软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, 2022,33(10):3793−3805 [doi: 10.13328/j.cnki.jos.006307]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 融合预训练语言模型的成语完形填空算法*

琚生根, 黄方怡, 孙界平

(四川大学 计算机学院, 四川 成都 610065)

通信作者: 孙界平, E-mail: sunjieping@scu.edu.cn

**摘 要**: 根据上下文语境选择恰当的成语, 是自然语言处理领域的重要任务之一. 现有的研究将成语完型填空任务看成是文本匹配问题, 虽然预训练语言模型能够在文本匹配研究上取得较高的准确率, 但也存在明显的缺陷: 一方面, 预训练语言模型作为特征提取器时, 会丢失句子间相互信息; 另一方面, 预训练语言模型作为文本匹配器时, 计算开销大, 训练时间和推理时间较长. 另外, 上下文与候选成语之间的匹配是不对称的, 会影响预训练语言模型发挥匹配器的效果. 为了解决上述两个问题, 利用参数共享的思想, 提出了 TALBERT-blank. TALBERT-blank 是将成语选择从基于上下文的不对称匹配过程转换为填空与候选答案之间的对称匹配过程, 将预训练语言模型同时作为特征提取器和文本匹配器, 并对句向量作潜在语义匹配. 这样可以减少参数量和内存的消耗, 在保持准确度的情况下, 提高了训练和推理速度, 达到了轻量高效的效果. 在 CHID 数据集上的实验结果表明: 作为匹配器, TALBERT-blank 相较于 ALBERT, 在保证准确率的情况下, 更大限度地精简了模型的结构, 计算时间进一步缩短 54.35%.

**关键词**: 成语完形填空; 文本匹配; 深度学习; 预训练语言模型
**中图法分类号**: TP18

中文引用格式: 琚生根, 黄方怡, 孙界平. 融合预训练语言模型的成语完形填空算法. 软件学报, 2022, 33(10): 3793−3805. http://www.jos.org.cn/1000-9825/6307.htm

英文引用格式: Ju SG, Huang FY, Sun JP. Idiom Cloze Algorithm Integrating with Pre-trained Language Model. Ruan Jian Xue Bao/Journal of Software, 2022, 33(10): 3793−3805 (in Chinese). http://www.jos.org.cn/1000-9825/6307.htm

## Idiom Cloze Algorithm Integrating with Pre-trained Language Model

JU Sheng-Gen, HUANG Fang-Yi, SUN Jie-Ping

(College of Computer Science, Sichuan University, Chengdu 610065, China)

**Abstract**: One of the crucial tasks in the field of natural language processing (NLP) is identifying suitable idioms due to context. The available research considers the Chinese idiom cloze task as a textual similarity task. Although the current pre-trained language model plays an important role in textual similarity, it also has apparent defects. When pre-trained language model is used as a feature extractor, it ignores the mutual information between sentences; while as a text matcher, it requires high computational cost and long running time. In addition, the matching between context and candidate idioms is asymmetric, which influences the effect of the pre-trained language model as a text matcher. In order to solve the above two problems, this study is motivated by the idea of parameter sharing and proposes a TALBERT-blank network. Idiom selection is transformed from a context-based asymmetric matching process into a blank-based symmetric matching process by TALBERT-blank. The pre-trained language model acts as both a feature extractor and a text matcher, and the sentence vector is utilized for latent semantic matches. This greatly reduces the number of parameters and the consumption of memory, improves the speed of train and inference while maintaining accuracy, and produces a lightweight and efficient effect. The experimental results of this model on CHID data set prove that compared with ALBERT text matcher, the calculation time is further shortened by 54.35 percent for the compression model with a greater extent under the condition of maintaining accuracy.

* 基金项目: 国家自然科学基金(61972270); 四川省新一代人工智能重大专项(2018GZDZX0039); 四川省重点研发项目(2019YFG0521)

收稿时间: 2020-09-26; 修改时间: 2020-12-08; 采用时间: 2021-01-13

**Key words**: Chinese idiom cloze task; text matching; deep learning; pre-trained language model

成语是一种特殊的汉语型式, 是汉语中最美丽、最独特的部分之一. 像其他语言中的俗语一样, 适当地使用成语可以使文章更加引人入胜. 现在仍有 7 000 多个成语在现代汉语、日文、韩文、越南文中被广泛使用[1]. 成语大多约定为四字结构, 短小精悍且能够表达完整的语义信息. 但是成语包含有隐含义、引申义和联想义, 从而导致其实际喻义与字面意思往往有所不同[2], 例如, “亡羊补牢”, 字面意思为“羊逃跑了再去修补羊圈, 还不算晚”, 但是实际喻义为“及时弥补错误”. 因此, 成语的特殊性以及复杂性给自然语言处理(natural language processing, NLP)的各种任务, 如机器翻译[3]、情感分析[4]、完形填空[1,5]等, 带来挑战和机遇. 近年来, 一些学者致力于成语在 NLP 任务中的研究, Wang 等人[2]构建了一个汉语成语知识库, 并对带成语语句进行自动情感分类. Wang 等人[6]为了帮助科学家对古代历史信息的研究, 构建了一个基于中国成语知识图的“古代中国科学工具包(STAC)”. 在机器翻译中, Shao 等人[3]发现含有成语的中英翻译比普通字的机器翻译性能要差, 并提出一种用黑名单来评估成语翻译性能的方法. Zheng 等人[5]在 2019 年公布了一个成语完型填空的公开数据集 CHID, 并用双向长短时记忆网络(bi-directional long short-term memory, Bi-LSTM)[7]和预训练语言模型 BERT[8]对数据集进行了实验.

成语完形填空任务是成语理解中的子任务, 主要是研究不同的语境应该搭配什么成语, 这是汉语学习者的挑战之一[9]. 成语完型填空中存在很多意思相似但不完全相同的近义成语, 学习者需要根据不同的上下文来甄别近义成语间的细微差距, 才能选择出最佳成语, 比如“侃侃而谈”和“口若悬河”. Liu 等人[10]第一次提出成语完型填空的问题. Jiang 等人[1]将注意力机制用于成语完型填空. Liu 等人[11]把成语完形填空问题看成是机器翻译问题, 根据成语的上下文自动翻译为成语的过程. Zheng 等人[5]第一次将预训练语言模型用于解决成语完型填空, 在预先学习了大量语言学特征的基础上微调权重, 选择最恰当的成语. Tan 等人[12]利用 BERT 为特征提取器, 然后将候选成语与填空处匹配, 同时也和上下文匹配. Long 等人[13]利用近义词图来辅助成语的选择. Wang 等人[14]同时利用上下文、成语的定义、成语的多个词嵌入来帮助选择正确的成语.

预训练语言模型的诞生, 是为了解决多义词在不同语境有不同含义的问题, 每一个字都有对应的特征向量, 这个特征与上下文密切相关. 其中, EMLO 模型[15]中是把双向多层循环神经网路(recurrent neural network, RNN)的隐藏层向量按不同权重相加作为特征向量; BERT 模型[8]是将多层 Transformer Block[16]叠加在一起, 利用自注意力机制得到特征向量; ALBERT 模型[17]是基于 BERT 的改进模型, 它通过词嵌入的因式分解和跨层参数共享等改进来提高预训练的效率和性能. 预训练语言模型是事先在大规模的语料库上训练得到的, 使得模型具有较好的泛化性, 因此, 在许多 NLP 任务上取得了很好的性能. 预训练语言模型在成语完型填空中的应用主要有两种方式[18]: 一种是作为匹配器[9], 将候选成语和上下文拼接输入, 通过深度神经网络捕获第 1 段文本中每个汉字与第 2 段文本中所有汉字之间的语义联系; 另一种是作为特征提取, 如 SBERT[19], 将上下文和候选成语分别映射到相同的向量空间, 再计算两个向量之间的距离. 预训练语言模型作为匹配器准确率虽然高, 但训练和推理时间长; 预训练语言模型作为特征提取器时间短, 但准确率明显低于前者[18,19]. 针对以上问题, 鉴于 Lan[17]、Qiao[18]、Reimers[19]的思想, 提出了 TALBERT-blank 模型, 使用 3 个级别的参数共享策略, 从小到大的范围包括 Transformer 层间参数共享、两特征提取器之间参数共享、特征提取器与匹配器之间参数共享, 并在 CHID 数据集上对 TALBERT-blank 模型的准确率和计算时间进行了验证.

