

基于类型间关系学习的细粒度实体分类^①席鹏弼^{②*} 靳小龙^{③*} 白 硕^{***}

(* 中国科学院计算技术研究所网络数据科学与技术重点实验室 北京 100190)

(** 中国科学院大学计算机科学与技术学院 北京 100408)

(***) 恒生电子股份有限公司 杭州 310053)

摘 要 细粒度实体分类旨在为构建知识图谱过程中所抽取的实体或实体提及确定一个或多个层次化、细粒度的类型,以便更好地为下游任务提供支持。现有细粒度实体分类方法存在细粒度分类精度不高、部分实体难以有效分类的问题。另一方面,直观来说,掌握细粒度类型之间的语义区别有助于实体的细粒度分类。但由于已有面向该任务的数据集缺少可用于学习细粒度类型间语义差别的数据,因此目前没有将细粒度类型之间的语义区别应用于细粒度实体分类的研究。为此,本文提出一种基于 Freebase 知识库学习细粒度类型语义区别的方法,并将学习到的语义信息应用在细粒度实体分类任务中。具体地,利用 SPARQL 从 Freebase 中获取类型之间的关系数据,据此学习细粒度实体类型之间的语义区别信息,进而结合实体提及及其上下文的文本信息进行细粒度实体分类。实验表明,本文提出的方法可以有效学习细粒度类型之间的语义区别,能够达到提升细粒度实体分类准确率的效果。

关键词 实体分类;细粒度类型;知识图谱构建;关系学习;多标签分类

0 引 言

随着知识图谱在应用领域的流行,需要图谱中实体具有更准确的语义信息。命名实体抽取任务得到的 PER、LOC、ORG 和 MISC 等少数几类实体类型^[1]难以满足应用上的要求。细粒度实体分类任务为实体确定若干个树形结构的细粒度类型,在知识图谱构建过程中发挥着重要作用,在领域知识图谱^[2-3]构建过程中体现得尤为明显。同时,也为知识图谱构建过程中的下游任务提供了重要信息。已有研究证明细粒度类型信息能大幅提升关系抽取^[3,4]、事件抽取和知识补全^[5,6]任务的性能。

细粒度实体分类一般采用多类别多标签分类方

法进行建模,把细粒度类型作为独立标签进行分类。实体信息学习方面的研究,从人工构造特征发展到神经网络学习特征,以及最近比较流行的 ELMo、Bert 语言模型^[7],对实体提及及其上下文的特征信息的学习越来越好,在数据集上提升了细粒度实体分类的性能。但这类模型忽视了细粒度类型之间语义上的区别信息,造成一部分细粒度类型没有预测出来或类型间相互错误预测。在模型的预测结果中,如实体的细粒度类型/location/structure/sports_facility、/person/political_figure 等多数预测不出来,或着类型/gpe/city 和/organization/corporation 相互之间错误预测,出现了细粒度分类精度不高、部分实体难以有效分类的问题。

在将细粒度类型间的关系应用到细粒度实体分

① 国家自然科学基金(U1911401,61772501,62002341,U1836206)资助项目。

② 男,1981 年生,博士生;研究方向:知识图谱;E-mail: xipengbi@ict.ac.cn。

③ 通信作者,E-mail: xipengbi@ict.ac.cn。

(收稿日期:2021-12-21)

类任务方面,最近有学者们采用细粒度类型之间的父子关系提出了一系列方法^[8-11]。该类方法使用了细粒度类型之间的父子关系,在类型信息学习不足的情况下,通过类型间的关系对细粒度类型的信息进行补充。例如类型/person、/person/artist、/person/artist/author 之间通过父子关系补充细粒度类型语义上的相似信息,因这类关系只包含类型之间的语义相似信息,无法学习到细粒度类型间的语义区别信息。

经调研发现,细粒度实体分类使用的细粒度类型集大多数参照 Freebase 知识库的实体类型系统构建,同时 Freebase 中细粒度类型的大量实体之间存在着丰富的关系,这些关系可以为细粒度类型间的语义区别信息的学习提供有效信息。例如,细粒度类型之间关系(姚明, Instance_of, /person/athlete), (姚明, Played_for, 上海大鲨鱼), (上海大鲨鱼, Instance_of, /organization/sports_team) 中,细粒度类型/person/athlete 和/organization/sports_team 对应的实体之间存在的 played_for 关系,该关系为细粒度类型/person/athlete、/person/doctor、/person/director 的语义区别信息的学习提供了有用的信息。

为了利用细粒度类型间语义区别信息解决细粒度类型预测错误的问题,本文提出了基于类型间关系学习的细粒度实体分类方法,以细粒度类型集为输入,设计算法在 Freebase 中获取细粒度类型之间的关系数据,从获取的关系数据学习细粒度类型间语义区别信息,然后结合实体提及及其上下文的特征信息,进行细粒度类型实体分类。本文的主要贡献概括如下。

(1) 提出了基于类型间关系学习的细粒度实体分类方法 TypeR-FET,该方法学习了细粒度类型间语义区别信息,并结合实体提及及其上下文的特征信息,进行细粒度类型实体分类,解决现有方法存在的细粒度分类精度不高、部分实体难以有效分类的问题。

(2) 提出了将细粒度类型之间的大量关系信息应用到细粒度实体分类任务的方法。以细粒度类型集为基础,从知识库中获取细粒度类型间的关系数据,采用知识表示学习方法学习细粒度类型及类型

之间关系的嵌入表示,为细粒度实体分类提供细粒度类型之间的语义区别信息。

(3) 用细粒度实体分类任务的常用 BBN, OntoNotes, Figer 数据集,对 TypeR-FET 方法进行细粒度实体分类实验,性能指标较当前方法均有提升。表明通过 TypeR-FET 方法学习了细粒度类型间的语义区别信息,能够达到提升细粒度实体分类准确率的效果。

