看出，加入 FastText 词向量后，模型性能也有约 0.3% 的提升，这是由于 Bert 等模型学习的主要单位为字，因此对词级别语义的捕捉不够充分，加入 FastText 模型后弥补了该缺陷。从第三列和第五列可以看出，加入 BiLSTM 后，模型性能也提升了约 0.3%，足以证明通过 BiLSTM 进一步提取词向量的上下文语义信息这一方法是有效的。

4.6 不同分词工具的对比

在 FastText 对应的词向量通道中，需要先将句子分词，再进行词向量的训练。由此可见，分词效果的不同对于后续模型的表现有着至关重要的作用，所以本节使用不同的分词工具进行实验，以验证分词效果对实验结果的影响。

由于可选择的分词工具较多，本文主要介绍 3 种：Jieba，Pkuseg，Hanlp。

Jieba[1)] 是最常见的分词工具，共有 3 种模式：精确模式、全模式、搜索引擎模式。本文采用的是精确分词模式。

Pkuseg 分词是北京大学语言计算与机器学习研究组进行开发并开源的项目，该工具匹配了专业领域的分词模型。由于本文所用数据集对应领域不包含在提供的专业领域中，因此实验使用通用模型。

HanLP 是由一系列模型与算法组成的 NLP 工具包，由大快搜索主导并完全开源，目标是普及自然语言处理在生产环境中的应用。对于本文的数据集，实验中采用 HanLP 对应的通用领域的中文分词模型。

使用 3 种分词工具对句子“微信没有微粒贷功能”和“为什么我用微信验证登录不了呢”进行分词效果展示，如表 7 所列。

表 7 不同分词工具的分词效果

Table 7 Word segmentation effects of different word segmentation tools

分词工具	分词效果
Jieba	微信 没有 微粒 贷 功能 为什么 我 用 微信 验证 登录 不了 呢
Pkuseg	微信 没有 微粒贷 功能 为什么 我 用 微信 验证 登录 不 了 呢
Hanlp	微信 没有 微粒贷 功能 为什么 我 用 微信 验证 登录 不 了 呢

3 种分词工具在 BQ 数据集上的实验结果如表 8 所列。从表中展示的实验结果来看，Jieba 分词工具的效果最好。

表 8 不同分词工具的实验效果

Table 8 Experiment effects of different word segmentation tools

(%)

分词工具	*ACC*	*F*1
Jieba	**85.42**	**85.40**
Pkuseg	85.20	85.17
Hanlp	85.25	85.24

4.7 FastText 维度实验

在本文模型中，词向量部分选择使用 FastText 模型。在 FastText 的训练过程中，可以对维度参数进行调整，而维度不同也会导致模型获取的语义信息不同。为了验证 FastText 模型不同维度的效果，本节对 FastText 模型的多个维度进行实验。实验中将模型超参数 hiddensize 分别设置为 100，384，768，实验结果如表 9 所列。

表 9 FastText 不同维度的实验效果

Table 9 Experiment effect of different dimensions of FastText

(%)

FastText 维度	*ACC*	*F*1
768	**85.42**	**85.40**
384	85.09	85.07
100	84.72	84.70

从表 9 可以看出，FastText 词向量在 768 维的时候效果最好；且随着维度的增加，实验效果也相应提高。本文分析可能是因为维度更高导致可获取的语义信息更多，进而使得整体模型的性能也有所提升。

4.8 案例分析

本节在 BQ 数据集上选取多条数据进行案例分析，以直观感受义原相似度矩阵和词向量通道的有效性，如表 10 所列。

表 10 案例分析(Bert 和 Ours 的对比)

Table 10 Case study(comparison between Bert and Ours)

	句子 1	句子 2	标签(括号内为真实标签)
Bert	我换手机号了 怎么才算是邀请朋友	如果我换手机怎么办 不是邀请的，要怎么开通	0(1) 1(0)
Ours	我换手机号了 怎么才算是邀请朋友	如果我换手机怎么办 不是邀请的，要怎么开通	1(1) 0(0)