本文的贡献主要包括:

(1) 提出了 TALBERT-blank 模型, 这是使用三重态网络结构对 ALBERT 模型进行修改. 特征提取器提取文本自身的语义特征, 匹配器提取两文本间的交互信息;

(2) 通过 3 个级别的参数共享策略, TALBERT-blank 模型实现了最少化参数量, 缩短了计算时间, 提高了模型的计算效率.

本文第 1 节介绍现有的完型填空的研究现状, 并对现有研究方法进行分析总结. 第 2 节形式化描述完型填空任务. 第 3 节详细阐述基于 ALBERT 的 TALBERT-blank 模型. 第 4 节介绍实验数据集、评估指标以及实

验结果与分析. 第 5 节总结全文, 并对未来研究工作进行展望.

## 1 相关工作

完形填空由于其简单的形式被广泛地用于考察阅读理解能力[20,21], 在一些中文完形填空数据集, 如 People Daily & Children's Fairy Tale[22]和 CMRC[23]中, 完形填空的正确答案的字面上与上下文关系密切、重叠性大. 比如"所谓的'傻钱'___, 其实就是买入并持有美国股票这样的普通组合. 这个策略要比对冲基金和其他专业投资者使用的更为复杂的投资方法效果好得多". 正确答案为在下文中出现的"策略". 而成语在字面上与上下文重叠性低, 几乎没有联系, 如"这次是小测验, 考不好没关系, 知道了自己的不足, 赶紧弥补, 以期升学考出好成绩, 这就是古语所说的_____, 为时不晚". 正确答案为"亡羊补牢", 成语中没有一个字在上下文中出现, 更需要将整个成语的语义和上下文语境匹配, 这使得成语完型填空更具有挑战性. 2018 年, Liu 等人[10]第一次提出研究成语完形填空的问题, 它是指根据上下文从候选成语集里面选出最佳成语作为参考成语, 并提出了基于 CNN 的 CLCE 模型. 对上下文信息和候选成语分别使用了全局和局部的嵌入表达, 其中, 全局词嵌入是利用不同宽度的 Convolution Filter 作为滑动窗口函数, 编码各种粒度的语义; 局部词嵌入是利用提示词来对成语所在句子的上下文进行建模. 但是 CNN 擅长获取局部的特征信息, 但不能很好地捕获长距离的信息. 同时, 由于成语本身过于精炼简洁, CNN 很难获取丰富的语义信息. 因此, Jiang 等人[1]通过对成语完整的定义进行编码, 利用成语的定义代替成语. 如"亡羊补牢"替换为"及时弥补错误", 与上下文中的"赶紧弥补"相对应. 再利用两个不同的 Bi-LSTM 网络分别对上下文文本和候选成语的定义进行编码, 将最后一个隐藏层作为候选成语的特征向量与上下文中每个词向量计算相关性, 得到注意力权重. 最后, 用 softmax 函数归一化得到每个候选成语填入填空处的概率. 但是, Bi-LSTM 仍不能很好地解决长序列信息缺失的问题. 2019 年, Liu 等人[11]提出了上下文文本和成语实际上是两个不同语言风格的问题. 上下文是由现代汉语书写的, 而成语大部分是由古汉语书写的. 因此, 他们将成语完形填空任务当作是从上下文到成语的双语机器翻译, 利用序列生成器 Seq2Seq 框架. 首先, Bi-LSTM 分别编码上下文和成语, 使用注意力机制得到注意力向量; 再使用 Bi-LSTM 依次生成下一个字, 生成的四字词语与候选成语用编辑距离计算相似度. 但是, 由于不同位置的填空处需要的成语是不一样的, 而上下文生成的成语却是相同的, 所以还缺乏位置信息来准确生成成语. 随着完形填空任务在自然语言研究中引起越来越多的关注, 公开的测试集成为算法评估的重要依据. 为此, Zheng 等人[5]公布了一个大规模成语完型填空数据集 CHID, 促进成语完形填空任务的研究, 也是第一篇将预训练语言模型用于成语完型填空的文章. 他们在预训练语言模型的基础上微调权重, 选择最恰当的成语. 预训练语言模型在自注意力层参考了位置信息, 使不同位置的相同汉字可以得到不同的注意力, 模型的准确率达到 72.71%. Tan 等人[12]利用 BERT 为特征提取器, 分别编码候选成语和上下文, 每一个候选成语有两个词向量, 其中一个词向量会和填空处的词向量匹配, 另一个向量会和上下文中所有位置的字向量匹配再进行池化. 这样可以实现对称匹配和非对称匹配的结合, 但参数量增加一倍. Long 等人[13]利用图注意力和门控机制为每个候选成语构建一个近义词图, 通过近义词图辅助成语的选择. 这里的关键在于图的构建, 构建好的近义词图也可以促进其他成语任务. Wang 等人[14]指出, 语料库中既包括正确使用的成语, 也包括很多被错误使用的成语. 他们利用成语的定义和成语预训练的词向量对上下文本进行矫正, 再选择出正确的成语.

预训练语言模型[8,24,25]是在大规模语料库上预训练后, 再作为特征提取器, 动态地调整包含上下文文本信息的特征表示, 具有很强的泛化能力. 如 ELMo 模型[15]是多层的两个反向独立的 LSTM 预训练, 证明了在多层的 RNN 结构中, 不同层学到的特征是有差异的. 只要为每一层设置不同的权重, 利用加权和得到的词向量可通过迁移学习指导下游任务. 但是 ELMo 本质上还是单向的语言模型, 只是将两个 LSTM 进行了拼接. GPT 模型[26]基于多层 Transformer Decoder Block[16], 属于生成式预训练模型, 使神经网络在给定上文的情况下预测下一个字的准确率提高. 但是 GPT 下游任务的模型结构必须和预训练模型结构相同, 适合文本生成任务. BERT 模型基于多层 Transformer Encoder Block[16], 通过前后的上下文预测填空词和根据上一句预测下一句两种方式进行预训练, 预训练好的 BERT 只需要进行微调就可以为多种任务提供基础模型. 由于预训练语

言模型的参数量非常大, 很多研究者开始转向如何减少预训练语言模型的参数量, 以实现轻量级的预训练语言模型的研究. TinyBert 模型[27]是用知识蒸馏的方式, 将“teacher”BERT 中预训练学习的知识迁移到“student” TinyBERT, 但是 TinyBERT 侧重于减少模型在应用部署时的存储和时间消耗, 训练模型所需要的硬件和时间成本反而增加. ALBERT 模型一方面采用低秩分解的方式对嵌入层部分降维达到减少参数的目的, 另一方面, Transformer Block 之间的参数共享, 层间的权重共享使层与层之间的传递更加平滑, 鲁棒性更好, 使之相较于 BERT 模型的参数量减少 89.7%, 训练时间缩短 35.9%.