1 相关工作

细粒度实体分类任务提出以后,学者在分类方法、句子文本信息的向量表示、细粒度类型的表示几个方面开展研究。

分类方法方面,2012 年文献[12]采用人工提取的实体提及及其上下文中 Tokens、Word Shape、Part-of-Speech tags、Length、Contextual unigrams 等自然语言特征,结合实体的类型形成特征向量,用感知机线性分类器方法预测实体类别,采用预测向量和真实向量的差对模型进行优化。根据细粒度类型的层级结构特性,2014 年文献[13]采用层级分类方法进行实体分类。用 local 和 flat 2 种计算方法进行实验。Local 为每个细粒度类型构建一个二类分类器,用策略保证了预测类型的一致性;Flat 则为所有细粒度类型训练一个多类别分类器预测实体的类型。在国内,研究人员针对细粒度实体类型的标注数据缺少的问题进行研究。2020 年文献[14]采用原型网络方法,基于细粒度类型的少量标注数据提出了细粒度实体分类方法。2020 年文献[15]针对缺少标注语料的实体分类任务,提出了基于迁移学习的细粒度实体分类方法。2020 年文献[16]提出了一种采用动态阈值的层级实体分类方法。在训练过程中,运用细粒度类型间的层级结构信息设计了正类型和负类型之间排序损失函数;在预测过程中,运用细粒度类型间的层级结构信息定义一个按父子关系搜索的解码器,该解码器将已预测类型的子类型作为候选类型进行实体类型的预测。该类研究的缺点是将细粒度类型看成相互独立的类型标签,造成细粒度类型之间的相关性学习不足。

句子文本信息的向量表示方面,2015 年文献[17]提出将实体特征向量和实体类型映射到同一低维空间,用实体特征和实体类型在低维空间的距离建模实体类别。2016 年文献[18]采用长短期记忆神经网络对实体及其上下文学习向量表示,用注意力机制学习上下文中每个字的权重,建模实体及其文本上下文的特征,通过逻辑斯蒂回归模型进行实体文本特征和实体类型间映射关系学习。2017 年文献[4]提出多级表示的学习方法,对实体在互补的 3 个层次上进行表示学习,即字符(在实体名称中的字符)、单词(在实体名称中单词的嵌入表示)和实体(实体的嵌入表示)。实验证实,层次间的表示信息可以互补,3 个级别的联合表示改进了现有的嵌入表示方式。2019 年文献[19]提出基于实体提及的注意力机制学习实体的低维度表示,在学习过程中,利用类型的二相关性和共现类型之间的隐含语义相关性学习了一个低维度表示向量,并用该向量重构类型向量。同时设计了一种混合类型分类器模型进行实体分类。这类研究的缺点是过于偏重学习好的向量表示,缺乏对细粒度类型间关系信息的学习。

在细粒度类型的表示方面,2017 年文献[20]结合符号语义和分布式语义提出一种新的无监督实体类型表示学习框架。从一般语义表示、特定上下文表示、基于知识库中知识的表示 3 个方面学习每个实体类型的表示。2018 年文献[21]提出一种利用实数和复数双线性映射方法学习细粒度类型间父子关系的模型,该模型和其他细粒度类型间父子关系学习方法比较,预测性能指标有一定的改进。该类研究存在的问题是只学习了细粒度类型间的共现关系、父子关系等体现语义相似性的关系,忽视了细粒度类型间的语义区别信息。

本文结合细粒度实体分类在分类方法、句子文本信息的向量表示、细粒度类型的表示等方面有效的最新研究成果,设计获取知识库中细粒度类型之间的关系信息算法,从获取关系中学习细粒度类型间的语义区别信息,进一步结合实体提及及其所在句子的上下文中特征,提出基于相似度排序的细粒度实体分类方法。

2 模型介绍

本节首先介绍模型中用到的各类信息的数学表示。细粒度实体分类任务给出标注了实体的句子,在树形结构的细粒度类型集上,根据学习到的实体提及及其上下文的语义信息,为实体确定若干个实体在句子中所具有的语义类型。标注了实体的句子 x 用三元组表示,即 $x = (W, l, r)$,其中 $W = (w_0, \dots, w_{n-1})$ 为实体提及所在的句子, n 为实体提及所在句子的长度, $[l, r]$ 为实体提及位置下标, l 为实体提及的左边位置序列在句子中的下标, r 为实体提及的右边位置序列在句子中的下标,句子的下标从 0 开始。实体提及用 $m = (w_l, \dots, w_r)$ 表示。细粒度类型实体分类任务的树型结构的细粒度类型集用 Y 表示,其结构如图 1 所示。

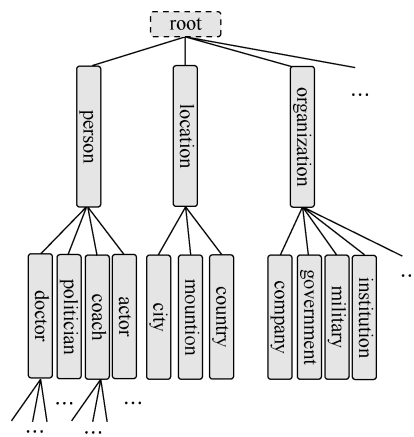


图 1 细粒度类型集

实体提及 m_i 在句子中的细粒度类型集合用 Y_i 表示, $Y_i = \{y_i | y \in Y\}$ 。

本文的模型框架如图 2 所示,从左到右包括实体特征的表示学习,细粒度类型间关系数据的获取及表示学习、实体特征和细粒度类型嵌入表示之间的相似度打分、实体特征和细粒度类型间关系的相似度打分和全局损失函数。实体特征的表示学习先通过预训练 ELMo (embeddings from language models) 语言模型得到实体提及的基础嵌入表示和实体提及上下文的基础嵌入表示,再采用注意力机制学习得到实体提及的最终嵌入表示 E_m 和实体提及

上下文的最终的嵌入表示 E_c , E_m 和 E_e 拼接形成实体的特征信息 F_e ; 细粒度类型间关系数据的获取以细粒度类型集为输入, 在 Freebase 知识库中获取细粒度类型实体之间关系 *relation*, 组成模型需要的三元组 ($head_{type}, relation, tail_{type}$) 信息, 其中三元组的头尾实体是 $head_{type} \in Y, tail_{type} \in Y$ 。关系的表示学习采用 *TransE* 方法训练得到细粒度类型的嵌入表示 E_l 和细粒度类型间关系的嵌入表示 E_r ; 实体特征采用 2 层全连接神经网络进行细粒度类型的语义提取后, 和

