arXiv:2302.10570v1 [cs.CL] 21 Feb 2023

Transformer与HowNet义原知识融合的协同驱动语义一致性识别

Fan Chen[1], Yan Huang[2]*、Xinfang Zhang[1]、Kang Luo[1]、Jinxuan Zhu[1],和Ruixian He[1]

[1] 华中科技大学机械科学与工程学院，中国武汉[2] 华中科技大学人工智能与自动化学院，中国武汉

摘要。 语义一致性识别旨在检测并判断两个文本句子的语义是否一致。然而，现有方法通常面临同义词、多义性以及难以理解长文本等挑战。为解决上述问题，本文提出了一种基于Transformer与HowNet义原知识融合的协同驱动语义一致性识别方法。首先，利用Transformer以数据驱动的方式对句子内部结构进行多层级编码；随后，引入义原知识库HowNet，以知识驱动的方式对句子对之间的语义知识关联进行建模。然后，利用软注意力进行交互式注意力计算，并将知识与义原矩阵融合。最后，采用双向长短期记忆网络（BiLSTM）编码概念语义信息，并推断语义一致性。本文在两个金融文本匹配数据集（BQ、AFQMC）以及一个用于释义识别的跨语言对抗数据集（PAWSX）上进行了实验。与DSSM、MwAN、DRCN等轻量级模型以及ERNIE等预训练模型相比，所提出的模型不仅能有效提升语义一致性识别的准确率（在BQ数据集上，相较于DSSM、MWAN和DRCN模型分别提升2.19%、5.57%和6.51%），还可减少模型参数量（约为16M）。此外，在HowNet义原知识的驱动下，所提出的方法有望适用于长文本场景。

关键词： 语义一致性 · 知网 · Transformer · 义原知识 · 知识融合。

1 引言

语义一致性识别（或文本语义匹配）任务是自然语言处理（NLP）领域的重要任务之一，可应用于多种下游任务，例如信息检索、问答、对话系统、机器翻译等。该任务的输入主要是句子对，不同于

* 通讯作者：Yan Huang（电子邮箱：platanus@hust.edu.cn）

旨在识别句子对之间语义关系（中立、蕴含、矛盾）的自然语言推理（以及文本蕴含），语义一致性识别的最终目标是判断句子对的语义是否一致或相似。

由于多义性和同义关系的特点，语义一致性识别是一项颇具挑战性的任务;这些特点在中文中尤为突出，可能导致歧义和误解。然而，大多数现有算法，包括词袋模型（BOW）、向量空间模型（VSM）、词频-逆文档频率（TF-IDF）以及 DSSM [2],MwAN [3], DRCN [4]，都无法准确捕捉语义含义。这些算法主要解决词汇层面的匹配或相似度问题，难以根据上下文准确理解文本语义。具体而言，基于词汇重合和数据驱动神经网络的现有文本匹配算法存在以下局限:

（1）词汇的语义多样性。同一个词在不同语境中可以表达不同含义，例如，“苹果”在菜单中可能指一种水果，而在电子产品市场中也可能指“苹果公司”。

（2）短语的含义取决于表达顺序。对于词汇短语，调换顺序后所表达的含义会完全不同，例如中文中的“奶牛（cow）”和“牛奶（milk）”，英文中的“one another（彼此，互相）”和“another one（另一个）”。

为解决上述问题，本文提出一种基于 Transformer [1] 和知网 [22], 的语义一致性识别方法，以拓展句子语义信息获取方面的研究。首先，我们利用 Transformer 通过数据驱动的方式进行多层级编码，以表征文本句子的内部结构和内在语义。随后，我们引入外部知识库——知网，对词汇之间的语义知识关联进行知识驱动建模。此外，我们利用软注意力计算相互注意力，并与语义矩阵实现知识融合。最后，结合双向长短期记忆网络，进一步编码文本概念层级的语义信息，并推断语义一致性。大量实验表明，与现有轻量级模型和预训练模型相比，融合知网义原知识并发挥 Transformer 在长文本处理方面的优势，能够在一定程度上提高语义一致性识别的准确率。

本文提出的模型具有以下创新之处:

– 为了解决语义一致性识别中文本同义词、多义性以及长文本难以理解的问题，提出了一种基于 Transformer 与 HowNet义原知识融合的协同驱动方法。除采用 Transformer 进行编码并通过数据驱动进行模型预训练外，还引入 HowNet义原，通过知识驱动嵌入与推理增强对同义词和多义性的理解。– 提出了一种语义知识融合技术方法。通过 Transformer 对句子对的多层次内部结构和语义信息进行编码，并引入外部知识库 HowNet，对义原序列之间的语义知识相似度进行建模。最后，利用软注意力

