融合对比学习的成语完形填空算法

张本文[1]，黄方怡[2]，琚生根[2]
(1. 四川民族学院理工学院，康定 626001；2. 四川大学计算机学院，成都 610065)

摘 要：成语完形填空是机器阅读理解(MRC)的一类子任务，旨在测试模型对中文文本中成语的理解和应用能力. 针对现有的成语完形填空算法忽视了成语的嵌入向量会出现表征崩溃的现象，并且模型在域外数据上的准确率低，泛化能力较差的问题，本文提出了 NeZha-CLofTN. 该算法由嵌入层、融合编码层、图注意力子网络和预测层等 4 部分组成. 其中融合编码层中利用对比学习迫使网络改变特征提取的方式，避免了网络输出恒定的嵌入向量，从而预防了表征的崩溃；预测层综合多个近义词图子网络的输出，以获得比其中单独的子网络更好的预测性能，增强模型的泛化能力. NeZha-ClofTN 在 ChID-Official 和 ChID-Competition 数据集上进行了实验验证，准确率分别达到 80.3%和 85.3%，并通过消融实验证明了各个模块的有效性.

关键词：成语完形填空；预训练语言模型；对比学习；近义词
中图分类号：TP391 **文献标识码**：A DOI：10.19907/j.0490-6756.2022.052003

An idiom cloze algorithm incorporating contrastive learning

ZHANG Ben-Wen[1], *HUANG Fang-Yi*[2], *JU Sheng-Gen*[2]
(1. School of Science and Engineering, Sichuan Minzu College, Kangding 626001, China;
2. College of Computer Science, Sichuan University, Chengdu 610005, China)

Abstract: Idiom cloze test is a subtask in Machine Reading Comprehension (MRC), which aim to test the model's ability to understand and apply idioms in Chinese text. The existing idiom cloze algorithms ignore the fact that the idiom embeddings suffer from representational collapse, which leads to low accuracy and poor generalization performance on out-of-domain data. In this paper, the authors propose the NeZha-CLofTN, which consists of four parts: embedding layer, fusion coding layer, graph attention subnetwork, and prediction layer. The fusion coding layer uses contrastive learning to force the network to change the feature extraction that avoids the network outputting a constant embedding vector, thus preventing the representational collapse. The prediction layer combines the output of multiple synonym subgraphs to obtain better prediction than a single subgraph and to enhance the generalization performance of the model. NeZha-ClofTN is used in the ChID-Official and ChID-Competition datasets with accuracy of 80.3% and 85.3%, and the effectiveness of each module was demonstrated by ablation experiments.

Keywords: Idiom cloze test; Pre-trained language model; Contrastive learning; Synonym idiom

收稿日期：2022-05-09
基金项目：国家自然科学基金重点项目(62137001)
作者简介：张本文(1978－)，男，副教授，研究方向为数据挖掘、粗糙集及代价敏感. E-mail：zhbwin@163.com
通讯作者：琚生根. E-mail：jsg@scu.edu.cn

1 引　言

成语是一种汉语形式,是汉语中最独特的部分之一,适当地使用成语可以使文章更具有意象和表现力.成语在文学作品、日常对话、科学文献中无处不在,大约有 7000 多个成语在现代汉语、日文、韩文、越南文中被广泛使用[1].成语比普通词更难学,人们很容易在错误的语境中使用成语.语言能力的形成需要长时间的练习和积累,如果很少使用成语或错误使用成语就会对成语的语义感到陌生,导致写文章时出现文字拼凑、表达不连贯等情况.成语完形填空可以帮助解决上述问题,该任务的目标是从训练样本中学习成语的上下文语境,使计算机能"理解"作者的写作意图,并推荐合适的成语为其提供参考.因此,成语完形填空是一个研究的问题.

成语完形填空相较于普通的完形填空任务难点有两个:(1) 成语既有字面含义也有隐喻,两种语义会搭配不同的语境;(2) 普通完形填空任务的正确答案往往就存在段落中,而成语字面上和上下文重叠性低,很难直接从段落抽取答案,难度更大.目前成语完形填空的主流模型可分为以下 3 类:基于 CNN 的方法[2]、基于 RNN 的方法[1,3,4]、基于以 Transformer 为基础的预训练语言模型(Transformer-based Pre-trained Language Mode-ls, T-PTLM)的方法[5-8].基于 CNN 的方法是最初的成语完形填空算法,利用 CNN 分别对段落和成语进行编码.但 CNN 通常擅长捕获局部信息,在语言中,相关的词语并不需要相邻,这使得 CNN 在语言处理方面并不是非常有优势.基于 RNN 的方法能弥补 CNN 的不足,研究者常选择双向 RNN 结合注意力机制来对段落与候选成语进行编码.虽然以 RNN 为基础的模型已经能表现出比较好的性能,但他们的缺点是不能并行化处理序列,无法捕获长程依赖,并且也没有实现成语与段落多维度、多步骤地匹配.基于 T-PTLM 的方法以 Transformer 为基础,实现成语与段落多维度、多步骤地匹配,并利用预训练从大量文本数据中学习了通用的语言表示,将这些背景知识转移到成语完形填空任务中.基于 T-PTLM 的模型已经在成语完形填空任务中取得了不错的效果,但仍然存在一些问题:(1) 不同位置的填空应该填入不同的成语,位置信息发挥了关键作用,需要好的位置嵌入方法;(2) T-PTLM 提取的成语的嵌入向量中错误成语与正确成语的向量之间具有很高的相似性,尤其是近义词成语.并且模型在域外数据上的准确率低,泛化能力较差.

本文在详细了解成语完形填空研究的基础上提出了融合对比学习的成语完形填空算法(NeZha-CLofTN).该算法可以防止成语嵌入的表征崩溃,避免恒定的输出,增强泛化能力,使模型在测试阶段的准确率得到显著提高,而模型的训练效率几乎不受影响.该算法引入对比学习,使成语的嵌入向量具有高对齐性或者高均匀性;实现了同时编码绝对位置和相对位置,且具有单调性、平移不变性、对称性和递推性;利用近义词图注意力网络的对比学习,缓解了成语字面意思和实际含义不一致现象,提升了算法在域外数据上的准确率.本文通过对比实验、消融实验以及可视化分析证明了算法的有效性和可解释性.此外,经过该算法提取的成语嵌入向量更加贴近成语的真实语义,可能促进成语其他的中文语言处理的研究.