细粒度类型嵌入表示通过内积运算进行相似度打分; 实体特征和细粒度类型之间的关系的相似度打分计算中, 先通过细粒度类型间关系的表示 E_r , 采用双层神经网络监督学习实体提及及其上下文中隐含的细粒度类型间关系的语义信息 E'_r , 再用学习得到的 E'_r 和细粒度类型的嵌入表示 E_l 之间采用内积计算相似度打分。最后, 用前面几步的损失目标函数, 通过权重加和形成全局损失函数, 整体对模型进行联合优化。

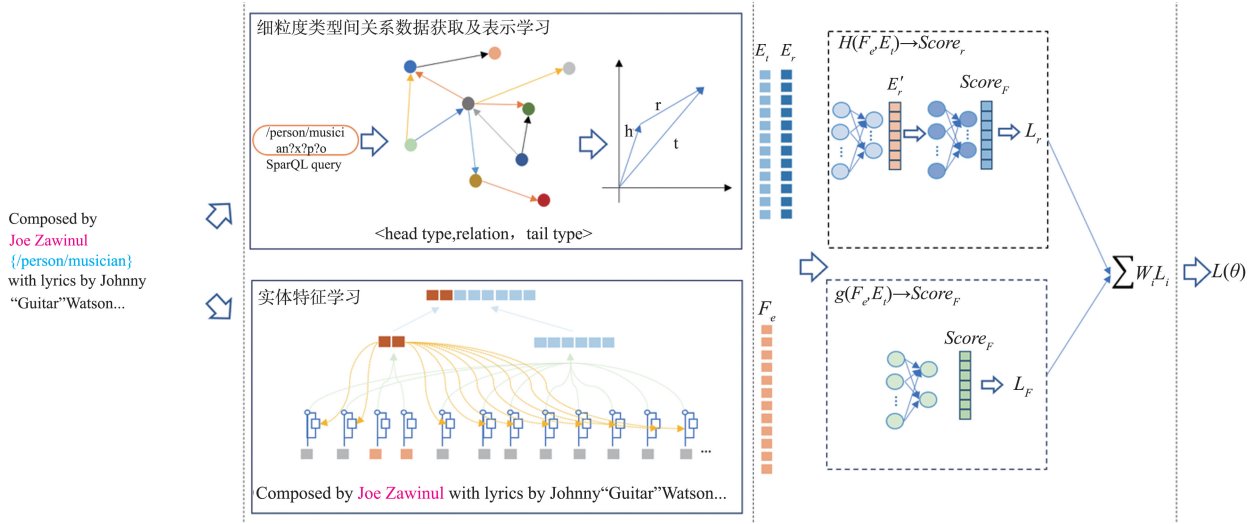


图 2 TypeR-FET 模型的整体框架

2.1 实体特征表示学习

实体特征是实体提及及其上下文文本中的语义信息。已有研究人员采用 Glove (global vectors for word representation) 词向量^[8]、ELMo 预训练模型^[22]、Bert (bidirectional encoder representation from transformers) 预训练模型进行文本信息的向量化表示对比研究^[19]。借鉴已有的研究成果, ELMo 预训练模型能较好地学习字符、词、短语多个粒度的语义信息, 因此采用 ELMo 预训练模型对实体提及及其上下文信息进行向量化表示。假设实体提及所在的句子由 n 个词组成, 采用 ELMo 语言模型进行向量化表示后生成句子的向量 $\{e_{w_1}, e_{w_2}, e_{w_3}, \dots, e_{w_n}\}$, 其中 $e_{w_i} \in \mathbb{R}^{d_w}$ 是句子中词的向量表示, d_w 是向量表示的维度。为了学习到实体提及中对细粒度类型比较重要的信息, 本文采用文献[16]提出的实体提出词注意力机制的类似方法。将实体提及向量表示后, 采用线性

映射学习表示的关键信息, 再通过 *maxpooling* 方法取每一维度最大值。形成实体提及的嵌入表示 E_m , 如式(1)所示。

$$E_m = \maxpooling(A_l e_{w_l}, \dots, A_r e_{w_r}) \quad (1)$$

其中, $A_1, \dots, A_r$ 为学习的线性映射权重。

为了学习到实体提及上下文中对细粒度类型比较重要的信息, 实体提及的上下文表示采用基于实体提及的注意力机制。注意力的权重参数 β_j 采用 *Softmax* 计算实体上下文中每个词的权重, 如式(2)所示。

$$\beta_j = \text{softmax}(B_c \tanh(W_c(e_{w_j} \oplus E_m))) \quad (2)$$

其中 $j \notin (l, r)$ 。实体的上下文嵌入表示如式(3)所示。

$$E_c = \sum_{j=s}^h \beta_j e_{w_j} \quad (3)$$

其中 s 为实体提及所在句子中去掉实体提及之后所剩词的最小位置坐标, h 为最大位置坐标。

将实体提及的嵌入表示和实体提及的上下文嵌入表示拼接起来形成实体特征的表示,即 $F_e = [E_m; E_c] \in \mathbb{R}^{2d_w}$ 。

2.2 细粒度类型间关系的获取及关系的表示学习

细粒度实体分类任务的数据中细粒度类型之间的关系是共现关系,有研究者在建模实体语义信息和细粒度实体类型间的映射关系的^[8-10,18]过程中,设计父子类型间语义信息传递规则,对细粒度类型树形结构上具有父子关系细粒度类型,学习细粒度类型之间语义上的相似信息。因缺少体现细粒度类型之间语义区别信息的数据,现有研究没有针对语义区别信息的学习方法。然而,细粒度实体分类任务需要对实体确定粒度更细的类型,显而易见,要求模型能辨别细粒度类型。细粒度类型间的语义区别信息能为模型辨别细粒度类型提供语义信息,通过学习这类信息解决细粒度分类精度不高、部分实体难以有效分类的问题。

针对缺少体现细粒度类型之间语义区别信息的数据问题,本文调研了细粒度类型分类任务数据集的构造过程,发现常用数据集的细粒度类型来自于 Freebase 知识库^[12-13,18]的类型系统。同时 Freebase 中有大量的类型间的关系数据。如果能获取并利用这类关系数据,可以为细粒度类型之间的语义区别信息学习提供解决方法。

