doi:10. 16055/j. issn. 1672-058X. 2022. 0004. 002

基于细粒度实体分类的对比研究

周 祺,陶 皖
(安徽工程大学 计算机与信息学院,安徽 芜湖 241000)

摘 要:细粒度实体分类是一项多类别多标签任务,能协助广泛的下游任务(关系抽取、共指消解、问答系统等)提高工作效率、优化准确率,已成为自然语言处理领域的一个研究热点。针对传统的细粒度实体分类方法人工标注大型语料库难度大,准确率偏低等问题,研究人员提出了基于神经网络的细粒度实体分类方法,不仅能够解决人工标注费时费力的问题,而且可以提高分类的准确率。然而现有的神经网络模型大多需要远程监督的参与,在此过程中会引入噪声标签等问题,通过噪声标签处理方法能够有效抑制噪声标签对分类结果的影响,进一步提升分类性能。在相同评测数据集下,根据相同评价指标对比各类细粒度实体分类方法的性能,可以发现在细粒度实体分类领域中采用 BiLSTM 处理实体指称上下文,并通过注意力机制提取更为重要的特征,有助于提高细粒度实体分类方法的准确率、Macro F1 值和 Micro F1 值。

关键词:细粒度实体分类;神经网络;远程监督;噪声处理
中图分类号:TP391. 1 **文献标志码**:A **文章编号**:1672-058X(2022)04-0009-10

0 引 言

在自然语言处理(Natural Language Processing, NLP)中,实体分类是指为每一个实体分配一个指定的标签,这是一项非常重要而又基础的工作,在知识图谱的构建中扮演着重要的角色,作为构建知识图谱的基础性工作,实体分类的质量好坏直接影响整个知识图谱的可信度和可用性。传统的命名实体识别[1](Name Entity Recognition, NER)作为实体抽取的子任务为后续的工作奠定了坚实的基础,即将文本中的指称(即实体在具体上下文中的一段指代)抽取出来,并判断其在上下文中的类型为人、位置、组织、其他等粗粒度类型的过程。近年来,传统的命名实体识别被扩展到更深层次的细粒度实体类型。由于上游分配粗粒度的实体类型,后续选取实体间的候选关系就会复杂,相应的关系抽取任务会变得愈加困难,于是就促进了细粒度实体分类任务的研究。通过细粒度实体分类概念的引入,有效地将粗粒度的实体类型标签细化、层次化,从而使得下游任务(关系抽取、事件抽取、问答系统、实体推荐等)的工作效率降低,提高工作效率。

细粒度实体分类[2](Fine-grained Entity Typing, FET)在给定实体指称的情况下,依据其上下文给实体指称赋予一个或多个实体类型。在 FET 中,能够对目标实体类型进行更细致地划分,同时保证类型之间存在一定的层次关系。细粒度的实体类型表示

收稿日期:2021-05-17;**修回日期**:2021-06-18.
基金项目:安徽省教育厅高校自然科学重点项目(KJ2019A0158, KJ2019ZD15);国家级大学生创新创业项目(202010363098, 201910363076).
作者简介:周祺(1997—),女,安徽宿州人,硕士研究生,从事云计算与大数据处理研究.
通讯作者:陶皖(1972—),女,安徽芜湖人,教授,硕士,从事大数据与数据分析研究. Email: taowan@ ahpu. edu. cn.

可以为其他 NLP 任务提供更多的语义信息,有助于增强后续关系抽取和事件抽取等任务的指示性,提高问答系统、实体推荐等下游任务的工作效率。

传统的 FET 任务通常采用人工标注语料的方式,费时费力。随着实体类型数据集的不断增大,人工标注类型标签的难度增加、代价高昂,而且容易出错。为此将深度学习中的神经网络方法[3-5]运用到细粒度实体分类领域,可充分利用深度学习方法从大量训练语料中学习不同语料的语义特征,代替人工标注,从而提高细粒度实体分类的准确率。然而现有的基于神经网络的细粒度实体分类模型大多需要远程监督[6](Distant Supervision)的参与,由于远程监督链接到知识库中实体指称的所有标签,召回过程不可避免地会引入噪声问题,但过多的噪声使得训练模型性能变差,影响分类精度。为了缓解噪声标签产生的负面影响,有学者提出细粒度实体分类的标签噪声处理方法,如剪枝噪声标签[7]、划分数据集[8]等,能够有效地改善 FET 任务处理噪声标签的鲁棒性,促进了细粒度实体分类的进一步研究。