本文的主要贡献可归纳如下:(1) 同时编码绝对位置和相对位置信息,实现了位置嵌入的单调性、平移不变性、对称性和递推性;(2) 对比学习让填空和正确成语尽可能靠近,和 Batch 内的其他嵌入尽可能远离,使成语的嵌入向量具有高对齐性或者高均匀性;(3) 通过训练两个近义词子网络输出的向量和概率分布尽可能接近,使模型的泛化能力更强.

2 相关工作

目前在成语完形填空领域的方法主要分为:基于 CNN 的方法,基于 RNN 的方法以及基于 T-PTLM 的方法.

(1) 基于 CNN 的模型. 2018 年 Liu 等[2]用 CNN 分别对段落和成语进行建模,段落提取全局向量和局部向量,拼接之后得到最终的上下文嵌入向量和成语的嵌入向量 ,利用余弦相似度选出最佳成语.但文献[2]并没有阐述选用 CNN 的原因.CNN 模型的优点是能够并行地训练,比 RNN 快;缺点是只能提取局部信息,不能处理长序列.

(2) 基于 RNN 的模型. 2018 年 Jiang 等[1]利用成语的定义作为成语的背景知识,借助双向长短时记忆网络(Bi-directional Long Short-Term Memory,Bi-LSTM)分别对段落和成语的解释进行编码,利用软注意力(Soft Attention)为段落中的每个词分配权重,并预测每个候选成语的匹配分数.但是缺点也非常明显,必须对成语的解释文本

进行编码，没有提供成语解释的成语并不能很好的提取语义信息. 2019 年 Liu 等[3]将成语看作文言文书写的，巧妙地将段落→成语的问题转换为现代文→文言文的翻译问题，借助文本生成的 Seq2Seq (Bi-LSTM Encoder-Decoder)框架进行解决，生成的目标文本再和候选成语计算编辑距离. 但是缺点在于文本生成本身存在暴露偏差(Exposure Bias)的问题，即预测模式下，生成过程中仅依靠先前预测的词语没有目标语言的“指导”(Teacher Forcing)，这会导致模型在预测模式下表现不佳. 另外利用文本生成的框架生成成语时，没有考虑成语所在的位置，同一段段落生成的成语总是相同的，但实际上不同位置的填空应该需要不同的成语. 2019 年 Zheng 等[4]发布了一个大规模成语完形填空任务的数据集 ChID(Chinese IDiom Dataset)，并测试了 Language Model(LM)、Attentive Reader(AR)、Stanford Attentive Reader(SAR)这 3 个模型在数据集上的表现.

(3) 基于 T-PTLM 的模型. 2020 年 Tan 等[5]提出了一个基于 BERT 的双嵌入模型，候选成语会有两个独立的嵌入向量，其中一个与填空处的隐藏表示进行匹配，另一个与段落中的所有词语进行匹配. 文献[5]认为局部的填空的嵌入向量能够捕捉成语的语法属性，而全局的整个段落的嵌入向量能够捕捉其主题含义. 2020 年 Tan 等[5]将每个候选成语与段落拼接起来，构建成单个序列输入 BERT，利用下一句预测(Next Sentemce Prediction，NSP)的方式提取成语和段落的关系向量，但他所需要的计算开销太大，训练时间和预测时间过长. 2020 年 Long 等[6]首先用实验证明了许多成语的字面意义与其语义相去甚远，而近义词可以缓解这种不一致. 41.5%的成语在近义词的帮助下更容易理解. 由此，文献[6]提出(Synonym Knowledge Enhanced Reader，SKER)，首先构造近义词图，再利用图注意力(Graph Attention Network，GAT)为不同的近义词分配不同权重，将近义词信息融入到相应的核心成语中. 构建好的近义词图也可以用于促进其他成语任务. 我们认为 SKER 的缺点在于没有考虑到候选成语集中本身也存在近义词，可能会有两个选项在同一近义词子图内，这样计算图注意力后，一个选项可能包含另一选项的语义信息，可能反而会增加问题难度. 2020 年 Wang 等[7]提出了一种有效整合成语各种外部信息的模型. 包括成语的段落、成语的解释以及成语的各种词嵌入，并提出属性注意力(Attribute Attention)来平衡不同信息的权重，添加了通过从整个成语词汇表中选择答案的额外损失. 但我们认为性能的提升在于增加更多的信息，如果没有增加更多的其他信息的成语便不能有比较好的表现. 2021 年徐家伟等[8]提出了增强型全局注意力机制(Global Enhanced Attention，GEA)，文献[8]在考虑候选成语和填空(局部)以及段落(全局)的相关度的同时，还利用选项的注意力因子调控对段落中的关注度，寻找不同候选成语之间的共性.

以上方法获取的词嵌入可能出现表征崩溃的现象，候选成语都被映射到空间中的一个小区域，彼此之间具有极高的相似性. 导致 T-PTLM 原本的可泛化性在微调阶段大幅减弱[9]，输出的成语词嵌入偏离成语的实际语义，近义词的干扰影响也愈加严重. 此外，现有模型在域外数据上泛化能力较差也是一个不可忽视的问题. 域外数据中序列更长、低频词更多及干扰成语更多的特点使模型在域外数据的准确率不高. 如何利用候选成语之间的关系更好地表征成语语义，增强模型在域外数据上的泛化能力，缓解模型在域内数据上的过拟合也是一个值得探究的点.

3　本文方法

3.1　任务描述

成语完形填空任务是给定一段段落，段落中间有一个空格，模型需要选择一个成语进行填空，每个填空有对应的成语候选集，其中包括正确答案、近义词成语和随机设置的成语. 我们定义给出三元组(P,Q,A)构成的训练示例，其中 P 表示段落，Q 表示填空，A 表示正确答案.

目标：学习预测器 f，将从候选成语集合中推断出成语填入填空.