表 10 所列的两个案例中，Bert 基础模型都无法获得正确的预测标签，而本文模型都可以正确预测。造成这一结果的原因是，Bert 模型在预测时更加关注不相似部分，例如“了”“如果”和“怎么办”等字词部分，模型预测的难度增大，使得不相似部分过于干扰模型中相似部分的语义表达；而本文模型融入义原相似度矩阵，使得模型更加关注相似字词部分，且降低了对不相似字词部分的注意力分数，模型学习倾向于对预测有益的语义信息。

4.9 大语言模型对比分析与时间性能实验

随着大语言模型的发展，NLP 领域某些任务的许多传统方案被取代，但由于大模型的部署成本较大，且推理时间较长，因此传统方法在实际应用中还有着一席之地。为了验证本文模型等传统方案的有效性，本节选取 Bert、本文模型、Qwen-7B 和 Baichuan2-7B 大模型进行对比实验。其中，Qwen-7B[29] 和 Baichuan2-7B[30] 为目前综合效果最好的两个中文开源大语言模型。由于服务器的显存限制，选择参数量为 7B 的版本。Qwen 大模型由阿里推出，Baichuan2 大模型由百川智能团队推出。本实验选择上述两个模型的 Chat 版本进行实验(base 版本需要进行 few-shot 微调，无法进行有效的公平对比)。由于大模型的 prompt 设置不同会导致输出结果不同，本实验设置多条 prompt 得到测试结果后取平均值，所以本实验的计算指标仅作为实验对照参考，不可作为实际指标值。实验对 BQ 和 LCQMC 数据集的测试集进行测试并计算 ACC 和 F1 指标，实验结果如图 7 和图 8 所示。

从实验结果来看，在类似文本匹配的具体领域上，没有进行微调的 Qwen 和 Baichuan2 大模型表现不如传统方案，但

1) http://github.com/fxsjy/jieba

Qwen 和 Baichuan2 大模型可以在多个数据集上不进行微调的情况下发挥出较为优异的性能。在实际应用中，对于训练语料较少甚至缺失的应用场景，大模型的性能可以完美发挥，而传统方案则对此无能为力。

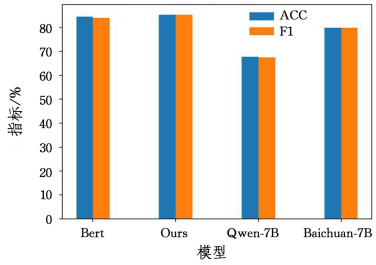

图 7 BQ 数据集测试指标

Fig. 7 Test metrics of BQ dataset

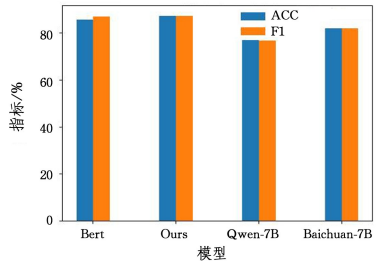

图 8 LCQMC 数据集测试指标

Fig. 8 Test metrics of LCQMC dataset

在实际应用中，文本匹配任务较多地用于智能问答、智能客服等领域，对推理时间要求较高。因此，对上述 4 个模型进行时间性能实验，单条数据的时间测试结果以及对 BQ 和 LCQMC 测试集进行测试的总耗时如图 9 和图 10 所示。其中，单条数据测试时间包含了加载模型的时间。

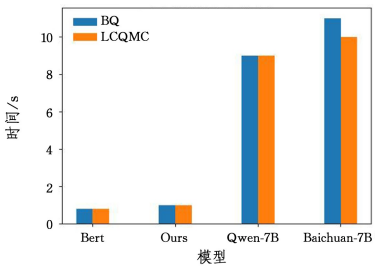

图 9 单条数据测试时间的对比

Fig. 9 Comparison of testing time of single data

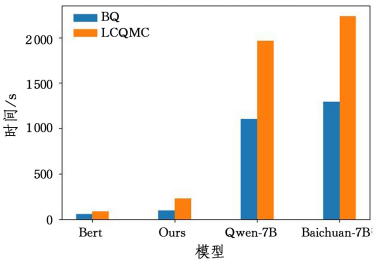

图 10 所有数据测试时间的对比

Fig. 10 Comparison of testing time of all data

由图 9 和图 10 可以得出，本文模型在推理所有测试数据时，花费的时间约为 Bert 模型的 2 倍；在单条数据推理时，花费时间也有所增多，但差异很小。因为本文模型在测试数据时，需要加载预训练好的相似度矩阵，所以存在速度差异。在实际应用中，文本匹配多用于单条以及少量数据推理，所以本文模型的时间消耗差异对实际应用并无明显影响。另外，从传统方法 Bert 和本文模型与大模型的推理速度对比来看，大模型 Qwen 和 Baichuan 的推理速度明显慢于传统方法，所以在某些对推理速度有严格要求的应用场景下，大模型的效果不如传统方法。

