计算机科学
COMPUTER SCIENCE

基于增强序列标注策略的单阶段联合实体关系抽取方法

朱秀宝, 周刚, 陈静, 卢记仓, 向怡馨

引用本文

朱秀宝, 周刚, 陈静, 卢记仓, 向怡馨基于增强序列标注策略的单阶段联合实体关系抽取方法[J]. 计算机科学, 2023, 50(8): 184-192.

ZHU Xiubao, ZHOU Gang, CHEN Jing, LU Jicang, XIANG Yixin. Single-stage Joint Entity and Relation Extraction Method Based on Enhanced Sequence Annotation Strategy [J]. Computer Science, 2023, 50(8): 184-192.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220700082

基于增强序列标注策略的单阶段联合实体关系抽取方法

朱秀宝　周　刚　陈　静　卢记仓　向怡馨
数学工程与先进计算国家重点实验室　郑州 450001
(freeline55@163.com)

摘　要　从非结构化文本中抽取实体和关系是自动构建知识库的基础工作。现有的工作主要采用联合学习方法来解决嵌套实体、重叠关系、冗余计算和曝光偏差等问题,但单个模型仅在部分问题上表现出色,尚无模型可以同时解决上述问题。因此,提出了一种基于增强序列标注策略的单阶段联合实体关系抽取方法(A Token With Multi-labels Entity and Relation Extraction,ATMREL)。首先,设计了一种增强序列标注策略,将文本中的每个单词标记为多个标签,标签包含每个单词在实体中的位置、关系类型和实体位置信息。然后,将每个单词的标签预测转化为多标签分类任务,同时将联合实体关系抽取转化为序列标注任务。最后,为增强实体对之间的依赖关系,引入实体相关矩阵,用于对抽取结果进行剪枝,以提升模型抽取效果。实验结果表明,与 CasRel 和 TPLinker 模型相比,ATMREL 模型在 NYT 和 WebNLG 数据集上的参数量减少了 $3.1\times10^6\sim5.4\times10^6$,平均推理速度提升了 2～4.2 倍,F1 值提升了 0.5%～2.1%。

关键词:联合实体关系抽取;序列标注;组合标签;相关矩阵
中图法分类号　TP391

Single-stage Joint Entity and Relation Extraction Method Based on Enhanced Sequence Annotation Strategy

ZHU Xiubao,ZHOU Gang,CHEN Jing,LU Jicang and XIANG Yixin
State Key Laboratory of Mathematical Engineering and Advanced Computing,Zhengzhou 450001,China

Abstract　Extracting entities and relations from unstructured text is the fundamental task of automatically constructing knowledge bases. Existing works mainly adopt joint learning to solve the problems of nested entities,overlapping relations,redundant computation,or exposure bias,but a single model only performs well on some issues,and no model can solve the above problems simultaneously. Therefore,a single-stage joint entity and relation extraction method based on an enhanced sequence annotation strategy called ATMREL is proposed. First,an enhanced sequence annotation strategy is designed to tag each word in the text with multiple labels,and the labels contain information about the position of each word in the entity,the relation type and the entity location. Second,the labels prediction of each word is transformed into a multi-label classification task,while the joint entity and relation extraction is transformed into a sequence annotation task. Finally,to enhance the dependencies between entity pairs,an entity correlation matrix is introduced for pruning the extraction results to improve the model extraction effect. Experimental results show that ATMREL model reduces the parameter volume by $3.1\times10^6\sim5.4\times10^6$,improves the average inference speed by 2～4.2 times,and improves the F1 value by 0.5%～2.1% compared with the CasRel and TPLinker models on the NYT and WebNLG datasets.

Keywords　Joint entity and relation extraction,Sequence annotation,Combined labels,Correlation matrix

1　引言

实体关系抽取旨在从非结构化文本中抽取实体和实体之间的语义关系,形成实体关系三元组,其一般的表达形式为(头实体,关系,尾实体),这也是知识图谱的组成三要素[1]。早期的实体关系抽取采用流水线方法[2-4],即首先使用命名体识别模型抽取文本中的实体,然后使用关系分类模型预测候选实体对之间的关系。该方法由于缺少两个子任务的交互,容易造成误差累积和冗余计算等问题。因此,后续的研究提出联合实体关系抽取方法[5-25]来同时抽取文本中存在的实体及关系,即使用深度神经网络编码句子,通过设计合理的标注策略,或者采用不同的向量融合方式及解码顺序,不断增强两个子任务之间的交互,从而提升模型的抽取效果。近年来,针对联合实体关系抽取的研究已取得长足进步,但仍然存在

到稿日期:2022-07-08　返修日期:2022-12-01
基金项目:河南省科技攻关项目(222102210081)
This work was supported by the Science and Technology Project of Henan Province(222102210081).
通信作者:周刚(gzhougzhou@126.com)

如下 4 个方面的问题。

(1)嵌套实体问题[26]，指在一个实体的内部包含一个或多个其他类型的实体。例如，“河南博物院”是一个类型为组织机构名的实体，而“河南博物院”中的“河南”同时也是类型为地名的实体。

(2)重叠关系问题[11]，根据实体关系三元组中实体的重叠程度可以将句子分为正常(Normal)、实体对重叠(Entity Pair Overlap，EPO)和单实体重叠(Single Entity Overlap，SEO)3 种类型。如表 1 所列，如果一个句子中的所有实体关系三元组都没有重叠的实体，则这个句子就属于正常类型；如果一个句子中相同的实体对之间存在多个不同的关系，则这个句子就属于实体对重叠类型；如果一个句子中的一个实体存在于多个实体关系三元组中，则这个句子就属于单实体重叠类型。

表 1 正常、实体对重叠和单实体重叠类型的例句

Table 1 Examples of normal，EPO，and SEO types

类型	示例
Normal	国家 [河南省]位于[中国]中东部
EPO	导演 [周星驰]自导自演了[《功夫》] 主演
SEO	作曲 [《夜曲》]由[方文山]作词，[周杰伦]作曲 作词

(3)冗余计算问题[27]，即模型训练或者推理阶段产生了大量无意义的运算。例如，TPLinker[24] 和 OneRel[25] 等模型在训练阶段通常需要预定义多个关系，并为每个关系建立一个矩阵。在推理阶段，不管文本中是否存在某个或某些关系，都要遍历所有预定义的关系矩阵，造成了冗余计算问题，而且预定义的关系数量越多，推理时间越长，占用内存越大。