输入：P 表示一段中文段落，其中成语被替换成空格，段落的总长度为 m；Q 表示段落 P 中的空格(需要填入成语的位置)；C 表示 Q 对应的候选成语集合；T 为集合中成语的个数. 其中 C_t 表示第 t 候选成语，候选成语的长度为 n.

输出：通过对每一项候选成语打分，概率最高的成语为最终答案.

一段文本段落 P 对应一个空格 Q 以及 T 个候选成语. 成语完形填空的实例如图 1 所示.

Passage: 最新研究显示，同样是为时2小时的健身运动，将其分为40分钟做一次，共做3次，所消耗的脂肪几乎是分为60分钟做一次，共做2次7倍。由于每次运动过后，其体内可能维持最高__速率至少12小时，体内囤积的脂肪也会在此时被迅速消耗。所以尽可能分段运动，减肥效果也会更好。

Candidates:

○ 天旋地转	○ 不省人事
○ 青黄不接	● 新陈代谢
○ 心力衰竭	○ 永垂不朽
○ 改天换地	○ 与世长辞
○ 一念之差	○ 不省人事

图 1　成语完形填空示例

Fig. 1　Example of the idiom cloze test

3.2　模型简介

图 2 是 NeZha-CLofTN 的整体框架图. 图 2 主要由 4 部分组成:(1) 嵌入层:NEZHA 将段落 P 和候选成语 C 中的字符分别转换为 d 维的嵌入向量表示;(2) 融合编码层:截取填空处,通过自注意力机制和余弦相似度捕获填空和候选成语之间的相关性;(3) 近义词图注意力子网络:利用图注意网络和门控机制提取带有近义词信息的成语词嵌入;(4) 预测层:计算候选成语的概率分布,输出概率最大的选项. 其中 $loss^1_{CL}$ 是计算填空和候选成语两个嵌入向量之间的差距;$loss^2_{CL}$ 是计算两个子网络的输出之间的差距,每个子网络的近义词被赋予不同权重,组合形成新的成语词嵌入,并补充到预测层中,虚线圆圈表示按一定概率被随机舍弃的神经元.

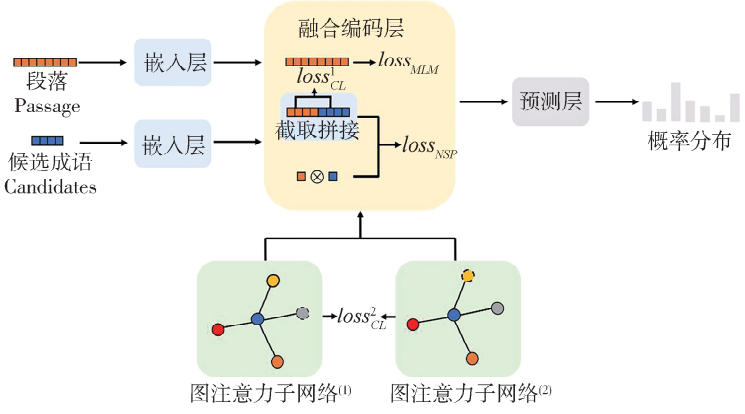

图 2　NeZha-CLofTN 模型整体框架图

Fig. 2　The framework of NeZha-CLofTN

3.3　嵌入层

首先,P 和 C_i 的头部和尾部分别加上特殊字符[CLS]和[SEP],再利用 WordPiece[10] 对原始序列进行分词,用中文词表将其转换为词表中 ID,然后,每个字符进行字符嵌入和位置嵌入,二者进行相加作为总的嵌入向量. 字符嵌入的嵌入矩阵都是随机初始化,将字符在词表中的 ID 转换为向量. 位置嵌入是参考 FLOATER[11](Flow-based Transformer),在 NEZHA[12] 三角函数式相对位置嵌入的基础上融入递推关系,如式(1)和式(2).

$$\widetilde{K}_i = E^C_i \odot W^K + \widetilde{E}^K_{ij} \tag{1}$$

$$K_i = \widetilde{K}_i + E^P_i = \widetilde{K}_i + p(0) + \int_0^{i\cdot\Delta t} h(\eta, p(\eta); W_h)\,\mathrm{d}\tau \tag{2}$$

其中,$\widetilde{K}_i$ 是 NEZHA 原始的 Key 的计算过程;$\widetilde{E}^K_{ij}$ 表示原始的位置 i 到位置 j 的相对位置嵌入. K_i 是新的 Key,只需 $\widetilde{K}$ 加上 i 的位置嵌入 E^P_i,即可实现递推性. 新的位置嵌入方法,是一种绝对位置嵌入和相对位置嵌入结合的方法,既可以编码相对位置差,拥有平移不变性;也可以保留序列从前往后的顺序性,拥有递推性. Value 和 Query 的计算方法和 Key 计算方法相同. 式(2)的求解是利用了常微分方程求解器[13],其采用了伴随方法求解梯度. 只需给定初始状态和常微分方程,可以找到任意时刻的位置嵌入.

WordPiece 的字符嵌入是学习与段落无关的表示,而 NEZHA 的嵌入是学习与段落相关的表示,两者相互补充. 先利用 WordPiece 将他们投影到维度大小为 e 的嵌入空间,然后利用 NEZHA 将其映射到维度大小为 d 的隐藏空间. NEZHA 由 l 层 Transformer[14] 编码器块组成,每层隐藏层的大小为 d.

段落和候选成语通过相同的嵌入层(共享参数),映射到相同的向量空间,然后融合编码层计算二者的距离,使它们在向量空间中的距离近似于现

实中的语义相似度.

3.4　融合编码层

融合编码层承接嵌入层，嵌入层的输出作为融合编码层的输入. 段落为$P'=[P'_i]_{i=1}^{m}$，$P'_i\in\mathbb{R}^d$，候选成语为$C'_t=[C'_{tj}]_{j=1}^{n}$，$C'_{tj}\in\mathbb{R}^d$，段落中的填空为$Q'=[Q'_b]_{b=1}^{m_b}$，$Q'_b\in\mathbb{R}^d$，m_b 表示填空的长度. C'_t 中 j 位置的向量($[C'_{tj}]_{j=1,j\neq[CLS]}^{n}\in\mathbb{R}^{(n-1)\times d}$)组合成语构成字的组合语义.