计算交互注意力，并通过义原矩阵进行知识融合。

– 在蚂蚁金融、银行金融场景的文本匹配以及多种释义识别应用中进行了实验。与 DSSM [2], 、MwAN [3], 、DRCN [4] 等轻量级模型以及 ERNIE 等预训练模型相比，所提出的方法不仅能有效提高文本语义一致性识别的准确率，还能减少模型参数。值得一提的是，我们的模型能够适用于长文本。代码将在 https://github.com/Platanus-hy/sememes_codriven_text_matching 发布。

2 相关工作

近年来，随着机器学习的快速发展，人们提出了大量方法来解决文本一致性识别问题。在语义信息获取方面，经典的短文本匹配模型 DSSM [2] 解决了潜在语义分析（LSA）和潜在狄利克雷分配（LDA）等方法中的词典爆炸问题，但由于采用词袋模型，也丢失了上下文信息。2016 年提出的 ESIM [5] 模型综合运用了双向长短期记忆网络和注意力机制，并首次在局部推理中对句子对进行交互。2018年提出的 DIIN [6] 模型使用卷积神经网络和长短期记忆网络进行特征提取，但作者在其输入层同时使用词向量和局部向量，输入了一些额外的句法特征，并使用密集连接网络进行特征提取。2018年提出的 DRCN 模型借鉴了 DenseNet [7] 在图像识别中的密集连接操作，通过与循环神经网络进行密集连接来保留文本最原始的信息，并通过多次循环持续向矩阵向量中添加交互信息，最终经由全连接层输出。2018年提出的 KIM [8] 模型使用外部知识库 WordNet [9] 推断句子对之间的逻辑关系，并将外部先验知识嵌入相似度矩阵。2018年提出的 MwAN [3] 模型中，作者使用多种注意力机制（拼接、双线性、点乘、减法）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加权并组合，输出最终概率。

在句子结构方面，CT-LSTM [10],于 2015 年提出，它引入了一种树形长短期记忆网络，以解决长短期记忆网络无法提取句子结构信息的问题，同时探讨了长序列依赖问题。与常用的循环神经网络序列建模不同，它将句子的依存关系作为长短期记忆网络的输入，这也为未来的研究带来了一些启发。

随着 BERT [11] 模型在2018年问世，预训练模型的浪潮席卷了整个自然语言处理领域，并在各大自然语言处理榜单中名列前茅。BERT 拥有完整的编码器-解码器框架。其基本组成部分是 Transformer，主要由多头注意力构成。它是一种纯粹基于注意力构建的模型，能够解决文本序列中的长距离依赖问题；也就是说，注意力机制能够对更长范围内的语义上下文进行建模和记忆

。BERT 模型的优势在于，它能从大型语料库中学习更多语法和语义信息，使输出的词向量更具代表性；更多的参数提升了模型的表达能力，使其在各种下游任务中表现出色。为了加快预训练模型的训练速度，并应对消费级 GPU 的高硬件要求，Tim Dettmers [12] 为 Transformer 提出了 LLM.Int8()，使普通消费级 GPU 无须牺牲性能即可运行规模极大的模型。除 Transformer模型 外，Hanxiao [13] 还提出了另一种简单且不依赖注意力的架构 gMLP。该架构在某些预训练指标上达到了与 Transformer模型 相同的水平，甚至在某些下游任务中表现更佳。

由于词汇的多义性较为特殊，文本匹配存在一定困难。不同词语往往表达相同的含义，例如“中国”和“华夏”，二者语义一致，但在字形上并无关联。为解决这一问题，许多研究者选择利用词性、依存句法等信息来计算相似度。例如，Yan [14] 等人尝试为文本标注词性，并仅保留名词、动词和形容词。他们结合依存句法分析获取词对，并以 PageRank [15]和度中心性为指标，为大量文本构建语法网络，提出了一种结合句法关系与词汇语义的文本相似度计算方法。Yan [16] 提出了一种基于主题约束的篇章级文本自动生成模型。该模型利用关键词集合的同义关系生成多个文章主题方案。Lin [17] 引入概念向量空间的概念，并将文档表示为概念词集合以构建向量空间，随后通过余弦相似度计算语义相似度；其效果优于结合 Word2Vec [18]的词袋模型（BOW）。

Boer [19] 于2021年提出的LET模型利用知网进行文本蕴含识别。他们首先通过图注意力对每个词汇词的所有义原的初始向量进行转换，然后通过注意力池化融合每个词的义原向量，并通过整合GRU和BERT词向量获得最终的词向量。词汇词通常具有歧义，只有少数相关义原对于识别正确语义至关重要，这会导致所获得的义原向量与实际句子不匹配，从而产生冗余语义信息。本文在将义原词纳入交互矩阵之前预先对其进行筛选，以避免引入冗余信息。