1 细粒度实体分类

Lee 等[9]首次在问答系统中对细粒度命名实体识别任务进行处理,提出利用条件随机场(Conditional Random Field,CRF)检测命名实体的边界,并使用最大熵(Maximum Entropy,ME)对实体进行分类,同时他们定义了 147 种细粒度的命名实体类型。但对于细粒度的语义命名实体识别与分类还没有系统的研究,因此 Ekbal 等[10]依赖大型文本语料库,获取细粒度的语义类型和实例,构建了细粒度命名实体识别与分类的数据集。为了扩展命名实体类型表示,Sekine[11]使用日本百科全书的知识创建了 200 种扩展命名实体类型,其中包含了扩展命名实体的丰富描述以及一系列的属性设计。与上述工作不同的是,Ling 等[2]针对细粒度实体分类任务,创建了经典数据集 FIGER,将本来只划分为 5~6 种类型的实体扩展到 112 种类型标签,通过远程监督的方式,获取维基百科词条中的实体类型信息,并根据 CRF 划分实体边界,最终由感知机算法完成多类别多标签任务,开辟了针对细粒度实体分类领域的新方法,为后续的研究工作提供了便捷。针对 FIGER 数据集中类型数量相对较少,一个实体通常只映射到一个类型的问题,Yosef 等[12]提出了在不同层次、数百种类型的基础上,利用层次分类法对来自 WordNet 中的大量实体类型自动计算扩展实体指称的类型,得到了非常精细的 505 种实体类型,形成了一个多标签的分级分类系统 HYENA。略显不足的是,HYENA 中的类型均来自 WordNet 中的子集,缺少重要的实体类型。为了弥补这一缺陷,FINET[13]不再限制实体类型,提取整个 WordNet 中超过 16 000 种类型,其中包括个人、组织和位置等。以往的研究大多依赖于人工标注的特征,而 Dong 等[14]首次采用深度学习的方式,使用词嵌入作为特征,通过监督方法将网页中的内容提取与现有知识库中的先验知识相融合,能够有效地提高识别实体类型的准确率。上述工作均为细粒度实体分类领域的研究奠定了基础,证实了其存在价值及重要意义,并为后续实验创建了基础实验数据集。

2 基于神经网络的细粒度实体分类

由于实体指称在知识库中所对应的类型较多,一般采用人工标注的方式保证样本的准确率,但这样人工成本耗费过多。随着知识库规模的急速增长,人工标注的方式已经无法跟上数据更新的速度,因此使用神经网络方法代替部分人工过程,以便提高细粒度实体分类的准确率和召回率。

2.1 基于卷积神经网络的细粒度实体分类

卷积神经网络(Convolutional Neural Network,CNN)[3]通常由输入层、隐藏层和输出层组成。首先在输入层处理多维数据,其次在隐藏层中可以利用卷积层、池化层和全连接层对输入数据进行特征提取、特征选择以及信息过滤,最后在输出层使用逻辑函数或 softmax 函数输出分类标签。

为提取实体信息用于知识库补全领域,Jia 等[15]提出一种学习实体指称及其上下文联合表示的卷积神经网络联合模型(Convolutional Neural Network Joint Model, CNNJM),在词嵌入平均化的基础上进行

一层卷积,通过最大池化操作获得最重要的特征,类似思想也用于句子分类[16]、事件抽取[17]领域中,CNNJM 更关注于实体本身的特征信息。然而实体指称的上下文蕴含着更丰富的信息,于是 Murty 等[18]通过对实体指称的上下文及位置信息进行卷积操作,之后进行最大池化处理,提取更多的上下文特征信息,有利于提高细粒度实体分类的准确率。

2.2　基于循环神经网络的细粒度实体分类

考虑 CNN 网络层次之间的关联性不强,且无法很好地学习自然语言数据的长距离依赖和结构化语法特征,因此卷积神经网络在后期自然语言处理中的应用要少于循环神经网络(Recurrent Neural Network,RNN)[4]。RNN 以序列数据为输入,在序列的演进方向进行递归操作,将所有循环单元按链式连接,主要包括输入层、隐藏层和输出层,隐藏层中添加了记忆细胞模块。与 CNN 有所不同的是,CNN 隐藏层之间的节点是无连接的,而 RNN 会对前面的信息进行记忆并应用于当前输出的计算中,即隐藏层之间的节点是有连接的。