首先是遮蔽语言模型(Masked Language Model，MLM)任务，段落中成语每个字符都被替换为[MASK]，因此需要并行预测段落中的多个遮蔽字符.

$$loss_{MLM}=-\sum_{i}^{m}m_i\left(\sum_{u}^{v}y_{iu}\log(\mathrm{Softmax}(W_{MLM}P'_i+b_{MLM}))\right)\tag{3}$$

式(3)中，$P'_i\in\mathbb{R}^d$ 为段落中 i 位置字符的嵌入层输出；W_{MLM} 和 b_{MLM} 是线性映射的参数，通过 Softmax 计算 P'_i 基于词表的概率分布；m 为段落的长度，$m_i\in\{1,0\}$ 表示 i 位置字符是否为[MASK]；v 为词表的大小；u 为词表中第 u 词；$y_{iu}\in\{0,1\}$ 表示字符 P_i 是否为 u. 甚至不需要额外的提示工程，完形填空任务与预训练阶段的 MLM 任务目标一致，通过序列中其余字符预测遮蔽字符.

其次是 NSP(Next Sentemce Prediction)，候选成语 C'_t 依次与填空 Q' 拼接，拼接后输入 NEZHA，预测二者的相关性. 融合编码层的 NEZHA 和嵌入层的 NEZHA 共享参数，目的是在同一向量空间获取序列之间的交互关系.

$$\xi(C''_{[CLS]})=\tanh(W_{NSP}C''_{[CLS]}+b_{NSP})\tag{4}$$

$$P_1(C'_t,Q')=\mathrm{Softmax}(W_0\xi(C''_{[CLS]})+b_0)\tag{5}$$

式(4)中，拼接后序列的头部特殊字符[CLS]在 NEZHA 最后一层隐藏层的向量 $C''_{[CLS]}$，通过前馈神经网络层 $\xi(\cdot)$，激活函数为 tanh 函数，W_{NSP} 和 b_{NSP} 是学习的参数. 式(5)中，$\xi(C''_{[CLS]})$ 通过一层线性映射，W_0 和 b_0 是线性映射的参数，利用 Softmax 函数计算每个候选成语填入空格的概率，C'_t 和 Q' 分别是候选成语和填空的嵌入层输出，$C''_{t[CLS]}$ 是成语和填空拼接后的[CLS]输出.

上述是从字符级别考虑填空的语义，缺乏从整体考虑其语义. 为此，本文将字符嵌入压缩成句向量(虽然不是完整的句子，但是习惯上称序列的向量为句向量)，从而实现字符级别和序列级别的多层次匹配. 本文尝试了最大值、平均值和加权平均值等多种压缩方法 $\rho(X)$，其中最大值的准确率最高，$\rho(X)$要求输入序列矩阵 X. 填空经过 $\rho(X)$得到的句向量$q'_b\in\mathbb{R}^d$，候选成语的[CLS]的嵌入层输出$C'_{t[CLS]}\in\mathbb{R}^d$，分别作为各自序列的整体语义表示. 文本采用先计算哈达玛积然后线性映射的方法衡量向量(q'_b 和$C'_{t[CLS]}$)之间的相关性如式(6).

$$P_2(C'_t,Q')=\mathrm{Softmax}(W_1(C'_{t[CLS]}\otimes q'_b)+b_1)\tag{6}$$

此外，本文将 $P'_{[CLS]}$ 与 q'_b 组合，利用融合函数 f_{mix}形成新的特征表示 $q\in\mathbb{R}^d$，如式(7).

$$q=W_2\begin{bmatrix}P'_{[CLS]}\\q'_b\\P'_{[CLS]}\otimes q'_b\\P'_{[CLS]}+q'_b\end{bmatrix}\tag{7}$$

其中，$P'_{[CLS]}\in\mathbb{R}^d$ 和$C'_{t[CLS]}\in\mathbb{R}^d$是特殊字符[CLS]在 NEZHA 最后一层的隐藏层向量表示，代表整个序列的信息；式中⊗是哈达玛积；W_1，W_2，b_1，$\in\mathbb{R}^{d\times 4d}$，是学习的参数. 候选成语采用同样的计算方法，得到候选成语新的向量表示 c_t.

最后，我们利用对比学习比较段落和候选成语组成的序列对来学习. 一个 Batch 样本数 K，$D=\{q_k,v_k^+,v_{k,1}^-,\ldots,v_{k,u}^-,\ldots,v_{k,U}^-\}_{k=1}^{K}$，每个样本由一个段落$q_k\in\mathbb{R}^d$、一个恰当的成语$v_k^+\in\mathbb{R}^d$(正样本)、$U$ 个不相关的向量$v_{k,u}^-\in\mathbb{R}^d$(负样本)组成. 本文的损失函数为 $L(q_k,v_k^+,v_{k,1}^-,\ldots,v_{k,u}^-,\ldots,v_{k,U}^-)$，为了阐述方便，简写为 L_k，如式(8).

$$L_k=-\mathrm{sim}(q_k,v_k^+)/\tau+\log\sum_{u=1}^{U}\exp(\mathrm{sim}(q_k,v_{k,u}^-)/\tau)\tag{8}$$

$$loss_{CL}^{1}=\frac{1}{K}\sum_{k=1}^{K}L_k\tag{9}$$

式(8)和式(9)中的 L_k 表示第 k 样本的损失函数，基于三重态网络的对比学习损失函数为式(9). 正确的成语是唯一正样本，剩下 Batch 内的$(K-1)$个段落和$(T\cdot K-1)$个候选成语全都为负样本，总数为$U=(T+1)\cdot K-2$，其中 T 表示段落对应的候选成语集合大小，减 2 是因为减去段落本身 q_k 以及正样本 c_k^+，$\tau=0.05$ 表示温度系数. 可以通过调整 Batch Size 改变负样本的数量.