本文将成语完型填空看作文本匹配问题, 根据与上下文的匹配度对候选成语进行排序. 现有的文本匹配主要是通过将两段文本映射到向量空间, 再对两个向量进行距离计算(如欧氏距离). 有研究者尝试将预训练 BERT 模型应用在文本匹配中, 主要分为两种方式[18]: 一种是将 BERT 作为特征提取器, 把提取的特征作为其他神经网络的输入, 如 BAS 模型[28], BAS 首先判断问题需要的答案类型, 把满足条件的答案做上特殊标记, 再输入 BERT, 最后将所有提取的特征词向量通过 CNN 或 RNN, 进行文本关系分类; 另一种方式是将 BERT 作为匹配器, 将需要检测的两文本先进行拼接, 再输入到 BERT 中, 使其提取文本关系的特征, 如 CEDR 模型[29]. CEDR 一方面直接利用 BERT 作为匹配器, 另一方面将 BERT 得到的字向量输入到 KNRM[30]中, 综合了预训练模型和传统排序模型各自的优势. Qiao 等人[18]对 BERT 作为特征提取器和匹配器两种方式进行了比较分析, 结论是: 将 BERT 作为特征提取器生成句向量, 再匹配的效果远不如将 BERT 直接作为匹配器的效果. 原因是 Transformer Block 擅长学习句子间细粒度的交互信息.

近年来, 孪生网络一直是文本匹配中的研究热点. 孪生网络[31]是基于权重共享使两个文本分别输入相同的网络结构, 训练过程中参数同时更新. 两个输入通过相同的神经网络 RNN[32−34]或 CNN[35,36]映射到相同的向量空间, 然后计算两个输出向量之间的距离. SBERT 模型[19]是第一个基于预训练的孪生网络模型, 将问题和答案分别输入到孪生 BERT 特征提取器, 利用特殊字符[*CLS*]的字符向量代表句向量, 再利用余弦距离得到两者的相关性. 相较于 BERT 作为匹配器, 孪生 BERT 特征提取器训练时间明显缩短, 但准确率也降低. 孪生网络中权值共享的目的有两个: (1) 减少参数量, 减小模型的复杂度; (2) 将两个待检测文本映射到同一个向量空间, 使数据分布保持一致, 在同一个空间维度对两个向量进行编码, 这样有利于对两个文本中相似特征的抽取. 孪生网络由于其对称的特征, 使其更适用于对称的文本匹配. 但是上下文和候选成语无论是从内容、长度还是结构上都存在较大的差异, 具有不对称、不平衡的特点, 因此需要对传统的孪生网路结构加以改进.

从上述研究可知, 预训练语言模型在成语完型填空的应用上存在两个问题.

(1) 预训练语言模型在文本匹配的应用上有两种方式(匹配器或特征提取器): 作匹配器的准确率更高, 但时间开销巨大; 作特征提取器的运行速度快, 但准确率不高;

(2) 上下文(问题)和候选成语(答案)之间的匹配是不对称的, 不适合传统的孪生神经网络 SBERT[19], 需要对其结构进行改进.

## 2 成语完型填空任务描述

完型填空任务是根据上下文的描述信息, 从多个候选选项中找出正确的结果; 而其余候选选项是答案的干扰项, 有的和答案意思相近, 有的和上下文完全不相关.

- 目标: 从候选成语集合中找到与上下文匹配度最高的成语.

输入: $Q$ (上下文): 一段文本中成语被替换成空格后, 剩余的文本;

$Q^S$ (上下文集合): 多个上下文文本组成的集合, $Q^S=\{Q_1,Q_2,...,Q_{l_q}\}$, 其中, $l_q$ 表示 $Q^S$ 中 $Q$ 的个数;

$Q_i$: $Q^S$ 中第 $i$ 个上下文, $Q_i=[q_m]_{m=1}^{l_m}$, 其中, $l_m$ 表示 $Q_i$ 的长度, $q_m$ 表示序列中第 $m$ 个汉字;

$C_i^s$ : $Q_i$ 对应的候选成语集合, $C_i^s=\{C_{i1},C_{i2},...,C_{il_t}\}$, 其中, $l_t$ 为 $C_i^s$ 中成语的个数;

$C_{ij}$: $C_i^s$ 中第 $j$ 个候选成语, $C_{ij}=[c_n]_{n=1}^{l_n}$, 其中, $l_n$ 为该候选成语的长度, $c_n$ 为成语中第 $n$ 个汉字;

将 $Q_i$ 和 $C_i^S$ 组合成序列对$\langle Q_i,C_{ij}\rangle$作为输入;

输出: 匹配得分最高的结果, 即为应该填入空格处的成语“深居简出”.

从图 1 中可以看出: 近义词“安分守己”的得分也较高, 但“安分守己”更侧重于形容老实本分, 而“深居简出”则侧重于少出门, 与后面的“不大敢露面”相关联.

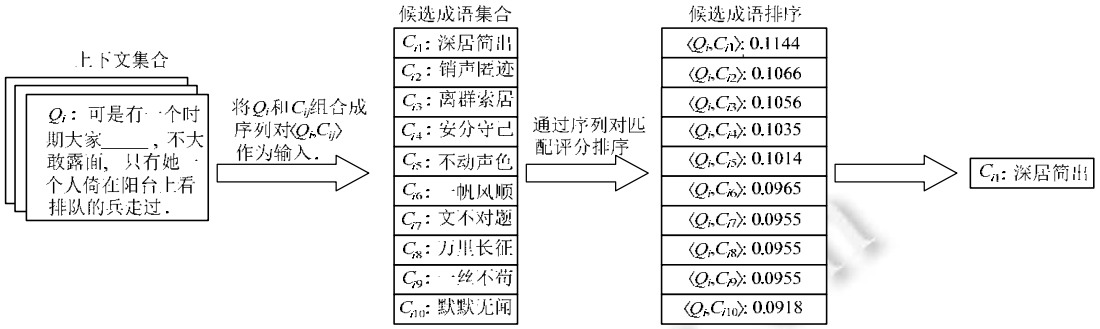


图 1 成语完形填空的基本流程

## 3 本文方法

上下文与候选成语之间无论是在内容、结构和长度上都存在明显差异, 如果直接将候选成语与上下文匹配, 上下文中的多余信息就会干扰匹配效果. 本文首先用特征提取器分别编码上下文和候选成语, 保证每一个字的特征表示都包含了所有位置的语义信息, 然后截取填空处与候选成语匹配. 为了控制参数量和计算时间, 本文提出了三重态网络(triple network). 三重态网络是指包含 3 个相同子结构的神经网络框架, 这里的“相同”是指具有相同的参数并且同步进行参数更新. 三重态网络的优势是: 通过参数共享实现最小化参数量, 以提高模型的计算效率. 本文基于三重态网络的思想设计出 TALBERT-blank 模型, 其中, TALBERT (triple ALBERT)即是 3 个完全相同的 ALBERT, blank 即是截取填空处与候选成语匹配, 这里的匹配包括预训练模型的匹配和句向量匹配两个部分. TALBERT-blank 的框架图如图 2 所示.

