大规模知识图谱预训练模型及电商应用

陈华钧[1,2]，张文[3]，黄志文[4]，叶橄强[1]，文博[1]，张伟[2,4]
1. 浙江大学计算机科学与技术学院，浙江 杭州 310007；
2. 阿里巴巴–浙江大学前沿技术联合研究中心，浙江 杭州 311121；
3. 浙江大学软件学院，浙江 杭州 310007；4. 阿里巴巴集团，浙江 杭州 311121

摘要
近年来，知识图谱因具有以统一的方式组织数据等优势，被广泛应用于许多需要知识的任务，并且在电子商务领域大放光彩。然而知识服务通常需要烦琐的数据选择和知识注入模型的设计，这会给业务带来不良影响。为了更好地解决这一问题，提出了“ 预训练+知识向量服务 ”的模式，并设计了知识图谱预训练模型（PKGM），在不直接访问商品知识图谱中三元组数据的情况下，以知识向量的方式为下游任务提供知识图谱服务。在商品分类、同款商品识别和商品推荐等知识图谱下游任务中进行测试，实验结果表明，知识图谱预训练模型能够有效地提高每个任务的性能。

关键词
知识图谱；预训练；电商

中图分类号：TP183　　文献标识码：A　　doi: 10.11959/j.issn.2096-0271.2021028

Large scale pre-trained knowledge graph model and e-commerce application

CHEN Huajun[1,2], ZHANG Wen[3], WONG Chi-Man[4], YE Ganqiang[1], WEN Bo[1], ZHANG Wei[2,4]
1. College of Computer Science and Technology, Zhejiang University, Hangzhou 310007, China
2. Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, Hangzhou 311121, China
3. School of Software Technology, Zhejiang University, Hangzhou 310007, China
4. Alibaba Group, Hangzhou 311121, China

Abstract
In recent years, knowledge graph has been widely applied to organize data in a uniform way and enhance many tasks that require knowledge. For example, it has been widely used in the field of e-commerce. However, such knowledge services usually include tedious data selection and model design for knowledge infusion, which might bring inappropriate results. Thus, to solve this problem, the method of first pre-training then providing knowledge vector service was put forward, and a pre-trained knowledge graph model (PKGM) was proposed for our billion-

scale e-commerce product knowledge graph, providing item knowledge services in a uniform way for embedding-based models without accessing triple data in the knowledge graph. PKGM was tested in three knowledge-related tasks including item classification, same item identification, and recommendation. Experimental results show PKGM successfully improves the performance of each task.

Key words
knowledge graph, pre-training, e-commerce

1 引言

知识广泛存在于文本、结构化及多种模态的数据中。除了通过抽取技术[1]将知识从原始数据中萃取出来以支持搜索、问答、推理、分析等应用，另外一种思路是利用数据中本身存在的基本信号对隐藏的知识进行预训练（pre-training）。随着GPT[2]、BERT[3]、XLNet[4]等预训练语言模型在多项自然语言处理领域任务上刷新了之前的最好效果，预训练受到了各界的广泛关注。预训练的核心思想是预训练和微调，例如文本预训练一般包含两个步骤：首先利用大量的自然语言数据训练一个语言模型，获取文本中包含的通用知识信息；然后在下游任务微调阶段，针对不同的下游任务设计相应的目标函数，基于相对较少的监督数据进行微调，即可得到不错的效果。

受预训练语言模型的启发，笔者将预训练和微调的思想应用到大规模商品知识图谱的表示学习与业务应用中。在阿里巴巴电商平台，包含千亿级三元组和300多万条规则的商品知识图谱被构建起来，并为语义搜索、智能问答、商品推荐等众多下游业务任务提供知识图谱服务。通常知识图谱提供服务的方式是直接给出原始的三元组数据，这会导致以下问题：①针对不同任务反复地进行数据选择和查询，存在大量重复性工作；②下游任务需要针对自己的任务重新设计知识图谱算法，从头训练模型，由于图谱规模庞大，业务应用迭代周期过长，导致效率低下；③商品知识图谱本身的不完整性风险会导致误差传导；④直接提供原始三元组存在数据公平性风险和隐私风险。

为了避免这个问题，使商品知识图谱更方便、更有效地为下游任务提供服务，笔者提出了“预训练+知识向量服务”的模式，并设计了知识图谱预训练模型（pre-trained knowledge graph model，PKGM），在不直接访问商品知识图谱中三元组数据的情况下，以知识向量的方式为下游任务提供知识图谱服务。在商品分类、同款商品对齐以及商品推荐等多个下游任务上，验证了PKGM的有效性，其中在推荐任务上达到了平均6%的提升，同时还证明了在困难数据尤其是样本较少的数据上提升效果更明显。此外，在电商业务的真实实践中，知识图谱预训练模型进一步被应用到商品图片分类、用户点击预测等任务中，任务效果均获得了提升。知识图谱预训练对于具有亿级别节点量级的阿里巴巴商品知识图谱而言极为重要，因为这能够避免对庞大的商品知识图谱进行重复训练，从而更高效快速地为下游任务场景提供服务。

本文首先介绍了背景知识，包括预训练语言模型和结构化上下文信息等；然后分别介绍了商品知识图谱静态预训练模型和动态预训练模型，详细阐述了这两者的模

型结构和具体的先预训练再微调模式；之后介绍了知识图谱预训练模型在阿里巴巴电商场景的各种知识图谱任务中的实验结果和具体应用，包括商品分类、同款商品对齐和商品推荐等任务；最后对本文的工作进行了总结。

2 相关工作

2.1 预训练语言模型

人类的语言是高度抽象且富含知识的，文本数据只是人类大脑进行信息处理后的一个载体，因此沉淀的文本数据本身具有大量有价值的信息。互联网上沉淀了大规模的自然文本数据，基于这些海量文本，可以设计自监督训练任务，学习好的表示模型，然后将这些表示模型用于其他任务。基于这样的思想，最近几年提出的预训练语言模型（pre-trained language model）[2-4]在许多自然语言处理任务上被证明是有效的，并且能够显著提升相关任务的实验结果。

预训练语言模型可以学习通用的语言表示，捕捉语言中内含的结构知识，特别是针对下游任务标注数据量少的低资源场景，采用预训练+微调的模式，能够带来显著的提升效果。预训练语言模型的输入通常是一个文本序列片段，神经编码器会编码输入序列，每个输入单元都会编码得到对应的向量表示。区别于传统的word2vec词向量[5]，预训练得到的向量表示是上下文相关的，因为向量是编码器根据输入动态计算得到的，所以能够捕捉上下文语义信息。

以BERT模型[3]为例，预训练语言模型首先在大型数据集上根据一些无监督任务进行训练，包括下一个语句预测（next sentence prediction，NSP）任务和掩码语言模型（masked language model）任务，这个部分被称作预训练。接着在微调阶段，针对后续下游任务，例如文本分类、词性标注、问答系统等，基于预训练好的语言模型进行微调，使得BERT模型只需调整输入输出数据和训练部分参数，就可以在不同的任务上达到很好的效果。图1展示了BERT模型的预训练阶段的结构，以及在多个不同数据集和任务上进行微调的结构。BERT模型具有很好的兼容性、扩展性，并在多种自然语言处理下游任务上达到顶尖的实验效果。

预训练语言模型的优点总结如下：

- 对庞大的文本语料库进行预训练，学习通用语言表示形式，并帮助完成下游任务；
- 预训练提供了更好的模型初始化，通常可以带来更好的泛化性能，并加快目标任务的收敛速度；
- 可以将预训练视为一种正则化，以避免对小数据过度拟合。

2.2 结构化上下文信息

给定一个知识图谱$\mathcal{G}=\{\mathcal{E},\mathcal{R},\mathcal{T}\}$，其中$\mathcal{E}$表示实体（entity）的集合，$\mathcal{R}$表示关系（relation）的集合，$\mathcal{T}$表示三元组（triple）的集合。每个三元组$(h,r,t)\in\mathcal{T}$由头实体（head）、关系和尾实体（tail）构成，于是三元组集合可以表示为$\mathcal{T}=\{(h,r,t)\mid h,t\in\mathcal{E},r\in\mathcal{R}\}$，其中头实体$h$和尾实体$t$都属于集合$\mathcal{E}$，关系$r$属于集合$\mathcal{R}$。

对于某个实体而言，包含了其若干个三元组的集合往往隐含这个实体丰富的结构和语义特征，例如（姚明,性别,男性）、（姚明,职业,篮球运动员）、（中国篮球协会,主席,姚明）等三元组能很好地刻画“姚明”这个实体。类似地，对于某个特