3 研究方法

本节主要介绍基于Transformer与HowNet义原知识融合的协同驱动文本语义一致性识别模型，并分析模型结构及其功能。所提出的模型结构如图1所示，共分为6层，分别是Transformer编码层、注意力层、BiLSTM层、池化层、全连接层和预测层。

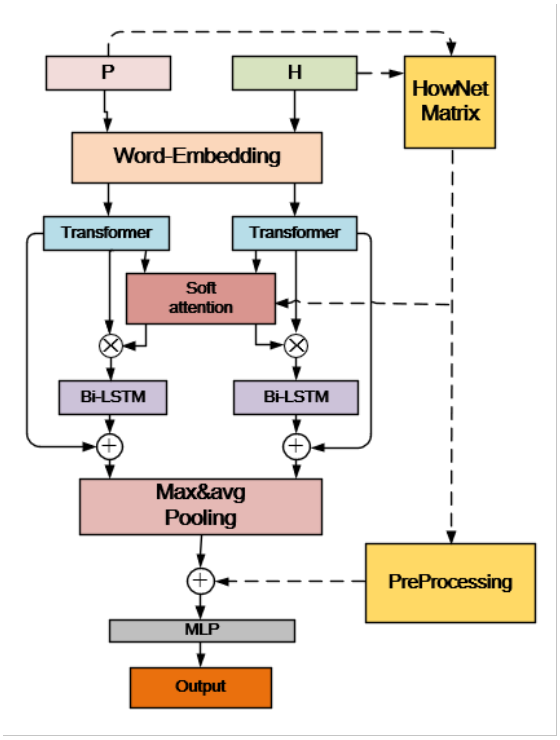


图 1. 模型结构。

3.1 Transformer 编码层

编码层的作用是通过神经网络对序列文本建模，从而获取深层语义信息。我们没有采用CNN、LSTM等常用神经网络，而是使用Transformer编码器架构进行句对编码，该架构主要由多头注意力模块构成，能够缓解梯度消失问题。多头注意力机制的形式化表示如下，见公式（1）至公式（4）：

$$Q = \mathrm{W}^{Q}X \tag{1}$$

$$K = \mathrm{W}^{K}X \tag{2}$$

$$V = \mathrm{W}^{V}X \tag{3}$$

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^{T}}{\sqrt{d_k}}\right)V \tag{4}$$

其中，X是输入句子，W^{Q}、W^{K}、$W^{V} \in R^{d_{\mathrm{model}} \times d^{k}}$是权重矩阵，$Q$、$K$、$V$分别是$X$在查询、键和值上的嵌入，即使用$Q$和$K$查询适当的键，然后通过 softmax分类器选择相应的值V。

此外，绝对位置编码按照公式（5）和公式（6）进行计算。

$$PE_{(p,2m)} = \sin\left(\frac{p}{10000^{2m/d}}\right) \tag{5}$$

$$PE_{(p,2m+1)} = \cos\left(\frac{p}{10000^{2m/d}}\right) \tag{6}$$

其中d 表示词嵌入的维度，p 表示词在句子中的索引。

3.2 注意力层

注意力层是文本一致性识别模型的重要组成部分，具有快速、高效且轻量的优点。近年来，人们提出了多种注意力机制，例如软注意力、硬注意力、自注意力等。Haili [20] 还提出了一种专注于细粒度情感的注意力机制。本文采用常用的软注意力机制，同时融入基于 HowNet义原知识生成的语义矩阵，并通过可训练权重 γ 对其进行平衡。

HowNet 是一个面向中英文的词汇义原知识库，揭示了概念与其义原之间的语义关系。本文主要利用 HowNet 获取句子对中各个词对应的所有词汇义原。也就是说，如果两个词共享相同的义原，则 HowNet 义原矩阵中对应位置的值设为 1，否则设为 0。图 2 展示了句子“我们都是中华儿女(We are all childrens of 中国)”和句子“华夏起源于先秦(Huaxia originated in the 先秦)”的义原交互计算示例。