(4)曝光偏差问题，即训练阶段和推理阶段子模型输入数据不一致。如图 1 所示，训练阶段各个组件的输入都来自真实的标注样本，而推理阶段各个组件的输入来自前面组件的预测结果，无论是头实体“三体”还是尾实体“刘慈欣”预测错误，都会导致后续组件抽取错误。

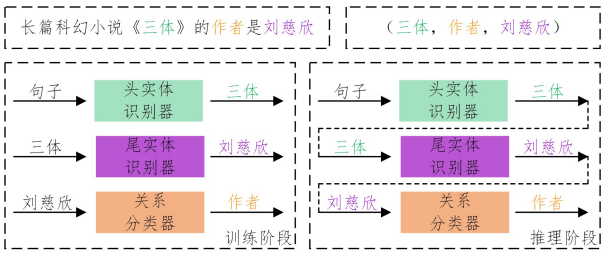


图 1 曝光偏差例子

Fig. 1 Example of exposure bias

为了同时解决上述问题，本文提出了一种基于增强序列标注策略的单阶段联合实体关系抽取方法(ATMREL)。首先，设计了一种增强序列标注策略，将文本中的每个单词标记为多个标签，标签包含每个单词在实体中的位置(即开始单词、内部单词或者非实体单词)、关系类型和实体位置信息(即头实体或者尾实体)。其次，将每个单词的标签预测转化为多标签分类任务，同时将联合实体关系抽取转化为序列标注任务，解码序列标注结果可以直接得到实体关系三元组。最后，为了增强实体对之间的交互，引入了实体相关矩阵，用于对抽取的结果进行剪枝，进而提升模型抽取效果。总体上，本文的主要贡献包括 3 个方面：

(1)设计了一种增强序列标注策略，将每个单词的标签预测转化为多标签分类任务，可以自然地解决嵌套实体、重叠关系等问题，有效解决了 BIO，BIOES 等传统序列标注策略只能将每个单词标记为单个标签的问题。

(2)将联合实体关系抽取转化为序列标注任务，该过程不包含任何相互依赖的步骤，实现了训练阶段和推理阶段子模型输入数据的一致性，从而解决了曝光偏差问题。另外，通过引入实体相关矩阵，有效减少了无意义的实体关系三元组的数量，由于矩阵参数与关系数量无关，因此计算量较小。

(3)在 NYT 数据集和 WebNLG 数据集上的实验结果表明，ATMREL 模型不仅参数量小、收敛速度快、推理速度快，还可以同时解决嵌套实体、重叠关系、冗余计算和曝光偏差等问题。

2 相关工作

在早期的联合实体关系抽取方法中[5-9]，两个子任务通过共享编码层的参数进行交互，取得了很好的抽取效果。然而，这些方法严重依赖于 NLP 工具来提取词性特征、语法特征、依存句法特征，存在较大误差。近年来，很多研究提出了基于深度神经网络的端到端实体关系抽取模型，这类模型可以同时编码实体和关系而不再依赖特征提取，不断刷新了公开评测数据集的最佳成绩，因此受到了广泛关注。

Zheng 等[10]首次将实体关系抽取创新性地转换成序列标注任务，该方法将每个单词标记为实体中的单词位置、关系类型和关系角色的组合标签以及“Other”标签，通过解码序列标注结果即可同时抽取实体和关系，但是由于一个单词只能属于一种标签，因此该方法不能解决嵌套实体和重叠关系问题。针对此类问题，Zeng 等[11]首次根据实体关系三元组中实体的重叠程度将句子分为正常、实体对重叠和单实体重叠 3 种类型，提出了一种基于复制机制的模型。为了进一步解决此问题，后续的研究提出了大量的方法，例如基于序列到序列的方法[12]、基于图神经网络的方法[13]、基于序列标注的方法[14]、基于强化学习的方法[15]、基于对比学习的方法[16]、基于表格填充的方法[17-18]、基于片段跨度的方法[19]、基于多头选择的方法[20]等，这些方法大多是先抽取实体，再判断实体对之间的关系，虽然取得了很好的抽取效果，但是由于很多实体之间并不存在关系，因而产生了冗余计算问题。

因此，很多研究通过采用不同的抽取顺序来缓解冗余计算问题。Wei 等[21]提出了一种新的级联二元标注框架，将关系建模成一个使头实体映射到尾实体的函数，即先抽取头实体再根据关系抽取对应的尾实体；Zheng 等[22]设计了一种基于潜在关系、全局矩阵和序列标注的模型，即先抽取关系再同时抽取头实体和尾实体；Ma 等[23]则提出了一种高效的级联双解码器，即先抽取关系再抽取头实体，最后抽取尾实体。

以上方法大多通过多个阶段完成联合实体关系抽取任务，虽然能够很好地处理嵌套实体、重叠关系和冗余计算问题，但是依然存在曝光偏差问题。因此，Wang 等[24]提出了新颖的握手标记方案，实现了头实体和尾实体的对齐，保证了训练阶段和测试阶段组件输入数据的一致性，实现了单阶段抽取实体和关系。Shang 等[25]将任务转化为表格填充问题，使用单独的关系矩阵学习头实体、关系和尾实体的依赖关系，有效减少了误差传播和冗余计算。然而，这些方法需要为每一个预定义的关系建立矩阵，该矩阵过于稀疏导致模型收敛速度慢、训练时间长，存在冗余计算问题。

综上所述，目前已经提出了大量的工作来解决嵌套实体、重叠关系、冗余计算和曝光偏差问题，而且在部分问题上表现出色，但尚无模型可以同时解决全部问题。基于此现状，本文提出了一种基于增强序列标注策略的单阶段联合实体关系抽取方法，从一个全新的角度同时解决上述问题。

3 方法描述

ATMREL 模型总共分为 4 个部分：文本序列编码器、增强序列标注组件、实体相关矩阵以及解码模块。模型的总体流程为：首先将文本序列输入文本序列编码器，得到每个单词的词向量表示；其次将所有单词的词向量表示同时输入增强序列标注组件和实体相关矩阵，得到预测结果的映射标签；然后在解码模块中对增强序列标注组件和实体相关矩阵预测结果的映射标签进行解码，得到具有相同关系的头实体和尾实体组合以及相关的开始单词组合；最后将相关的实体关系三元组保留，将不相关的删除，得到最终的抽取结果。图 2 给出了 ATMREL 模型的总体结构，本节将详细介绍各个组件的设计思路与实现方法。