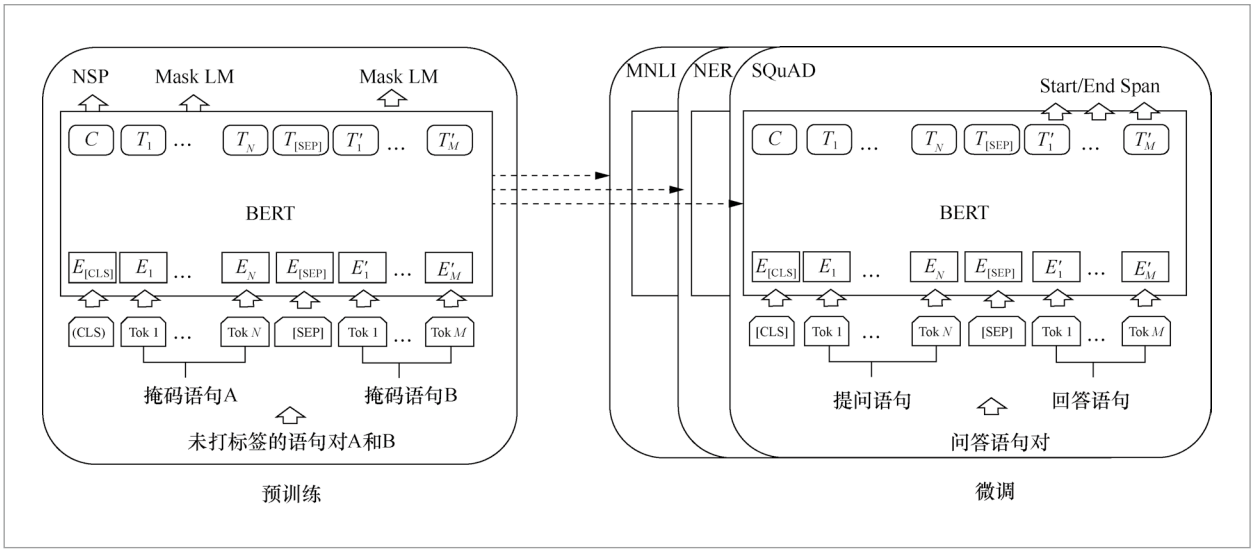


图1 BERT模型的预训练和微调过程的模型结构示意图

定的关系，知识图谱中也拥有丰富的包含了该关系的三元组集合。在此，可以将其定义为结构化上下文三元组（structure contextual triple）集合，简称为上下文三元组，并用$\mathcal{C}(x)$表示，其中x表示某个实体或者某个关系。因此不难看出，在知识图谱中有两种类型的上下文三元组：实体上下文三元组$\mathcal{C}(e)$和关系上下文三元组$\mathcal{C}(r)$。

实体上下文三元组$\mathcal{C}(e)$被定义为那些包含实体的三元组集合，无论实体e是某个三元组中的头实体还是尾实体，包含了e的三元组都可以被归入这个集合。用符号语言来表示就是：

$$\mathcal{C}(e)=\{(e,r,t)|(e,r,t)\in\mathcal{T},e,t\in\mathcal{E},r\in\mathcal{R}\}\cup\{(h,r,e)|(h,r,e)\in\mathcal{T},e,h\in\mathcal{E},r\in\mathcal{R}\} \quad (1)$$

类似地，关系上下文三元组$\mathcal{C}(r)$被定义为那些包含关系r的三元组集合，可以表示为：

$$\mathcal{C}(r)=\{(e_1,r,e_2)|(e_1,r,e_2)\in\mathcal{T},e_1,e_2\in\mathcal{E},r\in\mathcal{R}\} \quad (2)$$

为了更直观地展示上下文三元组在知识图谱中的结构，笔者画了一张简单的示意图来描述，如图2所示。图2中的圆代表实体，圆之间的短线代表关系。虚线框中的蓝色圆、橙色圆和粉色短线构成了一个特定三元组，分别代表头实体、尾实体和关系。对于头实体h（蓝色圆）来说，其上下文三元组$\mathcal{C}(h)$就是与蓝色圆相连的三元组，即图2中用蓝色短线连接起来的两两实体对组成的三元组再加上虚线框中的三元组得到的三元组集合。同理，尾实体t的上下文三元组$\mathcal{C}(t)$即图2中用橙色短线连接起来的三元组再加上虚线框中的三元组得到的三元组集合。而对于关系r的上下文三元组$\mathcal{C}(r)$，图2中用平行的、粉色的短线来表示同一种关系r，那么用这些粉色短线相连的三元组集合就是所期望的关系上下文三元组$\mathcal{C}(r)$。

3 商品知识图谱静态预训练模型

PKGM是基于预训练+知识向量服务的思路提出的，目的是在连续向量空间中提供服务，使下游任务通过嵌入计算得到必要的事实知识，而不需要访问知识图谱中的三元组。PKGM主要包含两个步骤，首先是商品知识图谱预训练，目标是使预

训练后的模型具有进行完整知识图谱服务的能力，其次是以统一的方式为下游任务提供知识向量服务。

具体来说，利用知识图谱中的结构化上下文信息进行预训练，从而为下游任务提供知识向量，利用知识图谱增强下游任务的效果。知识图谱静态预训练模型的静态体现在为下游任务提供预训练好的知识图谱嵌入向量表（embedding table），通过实体或者关系的ID能够直接查询并获取其对应的知识向量，该向量可以直接在下游任务中运用和参与计算。将预训练好的商品知识图谱模型作为知识增强任务的知识提供者，既能避免烦琐的数据选择和模型设计，又能解决商品知识图谱的不完整性问题。

3.1 PKGM预训练

预训练知识图谱模型中有两种常见的查询方式。

（1）三元组查询（triple query）

在给定头实体h、关系r的条件下，查询预测缺失的尾实体，于是该查询任务可以简写为$Q_{\text{tri}}(h,r)$。具体地，这个查询任务用SPARQL可以表示为：

```
SELECT ?x
WHERE {h r ?x}
```

（2）关系查询（relation query）

关系查询被用于查询一个项目是否具有给定的关系或属性。关系查询任务是针对给定的某个实体h，查询某个关系r是否与该实体相连，可以简写为$Q_{\text{rel}}(h,r)$。该查询任务用SPARQL可以表示为：

```
SELECT ?x
WHERE {h ?x ?y}
```

因此，考虑到商品知识图谱的不完整性问题，预训练知识图谱模型应该具有以下能力：

- 对于某一实体，显示该实体与其他实体之间是否存在某指定关系；
- 给定头实体和关系，查询对应的尾实体；
- 给定头实体和关系，如果查询不到尾实体，那么预测缺失的尾实体。

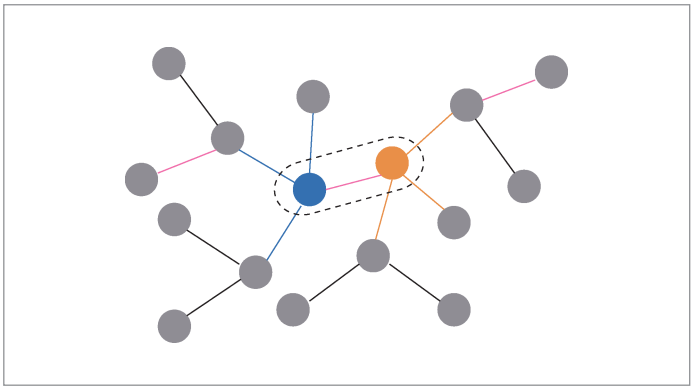


图 2 知识图谱中的上下文三元组

经过预训练，三元组查询模块和关系查询模块可以为任意给定的目标实体提供知识服务向量。更具体地说，一方面，关系查询模块可为目标实体提供包含不同关系信息的服务向量，如果目标实体具有或应该具有关系，则服务向量将趋于零向量；另一方面，三元组查询模块可为目标实体提供包含不同关系的尾实体信息的服务向量。

对于PKGM，在预训练知识图谱模型预训练好的基础上，通过向量空间计算为其他任务提供向量知识服务，具体如**图3**所示。在预训练阶段，首先会在10亿规模的商品知识图谱上对模型进行预训练，使预训练模型具备为三元组查询和关系查询提供知识信息的能力。在服务阶段，对于需要实体知识的任务，PKGM提供包含其三元组信息的嵌入向量，然后将其应用于基于嵌入的知识增强任务模型中。

3.2 PKGM查询模块

基于上述的关系查询和三元组查询两

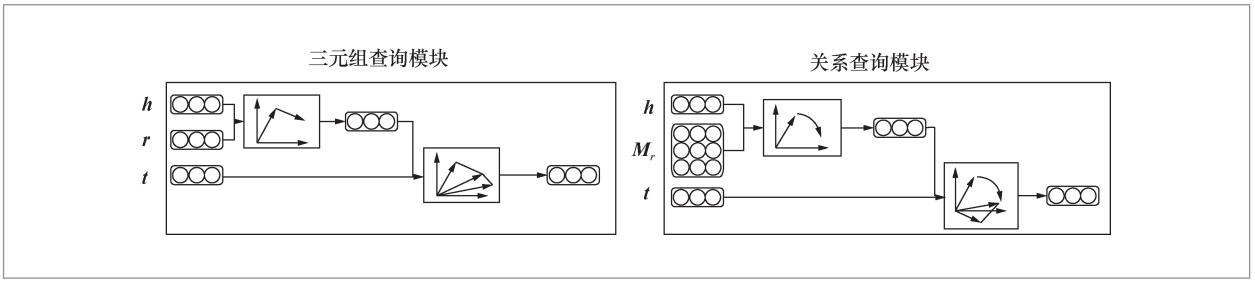


图 3 知识图谱静态预训练模型

种查询方式，可以构建对应的模块和评分函数用于模型预训练，因此PKGM主要包含两个查询模块：三元组查询模块和关系查询模块。

（1）三元组查询模块$\mathcal{M}_{\text{triple}}$

对于某个三元组查询需求$Q_{\text{triple}}(h,r)$，三元组查询模块$\mathcal{M}_{\text{triple}}$会生成一个服务向量，用于表示候选尾实体。这里可以认为，对于某个正确的三元组(h,r,t)，在向量空间中将头实体h和关系r进行组合，可以将其转化为尾实体t，并用评分函数$f_{\text{triple}}(h,r,t)$来表示。

自从知识图谱表示学习方法被提出，将实体和关系映射到向量空间的方法被大量的实验证明是有效的，因此在三元组查询模块$\mathcal{M}_{\text{triple}}$中，采用了表示学习中相对简单而有效的TransE模型[6]。每个实体$e\in\mathcal{E}$和每个关系$r\in\mathcal{R}$被编码为嵌入向量，那么头实体h、关系r和尾实体t对应的嵌入向量可以表示为$\boldsymbol{h}$、$\boldsymbol{r}$和$\boldsymbol{t}$。根据转换模型的假设，对于每个正确的三元组(h,r,t)，存在$\boldsymbol{h}+\boldsymbol{r}\approx\boldsymbol{t}$这样的关系，其中这些嵌入向量都是$d$维的向量，表示为$\boldsymbol{h}\in\mathbb{R}^d$、$\boldsymbol{r}\in\mathbb{R}^d$和$\boldsymbol{t}\in\mathbb{R}^d$。于是它们的评分函数可以表示为：

$$f_{\text{triple}}(h,r,t)=\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\| \tag{3}$$