顶部的方框表示作为词嵌入输入的P 和H 的分词句子，中间的方框表示每个词对应的多个义原信息，下方的方框表示两个词的义原信息之间的交互。HowNet交互矩阵按公式（7）和公式（8）计算。从图2可以看出，单词“China(中华)”的义原包括“China(中国)”、“country(国家)”、“local(地方)”、“Asia(亚洲)”和“politics(政)”，而单词“Huaxia(华夏)”的义原包括“borrow(借入)”、“finance(金融)”、“money(货币)”、“China(中国)”、“place(地方)”和“Asia(亚洲)”。两个单词的交集为“China(中国)”、“place(地方)”和“Asia(亚洲)”。

$$M_{i,j} = \begin{cases} 1 & \text{HowNet}\left(P_i, H_j\right) \neq 0 \\ 0 & \text{How Net}\left(P_i, H_j\right) = 0 \end{cases} \tag{7}$$

$$M = \begin{bmatrix} 0 \cdots 1 \\ \vdots \ddots \vdots \\ 1 \cdots 0 \end{bmatrix} \tag{8}$$

其中，P 和H 分别表示句子对经分词工具（如结巴、PKUseg、HanLP 等）处理后的嵌入结果。词汇

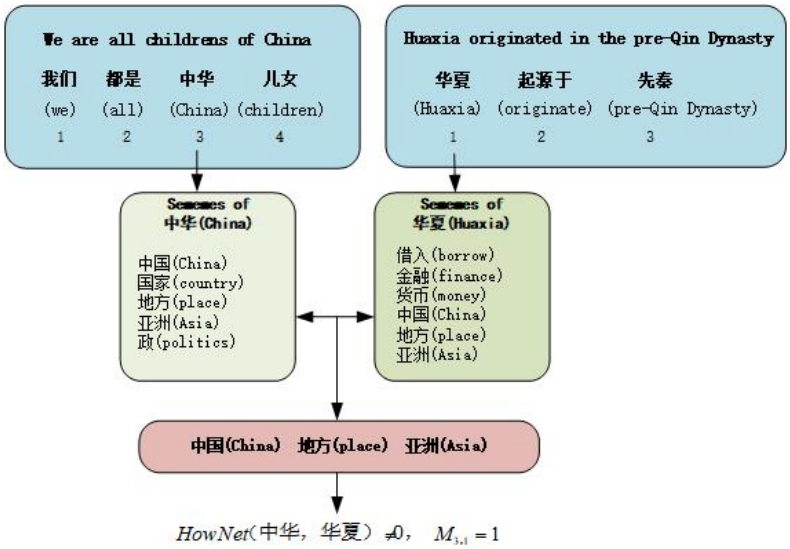


图2. 知网语义分析。

在不使用预训练时，通过随机初始化将其向量化。使用预训练时，则采用BERT 的词嵌入。注意力矩阵的生成过程可形式化表示为式 (9)：

$$e = PH^T + \gamma \cdot M \tag{9}$$

其中， γ 是一个可训练参数。注意力矩阵e 不仅整合了句子之间的结构和语义信息，还获得了句子间词对的同义和多义关系。矩阵变化的热力图如图4所示。注入义原信息后，一些交叉位置的权重增大，这表明这些位置反映了两个句子语义一致性的关键特征。结合注意力矩阵后，软注意力的计算如下：

$$\hat{P} = \sum_{j=1}^{l_h} \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{l_h} \exp\left(e_{ik}\right)} P_{tf}, \forall i \in \left[1, 2, \cdots, l_p\right] \tag{10}$$

$$\hat{H} = \sum_{j=1}^{l_p} \frac{\exp\left(e_{ij}\right)}{\sum_{k=1}^{l_p} \exp\left(e_{ik}\right)} H_{tf}, \forall i \in [1, 2, \cdots, l_h] \tag{11}$$

在公式（10）和公式（11）中，P_{tf} 和H_{tf} 分别是句子对经过Transformer编码后的嵌入矩阵。l_p 和l_h 表示句子的长度以及软注意力模块的输出。

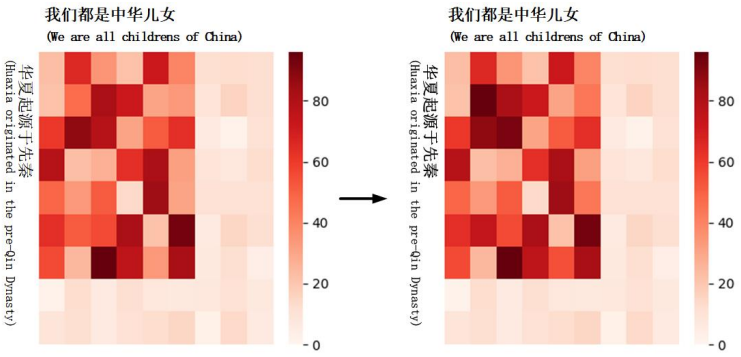


图 3. 注意力矩阵的变化。

3.3 双向长短期记忆网络层

该层用于处理经过软注意力机制后的Transformer层输出，即$\hat{P}$和$\hat{H}$。通过双向长短期记忆（BiL- STM），将前向编码和后向编码拼接起来，以获得上下文信息。