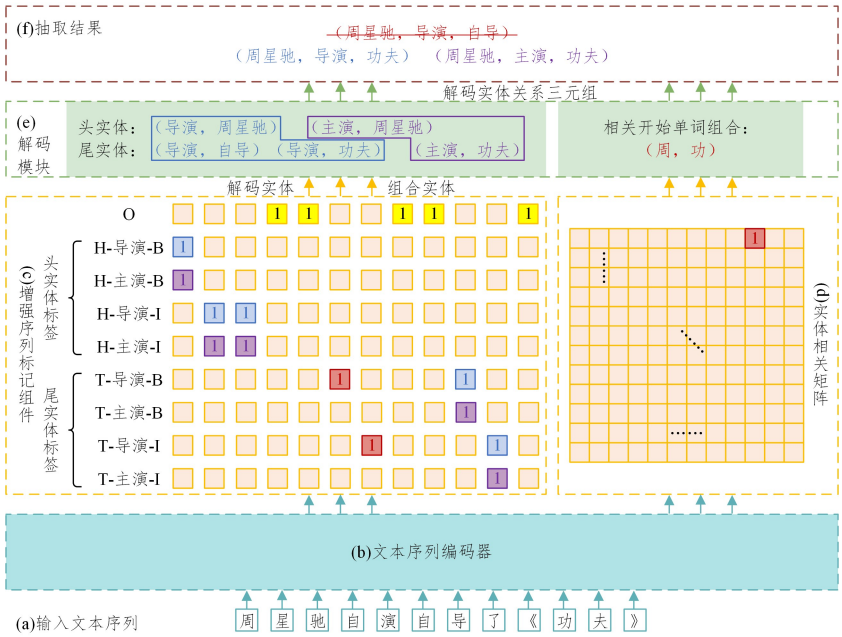


图 2 基于增强序列标注策略的单阶段联合实体关系抽取方法的总体结构(电子版为彩图)

Fig. 2 Overall structure of single-stage joint entity and relation extraction method based on enhanced sequence annotation strategy

3.1 文本序列编码器

在传统的 NLP 任务中，通常需要先使用语言模型对输入文本序列进行编码，得到每个单词的词向量表示。传统的语言模型有 Word2Vec[28]，GloVe[29]以及 fastText[30]等，由于这些方法得到的是固定的词向量表示，无法解决一词多义问题，因此，Devlin 等[31]于 2018 年 10 月提出了基于 Transformer 的双向编码表示模型 Bert，该模型采用遮蔽语言模型(Masked Language Model，MLM)和下一句预测(Next Sentence Prediction，NSP)两个任务，然后在大规模语料中以自监督的方式进行预训练，使得相同的单词可以在不同的上下文语境中具有不同的词向量表示，在 11 个 NLP 任务中刷新了最佳成绩。

本文使用 Bert 作为文本序列编码器来获取输入文本序列的词向量表示。给定一个长度为 N 的输入文本序列 $\boldsymbol{e}=[e_1,e_2,\cdots,e_N]$，其中 e_i 表示输入文本序列中的第 i 个单词，需要先将 $\boldsymbol{e}$ 转换为 Bert 模型输入需要的嵌入向量 $\boldsymbol{t}=[t_1,t_2,\cdots,t_N]$，该向量由词嵌入向量$\boldsymbol{W}_T$、分割嵌入向量$\boldsymbol{W}_S$和位置嵌入向量$\boldsymbol{W}_P$相加而得，计算式如式(1)所示。然后，将嵌入向量 $\boldsymbol{t}$ 输入 Bert 模型进行编码，其输出向量 $\boldsymbol{x}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N]$ 就是输入文本序列 $\boldsymbol{e}$ 的词向量表示，其中$\boldsymbol{x}_i$表示第 i 个单词的词向量表示，$\boldsymbol{W}_i\in\mathbb{R}^{1\times 768}$，其计算式如式(2)所示：

$$\boldsymbol{t}=\boldsymbol{W}_T+\boldsymbol{W}_S+\boldsymbol{W}_P \tag{1}$$

$$\boldsymbol{x}=Bert(\boldsymbol{t}) \tag{2}$$

3.2 增强序列标注组件

3.2.1 增强序列标注策略

在一个文本序列中，一个单词可能属于不同类型的实体，同一实体也可能参与不同关系类型的实体关系三元组。由于传统的序列标注策略只能将单词标记为一个标签，因此不能解决嵌套实体和重叠关系问题。针对此问题，本文设计了增强序列标注策略，将文本中的每个词标记为每个词在实体中的位置(开始词、内部词或者非实体词)、关系类型和实体位置信息(头实体或者尾实体)的组合标签，每个标签的顺序及含义如表 2 所列。由此可以计算出当输入文本序列长度为 N、预定义关系类型数量为 R 时，每个单词的标签总数为 $4\times R+1$，

该文本序列需要标注的标签总数为 $N\times(4\times R+1)$。

表 2 增强序列标注组件中标签的顺序及含义

Table 2 Order and meaning of tags in enhanced sequence annotation components

序号	标签	含义
0	O	非实体单词
1	H−R_1−B	头实体标签具有关系 R_1 开始单词
$R+1$	H−R_1−I	头实体标签具有关系 R_1 内部单词
$2\times R+1$	T−R_1−B	尾实体标签具有关系 R_1 开始单词
$3\times R+1$	T−R_1−I	尾实体标签具有关系 R_1 内部单词

如图 2(c)中的增强序列标注组件所示，给定一个文本序列“周星驰自演自导了《功夫》”，预定义的关系为“导演”和“主演”，则每个单词的标签总共有 9 个，详细的标签已在图 2(c)中列出。该文本序列标注了(周星驰，导演，功夫)和(周星驰，主演，功夫)两个实体关系三元组，第一个实体关系三元组的标注结果在图 2(c)中用蓝色方块标记，第二个实体关系三元组在图 2(c)中用紫色方块标记，其余非实体的单词在图 2(c)中用黄色方块标记。注意，图 2(c)中的红色方块标记是预测结果的标签映射，该结果会在 3.4 小节中介绍。