受到 RDF (resource description framework) 数据上类型推断方法^[23]研究的启发,本文使用基于 RDF 数据的查询语言 SPARQL (SPARQL protocol and RDF query language),设计了 Freebase 知识库中细粒度类型之间关系获取算法,查询获取细粒度类型之间的关系形成包含细粒度类型及细粒度类型间关系信息的数据集合。算法的输入 $t \in Y$,其中 Y 为细粒度类型集。算法的输出 G_t 是以细粒度类型为节点,细粒度类型间关系为边的知识图谱 G_t 。算法的核心包括 2 部分,先后分别解决了细粒度类型关系获取中的 2 个问题。第 1 个问题为将数据集中的细粒度实体类型映射为 Freebase 知识库的实体类型,根据文献^[12,13]中列出的数据集的细粒度类型构造过程,通过细粒度类型关键字可以进行细粒度实体类型的映射,由算法第 4~8 行完成。时间复杂度

为 $O(s)$,其中 s 为数据集的细粒度类型数。第 2 个问题是数据细粒度类型对应的 Freebase 知识库的实体类型构造查询语句查询细粒度实体类型间关系。由算法第 9~13 行完成。时间复杂度为 $O(s^2)$ 。因此,算法 1 的整体的时间复杂度为 $O(s^2)$ 。算法的计算代价主要为 SPARQL 查询的处理部分,选用 Freebase、DBpedia 等开放知识库在线查询服务使用的 Virtuoso 工具,已验证能为算法提供稳定的计算服务。根据分析算法 1 可以用于处理细粒度类型间关系获取。

算法 1 细粒度类型间关系的获取方法

输入:数据集上细粒度类型集合 Y

输出:细粒度类型间关系三元组集合 G_t

```

1  数据集上细粒度类型对应的 Freebase 中使用类型
   集  $F_t \leftarrow \emptyset$ 
2  数据集上类型和  $F_t$  中类型的映射字典  $M_t \leftarrow \emptyset$ 
3  为  $Y$  的所有粗类型构造超级父节点形成树  $T_Y$ 
4  for  $t \leftarrow$  深度优先遍历  $T_Y$  do
5      采用 type. object. type 关系关键字和  $t$  获取 Freebase
       中的对应类型  $t_{fb}$ 
6       $F_t \leftarrow t_{fb}$ 
7       $M_t \leftarrow t$  和  $t_{fb}$  对应关系
8  end for
9  for  $head_{type}, tail_{type} \leftarrow F_t$  do
10     生成头实体类型和尾实体类型为  $head_{type}, tail_{type}$  的
       SPARQL 查询
11     运行查询获取生成  $(head_{type}, r, tail_{type})$ 
12      $G_t \leftarrow (head_{type}, r, tail_{type})$ 
13 end for
```

算法 1 获得的 G_t 包含的大量的细粒度类型间关系数据,学习 G_t 中实体和关系的表示,可以学习到含有细粒度类型间语义区别信息的嵌入表示。

在知识表示学习方面的研究成果中,TransE 方法已在知识图谱补全、关系抽取等任务中取得了较好的成果。并且 QUOTE TransE 将知识图谱中的关系看作是头尾实体之间的某种平移向量,比较切合模型后续的实体特征和细粒度类型间相似度计算的场景,所以选用 TransE 方法学习 G_t 中细粒度类型和类型间关系的嵌入表示。在本文方法中,对三元组 $(head_{type}, r, tail_{type})$,TransE 将 r 的向量表示 E_r 作为

头实体向量表示 $E_{\text{head}_{\text{type}}}$ 和尾实体向量表示 $E_{\text{tail}_{\text{type}}}$ 之间的平移。学习目标函数如式(4)所示。

$$f(\text{head}_{\text{type}}, r, \text{tail}_{\text{type}}) = -\|\text{head}_{\text{type}} + r - \text{tail}_{\text{type}}\|_2 \quad (4)$$

学习得到细粒度类型集 Y 中类型的嵌入表示 E_i 和 G_i 中细粒度类型之间关系的嵌入表示 E_r 。

2.3 实体特征与细粒度类型的相似度计算

细粒度实体分类任务面临着细粒度类型标注数据少的难题,然而多标签分类方法需要大量的标注数据学习类型间的分类边界,该任务在数据方面的实际情况造成分类方法难以达到好的分类效果,现有直接采用分类模型的研究方法^[8-9,17-18]存在这方面的问题。最近学者们提出混合分类模型、排序模型^[16]等多种建模方式。本文结合最近研究的成果,从信息检索的角度建模细粒度实体分类任务,将实体特征看作是检索关键字,而将细粒度类型作为匹配的文档进行相似度排序,取排序最前的细粒度类型作为预测类型。具体而言,先根据细粒度类型树形结构设计相似度得分排序规则。根据细粒度类型树形结构,粒度更细的类型具有更准确的语义信息,采用排序规则让粒度较细的类型获取较高的相似得分。采用排序方法如式(5)所示。

$$\text{Score}(x, y) > \text{Score}(x, \hat{y}) > \text{Score}(x, y') \quad (5)$$

式中, $\text{Score}(x, y)$ 是实体特征与细粒度类型的相似度得分, y 是实体的正类型, $\hat{y}$ 是实体正类型的父亲类型, y' 是在细粒度类型集中抽样的负类型。在训练过程中,根据实体的正确类型,结合相似度得分排序规则训练细粒度类型相似度打分函数。在预测过程中,直接对实体特征计算每个细粒度类型的相似度得分,并取排序最前面的细粒度类型作为实体的预测类型。

在实体特征与细粒度类型的相似度得分计算方面,根据前几节获得的实体提及及其上下文的嵌入表示、细粒度类型的嵌入表示和细粒度类型间关系的嵌入表示,分实体特征和细粒度类型相似度得分、实体特征和细粒度类型间关系相似度得分部分计算。总的得分为

$$\text{Score}(x, y) = w_i G(\mathbf{F}_e, \mathbf{E}_i) + w_r H(\mathbf{F}_e, \mathbf{E}_r) \quad (6)$$