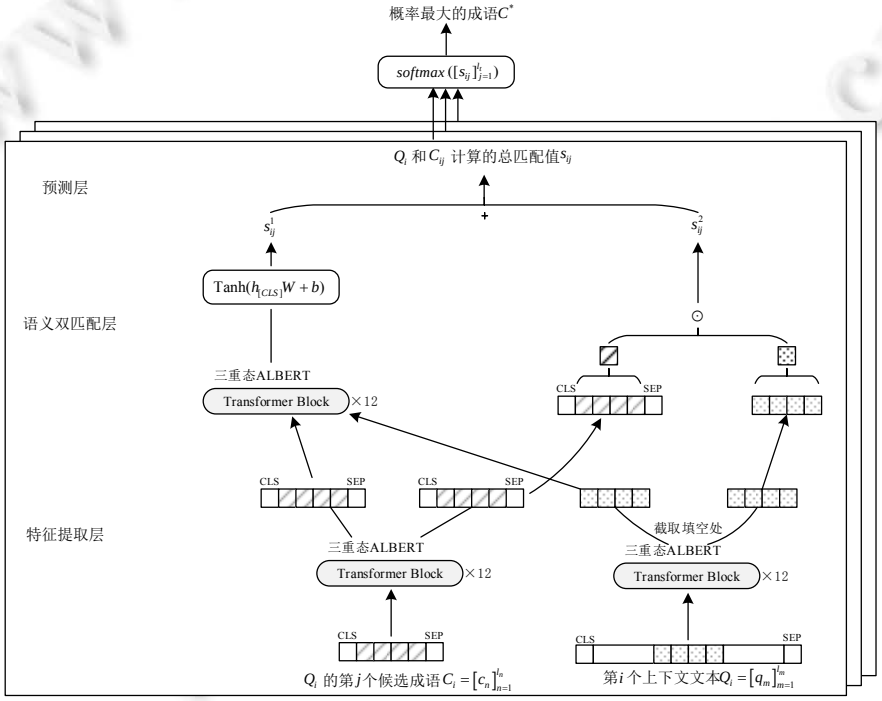


图 2 TALBERT-blank 模型结构

TALBERT-blank 包括特征提取层、语义双匹配层、预测层: 特征提取层分别对上下文和候选成语进行编码; 语义双匹配层判断填空处和候选成语在语义上的相关程度; 预测层归一化所有候选成语的匹配值, 输出概率最高的候选成语. 下面将对模型的各层给出详细的介绍.

### 3.1 多层权重共享的Transformer特征提取层

特征提取层结构如图 3 所示, 包括嵌入层和编码层: 嵌入层是将每个汉字转换为固定维度的向量表示, 编码层是将固定特征向量转换为包含其他位置语义信息的特征向量.

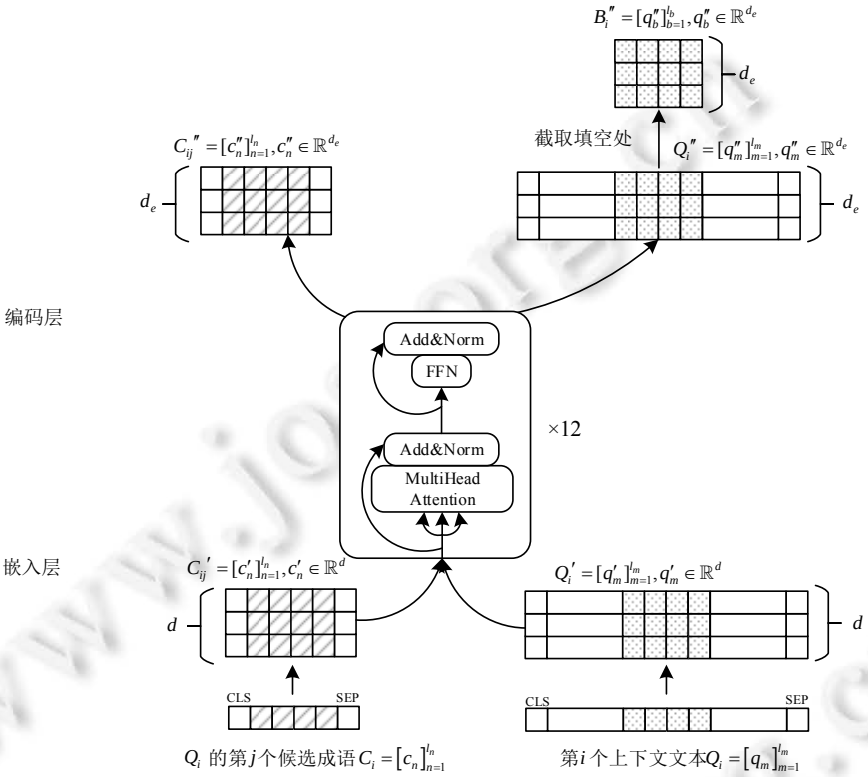


图 3 特征提取层

嵌入层包括字嵌入和位置嵌入, 嵌入层的输出为字向量和位置向量之和. 字嵌入是将序列中的每个汉字转换为字向量, 字向量嵌入矩阵的初始化是随机生成的. 位置嵌入是将位置索引转换为位置向量, 位置索引标识了汉字所在的位置, 代表句子的时间序列属性. 在序列的两端分别添加特殊字符[*CLS*]和[*SEP*], 分别表示整个序列的句向量和句子末尾. 公式(1)和公式(2)中, 上下文的嵌入矩阵 $Q_i' = [q_m']_{m=1}^{l_m}$, $q_m'$ 为汉字 $q_m$ 的嵌入向量, $q_m' \in \mathbb{R}^d$, $d$ 代表嵌入层的维度; 候选成语的嵌入矩阵 $C_{ij}' = [c_n']_{n=1}^{l_n}$, $c_n'$ 为汉字 $c_n$ 的嵌入向量, $c_n' \in \mathbb{R}^d$.

$$Q_i' = [q_m']_{m=1}^{l_m} = Embed_{token}(Q_i) + Embed_{position}(Q_i) \tag{1}$$

$$C_{ij}' = [c_n']_{n=1}^{l_n} = Embed_{token}(C_{ij}) + Embed_{position}(C_{ij}) \tag{2}$$

从嵌入层得到嵌入向量输入编码层, 编码层由 12 层 Transformer Block 构成, 所有层共享参数. 经过 12 层 Transformer Block, 上下文和候选成语进一步编码为 $Q_i''$ 和 $C_{ij}''$, $Q_i'' = [q_m'']_{m=1}^{l_m}$, $q_m'' \in \mathbb{R}^{d_e}$, $q_m''$ 是嵌入向量 $q_m'$ 基于编码层后的新向量, $d_e$ 表示编码层的维度. $C_{ij}'' = [c_n'']_{n=1}^{l_n}$, $c_n'' \in \mathbb{R}^{d_e}$, $c_n''$ 是嵌入向量 $c_n'$ 基于编码层后的新向量. Transformer Block 分为多头自注意力和前馈神经网络两个子模块, 子模块的输出都会经过残差网络和层归一化, 即 *LayerNorm*(*x*+*Sublayer*(*x*)), 其中, *x* 为子模块的输入, *Sublayer*(*x*)为子模块的输出. 下面详细介绍多头自

注意力层.

如图 4 所示, 以上下文的特征矩阵 $Q_i'$ 为例, 阐述多头自注意力层. 多头自注意力层的核心是: 在多个子空间平行地计算自注意力, 再将每个子空间的自注意力拼接起来.

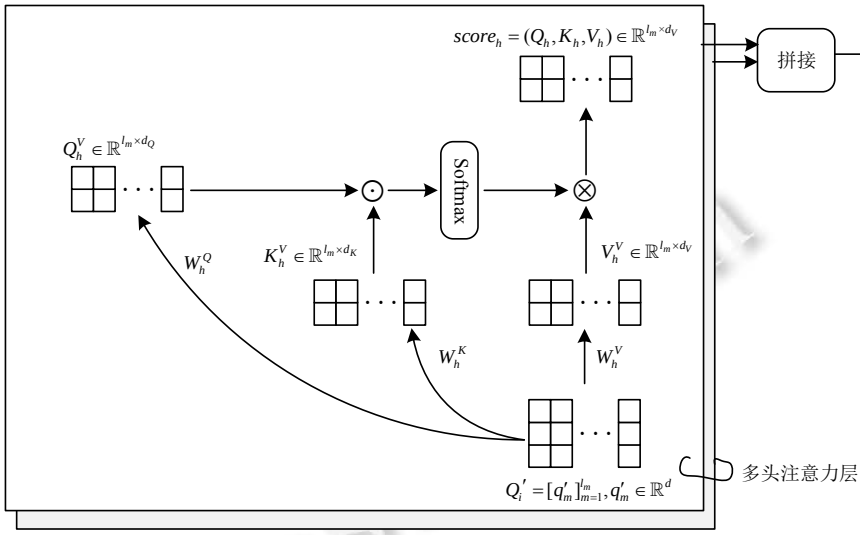


图 4 多头自注意力层

- 公式(3), $Q_i'\in\mathbb{R}^{l_m\times d}$ 在第 $h$ 子空间的自注意力的计算.