由于“周星驰”和“功夫”这两个实体共同出现在具有不同关系的实体关系三元组中，因此输入的文本序列属于实体对重叠类型。从标注结果来看，一个单词如果属于不同关系类型的实体，则会在不同的标签位置标记为 1，相互之间并不冲突，从而自然地标注了实体对重叠类型的句子。显然该策略还可以标注正常类型和单实体重叠类型的句子，与 BIO，BIOES 等传统的序列标注策略相比具有更强的表达能力，可以有效地解决嵌套实体和重叠关系问题。

3.2.2 增强序列标注组件实现

本文将每个单词的标签预测转化为多标签分类任务，而不是传统的多分类任务。首先，增强序列标注组件使用全连接神经网络来实现，激活函数使用 sigmoid。然后，将每个单词的词向量表示输入组件，输出每个单词所属标签的预测概率，计算式如式(3)所示：

$$\boldsymbol{p}_i=\mathrm{sigmoid}(\boldsymbol{W}_s\boldsymbol{x}_i+\boldsymbol{b}_s) \tag{3}$$

其中，$\boldsymbol{p}_i\in\mathbb{R}^{1\times(4\times R+1)}$，$R$ 为预定义关系的数量，$\boldsymbol{W}_s$ 表示可训练的权重矩阵，$\boldsymbol{b}_s$ 表示可训练的偏置常数。最后，如果每个单词所属标签的预测概率超过阈值，则映射结果为 1，否则为 0。

3.3 实体相关矩阵

增强序列标注组件不能有效表达头实体和尾实体的关联关系，受 Zheng 等[22]的启发，本文引入实体相关矩阵，用于增强头实体和尾实体的交互，减少输出无意义的实体关系三元组。假设模型最大输入文本序列长度为 M，则实体相关矩阵的维度为(M,M)，每一个元素表示两个单词作为头实体的开始单词与尾实体的开始单词之间的相关概率。首先将文本序列中单词的词向量两两组合，然后将其输入至全连接神经网络中计算单词之间的相关概率，计算式如式(4)所示：

$$p_{i,j}=\mathrm{sigmoid}(\boldsymbol{W}_m[\boldsymbol{x}_i;\boldsymbol{x}_j]+\boldsymbol{b}_m) \tag{4}$$

最后，如果相关概率超过阈值，则映射结果为 1，否则为 0。

3.4 解码模块

3.4.1 解码增强序列标注组件

算法 1 实体解码算法 DE

输入：文本序列 T，文本序列标签预测结果 tags，实体开始单词签索引 id，预定义关系数量 R

输出：实体列表 E

```
1. 初始化实体列表 E
2. for i←1 to len(tags) do
3.   if tags[i][id]==1 then //实体的开始单词
4.     j←0
5.     while i+j+1<len(tags) do
6.       if tags[i+j+1][id+R]! =1 then
7.         break
8.       j←j+1 //实体的内部单词
9.     add T[i:i+j] to E
10. return E
```

首先，根据增强序列标注组件的输出映射结果，解码出具有关系的头实体和尾实体。解码头实体可以根据头实体开始单词的标签索引 id，+R 找到头实体内部单词的标签索引；解码尾实体可以根据尾实体开始单词的标签索引 id，+R 找到尾实体内部单词的标签索引，详细过程如算法 1 所示。然后，将具有相同关系的头实体和尾实体两两组合即可生成实体关系三元组。

以图 2(c)所示的增强序列标注组件的映射结果为例，首先，遍历标记为 1 的头实体标签可以解码得到具有关系的头实体(导演，周星驰)和(主演，周星驰)；遍历标记为 1 的尾实体标签可以解码得到具有关系的尾实体(导演，自导)、(导演，功夫)和(主演，功夫)。然后，将关系为“导演”的头实体和尾实体两两组合，生成实体关系三元组(周星驰，导演，自导)和(周星驰，导演，功夫)，将关系为“主演”的头实体和尾实体两两组合生成实体关系三元组(周星驰，主演，功夫)。

3.4.2 解码实体相关矩阵

算法 2 实体关系三元组解码算法

输入：文本序列 T，文本序列标签预测结果 tags，实体相关矩阵标签预测结果 G，预定义关系数量 R

输出：实体关系三元组列表 S

```
1. 初始化实体关系三元组列表 S
2. for id←1 to R do
3.   hs←DE(T,tags,id,R) //解码头实体
4.   ts←DE(T,tags,id+2×R,R) //解码尾实体
5.   for h in hs do
6.     for t in ts do
7.       if (h[0],t[0]) in G then
8.         add(h,id,t) to S
9. return S
```

根据实体相关矩阵的输出映射结果，可以解码得到相关的头实体开始单词和尾实体开始单词的组合。最后将增强序列标注组件的解码结果和实体相关矩阵的解码结果

进行匹配，保留相关的的实体关系三元组，删除无意义的实体关系三元组，从而得到最终的抽取结果，详细过程如算法 2 所示。

如图 2(d)所示的实体相关矩阵的映射结果所示，遍历标记为 1 的所有元素可以解码得到相关开始单词组合(周，功)。如图 2(e)所示的解码模块所示，(周星驰，导演，自导)的开始单词组合为(周，自)，不是相关开始单词组合，结果删除；(周星驰，导演，功夫)的开始单词组合为(周，功)，是相关开始单词组合，结果保留；(周星驰，主演，功夫)的开始单词组合为(周，功)是相关开始单词组合，结果保留。因此，最终的实体关系抽取结果为(周星驰，导演，功夫)和(周星驰，主演，功夫)。