其中，$\|\boldsymbol{x}\|$表示向量$\boldsymbol{x}$的L1范式[7]。对于正确的三元组，$\boldsymbol{h}+\boldsymbol{r}$的和向量越接近$\boldsymbol{t}$越好；对于错误的三元组，$\boldsymbol{h}+\boldsymbol{r}$要尽可能远离$\boldsymbol{t}$。

（2）关系查询模块$\mathcal{M}_{\text{relation}}$

设置关系查询模块主要是为了编码某个实体h是否存在与之相连的某种关系r，评分函数可以写为$f_{\text{rel}}(h,r)$，并且用零向量$\boldsymbol{0}$表示存在这样的关系。如果实体h与关系r相连，函数$f_{\text{rel}}(h,r)$接近零向量$\boldsymbol{0}$，即$f_{\text{rel}}(h,r)\approx\boldsymbol{0}$；如果该实体$h$与关系$r$不存在相连的情况，那么函数$f_{\text{rel}}(h,r)$尽可能远离零向量$\boldsymbol{0}$。在细节上，对于每一个关系$r$，还定义了转化矩阵$\boldsymbol{M}_r$，可以将向量$\boldsymbol{h}$转化为向量$\boldsymbol{r}$，这样的方式可以使得正确的三元组中的$\boldsymbol{M}_r\boldsymbol{h}$尽可能接近$\boldsymbol{r}$，即$\boldsymbol{M}_r\boldsymbol{h}-\boldsymbol{r}\approx\boldsymbol{0}$。于是，评分函数可以表示为：

$$f_{\text{rel}}(h,r)=\|\boldsymbol{M}_r\boldsymbol{h}-\boldsymbol{r}\| \tag{4}$$

3.3 PKGM知识图谱服务

经过上述两个查询模块的训练后，可以利用知识图谱预训练模型中已经训练好的模型参数（包括头实体h、关系r和尾实体t的嵌入向量、转化矩阵$\boldsymbol{M}_r$等），为特定任务提供两类对应的知识服务。

（1）三元组查询服务$\mathcal{S}_{\text{triple}}$

给定头实体h和关系r，三元组查询服务$\mathcal{S}_{\text{triple}}$可以给出预测的候选尾实体：

$$\mathcal{S}_{\text{triple}}(h,r)=\boldsymbol{h}+\boldsymbol{r} \tag{5}$$

如果在知识图谱数据集$\mathcal{K}$中的确存

在三元组，即$(h,r,t)\in\mathcal{K}$，那么$\mathcal{S}_{\text{triple}}(h,r)$会非常接近尾实体$t$的嵌入向量$\boldsymbol{t}$；如果数据集中不存在包含$h$和$r$的三元组，那么$\mathcal{S}_{\text{triple}}(h,r)$会给出一个实体向量表示最有可能的尾实体$t$。这本质上就是三元组补全，作为被广泛使用和验证的知识图谱补全[8]任务的具体形式。

（2）关系查询服务$\mathcal{S}_{\text{rel}}$

类似于上述的三元组查询服务，关系查询服务$\mathcal{S}_{\text{rel}}$能够提供一个向量来表示实体h是否存在包含关系r的三元组：

$$\mathcal{S}_{\text{rel}}(h,r)=\boldsymbol{M}_r\boldsymbol{h}-\boldsymbol{r} \qquad (6)$$

这里会有以下3种情况：一是实体h显式地与关系r相连，即存在同时包含h和r的三元组，那么此时$\mathcal{S}_{\text{rel}}$会接近$\mathbf{0}$；二是实体h隐式地与关系r相连，即不存在直接包含h和r的三元组，但是在真实情况中实体h能够与关系r相连，此时$\mathcal{S}_{\text{rel}}$仍然接近$\mathbf{0}$；三是实体h与关系r不相连，数据集中不包含这样的三元组，真实世界中也不存在，那么$\mathcal{S}_{\text{rel}}$应该远离$\mathbf{0}$。

上述三元组查询模块和关系查询模块各自的预训练和服务阶段的函数见**表1**。从**表1**可以更清晰地看出它们的差别和联系。

给定头实体h和关系r，通过知识图谱静态预训练模型的查询服务得到的知识有着非常显著的优势：一方面，可以通过向量空间的运算间接地得到对应的尾实体t，这使得查询服务能够独立于数据本身，从而更好地保护数据，尤其是隐私数据；另一方面，通过给定的头实体h和关系r输入对，经过两个查询服务能够分别得到两个向量，而不是未经处理的三元组数据本身，能够以更简单的方式应用在多种特定任务上。除此以外，这两个查询服务模块还能够通过推理计算得到知识图谱数据集暂未包含的、但真实情况中存在的三元组，能够有效地解决知识图谱不完整性[9]的问题。

表1 知识图谱静态预训练模型的预训练阶段和服务阶段的函数

模块	预训练阶段	服务阶段
三元组查询模块	$f_{\text{triple}}(h,r,t)=\|\|\boldsymbol{h}+\boldsymbol{r}-\boldsymbol{t}\|\|$	$\mathcal{S}_{\text{triple}}(h,r)=\boldsymbol{h}+\boldsymbol{r}$
关系查询模块	$f_{\text{rel}}(h,r)=\|\|\boldsymbol{M}_r\boldsymbol{h}-\boldsymbol{r}\|\|$	$\mathcal{S}_{\text{rel}}(h,r)=\boldsymbol{M}_r\boldsymbol{h}-\boldsymbol{r}$

3.4 PKGM在下游任务的应用

在知识图谱中，通过某个给定的实体的上下文信息，可以生成来自三元组查询模块和关系查询模块的服务向量序列，分别表示为$\boldsymbol{S}^e_{\text{triple}}=\left[\boldsymbol{S}^e_1,\boldsymbol{S}^e_2,\cdots,\boldsymbol{S}^e_k\right]$和$\boldsymbol{S}^e_{\text{rel}}=\left[\boldsymbol{S}^e_{k+1},\boldsymbol{S}^e_{k+2},\cdots,\boldsymbol{S}^e_{2k}\right]$，其类似于自然语言处理领域中描述文本或者特征标签的单词嵌入向量序列。其中，从某个实体e的上下文三元组(h,r,t)中抽取出所有关系r，并组成核心关系集合$\mathcal{R}_e$，k表示核心关系集合$\mathcal{R}_e$中的第k个关系。

基于目标实体生成包含知识图谱结构化信息的两种服务向量位于同一个统一的、连续的向量空间中，便于满足后续多种知识增强任务的应用需求。根据目标实体输入模型的嵌入向量个数，可以将下游基于嵌入向量的模型分为两类，分别是输入嵌入向量序列的模型和输入单个嵌入向量的模型。

（1）嵌入向量序列模型的输入是多个向量，往往包含较多的信息，例如由某个实体的文本描述或者标签特征生成的向量序列，可以表示为$\boldsymbol{E}^e=\left[\boldsymbol{E}^e_1,\boldsymbol{E}^e_2,\cdots,\boldsymbol{E}^e_N\right]$。考虑到序列模块能够自动捕捉元素之间的交互信息，类似于BERT模型中使用的双向Transformer[10]模块，因此可以将基于某个实体e得到的$\boldsymbol{S}^e_{\text{triple}}$和$\boldsymbol{S}^e_{\text{rel}}$这两种服务向量序列，直接拼接到原

输入序列的尾部，从而让原先的文本单词信息与知识图谱信息自动融合、充分交互学习。此时，模型的输入就变为 $\widehat{\boldsymbol{E}^e}=\left[\boldsymbol{E}_1^e,\boldsymbol{E}_2^e,\cdots,\boldsymbol{E}_N^e,\boldsymbol{S}_1^e,\boldsymbol{S}_2^e,\cdots,\boldsymbol{S}_k^e,\boldsymbol{S}_{k+1}^e,\boldsymbol{S}_{k+2}^e,\cdots,\boldsymbol{S}_{2k}^e\right]$，即先加入三元组查询模块的服务向量 $\boldsymbol{S}_{\text{triple}}^e$，再加入关系查询模块的服务向量序列 $\boldsymbol{S}_{\text{rel}}^e$，如图4所示。

（2）单个嵌入向量模型是指只输入一个有关目标实体e的嵌入向量的模型。这里的单个向量指的是实体e在潜在向量空间中对应的向量，并将其表示为$\boldsymbol{E}^e$，如图4的原始模型部分所示。

考虑到整个原始模型的输入只有一个向量，需要在模型原始的输入向量和融合了知识的服务向量之间取一个平衡，因此这里将 $\boldsymbol{S}_{\text{triple}}^e$ 和 $\boldsymbol{S}_{\text{rel}}^e$ 融合为一个向量。具体来说，需要将基于相同关系但来源于不同模块的两个向量 $\boldsymbol{S}_i^e$ 和 $\boldsymbol{S}_{i+k}^e$ 一起考虑，这里直接将它们拼接成新的向量 $\widehat{\boldsymbol{S}_i^e}$：

$$\widehat{\boldsymbol{S}_i^e}=[\boldsymbol{S}_i^e;\boldsymbol{S}_{i+k}^e] \tag{7}$$

其中，i是1到k之间的一个整数，即$i\in[1,k]$，而$[\boldsymbol{x};\boldsymbol{y}]$表示由向量$\boldsymbol{x}$和向量$\boldsymbol{y}$拼接成的新的服务整合向量。

然后，将生成的向量序列进一步整合、平均池化为单个向量：

$$\boldsymbol{S}^e=\frac{1}{k}\sum_{i\in[1,k]}\widehat{\boldsymbol{S}_i^e} \tag{8}$$