长短期记忆（LSTM）网络的输出如下：

$$P_{bi-lstm} = \text{BiLSTM}(\hat{P}) \tag{12}$$

其中，$\hat{P}$表示句子P经软注意力机制得到的输出，而$P_{bi-lstm}$表示$\hat{P}$经BiLSTM模块得到的编码结果。

3.4 平均池化与最大池化

为了融合经过Transformer和双向长短期记忆网络处理后的文本信息，该模型通过最大池化和平均池化拼接多个输入。该层的目的是将两个句子的向量维度从$R^{l\times d}$转换为$R^{l\times d}$，以便后续输入全连接层：

$$P_o = \left[P_{tf}; P_{bi-lstm}\right] \tag{13}$$

$$P_{rep} = \left[\text{MaxPool}\left(P_o\right); \text{AvgPool}\left(P_o\right)\right] \tag{14}$$

其中，P_{tf}表示Transformer编码模块的输出，$P_{bi-lstm}$表示BiLSTM模块的输出，P_O是P_{tf}和$P_{bi-lstm}$的拼接结果，P_{rep}表示对P_O进行最大池化和平均池化后所得输出的拼接结果。

3.5 全连接层

在获得句对的完整句子向量表示，即P_{rep}和H_{rep}后，常用的向量拼接方法是将它们直接拼接并输入

到多层前馈神经网络中以获得结果。进行拼接时，会考虑 HowNet矩阵中的信息，并将 HowNet矩阵中两个不同维度的和分别记为HN_{row}（行求和）和HN_{col}（列求和）。最终，通过与P_{rep} 和H_{rep}拼接，得到前馈神经网络的输入H 。

$$\begin{aligned} HN_{\text{row}} &= \text{sum}(M, axis = 0) \\ HN_{\text{col}} &= \text{sum}(M, axis = 1) \\ H &= \text{concat}\left(P_{rep}; H_{\text{rep}}; P_{rep} - H_{rep}; HN_{\text{col}}; HN_{\text{row}}\right) \end{aligned} \tag{15}$$

其中，$sum(M,axis)$表示沿M 的轴向维度求和。例如，HN_{row} 表示沿第一维对知网矩阵求和的结果，即与句子P对应的知网信息。通过向量拼接，可以获得两个句子对应的语义信息，其中，$P_{rep} - H_{rep}$ 也表示两个句子向量之间的差异。

3.6 预测层

获得句子对的最终句子向量表示后，模型使用两层全连接神经网络和一个softmax层，将句子匹配结果分类为1（正向）或0（负向），如式（16）所示。交叉熵损失函数的计算如式（17）所示。

$$p = softmax(FFN(FFN(H))) \tag{16}$$

$$Loss = \frac{1}{N}\sum_{i}^{N} -[y_i \times \log(p_i) + (1 - y_i) \times \log(1 - p_i)] \tag{17}$$

其中，y_i 表示第i个样本（P_i,H_i）的标签，正样本的标签为1，否则为0。p_i 表示样本被预测为正样本的概率。

除了常用的交叉熵损失函数外，我们还尝试了 CoSent损失函数。该函数会强制使正样本对的相似度高于负样本，并尽可能拉大向量空间中正样本与负样本之间的距离。实验表明，使用 CoSent损失函数对 BERT 和 Sentence-BERT 等预训练方法有一定效果， [21], 能够缩短预训练模型的收敛时间。但对于本文提出的模型，在不进行预训练的情况下，与交叉熵损失函数相比，CoSent损失函数的效果有所减弱。

在训练阶段，我们使用 MultiStepLRto 动态调整学习率。当训练轮次达到某个里程碑（即实验的第 20、50、80、100 和 150 次迭代）时，各参数组的学习率便会按 0.5 的衰减率降低。随着迭代次数增加，动态调整学习率能够加快模型的收敛速度，其变化趋势如图 4 所示。

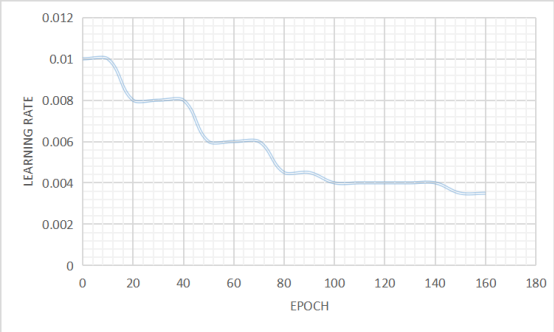


图 4. 学习率的衰减趋势。

4 实验与分析

4.1 数据集

表 1. 数据集大小

数据集名称	训练集大小	验证集大小	测试集大小
PAWSX	49401	2000	2000
AFQMC	34334	4316	3861
BQ语料库	100000	10000	10000