3.5 损失函数设计

损失函数 $\mathcal{L}_{all}$ 包含增强序列标签组件损失函数 $\mathcal{L}_{seq}$ 和实体相关矩阵损失函数 $\mathcal{L}_{matrix}$ 两个部分，计算式如式(5)－式(7)所示：

$$\mathcal{L}_{seq}=-\frac{1}{N\times(4\times R+1)}\sum_{i=1}^{N}\sum_{j=1}^{4\times R+1}(y_{i,j}\log p_{i,j}+(1-y_{i,j})\log(1-p_{i,j})) \tag{5}$$

$$\mathcal{L}_{matrix}=-\frac{1}{M\times M}\sum_{i=1}^{M}\sum_{j=1}^{M}(y_{i,j}\log p_{is,js}+(1-y_{i,j})\log(1-p_{is,js})) \tag{6}$$

$$\mathcal{L}_{all}=\mathcal{L}_{seq}+\mathcal{L}_{matrix} \tag{7}$$

其中，N 表示一个输入文本序列的长度，R 表示预定义关系的数量，M 表示输入文本序列的最大长度。

4 实验与分析

4.1 数据集描述

实验使用公开的 NYT 数据集[32] 和 WebNLG 数据集[33]，这两个数据集常用于评估实体关系抽取模型的性能。另外，这两个数据集各有两个版本，分别为 NYT，NYT* 和 WebNLG，WebNLG*，不同之处在于 NYT 和 WebNLG 数据集标注了整个实体，而 NYT* 和 WebNLG* 数据集仅标注了实体的开始单词。表 3 列出了 4 个数据集的训练集、验证集、测试集的数据量及关系数量、重叠关系类型、实体关系三元组数量等信息。由于部分句子既属于 EPO 类型，又属于 SEO 类型，因此 Normal 类型、EPO 类型和 SEO 类型句子的总数会略大于测试集的数量。

表 3 NYT*，NYT，WelNLG* 和 WebNLG 数据集的统计信息

Table 3 Statistical information of NYT*, NYT, WelNLG* and WebNLG datasets

Dataset	Train	Valid	Test	Relation	Overlapping Types			Number of triplets				
					Normal	SEO	EPO	$N=1$	$N=2$	$N=3$	$N=4$	$N\geqslant 5$
NYT* NYT	56195	4999	5000	24	2366	1297	978	3244	1045	312	291	108
WebNLG* WebNLG	5019	500	703	171 216	246	457	26	266	171	131	90	45

4.2 评价指标

由于 4 个数据集的实体标注策略不完全一致，因此 NYT* 和 WebNLG* 数据集使用部分匹配的评价方式，即只要抽取的实体关系三元组中头实体的开始单词、关系和尾实体的开始单词与真实标签数据全部匹配，就可以判断结果正确。NYT 和 WebNLG 数据集使用完全匹配方式，即预测的实体关系三元组必须和真实标签数据全部严格匹配才可以判断结果正确。实验采用的评价指标为精确率(Prec.)、召回率(Rec.)和 F1 值(F1-score，F1)[21]。

4.3 实验设置

模型使用 PaddlePaddle 2.3.0 搭建神经网络，使用 bert-base-cased 版本的 Bert 模型作为编码器，输入句子的最大长度为 100，NYT* 和 NYT 数据集使用的批大小为 24，WebNLG* 和 WebNLG 数据集使用的批大小为 6，使用 AdamW 优化器优化损失函数，使用线性衰减的方式动态调整学习率。设置学习率为 5×10^{-5}，学习率预热速率(warmup ratio)和权重衰减速率(decay ratio)为 1×10^{-4}，增强序列标注组件和实体相关矩阵的概率阈值为 0.5。模型使用 NVIDIA Tesla A100 80 显卡训练，使用 NVIDIA Tesla V100 32GB 显卡推理，总共训练 100 轮，训练过程中只保存在验证集上 F1 值最大的模型。实验选择了 8 个近年来用于解决嵌套实体、重叠关系、曝光偏差等问题的方法作为基线模型，分别为 NovelTagging[10]，CopyRE[11]，MultiHead[20]，GraphRel[13]，OrderCopyRE[15]，ETL-Span[14]，CasRel[21] 和 TPLinker[24]，基线模型的结果均取自 Wang 等[24]的实验。

4.4 结果分析

4.4.1 总体对比结果

表 4 列出了 ATMREL 模型与基线模型的总体对比结果，加粗数据表示最优结果。可以发现，ATMREL 模型在 4 个数据集上的各项评价指标均显著优于 NovelTagging，CopyRE，MultiHead，GraphRel，OrderCopyRE 和 ETL-Span 等模型，证明了 ATMREL 模型的优越性。与 CasRel 模型相比，ATMREL 模型在 NYT* 数据集上的 F1 值提升了 2.1%，在 WebNLG* 数据集上的 F1 值提升了 0.1%。与 TPLinker 模型相比，ATMREL 模型在 WebNLG 数据集上的 F1 值提升了 0.5%，在其他 3 个数据集上的准确率较高，但是召回率较低。总体来说，ATMREL 模型在 NYT 数据集上的效果比在 WebNLG 数据集上的效果好，这是因为 NYT 数据集关系的数量少，而且主要由 Normal 类型的句子组成，抽取难度较小。而 WebNLG 数据集不仅关系数量多，而且包含了大量的 EPO 和 SEO 类型的句子，抽取难度较大。另外，ATMREL 模型在 4 个数据集上的准确率相对较高，而召回率相对较低，这个差距是因 ATMREL 模型的实体相关矩阵非常稀疏造成的，因此，对抽取的实体关系三元组列表进行剪枝，虽然能够