先通过 3 个线性映射得到 $Q_h\in\mathbb{R}^{l_m\times d_Q}$, $K_h\in\mathbb{R}^{l_m\times d_K}$, $V_h\in\mathbb{R}^{l_m\times d_V}$, $d_Q$、$d_K$、$d_V$ 分别表示 *Query*(询问)、*Key*(键)、*Value*(值)的映射维度. 然后, 将 $Q_h$ 和 $K_h$ 矩阵相乘, 利用 softmax 函数完成归一化得到注意力矩阵, 注意力矩阵中重要位置的汉字权重更大. 在此过程中除以 $\sqrt{d_K}$, 目的是对点积注意力进行缩放, 使梯度得到有效传播. 最后, 再将注意力矩阵与 $V_h$ 进行矩阵乘法, 最终获得第 $h$ 子空间的注意力输出 $score_h$:

$$score_h(Q_h,K_h,V_h)=softmax\left(\frac{Q_hK_h^T}{\sqrt{d_K}}\right)V_h \tag{3}$$

- 整合子空间提取的不同语义信息.

如公式(4), 先将各个子空间得到的自注意力拼接, 再经过线性映射:

$$x=concat(score_1,score_2,\ldots,score_k)W_0 \tag{4}$$

其中, $W_0\in\mathbb{R}^{kd_V\times d}$, $k$ 代表子空间的数量. 多头注意力层的输出会经过残差网络和层归一化, 进入两层全连接神经网络, 再通过一次残差网络和层归一化, 得到最终每层 Transformer Block 的输出.

编码层的所有 Transformer Block 之间参数共享, 防止参数随着网络深度的增长而增长. 上下文和候选成语通过相同的编码层映射到同一向量空间, 使数据分布保持一致, 以利于后续的相似度计算. 在经过编码层之后, 为了去除冗余信息, 仅保留填空的特征矩阵和候选成语匹配. 由于填空处和候选成语在内容和长度上高度对称, 因此可以实现对称性匹配. 填空处的特征矩阵 $B_i''\in\mathbb{R}^{l_b\times d_e}$, $l_b$ 为填空处的长度, $d_e$ 代表编码层的维度; 候选成语的特征矩阵 $C_{ij}''\in\mathbb{R}^{l_n\times d_e}$, $l_n$ 为候选成语 $C_{ij}$ 的长度.

### 3.2 语义双匹配层

语义双匹配层的输入为候选成语和填空处组成的序列对 $\langle C_{ij}'',B_i''\rangle$, 它的作用是判断填空处和候选成语在语义层面是否密切. 匹配层通过两个方面进行判断: 一方面将候选成语与填空处的语义特征进行拼接, 利用 Transformer Block 匹配; 另一方面是得到两个序列的句向量, 再计算两个句向量之间的相似度.

(1) Transformer Block 作为匹配器

为了判断填空处和成语的相似度, 利用多头自注意力获取交互信息, 序列对 $\langle C''_{ij}, B''_i\rangle$ 按前后顺序拼接成矩阵 $E_{ij} \in \mathbb{R}^{l_s \times d_e}$, 如公式(5):

$$E_{ij} = concat(C''_{ij}, B''_i) \tag{5}$$

$B''_i$ 是指 $Q_i$ 中填空处的编码层输出, $C''_{ij}$ 为 $Q_i$ 对应的第 $j$ 个候选成语的编码层输出, $l_s$ 表示拼接后的序列长度, $l_s$=$l_n$+$l_b$.

$E_{ij}$ 由候选成语和填空处拼接形成, 所以加上序列标志区分两部分的来源, 序列标志为 0 代表来自候选成语, 序列标志为 1 代表来自填空处, 再将序列标志转换为嵌入向量表示. $E_{ij}$ 加上序列标志的嵌入向量后, 输入与编码层相同结构的 Transformer Block 匹配层, 输出 $E'_{ij} \in \mathbb{R}^{l_s \times d_e}$, $d_s$ 为匹配器的维度.

匹配器中的 Transformer Block 与特征提取层结构和权重相同. 特殊字符[*CLS*]的输出向量 $e'_{[CLS]}$ 作为两段序列的相互关系预测向量, $e'_{[CLS]} \in \mathbb{R}^{d_e}$. 再将 $e'_{[CLS]}$ 通过一层前馈神经网络层, 如公式(6), 激活函数为 *Tanh* 函数, 最后通过一层线性映射, 如公式(7):

$$BertPooler(e'_{[CLS]}) = Tanh(e'_{[CLS]}W_3 + b_3) \tag{6}$$

$$s^1_{ij} = BertPooler(e'_{[CLS]})W_4 + b_4 \tag{7}$$

$W_3$、$W_4$、$b_3$、$b_4$ 是学习的参数, $W_3 \in \mathbb{R}^{d_s \times d_s}$, $W_4 \in \mathbb{R}^{d_s \times 1}$, $s^1_{ij}$ 表示序列对 $\langle C''_{ij}, B''_i\rangle$ 通过 Transformer Block 匹配器得到的匹配值, $s^1_{ij} \in \mathbb{R}^1$.

(2) 句向量相似度计算

因为词向量能够表达的语义空间有限, 为了扩大语义空间的表示范围, 使其能够代表更多的语义信息, 将特征提取层得到的语义特征通过直接压缩的方式获得句向量, 句向量比词向量装载信息的能力更强. 从词向量序列获取句向量, 可以通过选择最大值、平均值、求和的方式, 本文将这几种方法都进行了比较, 发现求和方式最好. 将候选成语 $C''_{ij}$ 的 $l_n$ 个字向量直接相加, 得到候选成语 $C''_{ij}$ 的句向量表示 $v_{c_{ij}} = \sum_{n=1}^{l_n} c''_n$, $v_{c_{ij}} \in \mathbb{R}^{d_e}$. 将上下文 $Q''_i$ 的 $l_m$ 个字向量直接相加, 得到候选成语 $Q''_i$ 的句向量表示 $v_{q_i} = \sum_{m=1}^{l_m} q''_m$, $v_{q_i} \in \mathbb{R}^{d_e}$.

本文比较了余弦距离、欧氏距离、内积、哈达玛积这 4 种计算相似性的方法以评估两个句向量 $v_{c_{ij}}$ 和 $v_{q_i}$ 之间的相似性, 发现采用哈达玛积再线性映射的方式准确率最高. 因此, 文本采用先计算哈达玛积再线性映射的方式, 如公式(8):

$$s^2_{ij} = (v_{c_{ij}} \otimes v_{q_i})W_5 + b_5 \tag{8}$$

$v_{c_{ij}}$ 和 $v_{q_i}$ 计算哈达玛积, 然后通过线性转换为一维向量, 其中, $W_4$ 和 $b_4$ 为学习的参数, $W_5 \in \mathbb{R}^{d_e \times 1}$, $b_5 \in \mathbb{R}^1$, ⊗表示哈达玛积运算, 最终结果得到的是 $\langle C''_{ij}, B''_i\rangle$ 中两个句向量的相似度 $s^2_{ij}, s^2_{ij} \in \mathbb{R}^1$.