总结来说，在短文本语义匹配任务上，对比 Bert 等其他模型而言，本文模型加入了义原知识，因此对同义词的识别更加精确且补充了短文本的语义信息。另外，对于文本匹配任务来说，本文模型通过义原相似度矩阵对两个句子建立联系，帮助基础模型进一步交互。但义原信息的加入也有诸多局限，例如义原知识库的选择对本文模型的性能有影响，义原知识的预处理较为麻烦，以及时间性能略差于 Bert 等模型。上述分析与实验证明本文模型综合效果优于其他对比模型，且义原信息的引入对于短文本语义匹配任务有积极意义。

结束语 本文提出一种融合义原相似度矩阵与字词向量双通道协同优化的短文本语义匹配策略。首先，分别利用 Bert 和 FastText 模型组成字、词双通道并对文本进行编码，获取文本的字、词级别的语义信息。然后，将词级别通道加入 BiLSTM 模型，进一步提取上下文语义信息。为了解决同义词识别不精确以及未加入特定外部知识的问题，在双通道中分别加入多头注意力和协同注意力，同时引入义原相似度矩阵，将其注入注意力机制中。最后，将上述句子向量拼接并输入到分类器中进行语义一致性的判断。为了验证本文模型的有效性，在 BQ 和 LCQMC 数据集上进行了对比实验。实验表明，与多个模型对比，本文模型有着较为优秀的结果。通过对各个模块的组合进行消融实验，证明了各个模块的有效性。

未来，将继续探索更好地解决文本语义匹配任务中同义词识别不精确和语义信息不足问题的方案，并针对单字的语义歧义而引起的相似度数值误差问题提出解决方案。另外，由于预训练模型对不同语言的处理方式不同，还可对不同语言的数据集进行探索。

参 考 文 献

[1] DEVLIN J, CHANG M W, LEE K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [C]// North American Chapter of the Association for Computational Linguistics. Minneapolis, Minnesota: ACL, 2019: 4171-4186.

[2] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All You Need[J]. Advances In Neural Information Processing Systems, 2017, 30: 5998-6008.

[3] QI F, YANG C, LIU Z, et al. Openhownet: An Open Sememe-based Lexical Knowledge Base[J]. arXiv: 1901.09957, 2019.

[4] ARMAND J, EDOUARD G, PIOTR B, et al. Bag of Tricks for Efficient Text Classification[C]// Conference of the European Chapter of the Association for Computational Linguistics. Valencia, Spain: ACL, 2017: 427-431.

[5] HUANG Z H, XU W, YU K. Bidirectional LSTM-CRF Models for Sequence Tagging[J]. arXiv: 1508.01991, 2015.