3.5　近义词图注意力子网络

本文近义词图注意力子网络模块参照文献[6]构建. 步骤如下：(1) 构建高质量的成语词表. 本文参考了新华字典数据库、百度中文的成语词表，利用腾讯预训练的中文词汇的词嵌入语料库，抽取语

料库中成语的词嵌入,得到最终的成语的词嵌入表,共 22 648 个成语,词嵌入的维度为 200 维. 每一个成语都作为整体拥有一个词嵌入. (2) 近义词关系的定义. 一方面,利用上述字典中成语注释的近义词;另一方面,本文认为词语之间的余弦相似度超过一定的阈值即可视为近义词[15],阈值设置为 0.65;本文中我们将成语近义词个数的最大阈值设为 7. (3) 构建图. 每一个成语都有一个近义词无向图 $G=\langle V,E\rangle$,以核心成语为中心顶点,通过无向边连接着他所有的近义词,其中 V 表示所有成语组成的顶点集,E 表示近义词关系的边集.

门控的图注意力机制. 成语利用嵌入矩阵加载腾讯预训练的静态词嵌入,成语的初始词嵌入表示为 $\dot{s}$,$\dot{s}\in\mathbb{R}^{\hat{d}}$,$\hat{d}=200$. 由于核心成语与其近义词在语义上仍有细微差异,为避免引入不相关的噪声,本文仅保留与段落相关的语义,如式(10).

$$s'=(W_3\ \dot{s}\ )\otimes q \tag{10}$$

其中,W_3 是学习的参数,$W_3\in\mathbb{R}^{d\times\hat{d}}$;$\otimes$是两个向量的哈达玛积;$s'\in\mathbb{R}^d$即已经融入了段落信息的成语词嵌入;$q\in\mathbb{R}^d$表示填空的嵌入向量,包含了填空段落的段落信息,如式(11).

$$s_t=s'_t+ReLU(W_5(ReLU(W_4s'_t+b_4))+b_5) \tag{11}$$

式中,$s'_t\in\mathbb{R}^d$经过两层前馈神经网络和残差网络得到 s_t,$s_t\neq c_t$,s_t'是从静态词嵌入转换得到,s_t 是三重态网络的嵌入层输出转换得到;W_4 和 $W_5\in\mathbb{R}^{d\times d}$;$b_4$ 和 $b_5\in\mathbb{R}^d$是学习参数. s_t 及其近义词为 $S=\{s_t,s_{t,1},s_{t,2},\ldots,s_{t,t_n},\ldots,s_{t,t_N}\}$,$S\in\mathbb{R}^{d\times(t_N+1)}$,$t_N$ 表示近义词的个数,s_{t,t_n}表示 s_t 的第 t_n 近义词. 即使同为近义词,近义词与近义词之间的语义差距也可能非常大,可能对应核心成语的不同语义. 成语在不同语言环境下和近义词的相关程度不同,计算 s_t 和每一个近义词 s_{t,t_n}的相关性,如式(12)和式(13).

$$e_{t,t_n}=((W_6s_t)^TW_7s_{t,t_n})/\sqrt{d} \tag{12}$$

$$\alpha_{t,t_n}=\mathrm{Softmax}(e_{t,t_n})=\frac{\exp(e_{t,t_n})}{\sum_{t_k=1}^{t_N}\exp(e_{t,t_k})} \tag{13}$$

式(12)中,e_{t,t_n}表示近义词 s_{t,t_n}对核心词 s_t 的重要性;W_6 和 W_7 为学习的参数,使用 Softmax 函数对所有近义词的重要性进行归一化.

$$\tilde{s}_t=\alpha_{t,t_n}(W_8s_{t,t_n})^T \tag{14}$$

$$g=\mathrm{Sigmoid}(W_9\tilde{s}_t+b_9) \tag{15}$$

$$\hat{s}_t=g\cdot\tilde{s}_t+(1-g)\cdot s_t \tag{16}$$

实现了在降低潜在噪声的同时,有效利用近义词图. $\hat{s}_t\in\mathbb{R}^d$为融入了近义词信息的候选成语的整体词嵌入,每个近义词的贡献程度取决于他与核心成语的相关程度,W_8、W_9 和 b_9 为学习的参数.

3.6　预测层

本文受到 R-Drop[16] 和 SimCSE[17] 的启发,每一个候选成语的近义词图都经过两次门控图注意力子网络,输出 $\hat{s}_t^{(h)}$,h 表示是子网络编号. Dropout 使两个子网络丢弃的神经元不同,即使是输入相同的近义词图,两个子网络的输出也完全不同,$\hat{s}_t^{(1)}\neq\hat{s}_t^{(2)}$. 每个神经元被丢弃的概率为 p_{drop},与其他神经元被丢弃与否无关,本文最佳概率 $p_{\mathrm{drop}}=0.2$.

$$P_3^{(1)}(s'_t,Q')=\psi^{(1)}(\hat{s}_t^{(1)})=\mathrm{Softmax}(W_{10}\hat{s}_t^{(1)}+b_{2=10}) \tag{17}$$

$$P_3^{(2)}(s'_t,Q')=\psi^{(2)}(\hat{s}_t^{(2)})=\mathrm{Softmax}(W_{11}\hat{s}_t^{(2)}+b_{11}) \tag{18}$$

式(17)和式(18)中 W_{10}和$W_{11}\in\mathbb{R}^{1\times d}$,$b_{10}$,$b_{11}\in\mathbb{R}^1$,先将成语词嵌入映射为一个常数,然后经过 Softmax 函数,得到带近义词语义的第 t 候选成语填入填空的概率. Batch 内第 k 样本的第 t 候选成语填入的概率为 $P_{3,kt}^{(h)}(s'_{kt},Q'_k)$,缩写为 $P_{3,kt}^{(h)}$. 在训练过程中,本文希望最小化两个子网络输出的差距,包括两个概率分布间的 KL 散度和两个成语词嵌入之间的余弦相似度. 第 k 样本的基于 KL 散度的损失函数如式(19).

$$L_k^{kl}=\frac{1}{2T}\sum_{t=1}^{T}\begin{pmatrix}D_{KL}(P_{3,kt}^{(1)}\,||\,P_{3,kt}^{(2)})\\+D_{KL}(P_{3,kt}^{(2)}\,||\,P_{3,kt}^{(1)})\end{pmatrix} \tag{19}$$

式(19)中,$D_{KL}(P_{3,kt}^{(1)}||P_{3,kt}^{(2)})$表示基于子网络 2 的分布来拟合子网络 1 的分布所需的额外信息量. $D_{KL}(P_{3,kt}^{(2)}||P_{3,kt}^{(1)})$表示子网络 1 来拟合子网络 2 所需的额外信息量. 因为 KL 散度是非对称的,所以同时计算两个方向的拟合,取两个 KL 散度的平均数.