### 3.3 预测层

预测层的作用是根据匹配层输出的匹配值进行归一化, 输出概率最大的成语. 语义双匹配层得到的两个匹配值 $s^1_{ij}$ 和 $s^2_{ij}$ 先相加得到 $s_{ij} \in \mathbb{R}^1$, $s_{ij}$ 表示$\langle Q_i, C_{ij}\rangle$序列对最终的匹配得分. 由于 $Q_i$ 有 $l_n$ 个候选成语, 并分别组成序列对 $\{\langle Q_i, C_{i1}\rangle, \langle Q_i, C_{i2}\rangle, ..., \langle Q_i, C_{il_t}\rangle\}$, $l_t$ 为候选成语集中候选成语的个数, 将所有序列对的匹配得分进行拼接, 如公式(9), 得到 $S$:

$$S = concat(s_{i1}, s_{i2}, s_{i3}, ..., s_{il_t}) = \{s^1_{ij} + s^2_{ij}\}^{l_t}_{j=1} = \{s_{ij}\}^{l_t}_{j=1} \tag{9}$$

然后, 利用 softmax 函数实现归一化, 如公式(10), 使候选集合中候选答案的总概率和为 1, 得到 $Q_i$ 对应的所有候选成语, 填入填空处的概率分布 $\hat{y}_i$. $C^*$为概率最大的候选成语, 如公式(11):

$$\hat{y}_i = softmax(S) = softmax(\{s_{ij}\}^{l_t}_{j=1}) \tag{10}$$

$$C^* = \arg\max(\hat{y}_i) \tag{11}$$

$\hat{y}_i$ 是 $Q_i$ 对应的所有候选成语填入填空处的概率分布. 采用交叉熵损失函数进行训练, 如公式(12):

$$loss = -\sum_{i=1}^{I_t} y_{ij} \log(\hat{y}_{ij}) \tag{12}$$

$\hat{y}_{ij}$ 为预测的第 $j$ 个候选成语的概率值. $y_{ij}$ 标记了第 $j$ 个候选成语是否为真实的正确答案, 如果第 $j$ 个候选成语是正确答案, 则 $y_{ij}$=1; 反之为 0.

## 4 实验与结果

### 4.1 实验设置

- 数据集

本文使用的成语完形填空数据集是 CHID (Competition 版本), CHID 数据集包括了域内数据和域外数据: 域内数据有 1 319 030 段上下文序列, 域外数据集则有 52 644 段. 新闻和小说作为域内数据, 分为训练集、开发集和测试集, 域内数据涵盖 3 848 个汉语成语; 杂文作为域外数据用来测试, 以评估完形填空模型的泛化能力. 域内数据与域外数据存在下面一些差异: 域内数据的段落平均长度约为 100 个字, 而域外数据的段落平均长度为 127 个字; 段落的平均空白数也有所不同, 域内数据为 1.25, 域外数据为 1.49; 此外, 域外数据中成语出现的频率高于域内数据中成语出现的频率. 这些差异都使得域外测试集更具挑战性, 难度更大.

- 实验参数

本文的实验条件为 1 个 GTX1080Ti-11G, 使用 PyTorch (https://github.com/pytorch/pytorch)框架, BERT 和 RoBERTa 预训练语言模型采用科大讯飞基于中文维基和通用数据训练的 BERT-wwm-ext 和 RoBERTa-wwm-base-ext (https://github.com/ymcui/Chinese-BERT-wwm); ALBERT 预训练语言模型采用 Google 发布的基于中文语料库训练的 ALBERT-base (https://github.com/lonePatient/albert_pytorch), 学习率为 $8e^{-5}$, 嵌入层维度为 768; ALBERT 隐藏层维度为 768, batch size 为 16, 训练迭代轮次为 3 epochs, dropout 为 0.5.

- 评估指标

本文采用填空的准确率作为评估指标, 如公式(13)所示:

$$\text{准确率}=\text{正确的答案数}/\text{问题个数} \tag{13}$$

### 4.2 基线模型

为了更好地验证本文提出的 TALBERT-blank 模型的有效性, 我们以下面 3 个模型作为基线模型.

(1) Liu (2018)[10]: Liu 等人提出的基于 CNN、使用全局、局部信息表达的模型 GLCE;

(2) Jiang (2018)[1]: 利用两个不同的 Bi-LSTM 网络分别编码上下文文本和候选成语的定义, 再利用注意力匹配;

(3) Liu (2019)[11]: 利用 Seq2seq 框架生成的四字词语, 再用编辑距离进行匹配.

### 4.3 实验结果

(1) 实验结果对比和消融实验分析

表 1 中, 平均值为模型在验证集、测试集、域外集准确率的算术平均数. 训练速度和推理速度均以 ALBERT 为比较标准. Bi-LSTM+BERT 和 Seq2Seq+BERT 分别是将 Bi-LSTM 和 Seq2Seq 模型中的词嵌入用 BERT 代替. SALBERT (siamese ALBERT)两个孪生 ALBERT 特征提取器分别作为上下文和候选成语特征提取器, 再用句向量的余弦相似度进行匹配. 选择余弦相似度用于匹配, 是因为在文献[19]中提到: SBERT 中不同相似度度量方法的匹配结果间没有差异. TALBERT (triplet ALBERT)是在 SALBERT 的基础上再增加一个 ALBERT 匹配器, 是三重态网络在 ALBERT 上的实现. TALBERT-blank 和 TALBERT 都用到了三重态网络, 但二者的区别在于: TALBERT 是候选成语与整个上下文进行匹配; 而 TALBERT-blank 是候选成语与填空处匹配, 并增加句向量的匹配. 消融实验的目的是探究各个部分是否发挥了作用.

表 1 实验结果对比和消融实验分析

| | 验证集 | 测试集 | 域外集 | 平均值 | 训练速度 | 推理速度 |
|---|---|---|---|---|---|---|
| GLCE[10] | 57.22 | 58.37 | 47.55 | 54.38 | – | – |
| Bi-LSTM[1] | 60.32 | 59.90 | 49.74 | 56.65 | 2.94× | 3.12× |
| Bi-LSTM+BERT | 71.73 | 69.21 | 62.38 | 67.77 | 1.36× | 1.51× |
| Seq2seq[11] | 48.36 | 48.30 | 37.98 | 44.88 | **3.75×** | **3.89×** |
| Seq2Seq+BERT | 33.85 | 33.74 | 27.62 | 31.74 | 1.60× | 1.89× |
| ALBERT[17] | 74.55 | 74.36 | **67.33** | 72.08 | 1.0 | 1.0 |
| SALBERT | 70.90 | 70.53 | 62.37 | 67.93 | 3.34× | 3.52× |
| TALBERT | 73.97 | 73.96 | 65.91 | 71.28 | 0.78× | 0.81× |
| TALBERT-blank | **74.65** | **74.66** | 66.92 | **72.08** | 1.92× | 2.11× |

由表 1 可知: TALBERT-blank 的准确率能够明显高于 3 种模型基线模型, 并超过了增加了预训练因素的模型, 这表明所提出的模型的确能够更加准确地匹配成语信息. 将 Bi-LSTM+BERT 和 Bi-LSTM 进行比较, 说明预训练因素能够提高匹配的准确性. 将 Seq2Seq+BERT 和 Seq2seq 进行比较却得到相反的结论, 由此可以证明 BERT 不适合文本生成[37,38]. TALBERT 相较于 SALBERT 在测试集和域外集上的准确率提升了 4.9%和 5.7%, 域外数据集的准确率提升幅度高于测试集准确率的提升幅度, 表明 TALBERT 中的匹配器比 SALBERT 中的余弦相似度能够捕获更多的交互信息, 并具有更强的泛化能力. TALBERT-blank 是截取填空处与候选成语匹配, 这里的匹配包括预训练模型的匹配和句向量匹配两个部分. 测试集和域外集的准确率进一步提升了 0.9%和 1.5%, 说明填空处和候选成语在内容和结构上的对称形式能够促进三重态网络发挥作用, 使模型生成更接近标准值的答案. 综上, TALBERT-blank 模型的各个模块均对结果的提升起到正向作用, 消融实验的结果表明了模型的有效性.