为验证本文提出的基于 Transformer 和知网双重驱动的文本一致性识别模型的有效性，本文分别在三个公开数据集上进行了实验。这些数据集分别为 PAWSX 数据集、AFQMC 数据集和 BQ语料库数据集。

PAWSX数据集是谷歌发布的多语言定义对数据集。其特点是词汇高度重叠，有助于进一步提升模型对困难样本的判断能力，并可测试模型对相似样本的判断能力。AFQMC数据集是蚂蚁金融相似度数据集，包含34,334条训练数据、4316条验证数据和3861条测试数据。BQ语料库是银行与金融领域的问题匹配数据集，包含从一年的网上银行系统日志中提取的问题对。它是目前银行领域最大的问题匹配数据集，包含10000条训练数据、10000条验证数据和10000条测试数据。

4.2 实验设置

本文的实验在一台配备RTX2080ti的四卡GPU服务器上进行。用于模型训练的参数和软件如表3所示。

表2. BQ数据集示例

句子1	句子2	标签
微信消费算吗 （微信消费算不算）	还有多少钱没还 （还有多少货币未偿还）	0（负数）
下周有什么好产品 （下周有哪些好产品）	一月份有哪些理财产品 （有哪些金融产品 一月份）	1（正面）
能查账单吗 （可以查账单吗）	可以查询账单 （可以查询账单）	0（负数）
现在无法借款 （现在无法借入）	QQ有微粒贷平台 （QQ有微粒贷产品）	0（负面）

表 3. 模型的初始训练参数

参数	值
词嵌入维度	300
隐藏层数量	128
最大序列长度	100
批次 _size	64
Transformer 层数量	10
优化器	Adam
初始学习率	0.01
软件	版本
Python	3.6.13
PyTorch	1.10.2
OpenHowNet	2.0
transformer模型	4.18.0

12 FanChen、YanHuang 等。

4.3 实验结果比较

为验证本文所提模型的实际效果，我们选取了三种经典文本匹配模型进行比较，分别是 DSSM、MwAN 和 DRCN，且均未进行预训练。对于预训练模型，我们选择了 BERT-wwm-ext、BERT 和百度ERNIE。

所选数据集为 PAWSX、AFQMC 和 BQ语料库。为确保实验的一致性，所有模型在同一数据集上均使用相同的结巴词表，比较指标为准确率和 F1分数。

表4. BQ数据集上的实验结果

模型名称	预训练	Acc	F1
DSSM	✗	77.12	76.47
MwAN	✗	73.99	73.29
DRCN	✗	74.65	76.02
Ours	✗	**78.81**	**76.62**
提升	✗	+2.19%	+1.96%
BERT-wwm-ext	✔	84.71	83.94
BERT	✔	84.50	84.00
ERNIE	✔	84.67	84.20
Ours-BERT	✔	**84.82**	**84.33**
提升	✔	+0.177%	+0.464%

表5。 AFQMC数据集上的实验结果

模型名称	预训练	Acc	F1
DSSM	✗	57.02	30.75
MwAN	✗	65.43	28.63
DRCN	✗	66.05	40.60
Ours	✗	**66.62**	**42.93**
提升	✗	+0.86%	+5.7%
BERT-wwm-ext	✔	81.76	80.62
BERT	✔	81.43	79.77
ERNIE	✔	81.54	80.81
Ours-BERT	✔	**81.84**	**81.93**
提升	✔	+0.097%	+1.38%

从表4可以看出，所提出模型在BQ数据集上的准确率高于其他模型。如表5所示，从数据集的角度来看，三个模型在AFQMC上的结果都不太理想。初步分析表明，样本数据的语言规范化程度较低，例如句子不完整，

表6. PAWSX数据集上的实验结果

模型名称	预训练	Acc	F1
DSSM	✗	42.64	59.43
MwAN	✗	52.70	52.65
DRCN	✗	61.24	56.52
Ours	✗	**62.55**	**59.72**
提升	✗	+2.13%	+0.48%
BERT-wwm-ext	✔	77.23	76.52
BERT	✔	77.06	77.16
ERNIE	✔	78.02	77.59
Ours-BERT	✔	**78.33**	**77.96**
提升	✔	+0.397%	+0.476%