其中, $G(\mathbf{F}_e, \mathbf{E}_i)$ 是实体特征和细粒度类型之间的相

似度得分, $H(\mathbf{F}_e, \mathbf{E}_r)$ 是实体特征和细粒度类型间关系的相似度得分, w_i 和 w_r 是超参。

2.3.1 实体特征和细粒度类型的相似度得分

采用细粒度类型之间的关系数据学习得到细粒度类型表示 E_i 初始化细粒度类型的嵌入表示,结合训练数据中细粒度类型对应的实体特征更新细粒度类型嵌入表示,让细粒度类型的嵌入表示学习到数据集中本地化的语义信息,新的细粒度类型嵌入表示为 e_y 。

有学者研究发现,将实体提及及其上下文的嵌入表示拼接形成的实体特征表示 $\mathbf{F}_e$ 通过奇异值分解方法可以学习到共现类型之间的相互关系。受到该研究的启发,本文采用 2 层全连接神经网络学习细粒度类型之间的相互信息,计算过程如式(7)所示。

$$\bar{\mathbf{F}}_e = \tan(\mathbf{W}_{l2} \tan(\mathbf{W}_{l1} \mathbf{F}_e)) \quad (7)$$

其中 $\mathbf{W}_{l1}$ 和 $\mathbf{W}_{l2}$ 是全连接神经网络学习的权重矩阵。接着采用得到的新的实体特征的信息 $\bar{\mathbf{F}}_e$, 进行细粒度类型的相似度打分计算,计算方法如式(8)所示。

$$G(\mathbf{F}_e, \mathbf{E}_i) = \bar{\mathbf{F}}_e \cdot \mathbf{e}_y \quad (8)$$

2.3.2 实体特征和细粒度类型间关系的相似度得分

从图 1 可以看出细粒度类型之间的关系对细粒度实体分类提供了有用信息,这类信息需要通过实体对应的细粒度类型建立对应关系,因测试数据中看不到实体对应的细粒度类型而无法在细粒度实体分类任务中直接使用该信息。最近有研究人员在基于知识库中链接实体的注意力计算过程中面临类似的问题,提出了通过全连接神经网络模型从实体特征中学习链接实体嵌入表示方法。受到该研究的启发,采用 2 层全连接神经网络学习实体提及及其上下文的信息与知识图谱 G_i 中细粒度类型之间关系的映射函数,采用式(9)来表达。

$$\mathbf{E}'_r = \tan(\mathbf{W}_{r2} \tan(\mathbf{W}_{r1} \mathbf{F}_e)) \quad (9)$$

其中 $\mathbf{W}_{r1}$ 和 $\mathbf{W}_{r2}$ 为需要学习的参数矩阵。

在映射函数训练过程中,对于训练数据 x_i, x_i 中实体提及对应的细粒度类型为 $Y_i = \{t_1, \dots, t_s\}$, 其中 s 为实体提及的细粒度类型数。实体对应的细粒度

类型间关系集合 R 为每个细粒度类型 t_s 关系的并集,则实体对应的细粒度类型之间关系的表示 r 采用式(10)计算。

$$r = \sum_{j=1}^M a_j \mathbf{E}_r \quad (10)$$

为了训练实体提及及其上下文的信息和知识图谱 G_i 中细粒度类型之间关系的映射函数,采用平方损失目标函数 $loss_r$,如式(11)所示。

$$loss_r = - \sum \|r - \mathbf{E}'_r\|^2 \quad (11)$$

接着采用获得的新的实体特征的信息 $\bar{\mathbf{F}}_e$,进行知识图谱 G_i 中细粒度类型之间关系的相似度打分,打分函数如式(12)所示。

$$H(\mathbf{E}_r, y) = \mathbf{E}'_r \cdot \mathbf{e}_y \quad (12)$$

2.4 语义动态阈值的细粒度类型预测

在细粒度类型的预测过程中,根据实体特征计算的细粒度类型的相似度得分,可以直接取排序最前的细粒度类型。现有研究[9-11,17]多数采用这种方法。

本文的方法为了使模型以较大的概率确定具有更准确语义信息的层级树叶子节点类型,采用了式(5)的排序方法,仅仅使用这种策略可能造成预测出多余的叶子节点细粒度类型的问题。借鉴最近研究的动态阈值方法,在细粒度类型预测过程中采用和细粒度类型语义相关的动态阈值:

$$threshold(x, y) = S(x, p_y) \quad (13)$$

其中 p_y 为 y 在细粒度类型层级树的路径。既可以解决模型在细粒度类型预测上的缺陷,又通过动态阈值提高了模型预测方法上的泛化能力。结合实体特征和细粒度类型相似度计算步骤,设计式(14)来计算细粒度类型预测中动态阈值。

$$S(x, p_y) = Score(x, \hat{y}) + th \quad (14)$$

其中 th 为模型实验确定的超参, $Score(x, \hat{y})$ 为细粒度类型 y 的父类型的相似度得分。

实体的细粒度类型预测过程采用算法2。算法2采用多类别多标签层级分类思想,根据层级计算细粒度类型的动态阈值和细粒度类型预测。算法的时间复杂度为 $O(l)$,其中 l 为细粒度类型的层级数。

算法2 细粒度类型预测方法

输入:实体在细粒度类型集合 Y 上打分 $Score(x, \cdot)$

输出:实体的预测粒度类型集合 Y'

```

1   $Y \leftarrow \emptyset$ 
2  类型和类型动态阈值映射字典  $T_h \leftarrow \emptyset$ 
3  类型和父亲类型映射字典  $P_t$ 
4  细粒度类型层级和细粒度类型映射  $L_t$ 
5  for  $l, Y_l \leftarrow$  深层级  $l$  从高到低遍历  $L_t$  do
6     $t_p \leftarrow$  在  $P_t$  中取  $Y_l$  类型的父亲类型
7     $T_h \leftarrow l$  和  $t_p$  中类型  $Score(x, t_p) + th$  的对应关系
8  end for
9  for  $l, Y_l \leftarrow$  深层级  $l$  从低到高第遍历  $L_t$  do
10    $tsh_l \leftarrow$  在  $T_h$  中取  $l$  层细粒度类型阈值
11    $Sc_y \leftarrow Y_l$  中类型  $Score(x, t_p)$ 
12    $Y \leftarrow Sc_y$  中阈值超过  $tsh_l$  类型
13 end for
```