(2) 预训练语言模型在准确率以及计算速度上的差异

为了探究在成语完型填空任务中, 预训练语言模型作为特征提取器和匹配器在准确率及计算速度上的差异, 本文比较了 BERT 和 SBERT. 为了探究三重态网络能否扩展到其他预训练模型, 本文设计了 TBERT-blank 和 TRoBERTa-blank, 见表 2. SBERT 是孪生 BERT 特征提取器, 提取上下文和候选成语的句向量进行匹配. BERT、RoBERTa 和 ALBERT 是将候选成语与问题文本以句子对的形式输入, 将预训练语言模型作为匹配器. TBERT-blank 和 TRoBERTa-blank 类似于 TALBERT-blank, 是三重态网络和句向量匹配的方式在 BERT 和 RoBERTa 上的实现. 表 2 中, 训练速度和推理速度均以 ALBERT 为参考标准进行比较.

表 2 预训练语言模型在准确率以及计算速度上的差异

| | 验证集 | 测试集 | 域外集 | 平均值 | 训练速度 | 推理速度 |
|---|---|---|---|---|---|---|
| SBERT[19] | 72.71 | 72.37 | 64.65 | 69.91 | **2.41×** | **3.21×** |
| BERT (https://github.com/ewrfcas/bert_cn_finetune) | 82.20 | 83.04 | – | – | 0.80× | 0.86× |
| TBERT-blank | 81.08 | 80.79 | 74.81 | 78.89 | 1.37× | 1.56× |
| RoBERTa (https://github.com/ewrfcas/bert_cn_finetune) | **83.78** | **83.62** | – | – | 0.81× | 0.84× |
| RoBERTa-blank | 82.99 | 82.17 | **77.43** | **80.86** | 1.39× | 1.60× |
| ALBERT[17] | 74.55 | 74.36 | 67.33 | 72.08 | 1.0 | 1.0 |
| TALBERT-blank | 74.65 | 74.66 | 66.93 | 72.08 | 1.92× | 2.11× |

BERT 与 SBERT 比较, 可以说明两点: 第 1 点, 自注意力机制的低细粒度能够促进句子间交互信息的捕获, 但计算开销较大; 第 2 点, 预训练语言模型作为特征提取器, 将单个句子映射到某个向量空间, 信息的压缩程度大, 句子间的交互信息容易丢失, 匹配能力稍差.

TBERT-blank 相较于 BERT、TRoBERTa-blank 相较于 RoBERTa、TALBERT-blank 相较于 ALBERT, 准确率相差不大, 但计算速度都有所提升. 原因可以解释为:

- 第一, 文本的上下文内容含有噪声, 不相关的字会干扰选择出正确的成语. 截取填空处与候选成语的匹配方式, 从一定程度上减轻了嘈杂字的不利影响;
- 第二, 三重态网络中参数充分共享, 预训练模型既作为特征提取器又作为语义匹配器, 平衡了准确率和计算时间;
- 第三, 利用句向量隐式地表达填空处和候选成语的语义, 以便于语义相似的文本在距离上靠近, 结合

以上优势选出最佳成语.

三重态网络和句向量匹配的方式在其他预训练模型上同样能够发挥作用, 可在保持准确率的情况下缩短计算时间. 由此可知: 可将同样的思想扩展到更多的预训练模型以及预训练模型的改进版, 比如可扩展到基于 BERT 的双嵌入模型[12]和近义词图模型[13], 因此有可能进一步提高成语选择的准确率.

(3) 距离函数以及句向量提取方式比较分析

为了探究匹配层中句向量不同距离计算公式的影响, 除了计算哈达玛积外, 本文另外还比较了内积、余弦距离和欧氏距离, 见表 3, 结果提示: 欧氏距离的准确度最低, 平均值为 59.548; 哈达玛积、内积、余弦距离这 3 种距离计算方式的结果相差不大, 但均优于欧氏距离. 原因可能是后 3 种距离计算方式中都包含两个向量中的对应元素相乘, 这会使两个向量在同一维度上相似的特征都会被放大, 而同一维度上不相似的特征或者相似的低特征就会被缩小. 线性映射哈达玛积的方式可以通过权重让模型自己决定关注于哪些维度, 所以准确率高于内积和余弦距离. 从匹配层获取的语义特征序列获取句向量的方法也可能会影响最终的结果, 为此, 本文将句向量的提取方式进行了比较, 包括求和、最大值和平均值这 3 种方式, 见表 4. 容易发现, 求和的效果是最好的. 原因可能是求和的方式叠加了语义空间的表示范围, 使句向量能够代表更多的语义信息.

表 3 4 种距离函数的比较

| | 验证集 | 测试集 | 域外集 | 平均值 |
|---|---|---|---|---|
| 内积 | 65.20 | 65.41 | 57.81 | 62.81 |
| 哈达玛积 | 74.65 | 74.66 | 66.92 | 72.08 |
| 余弦距离 | 73.53 | 73.58 | 65.90 | 71.00 |
| 欧氏距离 | 62.17 | 62.63 | 53.85 | 59.55 |

表 4 3 种提取句向量方式的比较

| | 验证集 | 测试集 | 域外集 | 平均值 |
|---|---|---|---|---|
| 最大值 | 73.22 | 72.89 | 65.52 | 70.54 |
| 平均值 | 73.57 | 73.51 | 65.62 | 70.90 |
| 求和 | 74.65 | 74.66 | 66.92 | 72.08 |

## 5 总 结

成语完型填空是一项具有挑战性的任务, 学习者无法直接通过 4 个字简单拼凑出成语的含义, 也很难从字面上将成语和上下文匹配. 此外, 还存在很多意思相似但不完全相同的近义成语. 这些困难促使我们探究成语完形填空任务, 在提供删除了成语的上下文(问题)的情况下, 系统将从候选成语中自动推荐最佳成语(答案)以填补空白. 预训练语言模型可使成语完型填空的准确率达到较高的结果, 但预训练语言模型参数量庞大, 计算设备要求高, 训练时间过长. 为了解决这个问题, 本文 3 次利用参数共享策略, 包括 Transformer 层间共享、两语句的特征提取器共享、特征提取器和匹配器共享. 通过共享参数结合句向量匹配的方法在保持模型准确率的基础上, 减少了参数量. TALBERT-blank 模型在 CHID 数据集上较 ALBERT 在推理时间上缩短了 54.35%, 准确率相当. 另外, TALBERT-blank 中三重态网络和句向量匹配的方式能够扩展到其他预训练模型. 但 TALBERT-blank 选择的成语虽然能够符合上下文的主要意思, 但却忽略了转折等细节. 未来, 我们将进一步深入研究如何更好地获取填空处的语义而非整个上下文的语义[39,40].

**References**:

[1] Jiang Z, Zhang B, Huang L, *et al*. Chengyu cloze test. In: Proc. of the 13th Workshop on Innovative Use of NLP for Building Educational Applications. 2018. 154−158.

[2] Wang L, Yu S. Construction of Chinese idiom knowledge-base and its applications. In: Proc. of the 2010 Workshop on Multiword Expressions: From Theory to Applications. 2010. 11−18.

[3] Shao Y, Sennrich R, Webber B, *et al*. Evaluating machine translation performance on chinese idioms with a blacklist method. In: Proc. of the 11th Edition of the Language Resources and Evaluation Conf. 2018. 31−38.