减少大量错误的实体关系三元组，但是也会误删部分正确的实体关系三元组，从而导致召回率较低。

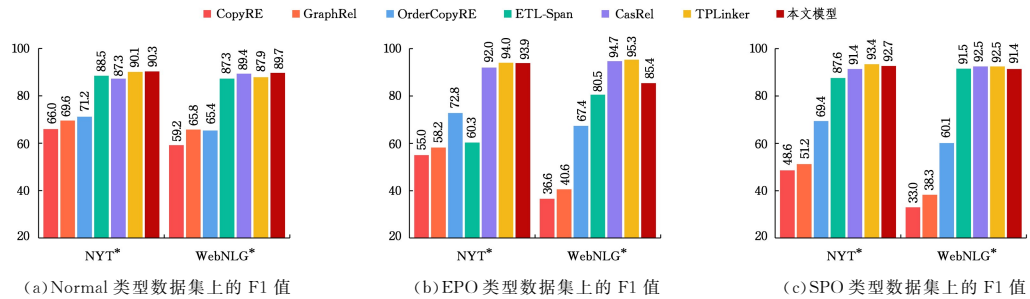
表 4 ATMREL 模型与基线模型的总体对比结果

Table 4 Overall comparison results of ATMREL model and baseline model

Model	NYT*			NYT			WebNLG*			WebNLG		
	Prec.	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*
NovelTagging[10]	—	—	—	32.8	30.6	31.7	—	—	—	52.5	19.3	28.3
CopyRE[11]	61.0	56.6	58.7	—	—	—	37.7	36.4	37.1	—	—	—
MultiHead[20]	—	—	—	60.7	58.6	59.6	—	—	—	57.5	54.1	55.7
GraphRel[13]	63.9	60.0	61.9	—	—	—	44.7	41.1	42.9	—	—	—
OrderCopyRE[15]	77.9	67.2	72.1	—	—	—	63.3	59.9	61.6	—	—	—
ETL-Span[14]	84.9	72.3	78.1	85.5	71.7	78.0	84.0	91.5	87.6	84.3	82.0	83.1
CasRel[21]	89.7	89.5	89.6	—	—	—	93.4	90.1	91.8	—	—	—
TPLinker[24]	91.3	**92.5**	**91.9**	**91.4**	**92.6**	**92.0**	91.8	**92.0**	**91.9**	88.9	84.5	86.7
ATMREL	**92.0**	91.4	91.7	**91.4**	91.1	91.2	**93.5**	88.9	91.2	**89.3**	**85.3**	**87.2**

4.4.2 复杂场景的结果

复杂场景包括不同重叠关系类型和不同实体关系三元组数量。ATMREL 模型在不同重叠关系类型数据集上的 F1 值对比结果如图 3 所示，可以发现，CopyRE，GraphRel 和 OrderCopyRE 模型在 EPO 和 SEO 类型数据集上的 F1 值均未达到 75%，而 ATMREL 模型在不同数据类型的 F1 值均超过了 85%，证明 ATMREL 模型可以有效地处理重叠关系问题。另外，与 ETL-Span，CasRel 和 TPLinker 模型相比，ATMREL 模型也取得了基本一致的对比结果。观察到 ATMREL 模型在 WebNLG * 数据集 EPO 类型数据集上的 F1 值仅有 85.4%，虽然比 ETL-Span 的 F1 值高 4.9%，但是相比 CasRel 和 TPLinker 等模型的 F1 值低 9.3%～9.9%。这是因为 EPO 类型数据集中包含大量头实体和尾实体相同的实体关系三元组。因此，使用稀疏的实体相关矩阵对实体关系三元组列表进行剪枝过滤时，会误删更多的实体关系三元组，导致整体的召回率偏低，进而导致 F1 值偏低。



(a) Normal 类型数据集上的 F1 值 (b) EPO 类型数据集上的 F1 值 (c) SPO 类型数据集上的 F1 值

图 3 ATMREL 模型与基线模型在不同重叠关系类型数据集上 F1 值的对比结果

Fig. 3 F1-score comparison of ATMREL model and baseline models on datasets of different overlapping relationship types

ATMREL 模型在不同实体关系三元组数量数据集上的 F1 值对比结果如表 5 所列，可以发现，CopyRE，GraphRel，OrderCopyRE 和 ETL-Span 模型在实体关系三元组数量较少的数据集上效果较好，但是随着实体关系三元组数量的增加，模型的效果逐渐变差。而 ATMREL 模型在不同实体关系三元组数据集上的 F1 值均超过了 86%。相比 CasRel 和 TPLinker 模型，三者的 F1 值差值基本保持在 2%以内，说明了三者都能较好地适用于复杂场景的实体关系抽取任务。

表 5 ATMREL 模型与基线模型在不同实体关系三元组数量数据集上 F1 值的对比结果

Table 5 Results of the comparison between the ATMREL model and the baseline model for F1 values on different datasets of the number of entityand relation triples

Model	NYT*					WebNLG*				
	$N=1$	$N=2$	$N=3$	$N=4$	$N\geqslant 5$	$N=1$	$N=2$	$N=3$	$N=4$	$N\geqslant 5$
CopyRE[11]	67.1	58.6	52.0	53.6	30.0	59.2	42.5	31.7	24.2	30.0
GraphRel[13]	71.0	61.5	57.4	55.1	41.1	66.0	48.3	37.0	32.1	32.1
OrderCopyRE[15]	71.7	72.6	72.5	77.9	45.9	63.4	62.2	64.4	57.2	55.7
ETL-Span[14]	85.5	82.1	74.7	75.6	76.9	82.1	86.5	91.4	89.5	91.1
CasRel[21]	88.2	90.3	91.9	94.2	83.7	**89.3**	**90.8**	94.2	92.4	90.9
TPLinker[24]	**90.0**	**92.8**	93.1	**96.1**	90.0	88.0	90.1	**94.6**	**93.3**	**91.6**
ATMREL	89.9	92.1	**93.4**	95.4	**91.1**	86.2	90.1	94.5	92.8	90.3

4.5 消融实验

ATMREL 模型通过引入实体相关矩阵来增强头实体和尾实体开始词的依赖关系，以对实体关系三元组列表进行剪枝，有效地减少了错误实体关系三元组的数量，提升了抽取效果。

为了证明实体相关矩阵的作用，设计了两组消融实验，第一组消融实验是移除实体相关矩阵，直接输出实体关系三元组列表。第二组消融实验使用潜在关系预测组件替换实体相关矩阵，使用输入文本的潜在关系列表对实体关系三元组列表进行剪枝。消融实验的对比结果如表 6 所列。

表 6 消融实验结果

Table 6 Ablation experiment results

Model	NYT*			NYT			WebNLG*			WebNLG		
	Prec.	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*	*Prec.*	*Rec.*	*F1*
ATMREL	**92.0**	**91.4**	**91.7**	**91.4**	91.1	**91.2**	**93.5**	88.9	91.2	**89.3**	85.3	87.2
第一组消融实验	88.4	**91.4**	89.8	87.7	**92.2**	89.9	93.0	**91.2**	**92.1**	88.6	**86.9**	**87.8**
第二组消融实验	85.4	89.5	87.4	85.6	90.1	87.8	93.0	87.9	90.4	88.6	84.3	86.4