最后将充分融合了结构化知识信息的向量$\boldsymbol{S}^e$和原始的嵌入向量$\boldsymbol{E}^e$拼接成一个向量，如图5所示。

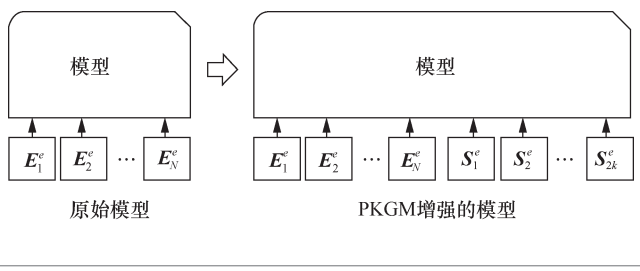


图4 将服务向量添加到嵌入向量序列模型尾部的示意图

4 商品知识图谱动态预训练模型

相对于静态预训练模型仅能为下游任务提供已经包含了结构化信息的嵌入向量表，知识图谱动态预训练模型能够根据下游任务的特征动态调整模型结构和模型参数，并根据下游任务对知识图谱中某些特征的倾向性进行微调和适配，具有更好的兼容性和扩展性。

4.1 上下文模块和整合模块

整个知识图谱动态预训练模型主要由上下文模块（contextual module，C-Mod）和整合模块（aggregation module，A-Mod）两部分构成。前者获取目标三元组的上下文三元组序列，并将每个上下文三元组的3个嵌入向量融合为一个向量；后者主要整合、交互学习上下文三元组向量序列，挖掘潜在的结构性特征，利用得分函数计算三元组分类任务的效果并用于训练。

（1）上下文模块

在上下文模块中，给定一个目标三元组 $\tau=(h,r,t)$，可以通过上述对结构化上下文信息的定义，得到该三元组的上下文三元组集合，即该目标三元组的头实体h、关系r和尾实体t各自的上下文三元组的并集。

$$\mathcal{C}(h,r,t)=\{\mathcal{C}(h)\cup\mathcal{C}(r)\cup\mathcal{C}(t)\} \tag{9}$$

然后，对于每一个上下文三元组c，例如目标三元组的第x个上下文三元组 $(h_x,r_x,t_x)\in\mathcal{C}(h,r,t)$，需要将原本对应的3个嵌入向量 $\boldsymbol{h}_x$、$\boldsymbol{r}_x$ 和 $\boldsymbol{t}_x$ 编码成一个向量 $\boldsymbol{c}_x$：

$$\boldsymbol{c}_x = \text{C-Mod}\left(\left\langle \boldsymbol{h}_x, \boldsymbol{r}_x, \boldsymbol{t}_x \right\rangle\right) \quad (10)$$

其中，$\left\langle \boldsymbol{h}_x, \boldsymbol{r}_x, \boldsymbol{t}_x \right\rangle$表示由向量$\boldsymbol{h}_x$、$\boldsymbol{r}_x$、$\boldsymbol{t}_x$组成的序列，并且满足$\boldsymbol{h}_x \in \mathbb{R}^d$、$\boldsymbol{r}_x \in \mathbb{R}^d$和$\boldsymbol{t}_x \in \mathbb{R}^d$。

对于C-Mod中的具体编码方式，可以有多种选择，比如简单的单层前馈神经网络。这里选择通过Transformer对向量序列进行学习和融合编码。将上下文三元组向量序列输入Transformer之前，需要在$\left\langle \boldsymbol{h}_x, \boldsymbol{r}_x, \boldsymbol{t}_x \right\rangle$序列前端加入特殊的标记[TRI]，生成得到一个新的序列$\left\langle [\text{TRI}], \boldsymbol{h}_x, \boldsymbol{r}_x, \boldsymbol{t}_x \right\rangle$，该序列对应的向量表示为$\left\langle \boldsymbol{k}_{[\text{TRI}]}, \boldsymbol{h}_x, \boldsymbol{r}_x, \boldsymbol{t}_x \right\rangle$，其中$\boldsymbol{k}_{[\text{TRI}]} \in \mathbb{R}^d$表示标记[TRI]对应的向量。在Transformer的最后一层，标记[TRI]对应位置上的向量为充分交互学习后融合了该三元组所有特征的向量，即向量$\boldsymbol{c}_x$。那么，头实体h、关系r和尾实体t各自的上下文三元组特征向量序列$\mathbf{seq}$可以表示为：

$$\begin{aligned} \mathbf{seq}_h &= \left\langle \boldsymbol{c}_h^1, \boldsymbol{c}_h^2, \cdots, \boldsymbol{c}_h^n \right\rangle \\ \mathbf{seq}_r &= \left\langle \boldsymbol{c}_r^1, \boldsymbol{c}_r^2, \cdots, \boldsymbol{c}_r^n \right\rangle \\ \mathbf{seq}_t &= \left\langle \boldsymbol{c}_t^1, \boldsymbol{c}_t^2, \cdots, \boldsymbol{c}_t^n \right\rangle \end{aligned} \quad (11)$$

其中，$\boldsymbol{c}_x^i$表示头实体h、关系r或者尾实体t中的某个$x \in \{h,r,t\}$的第i个上下文三元组特征向量，而n表示上下文三元组个数。

（2）整合模块

整合模块将目标三元组(h,r,t)的上下文三元组向量序列$\mathbf{seq}$整合编码输出为对应的整合向量$\boldsymbol{a}$，即：

$$\boldsymbol{a} = \text{A-Mod}\left(\mathbf{seq}_h, \mathbf{seq}_r, \mathbf{seq}_t\right) \quad (12)$$

为了增强目标三元组(h,r,t)中每个元素对应的上下文三元组在训练过程中的独立性，给每个三元组特征向量都加上一个段向量。具体地，总共有3种段向量：$\boldsymbol{s}_h$表示头实体h对应的上下文三元组的段向量，类似地，关系r和尾实体t对应的段向量为$\boldsymbol{s}_r$和$\boldsymbol{s}_t$。将上下文三元组特征向量加上段向量后生成新的特征向量：

$$\hat{\boldsymbol{c}}_x = \boldsymbol{c}_x + \boldsymbol{s}_x \quad (13)$$

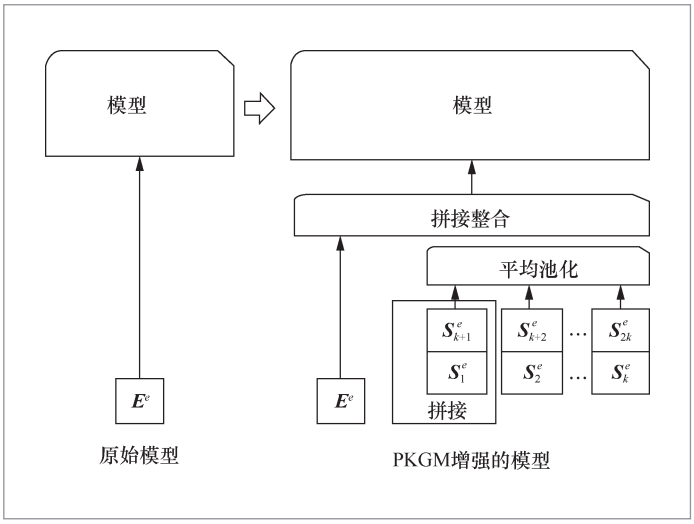


图 5 将服务向量添加到单个嵌入向量模型的示意图

其中，$x \in \{h,r,t\}$，那么三元组特征向量序列也更新为：

$$\widehat{\mathbf{seq}}_x = \left\langle \hat{\boldsymbol{c}}_x^1, \hat{\boldsymbol{c}}_x^2, \cdots, \hat{\boldsymbol{c}}_x^n \right\rangle \quad (14)$$

在将h、r、t的更新后的上下文三元组拼接特征向量序列输入整合模块之前，还需加入特定的标记来进一步区分它们。类似于上下文模块的[TRI]标签，这里引入[HEA]、[REL]和[TAI]标签，而它们对应的向量表示为$\boldsymbol{k}_{[\text{HEA}]}$、$\boldsymbol{k}_{[\text{REL}]}$和$\boldsymbol{k}_{[\text{TAI}]}$，将这3个向量分别加入头实体$h$、关系$r$、尾实体$t$的更新后的上下文三元组特征向量序列中，得到更新后的输入向量序列$\boldsymbol{i}$：

$$\boldsymbol{i} = \left\langle \boldsymbol{k}_{[\text{HEA}]}, \widehat{\mathbf{seq}}_h, \boldsymbol{k}_{[\text{REL}]}, \widehat{\mathbf{seq}}_r, \boldsymbol{k}_{[\text{TAI}]}, \widehat{\mathbf{seq}}_t \right\rangle \quad (15)$$

整合模块用另一个不同参数的多层双向Transformer来编码学习输入的向量序列$\boldsymbol{i}$，并在训练结束后，取出Transformer最后一层中[HEA]、[REL]和[TAI]对应的向量$\boldsymbol{a}_h$、$\boldsymbol{a}_r$和$\boldsymbol{a}_t$，这些向量表示经过充分整合交互学习后包含了丰富的知识图谱结构化信息的特征向量。

最后，将这3个向量拼接在一起，经过一个全连接层，融合为一个统一的整合向量：

$$\boldsymbol{a}_\tau = [\boldsymbol{a}_h;\boldsymbol{a}_r;\boldsymbol{a}_t]\boldsymbol{W}_{\text{agg}} + \boldsymbol{b}_{\text{agg}} \quad (16)$$

其中，$[\boldsymbol{x};\boldsymbol{y};\boldsymbol{z}]$表示将向量$\boldsymbol{x}$、向量$\boldsymbol{y}$和向量$\boldsymbol{z}$拼接在一起，$\boldsymbol{W}_{\text{agg}} \in \mathbb{R}^{3d\times d}$是该整合模块的权重矩阵，$\boldsymbol{b}_{\text{agg}} \in \mathbb{R}^d$是该整合模块的偏置向量。