[4] Liu P, Qian K, Qiu X, *et al*. Idiom-aware compositional distributed semantics. In: Proc. of the 2017 Conf. on Empirical Methods in Natural Language Processing. 2017. 1204−1213.

[5] Zheng C, Huang M, Sun A. CHID: A large-scale Chinese idiom dataset for cloze test. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. 778−787.

[6] Wang M, Xiao M, Li C, *et al*. STAC: Science toolkit based on Chinese idiom knowledge graph. In: Proc. of the Workshop on Extracting Structured Knowledge from Scientific Publications. 2019. 57−61.

[7] Hochreiter S, Schmidhuber J. Long short-term memory. Neural Computation, 1997, 9(8): 1735−1780.

[8] Devlin J, Chang MW, Lee K, *et al*. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of the 2019 NAACL-HLT, Vol.11. 2019. 4171−4186.

[9] Cieślicka A. Literal salience in on-line processing of idiomatic expressions by second language learners. Second Language Research, 2006, 22(2): 115−144.

[10] Liu Y, Liu B, Shan L, *et al*. Modelling context with neural networks for recommending idioms in essay writing. Neurocomputing, 2018, 275: 2287−2293.

[11] Liu Y, Pang B, Liu B. Neural-based Chinese idiom recommendation for enhancing elegance in essay writing. In: Proc. of the 57th Annual Meeting of the Association for Computational Linguistics. 2019. 5522−5526.

[12] Tan M, Jiang J. A BERT-based dual embedding model for Chinese idiom prediction. In: Proc. of the 28th Int'l Conf. on Computational Linguistics. 2020. 1312−1322.

[13] Long S, Wang R, Tao K, *et al*. Synonym knowledge enhanced reader for Chinese idiom reading comprehension. In: Proc. of the 28th Int'l Conf. on Computational Linguistics. 2020. 3684−3695.

[14] Wang X, Zhao H, Yang T, *et al*. Correcting the misuse: A method for the chinese idiom cloze test. In: Proc. of the Deep Learning Inside Out (DeeLIO): The 1st Workshop on Knowledge Extraction and Integration for Deep Learning Architectures. 2020. 1−10.

[15] Peters ME, Neumann M, Iyyer M, *et al*. Deep contextualized word representations. In: Proc. of the NAACL-HLT. 2018. 2227−2237.

[16] Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. In: Advances in Neural Information Processing Systems. 2017. 5998−6008.

[17] Lan Z, Chen M, Goodman S, *et al*. ALBERT: A lite BERT for self-supervised learning of language representations. In: Proc. of the ICLR. 2020.

[18] Qiao Y, Xiong C, Liu Z, *et al*. Understanding the behaviors of BERT in ranking. arXiv preprint arXiv: 1904.07531, 2019.

[19] Reimers N, Gurevych I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In: Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int'l Joint Conf. on Natural Language Processing (EMNLP-IJCNLP). 2019. 3973−3983.

[20] Jonz J. Cloze item types and second language comprehension. Language Testing, 1991, 8(1): 1−22.

[21] Tremblay A. Proficiency assessment standards in second language acquisition research: "Clozing" the gap. Studies in Second Language Acquisition, 2011, 33(3): 339−372.

[22] Cui Y, Liu T, Chen Z, *et al*. Consensus attention-based neural networks for Chinese reading comprehension. In: Proc. of the 26th Int'l Conf. on Computational Linguistics: Technical Papers (COLING 2016). 2016. 1777−1786.

[23] Cui Y, Liu T, Chen Z, *et al*. Dataset for the first evaluation on Chinese machine reading comprehension. In: Proc. of the 11th Int'l Conf. on Language Resources and Evaluation (LREC 2018). 2018. 2721−2725.

[24] Liu Y, Ott M, Goyal N, *et al*. Roberta: A robustly optimized Bert pretraining approach. In: Proc. of the ICLR. 2020.

[25] Yang Z, Dai Z, Yang Y, *et al*. Xlnet: Generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems. 2019. 5753−5763.

[26] Radford A, Narasimhan K, Salimans T, *et al*. Improving language understanding by generative pre-training. 2018. https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf

[27] Jiao X, Yin Y, Shang L, *et al*. Tinybert: Distilling BERT for natural language understanding. In: Proc. of the ICLR. 2020.

[28] Mozafari J, Fatemi A, Nematbakhsh MA. BAS: An answer selection method using BERT language model. Journal of Computing and Security, 2021, 8(2): 1−18.

[29] MacAvaney S, Yates A, Cohan A, *et al*. CEDR: Contextualized embeddings for document ranking. In: Proc. of the 42nd Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. 2019. 1101−1104.

[30] Xiong C, Dai Z, Callan J, *et al*. End-to-end neural ad-hoc ranking with kernel pooling. In: Proc. of the 40th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. 2017. 55−64.

[31] Ranasinghe T, Orasan C, Mitkov R. Semantic textual similarity with siamese neural networks. In: Proc. of the Int'l Conf. on Recent Advances in Natural Language Processing (RANLP 2019). 2019. 1004−1011.

[32] Mueller J, Thyagarajan A. Siamese recurrent architectures for learning sentence similarity. In: Proc. of the 30th AAAI Conf. on Artificial Intelligence. 2016. 2786−2792.

[33] Neculoiu P, Versteegh M, Rotaru M. Learning text similarity with siamese recurrent networks. In: Proc. of the 1st Workshop on Representation Learning for NLP. 2016. 148−157.

[34] Nicosia M, Moschitti A. Accurate sentence matching with hybrid siamese networks. In: Proc. of the 2017 ACM Conf. on Information and Knowledge Management. 2017. 2235−2238.

[35] Yin W, Schütze H. Convolutional neural network for paraphrase identification. In: Proc. of the 2015 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2015. 901−911.

[36] Severyn A, Moschitti A. Learning to rank short text pairs with convolutional deep neural networks. In: Proc. of the 38th Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval. 2015. 373−382.

[37] Dong L, Yang N, Wang W, *et al*. Unified language model pre-training for natural language understanding and generation. In: Advances in Neural Information Processing Systems. 2019. 13063−13075.

[38] Wang A, Cho K, Scholar CAG. BERT has a mouth, and it must speak: BERT as a Markov random field language model. In: Proc. of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation (ACL, 2019). 2019. 311−318.

[39] Zhang R, Yang XC, Ju SG, Liu NN, Xie ZW, Wang JY. Multi-level dynamic gated inference network for recognizing textual entailment. Journal of Sichuan University (Natural Science Edition), 2020, 57(2): 277−283 (in Chinese with English abstract).

[40] Gu YJ, Gui XL, Li DF, Shen Y, Liao D. Survey of machine reading comprehension based on neural network. Ruan Jian Xue Bao/Journal of Software, 2020, 31(7): 2095−2126 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6048.htm [doi: 10.13328/j.cnki.jos.006048]

**附中文参考文献**:

[39] 张芮, 杨煦晨, 琚生根, 刘宁宁, 谢正文, 王婧妍. 基于多层次动态门控推理网络的文本蕴含识别. 四川大学学报 (自然科学版), 2020, 57(2): 277−283.

[40] 顾迎捷, 桂小林, 李德福, 沈毅, 廖东. 基于神经网络的机器阅读理解综述. 软件学报, 2020, 31(7): 2095−2126. http://www.jos.org.cn/1000-9825/6048.htm [doi: 10.13328/j.cnki.jos.006048]

**琚生根**(1970−), 男, 博士, 教授, CCF 高级会员, 主要研究领域为数据挖掘, 自然语言处理, 知识图谱.

**孙界平**(1962−), 男, 副教授, 主要研究领域为智能信息处理, 智慧教育.

**黄方怡**(1996−), 女, 硕士生, 主要研究领域为数据挖掘, 自然语言处理, 知识图谱.