通过第一组消融实验的对比结果可以发现，引入实体相关矩阵能够显著提升精确率，在 NYT* 和 NYT 数据集上分别提升了 3.6% 和 3.7%，在 WebNLG* 和 WebNLG 数据集上分别提升了 0.5% 和 0.7%，证明了实体相关矩阵的有效性。然而，由于实体相关矩阵较为稀疏，导致在删除错误实体关系三元组时，也同时删除了部分正确的实体关系三元组，因此召回率相对较低，在 NYT 数据集上下降了 1.1%，在 WebNLG* 和 WebNLG 数据集上分别下降了 2.3% 和 1.6%。从 ATMREL 模型在 4 个数据集上的性能变化趋势来看，ATMREL 模型在 NYT* 和 NYT 数据集上的性能提升明显，下降缓慢，而在 WebNLG* 和 WebNLG 数据集上的性能提升缓慢，下降明显，这是因为 NYT* 和 NYT 训练集的数据量比 WebNLG* 和 WebNLG 训练集大近 10 倍，因此实体相关矩阵的训练更加充分，效果更好。

通过第二组消融实验的对比结果可以发现，引入潜在关系预测组件导致所有的评价指标下降，尤其在 NYT* 和 NYT 数据集上准确率下降明显，下降幅度分别为 6.6% 和 5.8%。通过对模型结构和推理结果的分析发现，ATMREL 模型在解码增强序列标注组件预测结果的过程中已经得到带有实体位置和关系类型的实体，因此引入潜在关系预测组件导致功能重复，反而影响了最终的抽取效果。

4.6 计算效率分析

计算效率主要包括训练收敛速度、参数量及推理时间。图 4 和图 5 给出了 ATMREL 模型在 NYT* 和 WebNLG* 数据集上训练时 F1 值的变化曲线。可以看出，ATMREL 模型与 CasRel 模型在 NYT* 数据集上训练 5 轮后基本收敛，在 WebNLG* 数据集上训练 22 轮后基本收敛，而 TPLinker 模型在 NYT* 训练集上训练 7 轮后才基本收敛，在 WebNLG* 数据集上训练 31 轮后才基本收敛。这是因为 ATMREL 模型和 CasRel 模型的编码和解码过程没有复杂的操作，因此模型训练收敛速度快、用时短；而 TPLinker 为了减小握手标记矩阵的稀疏性，将握手标记矩阵的下三角删除并映射到矩阵的上三角对应位置，然后将上三角铺平展开成序列进行训练，处理过程较为复杂，因此模型训练收敛速度慢、用时长。

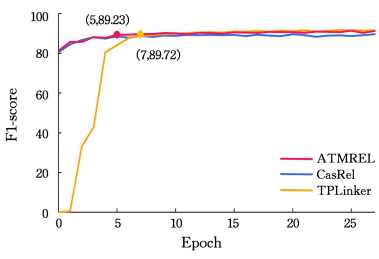


图 4 在 NYT* 数据集上训练模型时 F1 值的变化曲线

Fig. 4 Variation curves of F1 values when training models on NYT* dataset

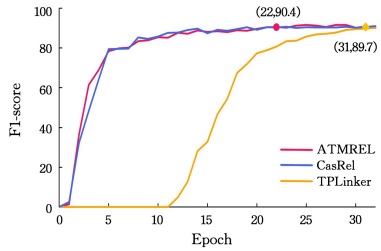


图 5 在 WebNLG* 数据集上训练模型时 F1 值的变化曲线

Fig. 5 Variation curve of F1 values when training models on WebNLG* dataset

表 7 列出了 ATMREL 模型的参数量和推理时间的对比结果。从参数量来看，ATMREL 模型在 3 个模型中的参数量最少，这是因为 CasRel 模型和 TPLinker 模型需要为每个预定义的关系维护一个关系矩阵，用于学习关系的特征，关系数量越多，引入的参数量就越多。而 ATMREL 模型中的组件参数与预定义的关系数量无关，因此引入的参数量较少。ATMREL 模型在 NYT* 数据集上的参数量比 CasRel 模型少 3.2×10^6，比 TPLinker 模型少 5.1×10^6；在 WebNLG* 数据集上的参数量比 CasRel 模型少 3.1×10^6，比 TPLinker 模型少 5.4×10^6。

表 7 计算效率分析结果

Table 7 Computational efficiency analysis results

Model	NYT*		WebNLG*	
	Params	Time (ms,24/1)	Params	Time (ms,24/1)
CasRel[21]	107.7×10^6	−/54.0	108.0×10^6	−/76.8
TPLinker[24]	109.6×10^6	15.2/82.7	110.3×10^6	25.6/112.6
ATMREL	104.5×10^6	6.4/27.3	104.9×10^6	6.5/26.9

实验使用推理时间来衡量模型的推理速度，并设置批大小为 24 和 1 两组参数。由于 CasRel 模型需要先预测头实体，再根据不同的关系预测出对应的尾实体，因此无法使用批处理，只能设置批大小为 1。从表 7 中可以看出，批大小为 24 时，ATMREL 模型在 NYT* 数据集上的推理速度是 TPLinker 模型的 2.4 倍，在 WebNLG* 数据集上的推理速度是 TPLinker 模型的 3.9 倍。批大小为 1 时，ATMREL 模型在 NYT* 数据集上的推理速度是 CasRel 模型的 2.0 倍，是 TPLinker 模型的 3.0 倍；在 WebNLG* 数据集上的推理速度是 CasRel 模型的 2.9 倍，是 TPLinker 模型的 4.2 倍。

另外，从表 7 中可以发现 CasRel 和 TPLinker 模型在 WebNLG* 数据集上推理速度比在 NYT* 数据集上的慢，主要原因是 WebNLG* 数据集的预定义的关系数量比 NYT* 数据集多，而且 CasRel 和 TPLinker 模型需要为每个预定义关系维护一个关系矩阵，在推理阶段只有遍历所有的关系矩阵才可以抽取出输入文本的全部实体关系三元组，因此造成

了大量的冗余计算。而 ATMREL 模型将每个词的标签预测任务转化为多标签分类任务，同时将联合实体关系抽取任务转化为序列标注任务，通过解码增强序列标注结果可以得到全部的实体关系三元组。另外，实体相关矩阵与关系无关，因此 ATMREL 模型在 NYT * 数据集和 WebNLG * 数据集上的推理速度不会随着关系数量的增加而变慢。

结束语 本文提出了一种基于增强序列标注策略的单阶段联合实体关系抽取模型，能够同时解决嵌套实体、重叠关系、计算冗余和曝光偏差问题。实验结果表明，ATMREL 模型与基线模型相比，具有模型参数量小、收敛速度快、推理速度快等优势。但是，本文引入的实体相关矩阵较为稀疏，在数据量较少的情况下会降低模型的抽取效果。未来，将对这一问题进行进一步的研究，从而取得更好的抽取效果。

参 考 文 献

[1] LIU Q, LI Y, DUAN H, et al. Knowledge graph construction techniques [J]. Journal of Computer Research and Development, 2016, 53(3): 582-600.