（3）评分函数和损失函数

根据上述上下文模块和整合模块，对于目标三元组$\tau=(h,r,t)$，可以将评分函数定义为：

$$\boldsymbol{s}_\tau = f(h,r,t) = \text{softmax}(\boldsymbol{a}_\tau \boldsymbol{W}_{\text{cls}}) \quad (17)$$

其中，$\boldsymbol{W}_{\text{cls}} \in \mathbb{R}^{d\times 2}$是分类权重矩阵，而经过softmax操作之后得到的$\boldsymbol{s}_\tau \in \mathbb{R}^2$是二维向量，并且满足预测为正确的得分$s_{\tau 1}$和预测为错误的得分$s_{\tau 0}$之和为1，即：

$$s_{\tau 0} + s_{\tau 1} = 1 \quad (18)$$

给定构造好的正样本三元组集合$\mathbb{D}^+$和负样本三元组集合$\mathbb{D}^-$，可以基于评分$s_{\tau 0}$、$s_{\tau 1}$和标签l_τ进行交叉熵计算，得到损失函数$\mathcal{L}$：

$$\mathcal{L} = \sum_{\tau\in\mathbb{D}^+\cup\mathbb{D}^-} l_\tau \cdot \log(s_{\tau 0}) + (1-l_\tau)\cdot\log(s_{\tau 1}) \quad (19)$$

其中，$l_\tau \in \{0,1\}$表示三元组τ是否是正确的标签，若三元组是正确的，或者说τ是正样本三元组集合$\mathbb{D}^+$的其中一个元素$\tau\in\mathbb{D}^+$，那么标签l_τ为1，否则标签l_τ为0。

4.2 预训练阶段和微调阶段

类似于自然语言处理中的预训练模型，知识图谱动态预训练模型也包括预训练和微调两个阶段。预训练阶段会对海量的数据进行无监督学习，而微调阶段就相对轻量，一方面根据特定任务的输入输出等要求调整模型结构并进行适配，另一方面基于相对较小的特定数据集，在预训练阶段模型参数的基础上再次训练和微调，使之在特定任务上能更快地获得更好的效果。

（1）预训练阶段

在预训练阶段，动态预训练模型利用三元组分类任务进行训练。三元组分类任务是无监督任务，将数据库中存在的三元组视为正样本，同时通过随机替换实体或者关系生成原本数据集中不存在的三元组，并将这些三元组作为负样本，训练目标为二分类任务，即判断该三元组是否正确。对于每一个输入的三元组，预训练模型都获取其上下文三元组并进行采样、聚合，通过三元组分类任务训练学习得到其中的结构化信息。预训练阶段输入的是三元组，而用输出的嵌入向量来判断三元组是正确的还是错误的。如**图6**所示，给定一个目标三元组(h,r,t)，找到它的上下文三元组并通过上下文模块和整合模块将它们输入知识图谱动态预训练模型中，最后得到聚合输出表示向量。

预训练阶段需要用到尽可能大的甚至全量的知识图谱数据集，这样才能更好地学习到知识图谱中的深层次结构化信息，才真正能够帮助下游任务。例如，BERT模型[3]使用了包含8亿个单词的BooksCorpus[11]数据集和25亿个单词的Wikipedia[12]数据集进行预训练，然后两个大小不同的模型（包括1.1亿个参数的$\text{BERT}_{\text{BASE}}$模型和3.4亿个参数的$\text{BERT}_{\text{LARGE}}$模型）分别在16个张量处理单元（tensor processing unit，TPU）上训练了4天才完成。

对于知识图谱的数据集，难以构造横跨多个不同知识图谱数据集的全量数据集，比如FB15k[6]、WN18、YAGO[13]等，甚至基于它们各自最原始的数据集Freebase[14]和WordNet[15]等都难以直接合并成一个数据集。这是因为每个数据集中的实体和关系都是以不同的文本和组织方式构建的，很难直接建立起不同数据集之间的联系。然而，笔者还是找到了合适的方法去间接构造一个足够大且丰富的知识图谱预训练数据集：利用包含真实世界描述的WordNet数据集（其中包含了名

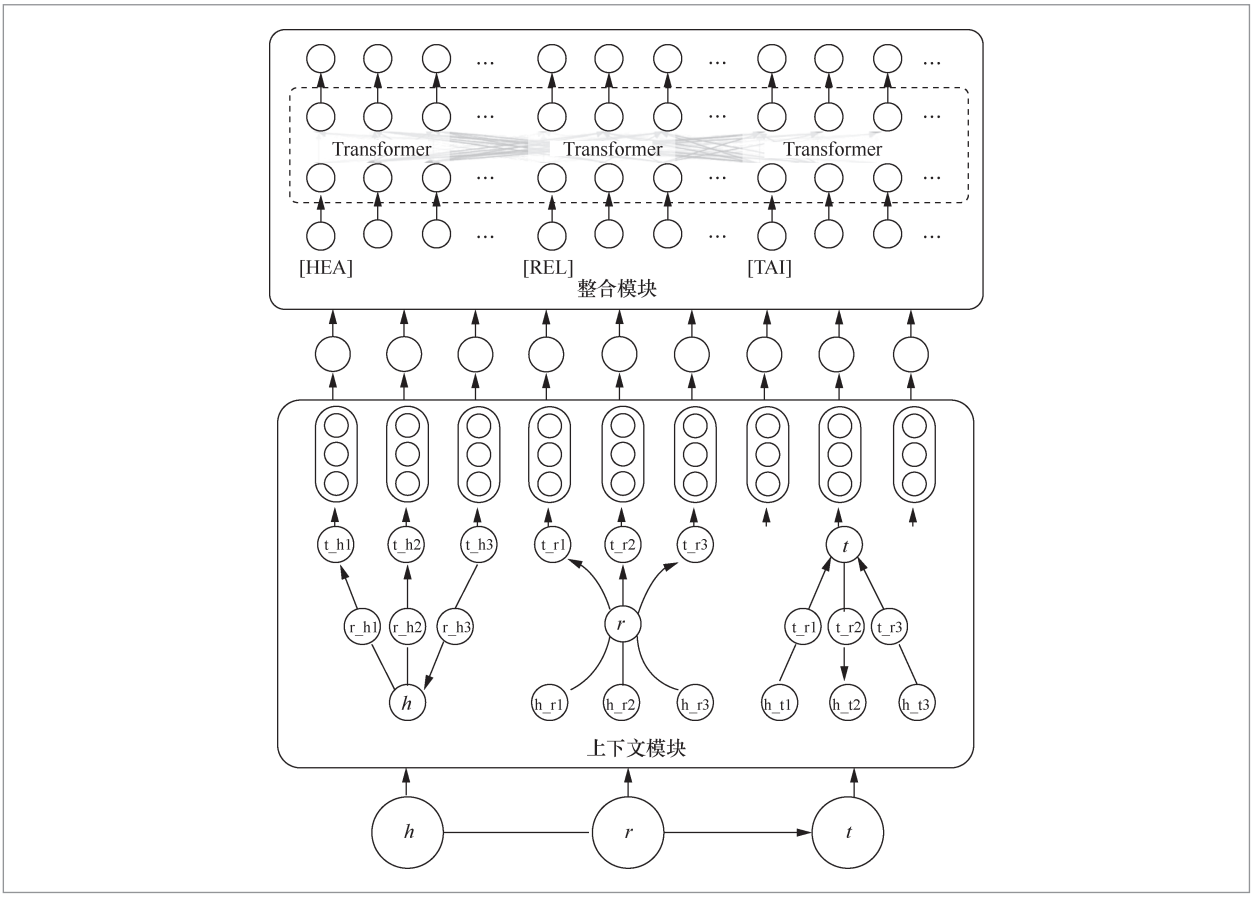


图 6 动态预训练模型结构示意图

词、动词、形容词和副词等词性的单个词语，最大程度地反映了真实场景和语言习惯），建立不同知识图谱数据集关联的桥梁。而其他知识图谱数据集中的实体或者关系往往是由多个单词构成的，可以利用类似于短语包含某些单词的关系构建起实体与实体之间的联系。而在阿里巴巴电商知识图谱上，可以直接利用海量商品的属性和属性值等三元组，用预训练模型学习商品知识图谱的结构化信息。商品知识图谱足够大，具有10亿节点级别的商品和千亿级别的三元组，可以支撑预训练的数据需求，并且能够在下游任务中很好地发挥出预训练模型的作用。

（2）微调阶段

在微调阶段，模型的输入输出结构会根据具体的任务和数据集特性进行调整，同时将调整后的模型在特定数据集上进行微调训练，最后得到符合该特定任务需求并有不错效果的模型，如**图7**所示。

例如，实体对齐任务的目标是在真实世界中找到本质上是同一个事物或者事件而在输入的知识图谱数据集中有两种或者多种表示的实体，比如中文语义下的实体对(漂亮的,美丽的)、(睡觉,睡眠)和(狗,犬)等，表达的是相同含义却有不同的文字描述。在这个实体对齐任务上，模型的输入从原来的三元组(h,r,t)变为头尾实体对(h,t)，即删除了关系r这一项元素，剩下前后两个实体，进一步来说，这两个实体就是判断是否具有相同含义的实体对(e_1, e_2)。相应地，模型的输出部分也需要替换为描述

两个实体是否对齐的训练函数，具体如**图7（c）**所示。

又如实体类型预测任务，需要找到某个实体所属的类别，而这个类别是存在于知识图谱中的另一个实体，即预测(实体,实体类型)中缺失的实体类型，比如(老虎,猫科动物)、(中文,语言)和(T细胞,淋巴细胞)等实体类型对。类似于上述的实体对齐任务，实体类型预测任务中的模型输入也变为一个实体对，而输出部分是判断这个实体类型对是否正确的评分函数，如**图7（b）**所示。