研究证明,RNN 对符合时间顺序、逻辑顺序等序列特性的数据十分有效,能挖掘数据中的时序信息以及语义信息,但是由于权重累加过大,无法进行长期记忆的学习,可能导致结果失真、运算效率降低,因此长短期记忆(Long Short-Term Memory,LSTM)[5]网络应运而生。LSTM 网络通过精妙的输入门、遗忘门和输出门控制将短期记忆与长期记忆结合起来,选择性地记录或遗忘输入的信息,有利于提取重要的特征信息,得到更好的实验结果。

为了达到更高精度识别实体、细化实体类型的效果,Shimaoka 等[19]创新地使用 LSTM 学习实体指称的上下文表示,同时引入注意力机制,为双向长短期记忆网络(Bi-directional LSTM,BiLSTM)编码的上下文序列计算注意力权重,识别更具表达类型标签的信息,并使分类行为更具可解释性。随后,Shimaoka 等[20]将先前未考虑到的人工标注特征与模型学习到的特征结合在一起形成互补的信息,再次提高细粒度实体分类任务的准确率和召回率。根据知识库(Knowledge Base,KB)中有关实体的丰富信息,Xin 等[21]提出了基于知识库的注意力神经网络模型。该模型将实体指称的上下文向量投入 BiLSTM,通过计算注意力权重,输出上下文表示。与此同时,还将注意力机制运用到实体指称表示和来自知识库的实体表示,既考虑了实体指称与上下文的关系,也能够把实体指称与知识库中相关实体的关系代入其中。鉴于 FIGER、OntoNotes 中的实体类型仍不够精细,Choi 等[22]提出了超细粒度实体分类(Ultra-Fine Entity Typing,UFET),采用两层独立的 BiLSTM 处理上下文,并通过注意力机制和多层感知机(Multi-layer Perceptron,MLP)算法生成实体指称的上下文表示,有效地改进了细粒度实体分类的效果。同时创建了三层的超细粒度实体类型数据集 UFET,包括 9 种通用类型、121 种细粒度类型和 10 201 种超细粒度类型。

由于 LSTM 的强大功能,将其应用到自然语言处理领域的效果良好,此后的细粒度实体分类任务大多采用 LSTM 处理实体指称的上下文向量,以获取重要的上下文语义特征,为实体指称分配细粒度实体类型提供指示性信息。

3　基于噪声处理的细粒度实体分类

现有的大多数细粒度实体分类模型采用基于神经网络的实体分类模型,利用远程监督方法首先将语句中的实体指称链接到知识库中的实体,再把 KB 中实体的所有类型标签分配给实体指称的候选类型集。由于采用远程监督方法,分配类型标签时未考虑实体指称的上下文,会将无关的实体类型标签引入训练数据中,把这些无关的类型标签视为标签噪声。远程监督在对实体指称进行细粒度实体分类时会受到标签噪声和相关类型的限制,从而加大了后续分类模型对实体指称进行正确分类的难度,严重影响了细粒度实体分类模型的准确性和可信性。因此,对标签噪声进行有效处理,能净化训练数据集,使分类模型训练时能够高效学习实体类型标签,优化分类模型的准确性。本节介绍基于噪声处理的细粒度实体分类,主要分为基于规则划处理数据集、优化损失函数两部分。