2.5 目标优化

在训练过程中,采用 hinge loss 计算损失函数,具体为

$$loss(x_k, Y_k, Y'_k) = \sum_{y_l \in Y_k} \sum_{y'_l \in Y'_k} \max \{0, \varepsilon - F(x_k, y_l) + F(x_k, y'_l)\} \quad (15)$$

其中, Y_k 是实体 x_k 在训练数据中的正确的细粒度类型集合, Y'_k 是在细粒度类型集上根据 Y_k 中细粒度类型的层级为实体 x_k 抽取的错误细粒度类型集合。例如, /person 的负样例取同层级的其他 /location /organization /gpe /event 等类型,而 /person/actor 的负样例取同一层的 $\text{/person/politician}$ /location/country 等同一级别的类型。

为了在实体特征与细粒度类型的相似度计算过程中,按照式(5)排序方法对细粒度类型进行打分,采用 hinge loss 方法设计了父子类型之间的损失函数,公式为

$$loss(x_k, Y_k, \hat{Y}_k) = \sum_{y_l \in Y_k} \sum_{\hat{y}_l \in \hat{Y}_k} \max \{0, w_s \varepsilon - F(x_k, y_l) + F(x_k, \hat{y}_l)\} \quad (16)$$

其中 $w_s \in (0, 0.5)$ 范围内取值的超参, $\hat{Y}_k$ 是实体

x_k 在训练数据中标注的细粒度类型集合的父类型集合。

综合式(15)和(16)2个损失函数、实体提及及其上下文的信息与知识图谱 G_i 中细粒度类型之间关系的映射函数的损失函数式(11),最终的损失函数为

$$L(\theta) = \text{loss}(x_k, Y_k, Y'_k) + \text{loss}(x_k, Y_k, \hat{Y}_k) + \text{loss}_r \quad (17)$$

采用文献[24]提出的 AdamW 优化方法训练模型,联合学习实体特征和细粒度类型表示的相似度打分、实体特征和细粒度类型关系信息的相似度打分及实体信息和细粒度类型间关系语义相关性学习。

3 实验

3.1 实验数据和参数设置

本文使用细粒度实体分类任务上比较通用的3个数据集 BBN、OntoNotes 和 Figer 进行实验。数据集提出后出现了多种版本,为了对比实验的严谨性,本实验采用文献[18]、[16]实验中使用的版本,其中数据集的统计信息如表1所示。

表1 各数据集统计信息

数据集名称	训练集	验证集	测试集	类型数
BBN	84 078	2000	13 766	56
OntoNotes	251 039	2202	8963	89
Figer	2 000 000	10 000	563	113

首先,本文的方法获取了各个数据集上细粒度类型在 Freebase 中的类型间的关系数据,具体通过存储了 Freebase 数据的开源 SPARQL 语言查询工具 Virtuoso^[25],对3个数据集中的细粒度类型运行2.1小节算法获取类型间的关系,形成细粒度类型之间的关系数据 G_i 。为了本文方法能通过类型间关系学习到更加丰富的语义信息,算法遍历细粒度类型之间的所有可能组合,获取 Freebase 知识库中尽可能全面的类型间关系。关系数据统计信息如表2所示。

表2 各数据集类型间关系统计

数据集名称	数据集类型数	获取的细粒度类型间关系数	关系的种类
BBN	56	118 533	2943
OntoNotes	89	374 864	3837
Figer	113	421 257	4343

模型的参数包括实体上下文的分布式表示维度 d 、细粒度类型表示维度 d_t 、细粒度类型间关系表示维度 d_r 、hinge loss 函数所用的间隔参数 ε 和模型训练的迭代次数 p 。其中实验使用的最佳参数如表3所示。

表3 模型使用参数

数据集名	d	d_t	d_r	ε	w_s	p
BBN	1024	1024	1024	1	0.1	35
OntoNotes	1024	1024	1024	1	0.25	25
Figer	1024	1024	1024	1	0.1	30

3.2 基于类型间关系信息的细粒度实体分类

在模型评估方面,本文沿用了细粒度实体分类任务^[12]中常用的严格准确率(Acc)、宏平均 F1 (Ma-F1)和微平均 F1 (Mi-F1)3个指标。严格准确率(Acc)是测试集中实体的正确类型和预测类型完全相同的数据量在整个测试集中的占比。宏平均 F1 (Ma-F1)是测试集上每个实体的 F1 值在整个测试集上的平均值。微平均 F1 (Mi-F1)为先计算整个测试集上准确率和召回率的平均值,然后再计算 F1。表4具体展示了本文的模型和最近研究的其他模型在3个数据集中的性能指标实验对比情况。从表中可以看出,和当前最新的研究相比较,本文模型性能指标在各个数据集上都有所提升。证明从细粒度类型之间的关系数据中学习的细粒度类型间区别语义信息,能对细粒度类型实体分类任务提供有效的语义信息。

3.2.1 用细粒度类型间关系学习的细粒度类型嵌入表示进行细粒度实体分类分析

为了分析从细粒度类型之间关系数据 G_i 学习的嵌入表示在细粒度实体分类的应用效果,通过 G_i 中细粒度类型之间的关系学习到细粒度类型表示后,固定细粒度类型的嵌入表示,本文模型在后续训练

过程中不更新细粒度类型的嵌入表示,并且只采用实体特征和细粒度类型的相似度进行实验,实验的性能指标如表 5 所示。从表中可以看出,只使用了细粒度类型间关系学习的细粒度类型的嵌入表示,本文方法(表中的 TypeR-FET-Eonly)实验结果优于 2019 及以前的研究,但是比 Chen 等人^[16]的方法差,分析原因主要是 Chen 等人的方法通过实体在数据集中的实体提及及其上下文语义信息学习,获得了数据集中数据来源方面的本地化的语义信息;而本文方法仅用 Freebase 知识库中类型关系数据,学习得到细粒度类型比较通用的语义信息。同时,证明在细粒度实体分类任务中,数据集本地化的语义信息对任务有一定的影响。为了任务达到更好的效果,在具体数据集中需要学习数据集中本地化的语义信息。