5 应用实践和实验结果

在删除了出现次数较低的实体后的商品知识图谱上对PKGM进行预训练。预训练完成后，在多个对知识图谱有需求的下游任务进行效果验证，不仅包括商品分类、同款商品对齐、商品推荐等以图谱数据服务为基础的任务，还包括可以利用知识图谱增强效果的一些NLP任务，例如商品实体识别、商品属性补齐和关系抽取、商品标题生成等。这里重点介绍了商品分类、同款商品对齐、商品推荐3个任务。在实验中，将只提供三元组服务向量的标记为PKGM-T，只提供关系服务向量的标记为PKGM-R，两类服务向量都提供的标记为PKGM-all。

5.1 基于知识图谱预训练的商品分类

亿级的商品数据组织依赖于良好的类目体系，因此商品分类在阿里巴巴电商平台是一项常见且重要的任务，其目标是将给定的商品分类到类目中对应的类别。商品的标题往往包含了密集的商品信息，因此也常被用作商品分类的原始信息，基于商品标题，商品分类任务可对应为文本多分类任务[16]，鉴于目前语言预训练模型在文本分类任务上取得了很好的效果，这里将BERT作为基准模型。**图8（a）**展示了基准模型BERT，**图8（b）**展示了PKGM增强的BERT模型，这里采用了为序列嵌入向量模型提供知识图谱服务的方式。

从阿里巴巴电商真实场景中抽取出1 293个类别和这些类别下的商品，生成正样本和负样本为1:1的数据集，具体见**表2**。为了更好地证明结合文本的知识图谱预训练模型的能力，在数据准备过程中将每个类别的实例（商品）限制在100个以下，展现出较少的训练样本数据情况下下游任务的实验效果。为此还特意生成每个类别不同实例个数的3种数据集dataset-20、dataset-50

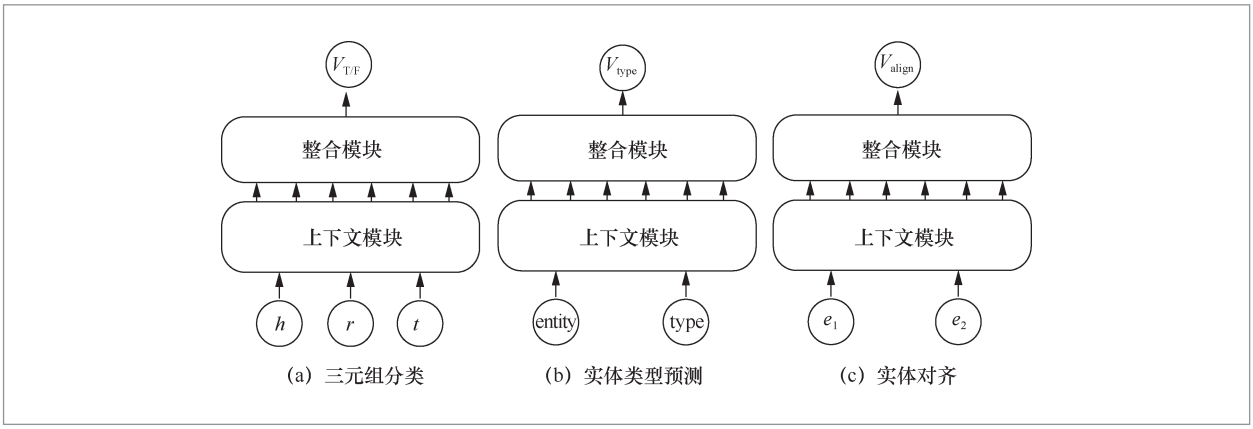


图 7 在微调阶段，图中 3 个模型结构对应于 3 个不同的训练任务

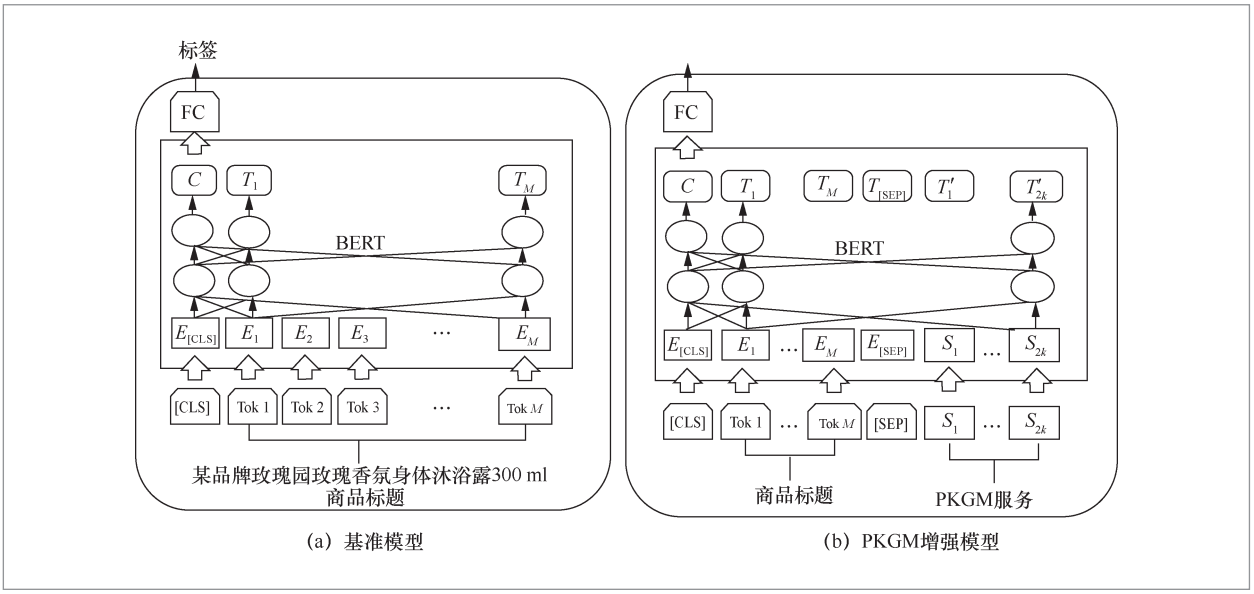


图 8 商品分类任务模型

和dataset-100，分别表示每个类别只有20、50或者100个实例个数的数据集。**表2**中的#Train、#Test和#Dev分别表示由三元组构成的训练集、测试集和验证集。

在实验中，采用预训练语言模型$BERT_{BASE}$在中文语言数据集上的训练模型作为基准模型，其中包含12层Transformer、12个注意力头（attention head）和大小为768的向量维度。类似于BERT模型，在输入数据序列前端加上特殊的分类符[CLS]，其在最后一层模型对应位置的嵌入向量用于表示整合了这个输入序列的向量。这里将整个序列长度固定为128，包含一个[CLS]分类符和长度为127的标题序列，若原始标题文本长度不够则补零，若超出则截取最前面127个字符序列。

基于知识图谱预训练得到的服务向量，可以得到PKGM增强的模型$BERT_{PKGM-all}$，具体步骤为：将基准模型BERT输入序列的最后2k个向量替换为k个关系查询模块的服务向量序列和k个三元组查询模块的服务向量序列，然后进行微调阶段的训练。类似地，只将输入序列中最后k个向量替换为k个三元组查询模块服务向量序列的模型，写为$BERT_{PKGM-T}$，而替换为k个关系查询模块服务向量的模型，写为$BERT_{PKGM-R}$。

表 2 商品分类任务的数据集

数据集	#Category	#Train	#Test	#Dev
数量/个	1 293	169 039	36 225	36 223

在训练批量大小（batch size）为32、学习率（learning rate）为$2e^{-5}$的参数条件下，对PKGM训练了3个轮次（epoch），其中来自知识图谱预训练的服务向量是固定不变的，而BERT模型中的相关参数会在训练中被调整优化，最终得到的商品分类任务实验结果见**表3**。**表3**给出了商品分类的预测准确率（accuracy，AC）和前k个预测值的命中率Hit@k，其中Hit@k表示在所有的测试数据集中预测正确的类别在所有商品类别的预测值序列中排名前k个的百分比，其中k包括1、3和10这3个候选值。

从**表3**可以看到，在预测准确率和Hit@k指标上，融入了知识服务向量的模型$BERT_{PKGM}$在这3个数据集上都要优于

表 3 商品分类任务的结果

数据集	模型	Hit@1	Hit@3	Hit@10	AC
dataset-100	BERT	71.03%	84.91%	92.47%	71.52%
	$BERT_{PKGM-T}$	71.26%	85.76%	93.07%	72.14%
	$BERT_{PKGM-R}$	71.55%	85.43%	92.86%	**72.26%**
	$BERT_{PKGM-all}$	**71.64%**	**85.90%**	**93.17%**	72.19%
dataset-50	BERT	60.98%	78.99%	89.21%	59.06%
	$BERT_{PKGM-T}$	61.47%	79.04%	90.08%	62.74%
	$BERT_{PKGM-R}$	61.52%	**80.09%**	**90.39%**	**62.98%**
	$BERT_{PKGM-all}$	**61.54%**	79.89%	90.36%	62.71%
dataset-20	BERT	30.26%	46.97%	68.12%	28.90%
	$BERT_{PKGM-T}$	30.65%	47.80%	67.40%	29.62%
	$BERT_{PKGM-R}$	31.47%	**50.69%**	69.07%	30.63%
	$BERT_{PKGM-all}$	**32.09%**	50.19%	**70.07%**	**30.91%**

基准模型BERT。具体来说，一方面，同时融入了两种服务向量的$BERT_{PKGM-all}$模型在Hit@1指标上都有最好的效果；另一方面，在Hit@3、Hit@10和预测准确率这3个指标上，$BERT_{PKGM-all}$和$BERT_{PKGM-R}$这两个模型有较好的效果，而且它们中的一个能达到特定条件下最好的实验效果。这也证明了知识图谱预训练模型和提供相应的查询服务向量的有效性，并且其中关系查询模块往往发挥着比三元组查询模块更重要的作用。