另外将两个子网络输出$\hat{s}_{kt}^{(1)}\in\mathbb{R}^d$和$\hat{s}_{kt}^{(2)}\in\mathbb{R}^d$作为正样本,将小批处理中的另外 $2(K\cdot T-1)$数据点作为负样本. 让模型学习区分正样本和负样本,最大化子网络的输出向量之间的一致性,如式(20). 用余弦相似度衡量向量之间的相关性,$\tau=0.05$ 表示温度系数.

$$L_k^d = -\frac{1}{T}\cdot \sum_{t=1}^{T}\log\left(\frac{\exp(\mathrm{sim}(\hat{s}_{kt}^{(1)},\hat{s}_{kt}^{(2)})/\tau)}{\sum_{j\cdot(j\neq kt,h\in\{1,2\})}^{K\cdot T}\exp(\mathrm{sim}(\hat{s}_{kt}^{(1)},\hat{s}_{j}^{(h)})/\tau)}\right) \tag{20}$$

缩小两个子网络输出的差距，既包括预测的概率分布之间的距离，也包括成语词嵌入在语义上的相似度. 因此，将这两个损失之和作为基于近义词图的总对比学习损失，如式(21).

$$loss_{CL}^2 = \frac{1}{K}\sum_{k=1}^{K}(L_k^{kl} + L_k^{cl}) \tag{21}$$

融合编码层得到的概率加上子网络预测的平均概率，得到最终的概率分布，如式(22)，其中 $\zeta\in\mathbb{R}^T$ 训练过程中可学习，初始化时每一维度上的值均为 2.0，为了简单，将 $P_1(C_t',Q')$ 和 $P_2(C_t',Q')$ 分别简写为 P_1 和 P_2. 总的损失函数为式(23)，α，β，γ，δ 都为可学习参数，初始值分别为 1.0、1.0、0.5、0.3，维度为 1 维.

$$P_{总} = P_1 + P_2 + \frac{\zeta}{2}\cdot(P_3^{(1)} + P_3^{(2)}),P_{总}\in\mathbb{R}^T \tag{22}$$

$$loss_{总} = \alpha\cdot loss_{MLM} + \beta\cdot loss_{NSP} + \gamma\cdot loss_{CL}^1 + \delta\cdot loss_{CL}^2 \tag{23}$$

4　实　验

4.1　数据集及评估函数

本文是在公开的大规模成语完形填空数据集 ChID[4] 上进行的实验. 这个数据集的数据来源是互联网的小说、散文及新闻文本，抽取其中包含成语的语句. ChID 属于完形填空式多项选择类的机器阅读理解数据集，包含了约 58 万个段落，段落中的成语都被空格代替，共约 73 万个空格，每个空格都搭配一个有限的、多项的候选成语集合，要求模型从这个集合中选择最好的答案填入空格. 和其他完形填空数据集不同的是 ChID 数据集的正确答案通常不会在段落中出现，所以难度更大. ChID 又分为 ChID-Official 和 ChID-Competition 两个版本. 第一个版本是 ChID-Official：含一个训练集(Train)、一个验证集(Dev)、四个测评集(Test、Ran、Sim、Out). 训练集段落个数为 520 711. 验证集段落为 20 000；测试集段落个数为 20 000；随机集段落个数为 20 000，随机集的候选成语都是从随机抽取的；近义词集的段落个数为 20 000，近义词集的候选成语是从与正确答案最相似的前 10 个成语中抽选的；域外集段落个数为 20 096，域外集的候选成语都是从域外数据中抽取出来的. 第二个版本是 ChID-Competition：成语完形填空的竞赛数据，是 ChID-Official 的修改版本. 多个段落分为一组，同组的段落会共享一组候选成语集，每个候选成语只能被选择一次. 模型需要区分具相似语义的段落所需成语的差异，以做出正确的判断. ChID-Competition 数据集分为训练集、验证集、测试集、域外集，分组个数分别为 84709、3218、3231、3754. 本文使用准确率作为本文评价指标.

$$准确率 = \frac{预测正确的样本数}{样本总数}\times 100\% \tag{24}$$

4.2　参数设置

本文的实验环境为：CPU 为 AMD Ryzen 5 3500X 6-Core Processor；内存大小为 32 G；GPU 为 1 个块 GeForce GTX 1080 Ti LIGHTNING Z，显存大小为 11 G；操作系统为 Ubuntu 18.04.6；开发环境为专业版 PyCharm Professional；环境管理器为 Anaconda 4.10.3；语言 Python 3.8.12；深度学习框架 PyTorch 1.8.0、预训练语言模型框架 HuggingFace Transformers 4.12.5.

选用基于中文语料库预训练语言模型，下面前三个贴出模型权重的下载地址：NeZha：https://github.com/lonePatient/NeZha_Chinese_PyTorch 中的 nezha-base-wwm. 预训练语言模型只要没有指明 Base 或 Large 的都是默认 Base 版本. 实验的超参数：隐藏层维度为 768，预训练语言模型的 Dropout 均为 0.1，学习率为 7e-5，Batch Size 为 64，3 个 Epochs，优化器为 BertAdam(学习率调整方法为 Warmup Linear，Warmup 比例为 6%).

4.3　实验结果

表 1 是 ChID-Official 数据集上的实验结果. 表 1 中，Human 和基线模型 EAR、SKER、EAR-RoBERTa、BERT-BL 是取参考文献中值，其中“—”表示论文中无对应值，Human 为 ChID-Official 数据集创建时采用人工评估的方式，从测试集中抽取了 200 个段落：Test、Ran、Sim 和 Out，聘请 3 名注释者来完成 800 个完形填空测试，注释者都是汉语非常好的大学一年级或二年级的学生. Base 版 NeZha-CLofTN 和 Large 版 NeZha-CLofTN 只是 NeZha 的大小不同. 前者的隐藏层大小为 768，Transformer 层数为 12 层，注意力头 12 个，参数量为 132 M；后者的隐藏层大小为 1024，层数 24 层，注意力头 16 个，参数量 367 M.