[2] ZELENKO D, AONE C, RICHARDELLA A. Kernel methods for relation extraction[C]// Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing. Philadelphia: ACL, 2002: 71-78.

[3] CHAN Y S, ROTH D. Exploiting syntactico-semantic structures for relation extraction[C]// The 49th annual Meeting of the Association for Computational Linguistics. Portland: ACL, 2011: 551-560.

[4] GORMLEY M R, YU M, DREDZE M. Improved relation extraction with feature-rich compositional embedding models [C]// Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing. Lisbon: ACL, 2015: 1774-1784.

[5] MIWA M, BANSAL M. End-to-end relation extraction using lstms on sequences and tree structures[C]// Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Berlin: ACL, 2016: 1105-1116.

[6] YU X F, LAM W. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach[C]// International Conference on Computational Linguistics. Beijing: Chinese Information Processing Society of China, 2010: 1399-1407.

[7] LI Q, JI H. Incremental joint extraction of entity mentions and relations[C]// Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. Baltimore: ACL, 2014: 402-412.

[8] MIWA M, SASAKI Y. Modeling joint entity and relation extraction with table representation [C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha: ACL, 2014: 1858-1869.

[9] REN X, WU Z Q, HE W Q, et al. Cotype: joint extraction of typed entities and relations with knowledge bases[C]// Proceedings of the 26th International Conference on World Wide Web. Perth: ACM, 2017: 1015-1024.

[10] ZHENG S C, WANG F, BAO H Y, et al. Joint extraction of entities and relations based on a novel tagging scheme[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: ACL, 2017: 1227-1236.

[11] ZENG X R, ZENG D J, HE S Z, et al. Extracting relational facts by an end-to-end neural model with copy mechanism[C]// Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics. Melbourne: ACL, 2018: 506-514.

[12] SUI D B, CHEN Y B, LIU K, et al. Joint entity and relation extraction with set prediction networks [J]. arXiv: 2011. 01675, 2020

[13] FU T J, LI P H, MA W Y, et al. GraphRel: modeling text as relational graphs for joint entity and relation extraction[C]// Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence: ACL, 2019: 1409-1418.

[14] YU B W, ZHANG Z Y, SHU X B, et al. Joint extraction of entities and relations based on a novel decomposition strategy[C]// ECAI 2020-24th European Conference on Artificial Intelligence. Santiago de Compostela: IOS Press, 2020: 2282-2289.

[15] ZENG X R, HE S Z, ZENG D J, et al. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning[C]// Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. Hong Kong: ACL, 2019: 367-377.

[16] YU K Q, HUANG F, WU Q, et al. Joint Extraction Method for Chinese Entity Relationship Based on Bidirectional Semantics [J]. Computer Engineering, 2023, 49(1): 92-99, 112.

[17] WANG Y J, SUN C Z, WU Y B, et al. Unire: a unified label space for entity relation extraction[C]// Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. ACL, 2021: 220-231.

[18] YAN Z H, ZHANG C, FU J L, et al. A partition filter network for joint entity and relation extraction[C]// Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Punta Cana: ACL, 2021: 185-197.

[19] JI B, YU J, LI S S, et al. Span-based joint entity and relation extraction with attention-based span-specific and contextual semantic representations[C]// Proceedings of the 28th International Conference on Computational Linguistics. Barcelona: International Committee on Computational Linguistics, 2020: 88-99.

[20] BEKOULIS G, DELEU J, DEMEESTER T, et al. Joint entity recognition and relation extraction as a multi-head selection problem [J]. Expert Systems with Applications, 2018, 114: 34-45.

[21] WEI Z P, SU J L, WANG Y, et al. A novel cascade binary tagging framework for relational triple extraction[C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL, 2020: 1476-1488.

[22] ZHENG H Y, WEN R, CHEN X, et al. PRGC: potential relation and global correspondence based joint relational triple extraction [C]// Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing. ACL, 2021:

6225-6235.
[23] MA L B, REN H M, ZHANG X L. Effective cascade dual-decoder model for joint entity and relation extraction [J]. arXiv: 2106.14163, 2021.
[24] WANG Y C, YU B, ZHANG Y Y, et al. TPLinker: single-stage joint extraction of entities and relations through token pair linking[C]// Proceedings of the 28th International Conference on Computational Linguistics. Barcelona: International Committee on Computational Linguistics, 2020: 1572-1582.
[25] SHANG Y M, HUANG H Y, MAO X L. Onerel: joint entity and relation extraction with one module in one step [J]. arXiv: 2203.05412, 2022.
[26] WANG J, SHOU L D, CHEN K, et al. Pyramid: a layered model for nested named entity recognition [C]// Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. ACL, 2020: 5918-5928.
[27] HUANG H Y, SHANG Y M, SUN X, et al. Three birds, one stone: a novel translation based framework for joint entity and relation extraction [J]. Knowledge-Based Systems, 2022, 236: 107677.
[28] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space [J]. arXiv: 1301.3781, 2013.
[29] PENNINGTON J, SOCHER R, MANNING C D. Glove: global vectors for word representation[C]// Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. Doha: ACL, 2014: 1532-1543.
[30] JOULIN A, GRAVE E, BOJANOWSKI P, et al. Bag of tricks for efficient text classification [C]// Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics. Valencia: ACL, 2017: 427-431.
[31] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [C]// Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics. Minneapolis: ACL, 2019: 4171-4186.
[32] RIEDEL S, YAO L M, MCCALLUM A. Modeling relations and their mentions without labeled text[C]// Proceedings of Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Barcelona: Springer, 2010: 148-163.
[33] GARDENT C, SHIMORINA A, NARAYAN S S, et al. Creating training corpora for nlg micro-planners[C]// Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. Vancouver: ACL, 2017: 179-188.

ZHU Xiubao, born in 1995, master candidate. His main research interests include knowledge graph and data mining.

ZHOU Gang, born in 1974, Ph. D, professor. His main research interests include big data analysis, knowledge graph and massive data processing.

(责任编辑:喻藜)