3.2.2 基于细粒度类型间的关系的嵌入表示进行细粒度实体分类分析

采用关系表示学习方法学习细粒度类型间关系数据 G_t ,得到 G_t 中细粒度类型之间关系的嵌入表示 E_r 。为了分析细粒度类型之间关系的嵌入表示 E_r 对细粒度实体分类的影响,在模型中先通过实体所在

的句子文本信息预测 G_t 中细粒度实体类型间关系综合嵌入表示 E'_t ,再通过 E'_t 预测细粒度类型。整个建模过程直接应用实体特征和细粒度类型间关系的相似度得分进行细粒度实体分类,不使用 TypeR-FET 模型的整体框架(图 2)中 $G(F_e, E_t)$ 的相似度打分。实验的性能指标如表 6 所示。从表中可以看出,通过 G_t 中细粒度类型间关系的嵌入表示进行细粒度实体分类,只有在 OntoNotes 数据集上稍好于 2016 的研究成果,总体上差于直接学习实体特征和细粒度类型间映射关系的方法。证明细粒度实体分类任务解决方法的步长对实验性能影响较大,虽然能学习到额外的语义信息,却因为问题建模过程长而丢掉了一些有用信息。

针对 3 个数据集上实验结果表现差别比较大的情况,结合训练数据中统计信息(如表 7 所示)进行分析,发现实验效果和数据集类型 2 层及以上的类型数目成正相关,BBN 为 39,OntoNotes 为 81,Fi-ger 为 66。粒度较细的类型多的效果更好。证明通过 G_t 中细粒度类型间的关系的学习,让 2 层及以上的细粒度类型学习到更多的语义信息,提高了细粒度实体分类任务中较细粒度的性能。

表 4 模型性能指标

数据集	BBN			OntoNotes			Figer		
评估指标	Acc	Ma-F1	Mi-F1	Acc	Ma-F1	Mi-F1	Acc	Ma-F1	Mi-F1
FIGER 2012 ^[12]	—	—	—	—	—	—	0.532	0.699	0.693
Yogatama 2015 ^[17]	—	—	—	—	—	0.729	—	—	0.723
Proto-HLE 2016 ^[26]	0.694	0.745	0.743	0.493	0.682	0.612	0.504	0.666	0.652
Shimaoka 2017 ^[18]	—	—	—	0.517	0.709	0.649	0.589	0.779	0.749
Jin 2019 ^[11]	—	—	—	—	—	—	0.570	0.798	0.836
Chen 2020 ^[16]	0.752	0.797	0.805	0.556	0.714	0.671	0.593	0.777	0.789
TypeR-FET	0.753	0.800	0.806	0.558	0.725	0.686	0.614	0.772	0.755

表 5 细粒度类型间关系的学习嵌入表示进行细粒度实体分类

数据集	BBN			OntoNotes			Figer		
评估指标	Acc	Ma-F1	Mi-F1	Acc	Ma-F1	Mi-F1	Acc	Ma-F1	Mi-F1
Proto-HLE 2016 ^[26]	0.694	0.745	0.743	0.493	0.682	0.612	0.504	0.666	0.652
Shimaoka 2017 ^[18]	—	—	—	0.517	0.709	0.649	0.589	0.779	0.749
Jin 2019 ^[11]	—	—	—	—	—	—	0.570	0.798	0.836
TypeR-FET-Eonly	0.742	0.797	0.791	0.552	0.660	0.706	0.580	0.488	0.423

表 6 模型性能指标

数据集	BBN			OntoNotes			Figer		
评估指标	Acc	Ma-F1	Mi-F1	Acc	Ma-F1	Mi-F1	Acc	Ma-F1	Mi-F1
Proto-HLE 2016 ^[26]	0.694	0.745	0.743	0.493	0.682	0.612	0.504	0.666	0.652
TypeR-FET-Ronly	0.193	0.484	0.489	0.512	0.625	0.558	0.381	0.488	0.423

表 7 数据集中细粒度类型统计

数据集	BBN			OntoNotes			Figer	
类型层级	层 1	层 2	层 1	层 2	层 3	层 1	层 2	层 2
类型数	17	39	3	41	40	36	66	66

4 结 论

针对现有细粒度实体分类方法存在的细粒度分类精度不高、部分实体难以有效分类的问题,本文提出了一种基于细粒度类型间关系学习的细粒度实体分类方法。该方法基于细粒度类型之间的语义区别信息可为细粒度实体分类提供有效消息的思考,利用 Freebase 知识库解决细粒度实体分类中缺乏学习细粒度类型间语义区别信息的数据的问题,获取能用于细粒度类型语义区别信息学习的类型间的关系数据;采用关系表示学习方法学习了细粒度类型的区别语义信息,并设计模型联合学习细粒度实体类型之间的关系信息、实体提及及其上下文特征信息进行细粒度实体分类。实验表明,本文提出的方法有效利用了细粒度类型间语义区别信息,提升了细粒度实体分类的性能。此外,本文提出的基于细粒度类型集从 Freebase 中获取细粒度类型之间的关系的方法,为细粒度实体分类任务引入了细粒度类型间的关系信息。在实际应用中,该方法针对应用领域细粒度类型的标注数据较少问题,可从细粒度类型间的关系数据方面为细粒度实体分类补充有用信息。

缺乏高质量标注训练数据的情况下,提升细粒度实体分类的性能是该任务一直尝试解决的难题。本文提出从外部知识库获取类型间关系数据的方法,为将类型间关系信息应用于细粒度实体分类提供了新思路。同时,为借助关系推理、知识表示学习等方面的研究成果提高细粒度实体分类的性能提供了探索方向,可以作为今后在细粒度实体分类任务上研究的新方向。

参考文献

[1] MANNING C, SURDEANU M, BAUER J, et al. The stanford core NLP natural language processing toolkit [C]//Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations. Baltimore: Association for Computational Linguistics, 2014: 55-60.