表 1　ChID-Official 数据集实验结果

Tab. 1　Experimental results on ChID-Official dataset

模型	准确率/%					加速比
	验证集	测试集	随机集	近义词集	域外集	
Human[4]	—	87.1	97.6	82.2	86.2	—
EAR[7]	74.6	74.5	84.4	67.9	65.5	—
SKER[6]	76.0	76.1	87.0	68.6	68.3	—
EAR-RoBERTa[7]	78.7	79.2	90.5	71.7	72.3	—
BERT-BL[5]	79.3	79.4	88.8	72.9	73.1	0.8×
Base 版 NeZha-CLofTN	81.0	81.4	90.2	74.8	73.9	1.4×
Large 版 NeZha-CLofTN	81.5	82.0	90.2	75.6	76.6	0.5×

表 1 中,近义词集与域外集的正确进行成语完形填空的难度较大,各种模型在这两个测试集上与人类准确率的差距相较于其他数据集要大得多. 究其原因,近义词集的候选成语均为与正确答案语义极为相近的成语,模型很难区分它们的差别,特别是当成语的嵌入向量都崩溃成一个常数项,那么就更难区分近义词的差别. 域外集的数据来源是散文,其段落更长,成语更多,低频成语出现的更多,因此域外集的正确进行成语完形填空的难度更大也更能评估模型的泛化能力. Large 版 NeZha-CLofTN 相较于 BERT-BL 在近义集(+3.7%)和域外集(+4.8%)的提升大于其他三个集合(验证集+2.8%,测试集+3.3%,随机集+1.6%). Large 版 NeZha-CLofTN 的参数量几乎是 Base 版的 3 倍,训练时间增加了近两倍,但准确率仅提升了 1.1%. 显然 Base 版平衡了性能和成本,是更值得研究的模型,后续的实验中的 NeZha-CLofTN 都指 Base 版.

表 2　ChID-Competition 数据集实验结果

Tab. 2　Experimental results of ChID-Competition dataset

模型	准确率/%			加速比
	验证集	测试集	域外集	
BERT-GEA[8]	77.3	75.8	70.3	—
BERT-BL	82.2	83.0	—	0.8×
LIMIT-BERT-char[18]	83.8	83.2	—	—
RoBERTa-BL	83.8	83.6	—	0.8×
NeZha-CLofTN	87.0	86.8	82.2	1.1×

表 2 是我们在 ChID-Competition 数据集上的实验. 与 ChID-Official 数据集不同,ChID-Competition 数据集中只有 3 个用于测评的子集. BERT-GEA 和 LIMIT-BERT-char 模型准确率取自参考文献值,BERT-BL 和 RoBERTa-BL 取自代码库 https://github.com/ewrfcas/bert_cn_finetune 中的值.

从表 2 可以得到与表 1 相似的结论,NeZha-ClofTN 是训练速度差不多的几种模型中准确率最高的,能够在比较少的计算时间下,取得比较好的效果.

表 3 是在 ChID 两个数据集上做的消融实验. 为了更简洁和直观地展示网络的每个部分是否发挥作用,本文直接比较了两个数据集上的平均值. 为便于描述,表 3 中的序号与正文中的模型序号一一对应.

表 3　消融实验结果

Tab. 3　Results of ablation experiments

序号	模型	准确率/%	
		ChID-Official 平均值	ChID-Competition 平均值
(1)	嵌入层	67.7	71.9
(2)	+融合编码层	79.5	83.4
(3)	+近义词图注意力子网络	79.9	84.9
(4)	+预测层的 $loss_{CL}$	80.3	85.3

表 3 中,模型(1)只保留三重态网络的嵌入层(孪生 NeZha). 首先用两个结构和参数相同的嵌入层分别提取段落和候选成语的嵌入向量,然后计算两个向量之间的余弦相似度,进而选择最恰当的成语. Reimers 等[19]发现孪生 BERT 中余弦相似度、欧几里得距离、曼哈顿距离几种不同距离度量方法几乎没有差异,因此本文直接选择余弦相似度. 模型(2)在模型(1)的基础上增加融合编码层,和孪生

NeZha 相比，模型(2)的参数量增加了 4.8%，训练时间延长了 40%，准确率提高了 17.4% 和 15.9%. 模型(3)在模型(2)的基础上增加了近义词的门控图注意力网络，准确率提升了 0.3% 和 1.8%，说明成语的近义词关系能促进模型对成语语义的理解，模型借助近义词的语义提取特征. 模型(4) NeZha-CLofTN 比模型(3)，准确率分别提升了 0.5%和 0.5%，其中在 ChID-Competition 域外集上准确率提升了 1.2%. 域外集就是用来衡量模型的泛化能力的，由此说明增强了模型的泛化能力. 综合以上的分析，NeZha-CLofTN 的各个部分都发挥了不同的作用，相互补充，促进模型选出正确的成语.

4.4　准确率实验分析

我们对比学习损失函数的对齐性和均匀性两大特性. 本文在 ChID-Official 数据集上测试了 45 次实验，不同实验之间不同的是温度系数 τ、Epoch 次数、学习率、Batch Size 等，探究对齐性和均匀性两个性质与准确率之间的相关性. 对齐性和均匀性评价指标(Align 和 Uniform)的定义出自 Wang 等[7]的论文，Align 和 Uniform 的值都是越小越好，小 Align 对应高对齐性，小 Uniform 对应高均匀性.

经过多次实验，结果如图 3 所示，横坐标表示 Align，纵坐标表示 Uniform，每一个点表示一次实验结果，散点颜色的深浅表示准确率的高低，越深的颜色代表越高的准确率.