3.1 基于规则处理数据集

由于FIGER[2]和HYENA[12]的训练集和测试集都是从Wikipedia中利用远程监督自动获取的,未经过任何的过滤和挑选。因此Gillick等[7]通过在训练集上采用启发式剪枝的方法来解决训练数据中出现的多余标签,用于完善训练数据的启发式方法删除了与单个实体关联的同级类型,仅保留了父类型;删除与在该类型集上训练的标准粗粒度类型分类器的输出不一致的类型;删除出现次数少于文档中的最小次数的类型。经过启发式规则能有效地改善人工标注数据的性能。但通过启发式规则剪枝噪声,会导致训练数据样本量减少,影响模型的整体性能,因此Ren等[8]提出自动细粒度实体分类模型(Automatic Fine-grained Entity Typing, AFET),对带有正确类型标签的实体指称和带有噪声标签的实体指称分别进行建模训练,另外还设计了一种新的部分标签损失算法,能利用噪声候选类型集中与实体指称相关的候选类型建模真实类型,并利用为指称所提取的各种文本特征逐步估计出最佳类型。然而,去噪过程和训练过程没有统一,这可能会导致误差传播,带来更多的复杂性。于是Zhang等[23]提出一种基于路径的注意力神经网络模型(Path-based Attention Neural Model, PAN)可以选择与每种实体类型相关的语句,动态减少训练期间每种实体类型的错误标记语句的权重,通过端到端的过程有效地减少类型标签噪声,并能在有噪声的数据集上实现更好的细粒度实体分类性能。为进一步改进噪声数据处理的效果,Abhishek等[24]参考AFET,构建了AAA模型,将训练数据分为干净集和噪声集,若训练数据实体的多个标签属于同一类别将其分为到干净集,反之则划分到噪声集。同时能联合学习实体指称及其上下文表示,并且在训练数据时使用变形的非参变量铰链损失函数,还运用迁移学习提高模型的有效性。

3.2 优化损失函数

3.2.1 铰链损失函数

铰链损失(Hinge Loss)函数是机器学习领域中的一种损失函数,可用于"最大间隔(Max-margin)"分类,经典公式如下:

$$L(y,y')=\max(0,\text{marg in}+y'-y) \tag{1}$$

其中,y是正例标签的得分,y'是负例标签的得分,两者间的差值用来预测两种预测结果的相似关系。

以往对细粒度实体分类中训练数据的噪声进行处理时将其剪枝或划分为不同的数据集,但是未充分考虑到细粒度实体分类系统处理噪声数据时的鲁棒性。于是Yogatama等[25]在WSABIE[26]的基础上,提出了学习特征和标签联合表示的模型K-WSABIE,将特征向量和标签映射到同一低维空间,学习特征和标签的联合表示。与此同时,在K-WSABIE中引入铰链损失函数,如下:

$$L(y,y')=\Re(rank(y))\max(0,1-y+y') \tag{2}$$

其中,y和y'含义如上,$\Re(rank(y))$使得正例标签的得分高于负例标签,彼此之间不产生竞争,有效提高模型应对噪声数据的鲁棒性。

为减少与上下文无关的噪声标签影响,Dai等[27]利用实体链接[28-29]改进细粒度实体分类模型,根据上下文、指称的字符以及用实体链接从知识库中获得的类型信息结合在一起灵活地预测类型标签,同时设计了一个变形的铰链损失函数防止训练后的模型过拟合弱标记数据,如下:

$$L(y,y')=\max(0,1-y)+\lambda\cdot\max(0,1+y') \tag{3}$$

其中,y和y'含义如上,λ为超参数,灵活地调整对负例标签的惩罚。

由于以往方法对实体指称独立建模,仅依据上下文分配实体类型标签,可能会妨碍信息跨越句子边界传递信息,为此Ali等[30]提出了一个基于边缘加权的注意力图卷积网络(Fine-Grained Named Entity Typing with Refined Representations, FGET-RR)。FGET-RR不仅分析具体的上下文信息,还侧重于对语料库中特定标签的上下文进行分析。另外,对于干净数据和含噪声数据分别设计铰链损失函数,如下:

$$L_{\text{clean}}=\text{Re}\ LU(1-y)+\text{Re}\ LU(1+y') \tag{4}$$

$$L_{\text{noisy}}=\text{Re}\ LU(1-y^*)+\text{Re}\ LU(1+y')$$

$$y^*=\arg\max\ y \tag{5}$$

3.2.2 交叉熵损失函数

交叉熵损失函数(Cross Entropy Loss)在机器学

习中主要用于衡量真实概率分布与预测概率分布之间的差异性,交叉熵的损失值越小,代表模型的预测效果就越好,如下:

$$L(p,q) = -\sum_{x} p(x)\log q(x) \tag{6}$$

其中,p 为真实概率分布,q 为预测概率分布。

与前人不同的是,Xu 等[31]对原本细粒度实体分类的多标签分类问题,转换为单标签分类问题,并且使用变形的交叉熵损失函数和分层损失函数来分别处理无关噪声标签以及过于具体的标签。变形的交叉熵损失函数根据实体指称的上下文自动过滤不相关的类型,如下:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\log y_i^{*}, y_i^{*} = \arg\max y \tag{7}$$