当然在一定程度上，$BERT_{PKGM-R}$在不少时候比$BERT_{PKGM-all}$有更好的效果，打破了人们对“有更多知识图谱特征向量往往能有更好效果”的传统认知。这很可能是因为在商品分类任务上，那些被三元组服务向量序列替换掉的文本序列比替换它们的三元组服务向量序列更重要，在这些特定指标上，文本序列本身比判断三元组是否成立的信息更有价值。

5.2 基于知识图谱预训练的同款商品对齐

阿里巴巴电商平台上的商品数量数以亿计，给商品管理带来了巨大挑战，其中一个挑战就是同款商品挖掘。商品在商品知识图谱中以实例的形式存在，因此商品同款的本质是商品对齐任务，其目标是找到本质上是相同的，但在平台上的拥有不同商品ID的商品，这种同款商品一般被定义为同一个产品。产品指由相同厂商生产的、具有相同款式相同属性而又与具体销售店铺无关的物品，商品定义为不同销售店铺或者商家在平台上设置上传并销售的、可能是相同产品也可以是不同产品的物品，每个商品都有自己唯一的ID。比如，平台上绿色、256 GB容量的某品牌某型号手机有很多，由不同商家售卖，因此这些商品在电商平台上被存储为不同的商品，但从产品的角度或者销售的商品本身而言，它们是同一款产品。检测两个商品是否是同一产品的任务在阿里巴巴电商场景的日常业务中非常重要。例如，用户想购买一台绿色、256 GB容量的某品牌某型号手机，在搜索框输入具体商品的需求后，能够显示所有属于该产品的商品，有助于用户方便、深入地比较销售价格及售后服务等。更重要的是，产品的数量远小于商品数量，因此从产品的角度来组织商品有助于减少数据管理和挖掘的工作量。

正因为商品来源不同，对齐同款商品成为提高数据有效性的重要任务，其目标是判断给定的两个商品是否为同款商品。商品信息用标题表示，这个任务可对应于同义句识别，基准模型的输入类似于BERT模型的下游任务，分别输入两个句子的文本，然后做分类任务，具体细节与商品分类任务相似，如**图9(a)**所示；而在PKGM增强的BERT模型中，在每个句子文本序列后面分别加入[SEP]标签和与该商品对应的包含知识信息的服务向量序列，如**图9(b)**所示。

从商品知识图谱中抽取出女装衬衫(category-1)、头发饰品(category-2)和儿童袜类(category-3)这3个类别的三元组集合，作为商品对齐任务的实验数据集。

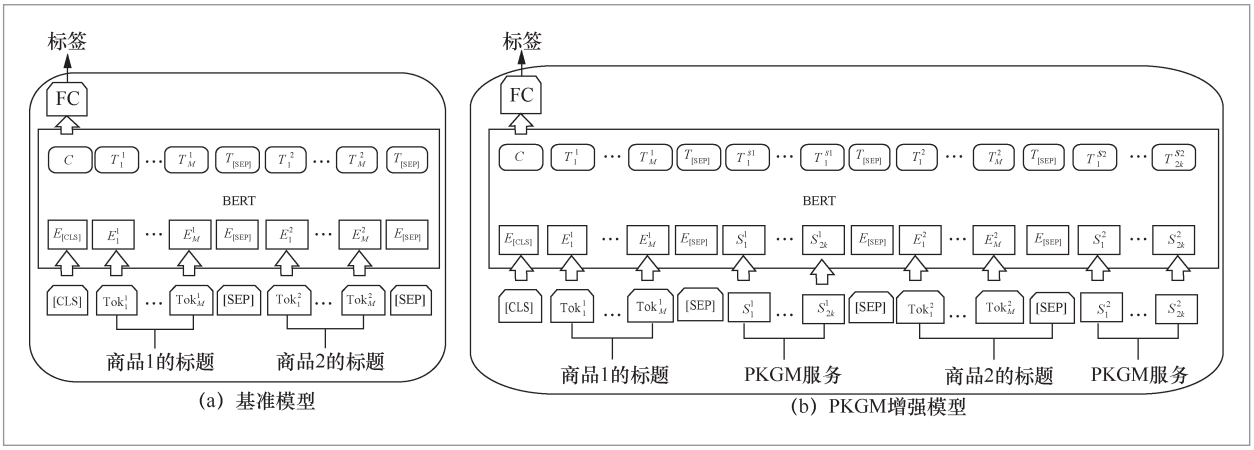


图 9 商品对齐任务模型

每个数据集中都有上千个样本，每个样本中包含两个商品各自的商品标题和判断这两者是否对齐的标签，标签1表示两个商品对齐，而标签0表示两个商品没有对齐。将包含正负样本的所有样本集合按照7:1.5:1.5的比例分配成训练集#Train、测试集#Test-C和验证集#Dev-C，用于训练和同款商品分类指标的测量，但是为了测试前k个预测值的命中率Hit@k，需要从中提取出只包含正样本的数据集并排序，因此得到相应的测试集#Test-R和验证集#Dev-R，具体见**表4**。

类似于第5.1节中的商品分类任务，同款商品对齐任务将BERT作为基准模型，并且输入格式与商品分类任务相同，只是在输入数据上略有不同。每个输入数据由两个商品的标题文本嵌入向量序列组成，在整个序列的第一个位置加入[CLS]标签，在每个标题序列后加入[SEP]标签，并用类似于第5.1节的方法归一化商品标题长度。**表5**展示了商品对齐任务的Hit@k结果，在3个数据集上，$\text{BERT}_{\text{PKGM-all}}$模型的Hit@3和Hit@10指标都优于基准模型BERT，并且在category-2和category-3这两个数据集上的所有指标上都有最好的效果，展示了知识图谱预训练模型对商品对齐任务的有效性，并且提升了预测准确率。在数据集category-1的Hit@1指标上，基准模型BERT略优于$\text{BERT}_{\text{PKGM-all}}$模型，很可能是因为该类别的数据集较大。可以说，足够的标题文本序列对商品对齐任务是有帮助的，而知识图谱预训练模型在少样本数据集上能发挥出更大的作用。

同时，比较了结合知识图谱预训练模型产生的两种查询服务向量不同组合方式的实体对齐任务的预测准确率，具体见**表6**。

表 4 商品对齐任务的数据集

对比项	#Train	#Test-C	#Dev-C	# Test-R	#Dev-R
category-1/个	4 731	1 014	1 013	513	497
category-2/个	2 424	520	519	268	278
category-3/个	3 968	852	850	417	440

表 5 商品对齐任务的 Hit@k 指标的实验结果

数据集	模型	Hit@1	Hit@3	Hit@10
category-1	BERT	**65.06%**	76.06%	86.68%
	$\text{BERT}_{\text{PKGM-all}}$	64.75%	**77.50%**	**87.43%**
category-2	BERT	65.86%	78.07%	87.59%
	$\text{BERT}_{\text{PKGM-all}}$	**66.13%**	**78.19%**	**87.96%**
category-3	BERT	49.64%	66.18%	82.37%
	$\text{BERT}_{\text{PKGM-all}}$	**50.60%**	**67.14%**	**83.45%**

表 6 商品对齐任务的准确率指标结果

模型	category-1	category-2	category-3
BERT	88.94%	89.31%	86.94%
$BERT_{PKGM-T}$	88.65%	89.89%	87.88%
$BERT_{PKGM-R}$	89.09%	89.60%	87.88%
$BERT_{PKGM-all}$	**89.15%**	**90.08%**	**88.13%**

从**表6**可以很明显地看出，$BERT_{PKGM-all}$模型在3个数据集上都有最好的效果，有效提升了实体对齐任务的预测能力。

5.3 基于知识图谱预训练的商品推荐

商品推荐是除搜索外将适合的商品呈现在用户面前的重要方式，因此商品推荐也是一项重要的任务。针对预测商品和用户交互的下游任务进行实验，实验中将用户和商品的交互记录图作为输入并预测潜在的交互，这是典型的链接预测任务。采用神经协同过滤（neural collaborative filtering，NCF）算法[17]作为基准模型。广义矩阵分解（generalized matrix factorization，GMF）层和多层感知机（multi-layer perceptron，MLP）层能够对用户和商品的交互数据进行建模，其中广义矩阵分解层使用线性核来模拟潜在的特征交互，而多层感知机层使用非线性核函数从数据中学习交互函数。**图10（a）**展示了基准模型NCF，**图10（b）**展示了PKGM增强的NCF模型，这里采用为单个嵌入向量模型提供知识图谱服务的方式。

在从淘宝真实记录中采样得到的数据集上进行测试，**表7**展示了商品推荐任务的具体细节，其中包括两万多个用户（#Users）和3万多个商品（#Items），以及44万条用户-商品交互记录（#Interactions）。数据集中保证每个用户的交互记录至少有10条，不至于太过稀疏。

基于上述数据集进行实验，实验中采用“leave one out”进行推荐效果评估。对于每个用户的数据，将其最近一次的交互作为测试集，其余作为训练集。在测试过程中，随机采样100个未观测到的负样本，将这些负样本同真正的测试正样本进行排序，通过这样的方式统计排名前k个命中率HR@k以及归一化累计增益NDCG@k，并将其作为评估指标，其中k的取值范围是{1,3,5,10,30}，对于每一个测试用户，分别计算这两种评价指标，并求出其在所有测