[2] ERKAN G, RADEV R. LexRank: graph-based lexical centrality as salience in text summarization[J]. Journal of Artificial Intelligence Research, 2004, 22: 457-479.

[3] MINTZ M, BILLS S, SNOW R, et al. Distant supervision for relation extraction without labeled data [C] // Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-ACL-IJCNLP '09. Singapore: Association for Computational Linguistics, 2009: 1003-1011.

[4] YAGHOOBZADEH Y, ADEL H, SCHÜTZE H. Noise mitigation for neural entity typing and relation extraction [C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers. Valencia: Association for Computational Linguistics, 2017: 1183-1194.

[5] DONG X, GABRILOVICH E, HEITZ G, et al. Knowledge vault: a web-scale approach to probabilistic knowledge fusion[C]//Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: Association for Computing Machinery, 2014: 601-610.

[6] LYU S, CHEN H. Relation classification with entity type restriction[C]//Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021. Virtual: Association for Computational Linguistics, 2021: 390-395.

- [7] XIN J, LIN Y, LIU Z, et al. Improving neural fine-grained entity typing with knowledge attention[C] // Proceedings of the AAAI Conference on Artificial Intelligence. New Orleans; AAAI Press, 2018: 1-8.
- [8] REN X, HE W, QU M, et al. AFET: automatic fine-grained entity typing by hierarchical partial-label embedding[C] // Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. Austin; Association for Computational Linguistics, 2016: 1369-1378.
- [9] REN X, WU Z, HE W, et al. CoType: Joint extraction of typed entities and relations with knowledge bases[EB/OL]. (2016-10-27) [2021-12-03]. <https://arxiv.org/pdf/1610.08763.pdf>.
- [10] XU P, BARBOSA D. Neural fine-grained entity type classification with hierarchy-aware loss[C] // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). New Orleans; Association for Computational Linguistics, 2018: 16-25.
- [11] JIN H, HOU L, LI J, et al. Fine-grained entity typing via hierarchical multi-graph convolutional networks[C] // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong; Association for Computational Linguistics, 2019: 4969-4978.
- [12] LING X, WELD D S. Fine-grained entity recognition[C] // The 26th AAAI Conference on Artificial Intelligence. Toronto; AAAI Press, 2012.
- [13] GILLICK D, LAZIC N, GANCHEV K, et al. Context-dependent fine-grained entity type tagging [EB/OL]. (2014-12-03) [2021-12-03]. <https://arxiv.org/pdf/1412.1820.pdf>.
- [14] 任权. 基于原型网络的细粒度实体分类方法[J]. 中文信息学报, 2020, 34(12): 65-72.
- [15] 冯建周, 马祥聪. 基于迁移学习的细粒度实体分类方法的研究[J]. 自动化学报, 2020, 36(8): 1759-1766.
- [16] CHEN T, CHEN Y, VAN DURME B. Hierarchical entity typing via multi-level learning to rank[C] // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. Online; Association for Computational Linguistics, 2020: 8465-8475.
- [17] YOGATAMA D, GILLICK D, LAZIC N. Embedding methods for fine grained entity type classification[C] // Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). Beijing; Association for Computational Linguistics, 2015: 291-296.
- [18] SHIMAOKA S, STENETORP P, INUI K, et al. Neural architectures for fine-grained entity type classification [EB/OL]. (2016-06-03) [2021-12-03]. <https://arxiv.org/pdf/1606.01341.pdf>.
- [19] LIN Y, JI H. An attentive fine-grained entity typing model with latent type representation[C] // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong; Association for Computational Linguistics, 2019: 6196-6201.
- [20] HUANG L, MAY J, PAN X, et al. Liberal entity extraction: rapid construction of fine-grained entity typing systems[J]. Big Data, 2017, 5(1): 19-31.
- [21] MURTY S, VERGA P, VILNIS L, et al. Hierarchical losses and new resources for fine-grained entity typing and linking[C] // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Melbourne; Association for Computational Linguistics, 2018: 97-109.
- [22] ALI M A, SUN Y, LI B, et al. Fine-grained named entity typing over distantly supervised data based on refined representations[EB/OL]. (2020-04-07) [2021-12-03]. <https://arxiv.org/pdf/2004.03554.pdf>.
- [23] PAULHEIM H, BIZER C. Type inference on noisy RDF data. [C] // Advanced Information Systems Engineering. Berlin, Heidelberg; Springer, 2013: 510-525.
- [24] LOSHCHILOV I, HUTTER F. Decoupled weight decay regularization[EB/OL]. (2017-11-04) [2021-12-03]. <https://arxiv.org/pdf/1711.05101.pdf>.
- [25] OpenLink Software. Virtuoso[EB/OL]. [2021-12-03]. <https://dbpedia.org/sparql/>.
- [26] MA Y, CAMBRIA E, GAO S. Label embedding for zero-shot fine-grained named entity typing[C] // Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics. Osaka; Association for Computational Linguistics, 2016: 171-180.

Type relation learning for fine grained entity type classification

XI Pengbi^{* **}, JIN Xiaolong^{* **}, BAI Shuo^{***}

(^{*} Key Laboratory of Network Data Science and Technology,

Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

(^{**} School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 101408)

(^{***} Hundsun Technologies Inc, Hangzhou 310053)

Abstract

In order to support downstream tasks better, the aim of fine grained entity type classification is to assign one or more hierarchical and fine grained types to the entity or mention extracted in the processing of knowledge construction. Prior work has the problems of lower accuracy or being unable to assign the right types. Intuitively, the semantic distinction is useful of fine grained entity typing. There is no research on this aspect since the datasets of fine grained entity type contain no data for learning the semantic distinction between types. A method which learns the semantic distinction of fine grained types using the knowledge base Freebase is proposed, and the learned semantic distinction is used in the task of fine grained entity type classification. Specifically, SPARQL is used to acquire the relations data from Freebase, and the semantic distinction is learned from those acquired data, then the fine grained type is assigned to entity according to the learned semantic distinction of types together with the semantic information learning from mention and context. Experimental results and analysis on datasets demonstrate that the proposed model can learn the semantic distinction of types, and the results of the proposed model outperforms other state-of-the-art methods.

Key words: entity typing, fine grained type, knowledge graph construction, relation learning, multi-labels classification