[6] LU J, YANG J, BATRA D, et al. Hierarchical Question-Image Co-Attention for Visual Question Answering[C]// Conference on Neural Information Processing Systems. 2016: 289-297.
[7] HUANG P S, HE X D, GAO J F, et al. Learning Deep Structured Semantic Models for Web Search Using Click through Data[C]// International Conference on Information and Knowledge Management. 2013: 2333-2338.
[8] CHEN Q, ZHU X D, LING Z H, et al. Enhanced LSTM For Natural Language Inference[C]// Annual Meeting of the Association for Computational Linguistics. 2017: 1657-1668.
[9] GONG Y C, LUO H, ZHANG J. Natural Language Inference over Interaction Space[J]. arXiv: 1709.04348, 2017.
[10] TAN C, WEI F, WANG W H, et al. Multiway Attention Networks for Modeling Sentence Pairs[C]// International Joint Conference on Artificial Intelligence. 2018: 4411-4417.
[11] LAN Z Z, CHEN M, GOODMAN S, et al. Albert: A Lite Bert for Self-supervised Learning of Language Representations[C]// International Conference on Learning Representations. 2020.
[12] LIU Y H, OTT M, GOYAL N, et al. Roberta: A Robustly Optimized Bert Pretraining Approach[J]. arXiv: 1907.11692, 2019.
[13] ZHANG Z S, WU Y W, ZHAO H, et al. Semantics-aware BERT for Language Understanding[C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2020: 9628-9635.
[14] ZHANG Z Y, HAN X, LIU Z Y, et al. ERNIE: Enhanced Language Representation with Informative Entities[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Italy: ACL, 2019: 1441-1451.
[15] LIU W J, ZHOU P, ZHAO Z, et al. K-Bert: Enabling Language Representation with Knowledge Graph[C]// AAAI Conference on Artificial Intelligence. 2020: 2901-2908.
[16] HE P C, LIU X D, GAO J F, et al. DeBERTa: Decoding-enhanced BERT with Disentangled Attention[C]// International Conference on Learning Representations. 2021.
[17] LYU B, CHEN L, ZHU S, et al. Let: Linguistic Knowledge Enhanced Graph Transformer for Chinese Short Text Matching [C]// AAAI Conference on Artificial Intelligence. 2021: 13498-13506.
[18] BAI J G, WANG Y J, CHEN Y R, et al. Syntax-BERT: Improving Pre-trained Transformers with Syntax Trees[C]// Conference of the European Chapter of the Association for Computational Linguistics. Minneapolis, Minnesota: ACL, 2021: 3011-3020.
[19] LI Y L, ZHOU Y P. Text Similarity Matching Based on Twin Network and Char-Word Vector Combination[J]. Applications of Computer Systems, 2022, 31(10): 295-302.
[20] LYU X F, ZHAO S L, GAO H D, et al. Short Texts Feautre Enrichment Method Based on Heterogeneous Information Network [J]. Computer Science, 2022, 49(9): 92-100.
[21] YU E, DU L, JIN Y, et al. Learning Semantic Textual Similarity via Topic-informed Discrete Latent Variables[C]// Conference on Empirical Methods in Natural Language Processing. 2022: 4937-4948.
[22] WANG S, LIANG D, SONG J, et al. DABERT: Dual Attention Enhanced BERT for Semantic Matching[C]// International Conference on Computational Linguistics. 2022: 1645-1654.
[23] ZOU Y C, LIU H W, GUI T, et al. Divide and Conquer: Text Semantic Matching with Disentangled Keywords and Intents[C]// Annual Meeting of the Association for Computational Linguistics. Findings of the Association for Computational Linguistics. Dublin, Ireland: ACL, 2022: 3622-3632.
[24] CHEN M Y, JIANG H Y, YANG Y J. Context Enhanced Short Text Matching using Clickthrough Data[J]. arXiv: 2203.01849, 2022.
[25] ZHANG H Y, DUAN L G, WANG Q C, et al. Long Text Multi-entity Emotion Analysis Based on Multi-task Joint Training[J]. Computer Science, 2024, 51(6): 309-316.
[26] JIANG K X, ZHAO Y H, JIN G Z, et al. KETM: A Knowledge-Enhanced Text Matching Method[C]// International Joint Conference on Neural Networks. 2023: 1-8.
[27] WU Z B, PALMER M. Verb Semantics and Lexical Selection [C]// Annual Meeting of the Association for Computational Linguistics. 1994: 27-30.
[28] CUI Y M, CHE W X, LIU T, et al. Revisiting Pre-Trained Models for Chinese Natural Language Processing[C]// Findings of the Association for Computational Linguistics: EMNLP. 2020: 657-668.
[29] BAI J, BAI S, CHU Y F, et al. Qwen Technical Report[J]. arXiv: 2309.16609, 2023.
[30] YANG A Y, XIAO B, WANG B N, et al. Baichuan2: Open Large-scale Language Models[J]. arXiv: 2309.10305, 2023.

LIU Dongxu, born in 1999, postgraduate. His main research interests include text matching and so on.

DUAN Liguo, born in 1970, Ph.D, professor, postgraduate supervisor, is a member of CCF(No. 15823S). His main research interests include natural language processing and so on.

(责任编辑:柯颖)