例如“可以帮我冻结花呗吗（可以冻结我的花呗账户吗）”和“冻结花呗额度”。标签一致，导致模型在训练集和测试集上的表现不佳。然而，本文提出的模型融合了Transformer和知网义原知识库，效果更好。结果表明，对于非规范文本，通过获取部分词语的语义信息并进行匹配，可以提升模型性能。如表6所示，对于数据样本难以采样的PAWSX数据集，传统的DSSM模型无法获取交互信息和上下文信息，因此效果较差。对于困难负样本，高度相似的句子对会导致语义信息过于相似。每对句子生成的HowNet矩阵几乎完全相同，因此，获取语义知识的方法不利于判断困难样本中的正负样本。

从误差分析的角度来看，由于我们直接使用结巴分词对文本进行预处理，分词产生的错误对实验结果造成了不同程度的影响。尽管存在分词错误，但对于同一数据集，所有模型都使用相同的词汇表，而且本文提出的模型比其他模型更有效。

4.4 消融实验

本节针对所提出模型的不同结构，在BQ数据集上开展消融实验，以评估模型各模块的必要性和有效性。表7展示了使用或不使用知网时，采用不同分词工具所产生影响的评估结果。表8展示了不同序列长度的影响评估结果。此外，表9评估了用于文本编码的Transformer层数。

从表7的实验结果可以看出，使用知网能够在一定程度上提升模型的性能。与不使用知网相比，各种分词工具的准确率均有所提高。如果数据样本中存在一些语义复杂的词语，引入外部知识库

表7. 使用不同分词器并结合或不结合知网时，在BQ数据集上的语义一致性识别结果。

分词器	知网	准确率	F1
结巴	✔	**0.7881**	**0.7662**
	✗	0.7783	0.7624
PKUseg	✔	**0.7869**	**0.7653**
	✗	0.7792	0.7611
HanLP	✔	**0.7853**	**0.7599**
	✗	0.7735	0.7512

能够显著提高模型对多义性和同义词的敏感度，并显著提升模型的性能。

表8. 在不同序列长度下，使用或不使用知网时，BQ数据集上的语义一致性识别结果。

序列长度	知网	准确率	F1
1~15	✔	**0.7869**	**0.7662**
	✗	0.7763	0.7521
15~50	✔	**0.7884**	**0.7684**
	✗	0.7792	0.7545

根据表8中的结果，知网能够有效提升长度小于15的文本以及长度大于15且小于50的文本的表现，并且能够从较长文本中获取更多有效的语义信息，从而取得更好的结果。在数据集中进行了最长文本片段实验后，在保证实验效果的基础上，该模型能够处理的最长文本长度为50。

表9. BQ数据集上不同Transformer编码层数量的语义一致性识别结果。

Transformer层数量	Acc	F1
2	0.7433	0.7353
4	0.7648	0.7572
6	0.7752	0.7731
8	0.7853	0.7658
10	0.7881	0.7662

从表9的结果数据可以看出，Transformer层数越多，模型效果越好。通过堆叠Transformer编码层，可以在一定程度上提升模型性能，但与此同时，模

型的参数数量和训练时间也会显著增加，收敛速度也会明显变慢。我们选取具有6个编码层的模型作为最优模型，其参数量为16M，而在未进行预训练的模型中表现最佳的DRCN模型参数量为19M。训练过程中，可训练参数的变化如下：/par

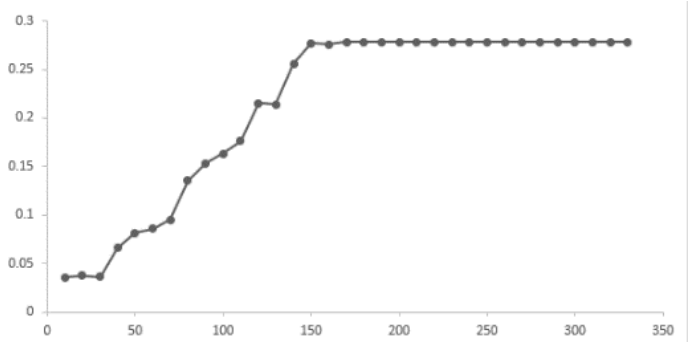


图5。 可训练参数变化图

从图5可以看出，在实验过程中，通过观察注意力矩阵中可训练参数的变化，发现知网获得的义原信息矩阵在注意力矩阵中的权重随着迭代次数的增加而逐渐增大，这表明原文本生成的知网义原信息对提升模型效果具有积极作用。