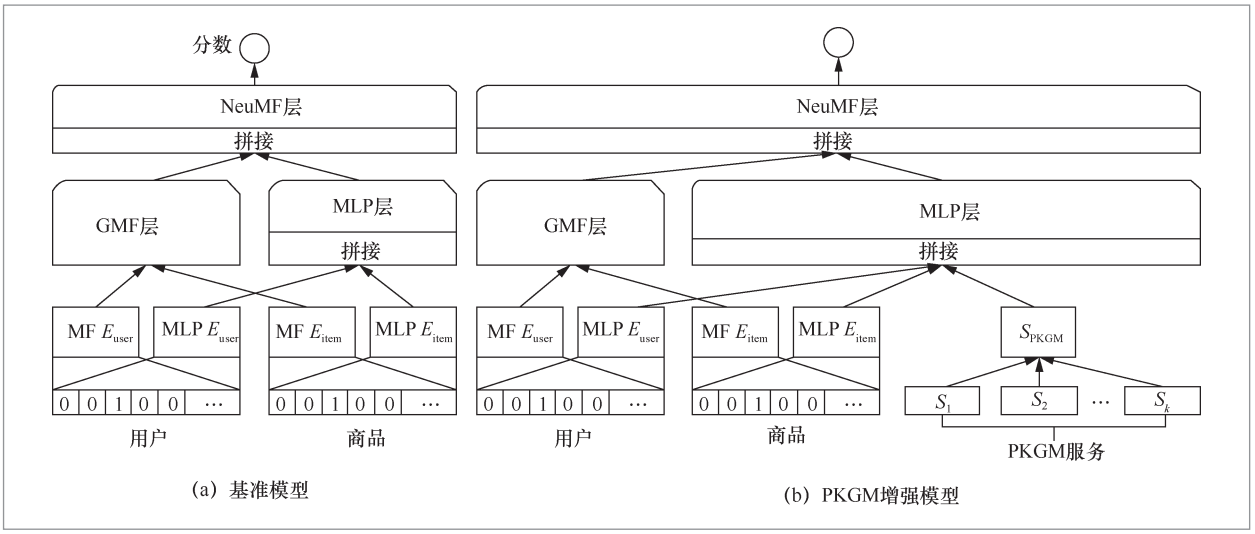


图 10 商品推荐任务的模型

试用户上的均值作为最终评估指标。

为每个用户随机采样一个正样本交互作为验证集，以求得模型的最优超参数。对于广义矩阵分解层，用户嵌入和商品嵌入的维度都为8。在多层感知机层中，用户嵌入和商品嵌入的维度设置为32。对于基准模型和知识增强模型，3个隐藏层的维度依次为32、16和8。对于知识增强模型，输入增强的特征，并与多层感知机层的用户嵌入和商品嵌入进行拼接，并且为广义矩阵分解层和多层感知机层中的用户嵌入和商品嵌入加了L2正则化惩罚，惩罚系数选择为0.001。学习率设置为0.000 1，预测层的维度为16，预测层的输入是由两个8维向量拼接而成的，分别是广义矩阵分解层的输出和多层感知机层的输出。在实验中，采用的负采样比例为4，即为每个正样本采样4个负样本。为了更加简洁和有效，基线模型和知识增强模型均采用了非预训练版本的神经协同过滤模型。

最终的实验结果见**表8**，有NCF_{PKGM-T}标识的神经协同过滤模型表示仅加入了基于知识图谱预训练的三元组查询服务向量的知识增强模型，有NCF_{PKGM-R}标识的神经协同过滤模型表示仅加入了关系查询服务向量的知识增强模型，有$NCF_{PKGM-all}$标识的神经协同过滤模型表示融合了以上两种服务向量的知识增强模型。

从**表8**可以看出：首先，相对于基准模型来说，所有的知识增强模型在所有评价指标上均有提升效果。对于NCF_{PKGM-T}模型来说，它在HR@*k*指标上比基线模型平均提升了0.37%，而在NDCG@*k*指标上比基线模型平均提升了0.002 3。对于NCF_{PKGM-R}模型来说，它在HR@*k*指标上比基线模型平均提升了3.66%，而在NDCG@*k*指标上比基线模型平均提升了0.034 3。对于$NCF_{PKGM-all}$模型来说，它在HR@*k*指标上比基线模型平均提升了3.47%，而在NDCG@*k*指标上比基线模型平均提升了0.032 4。提升的结果证明了预训练的知识增强模型能够有效提供仅从用户-商品交互不能分析出的额外信息，从而提升了下游任务（如电商推荐任务）的效果。

表 7 商品推荐任务数据集

数据集	#Items	#Users	#Interactions
商品推荐任务	37 847	29 015	443 425

其次，NCF_{PKGM-R}模型的效果要优于NCF_{PKGM-T}模型的效果，说明预训练模型提供的不同特征的侧重点不同。因此在商品推荐任务中，NCF_{PKGM-R}模型提供的特征相比于NCF_{PKGM-T}模型提供的特征要更加有用，这很有可能是因为描绘用户商品交互时，属性关系往往要比属性实体更有效。

6 结束语

将知识预先训练好，然后融入各种深度模型或下游任务中或许是未来知识图谱数据应用方式的一种新的发展趋势。本文介绍了大规模知识图谱预训练及电商应用的初步实践，通过三元组和关系模块的设

表 8 商品推荐任务的实验结果

模型	HR@1	HR@3	HR@5	HR@10	HR@30	NDCG@1	NDCG@3	NDCG@5	NDCG@10	NDCG@30
NCF	27.94%	44.26%	52.16%	62.88%	81.26%	0.279 4	0.374 4	0.406 9	0.441 5	0.485 3
NCF_{PKGM-T}	27.96%	44.83%	52.43%	63.51%	81.62%	0.279 6	0.377 8	0.409 1	0.444 9	0.488 0
NCF_{PKGM-R}	**31.01%**	**47.99%**	**56.10%**	**66.98%**	**84.73%**	**0.310 1**	**0.409 1**	**0.442 4**	**0.477 7**	**0.520 0**
$NCF_{PKGM-all}$	30.76%	47.92%	55.60%	66.84%	84.71%	0.307 6	0.407 9	0.439 5	0.475 8	0.518 5

计，PKGM模型具有在向量空间为下游任务提供知识图谱服务的能力，具有较好的知识图谱数据保护性以及对下游任务的兼容性，同时解决了知识图谱本身的不完整性问题。3种类型的知识图谱下游任务实验证明了PKGM模型能够提高这些任务的性能。在未来的工作中，希望将PKGM模型应用到更多的下游任务中，并探索应用服务向量的其他候选方法。

参考文献:

[1] PECHSIRI C, PIRIYAKUL R. Explanation knowledge graph construction through causality extraction from texts[J]. Journal of Computer Science and Technology, 2010, 25(5): 1055–1070.

[2] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre–training[Z]. 2018.

[3] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre–training of deep bidirectional transformers for language understanding[J]. arXiv preprint, 2018, arXiv:1810.04805.

[4] YANG Z L, DAI Z H, YANG Y M, et al. XLNet: generalized autoregressive pretraining for language understanding[J]. arXiv preprint, 2019, arXiv:1906.08237.

[5] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint, 2013, arXiv:1301.3781.

[6] BORDES A, USUNIER N, GARCIA–DURÁN A, et al. Translating embeddings for modeling multi–relational data[C]// Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS). [S.l.]: Curran Associates Inc., 2013: 2787–2795.

[7] PARK M Y, HASTIE T. L1–regularization path algorithm for generalized linear models[J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2007, 69(4): 659–677.

[8] JI S X, PAN S R, CAMBRIA E, et al. A survey on knowledge graphs: Representation, acquisition and applications[J]. arXiv preprint, 2020, arXiv:2002.00388.

[9] MELO A, PAULHEIM H. Detection of relation assertion errors in knowledge graphs[C]// Proceedings of the 2017 Knowledge Capture Conference. New York: ACM Press, 2017: 1–8.

[10] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. arXiv preprint, 2017, arXiv:1706.03762.

[11] ZHU Y K, KIROS R, ZEMEL R, et al. Aligning books and movies: towards story–like visual explanations by watching movies and reading books[C]// Proceedings of the 2015 IEEE International Conference on Computer Vision. Piscataway: IEEE Press, 2015: 19–27.

[12] VÖLKEL M, KRÖTZSCH M, VRANDECIC D, et al. Semantic Wikipedia[C]// Proceedings of the 15th International Conference on World Wide Web. New York: ACM Press, 2006: 585–594.

[13] SUCHANEK F M, KASNECI G, WEIKUM G. YAGO: a core of semantic knowledge[C]// Proceedings of the 16th International Conference on World Wide Web. New York: ACM Press, 2007: 697–706.

[14] BOLLACKER K, EVANS C, PARITOSH P, et al. Freebase: a collaboratively created graph database for structuring human knowledge[C]// Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data. New York: ACM Press, 2008: 1247–1250.

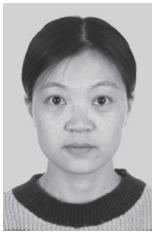
[15] MILLER G A, BECKWITH R, FELLBAUM C, et al. Introduction to WordNet: an on–line lexical database[J]. International Journal of Lexicography, 1990, 3(4): 235–244.

[16] SINOARA R A, CAMACHO–COLLADOS J, ROSSI R G, et al. Knowledge–enhanced document embeddings for text classification[J]. Knowledge–Based Systems, 2019, 163: 955–971.

[17] HE X N, LIAO L Z, ZHANG H W, et al. Neural collaborative filtering[C]// Proceedings of the 26th International Conference on World Wide Web. [S.l.:s.n.], 2017: 173–182.

作者简介

陈华钧（1978– ），男，浙江大学计算机科学与技术学院教授，主要研究方向为知识图谱、自然语言处理、大数据系统。



张文（1992– ），女，博士，浙江大学软件学院助理研究员，主要研究方向为知识图谱、知识表示和知识推理。

黄志文（1993– ），男，阿里巴巴集团商品知识图谱团队算法工程师，主要研究方向为深度学习和知识图谱。

叶橄强（1996– ），男，浙江大学计算机科学与技术学院硕士生，主要研究方向为知识图谱表示学习和预训练。

文博（1994– ），男，浙江大学计算机科学与技术学院硕士生，主要研究方向为知识图谱和推荐计算。

张伟（1983– ），男，博士，阿里巴巴集团资深算法专家，主要研究方向为自然语言处理和知识图谱。

收稿日期： 2021-04-01

通信作者： 张文，wenzhang@zju.edu.cn

基金项目： 国家自然科学基金资助项目（No.91846204，No.U19B2027）

Foundation Items: The National Natural Science Foundation of China (No.91846204, No.U19B2027)