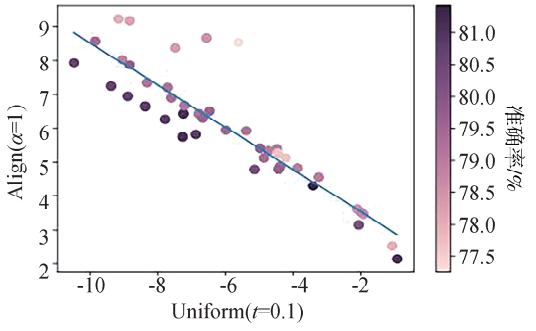

图 3　准确率的散点图

Fig. 3　Scatterplot of accuracy

本文发现，在准确率相差不大的情况下，Align 和 Uniform 呈现反相关，可以粗略地拟合为一条直线. 分析原因：Align 是衡量填空和正确成语(正样本)之间的距离，Uniform 衡量的是填空与 Batch 内其他嵌入(负样本)之间的距离. 当填空的嵌入与正确成语的嵌入之间的距离非常接近时，实际上与其他嵌入的距离也会非常接近，这对应了崩溃现象，实验点都聚集在图 3 中的右下角，此时 Align 小 Uniform 大. 当填空的嵌入与正确成语的嵌入之间的距离非常远时，嵌入向量的分布地越均匀，越不会崩溃，实验点聚集在图 2 中的左上角，此时 Align 大 Uniform 小. 这两种情况下，模型都可以取得较高的准确率，但是嵌入向量还不能同时拥有高对齐性和高均匀性. 这样很难判断模型是否真正地理解了段落或成语的语义，模型可能只是简单地区分正确嵌入和其他嵌入之间的差别.

5　结　论

本文提出了 NeZha-CLofTN 模型成语完形填空，该算法可以防止成语嵌入的表征崩溃，避免恒定的输出，增强泛化能力，使模型在测试阶段的准确率得到了显著的提高，而模型的训练效率几乎不受影响. 该算法引入对比学习，并利用去耦合与调整温度系数优化对比学习，使成语的嵌入向量具有高对齐性或者高均匀性；利用近义词图注意力网络的对比学习，缓解了成语字面意思和实际含义不一致现象，提升了算法在域外数据上的准确率. 本文通过对比实验、消融实验以及可视化分析证明了算法的有效性和可行性. 此外，经过该算法提取的成语嵌入向量更加贴近成语的真实语义，可能促进成语其他的中文语言处理的研究.

参考文献：

[1] Jiang Z, Zhang B, Huang L, *et al*. Chengyu cloze test [C]// Proceedings of the Thirteenth Workshop on Innovative Use of {NLP} for Building Educational Applications @ NAACL-HLT. New Orleans: ACL, 2018.

[2] Liu Y, Liu B, Shan L, *et al*. Modelling context with neural networks for recommending idioms in essay writing [J]. Neurocomputing, 2018, 275: 2287.

[3] Liu Y, Pang B, Liu B. Neural-based Chineseidiom recommendation for enhancing elegance in essay writing [C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019.

[4] Zheng C, Huang M, Sun A. ChID: A large-scale Chinese idiom dataset for cloze test [C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019.

[5] Tan M, Jiang J. A BERT-based dual embedding

model forchinese idiom prediction [C]//Proceedings of the 28th International Conference on Computational Linguistics. Barcelona: ICCL, 2020.

[6] Long S, Wang R, Tao K, *et al*. Synonym knowledge enhanced reader for chinese idiom reading comprehension [C]//Proceedings of the 28th International Conference on Computational Linguistics. Barcelona: ICCL, 2020.

[7] Wang X, Zhao H, Yang T, *et al*. Correcting the misuse: a method for the chinese idiom cloze test [C]//Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures. Online: ACL, 2020.

[8] 徐家伟, 刘瑞芳, 高升, 等. 面向中文成语的阅读理解方法研究[J]. 中文信息学报, 2021, 35: 118.

[9] Aghajanyan A, Shrivastava A, Gupta A, *et al*. Better fine-tuning by reducing representational collapse [C]// Proceedings of the 2020 International Conference on Learning Representations. Austria: OpenReview. net, 2020.

[10] Schuster M, Nakajima K. Japanese andkorean voice search [C]// Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Kyoto: IEEE, 2012.

[11] Liu X, Yu H F, Dhillon I, *et al*. Learning to encode position for transformer with continuous dynam-ical model [C]//Proceedings of the International Conference on Machine Learning. Vienna: PMLR, 2020.

[12] Wei J, Ren X, Li X, *et al*. NEZHA: neural contextualized representation for Chinese language understanding [EB/OL]. (2019-8-21) [2022-02-23]. https://arxiv. org/abs/1909. 00204/.

[13] Chen R T Q, Rubanova Y, Bettencourt J, *et al*. Neural ordinary differential equations [C]//Proceedings of the 32nd International Conference on Neural Information Processing Systems. Montreal: Curran Associates Inc, 2018.

[14] Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need [C]//Proceedings of the 31st Internat-ional Conference on Neural Information Processing Systems. Long Beach: Curran Associates Inc, 2017.

[15] Mikolov T, Chen K, Corrado G, *et al*. Efficient estimation of word representations in vector space [C]//Proceedings of the 2013 International Conference on Learning Representations. Scottsdale: OpenReview. net, 2013.

[16] Wu L, Li J, Wang Y, *et al*. R-Drop: regularized dropout for neural networks [C]. //Proceedings of the 35th International Conference on Neural Information Processing Systems. [S. l.]: OpenReview. net, 2021.

[17] Gao T, Yao X, Chen D. SimCSE: simple contrastive learning of sentence embeddings [C]//Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Punta Cana: ACL, 2021.

[18] Li Z, Zhou J, Zhao H, *et al*. Neural character-level syntactic parsing for Chinese [J]. J Artif Intell Res, 2022, 73: 461.

[19] Reimers N, Gurevych I, Reimers N, *et al*. Sentence-BERT: sentence embeddings using siamese BERT-networks [C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics. Hong Kong: ACL, 2019.

引用本文格式:

中　文: 张本文, 黄方怡, 琚生根. 融合对比学习的成语完形填空算法[J]. 四川大学学报: 自然科学版, 2022, 59: 052003.

英　文: Zhang B W, Huang F Y, Ju S G. An idiom cloze algorithm incorporating contrastive learning [J]. J SichuanUniv: Nat Sci Ed, 2022, 59: 052003.