5 结论

本文提出了一种基于 Transformer 和知网的新型文本语义一致性识别模型，利用知网义原知识解决句子对语义匹配中的同义词和多义性问题。在 BQ、AFQMC 和 PAWSX 等数据集上进行的实验表明，与 DSSM、MwAN 和 DRCN 等未使用预训练的模型，以及 ERNIE 等预训练模型相比，本文提出的方法取得了一定提升。通过堆叠 Transformer 层，可以有效提升语义一致性识别性能，但也会在一定程度上增加参数量。相比之下，本文提出的模型参数更少，性能更优。与同样将知网用作外部信息库的 LET 相比，本文在利用知网义原信息的同时过滤掉无关的语义信息，避免冗余义原对结果造成影响，因而更加准确直观。实验表明，利用知网义原知识，本文提出的模型能够有效提升一致性识别的准确率，

并且能够适用于 50 个字符以内的长文本场景。无论是轻量级模型还是预训练模型，均有明显提升。

未来，我们将深入研究语言表达的知识局限性，即某个句子在常识层面上不合逻辑，但在形态学和句法层面上却是正确的。例如，“太阳绕着地球运行”等。因此，我们还需要做更多工作来补充外部知识，并将继续利用常识知识，进一步提升对原始文本的语义理解。

参考文献

1. Vaswani A、Shazeer N、Parmar N 等. Attention Is All You Need. arXiv, 2017. 2. Huang P S、He X、Gao J 等. 利用点击数据学习用于网络搜索的深度结构化语义模型，ACM 第22届信息与知识管理国际会议论文集. ACM，2013. 3. Tan C、Wei F、Wang W 等. 用于句子对建模的多路注意力网络，第二十七届国际人工智能联合会议. IJCAI-18. 2018. 4. Kim S、Kang I、Kwak N. 利用密集连接循环与协同注意信息进行语义句子匹配. 2018. 5. Chen Q、hu X、Ling Z 等. 用于自然语言推理的增强型长短期记忆网络. 2016. 6. Yichen G、Heng L、Jian Z 等. 在交互空间中进行自然语言推理，ICLR 2018. 7. Huang G、Liu Z、Laurens V 等. 密集连接卷积网络，IEEE计算机学会. IEEE计算机学会，2016. 8. Qian C、Zhu X、Ling Z H 等. 利用外部知识增强的神经自然语言推理模型，计算语言学协会第56届年会论文集（第1卷：长篇论文）. 2018. 9. Lin D. 书评：《WordNet：电子词汇数据库》，计算语言学，2002. 10. Tai K S、Socher R、CD Manning. 基于树结构长短期记忆网络的改进语义表示，计算机科学，2015，5(1):: 36. 11. Raffel C、Shazeer N、Roberts A 等. 探索统一文本到文本转换器的迁移学习极限，2019. 12. Dettmers T、Lewis M、Belkada Y 等. LLM.int8：面向大规模 Transformer 的 8 位矩阵乘法. arXiv 电子预印本，2022. 13. Hanxiao Liu、Zihang Dai、David R. So 等. Pay Attention to MLPs. NeurlPS 2021. 14. Jiao Y、Jing M、Kang F. 基于融合共现距离的句法网络文本语义相似度计算，现代图书情报技术，2019，003 012 93 100( ): - . 15. Page、Lawrence、Brin 等. PageRank 引文排序：为网络带来秩序. 斯坦福数字图书馆工作论文，1998. 16. Yan H、Haili S、Ke X 等. 基于主题约束的文本生成方法. 北京大学学报（自然科学版），2020，56(1)：9-15. 17. Lin L、Hui L. 基于概念向量空间的文本相似度计算方法. 数据分析与知识发现，2018，2(5)：11. 18. Mikolov T、Chen K、Corrado G 等. 向量空间中的词表示的高效估计. 计算机科学，2013.

19. Boer Lyu、Lu Chen 等. LET：语言知识增强图 Transformer，用于中文短文本匹配. AAAI 人工智能会议论文集 . 2021. 20. Haili Sun 、 Yan Huang 、 Songfeng Lu. 改进细粒度文本情感迁移，以生成多样化评论. 载于：2020 IEEE 人工智能与计算机应用国际会议论文集（ICAICA 2020），中国大连，2020年6月27日至29日. IEEE，2020：261-266. 21. Reimers N、Gurevych I. Sentence-BERT：使用孪生 BERT 网络进行句子嵌入. 2019年自然语言处理经验方法会议暨第9届自然语言处理国际联合会议（EMNLP-IJCNLP）论文集. 2019. 22. Zhendong Dong、Qiang Dong. 知网——混合语言与知识资源. 自然语言处理与知识工程国际会议论文集，2003. 2003年，中国北京，2003年，第820-824页，doi：10.1109/NLPKE.2003.1276017.