其中,N 为实体指称的数量,$p(y_i)$ 为预测的概率分布,当实体指称对应多个类型标签时,只选取具有最高概率的标签。分层损失函数能调整预测相关类型的步骤,使模型了解实体类型的层次结构,预测真实类型的父类型会比其他不相关的类型效果好,从而减轻过于具体标签的消极影响。

在 NFETC[31]的基础上,为避免文献[8、24、31]中使用部分标签损失的确认误差累积影响,Chen 等[32]提出使用压缩隐空间簇(NFETC-Compact Latent Space Clustering,NFETC-CLSC)来规范远程监督模型。对于干净的数据,压缩相同类型的表示空间;对于有噪声的数据,通过标签传播和候选类型约束来推断它们的类型分布,激发出更好的分类性能。以 KL 散度计算远程监督损失值,如下:

$$L = -\frac{1}{B}\sum_{i}^{B}\sum_{j}^{J} y_{ij}\log(1-y_{ij})_J \tag{8}$$

其中,B 为干净数据训练时的批大小,J 为目标类型数,y_{ij} 为预测类型分布。

针对文献[31]将细粒度实体分类转化为单标签分类问题,此方法未必完全正确,于是 Zhang 等[33]提出了一种统一处理所有训练样本的基于概率自动重标记的方法(NFETC-Automatic Relabeling,NFETC-AR)。在训练过程中为每个样本分配所有候选标签上的连续标签分布,并且将连续标签分布作为训练参数的一部分通过反向传播算法进行更新,达到预测分布与伪真标签分布之间的最小化 KL 散度(Kullback – Leibler Divergence)的目的,最后取伪真标签分布中值最大的标签作为唯一的伪真标签,具体 KL 散度如下:

$$L = -\frac{1}{N}\sum_{i}^{N}\sum_{j}^{|T|} p_{ij}\log\left(\frac{p_{ij}}{p_j(y_i)}\right)_J \tag{9}$$

其中,N 为实体指称的数量,T 为类型数,p_{ij} 为连续标签分布。

不仅要考虑标签的层次结构,Xin 等[34]从语言角度提出了以无监督的方式,运用标签含义衡量上下文句子与每个远程监督获得的标签之间的兼容性,将模型分为两部分:实体分类模型(Entity Typing Module,ET)和语言增强模型(Language Model Enhancement,LME)。ET 通过交叉熵函数,最小化真实类型概率与预测类型概率的差异,如下:

$$J(\theta) = -\frac{1}{N}\sum_{i}^{N} y_i^{*}\log y_i + (1-y_i^{*})\log(1-y_i) \tag{10}$$

LME 利用一个语言模型和一组标签嵌入来判断标签与上下文句子之间的兼容性,减少由远程监督产生的噪声。

Lin 等[35]从特征提取和类型预测两方面改进细粒度实体分类,采用 ELMo[36]代替原来固定的词嵌入,获取在不同句子上下文中单词的语义信息;并且提出感知实体指称的注意力机制,使得模型集中于指称和上下文中重要的单词。在设计交叉熵损失函数时,以 $\tilde{y}_i$ 为预测概率分布,y_i 为真实概率分布,通过最小化损失函数实现目标函数的优化,如下:

$$J(\theta) = -\frac{1}{N}\sum_{i}^{N} y_i\log\tilde{y}_i + (1-y_i)\log(1-\tilde{y}_i) \tag{11}$$

4　常用数据集及评价指标

在本节将详细描述细粒度实体分类任务中所用到的实验数据集、评价指标以及部分文献的实验结果对比。

4.1　常用数据集

在细粒度实体分类任务中,主要用到以下 3 个

数据集:FIGER[2]、OntoNotes[7]、BBN[37]。其中 FIGER 和 BBN 为 2 层数据集,OntoNotes 为 3 层数据集,其他数据如表 1 所示。

表 1 细粒度实体分类中常用的数据集

Table 1 A dataset commonly used in fine-grained entity classification

数据集	数据来源	数据内容	第一层	第二层	第三层
FIGER	Wikipedia	1.5 MB 语句	22	28	—
BBN	华尔街日报	2 311 篇文章	15	24	—
Onto Notes	新闻文档	13 109 文档	4	34	21

4.2 评价指标

评价细粒度实体分类任务沿用 Ling 等[2]提出的 3 个指标:准确率(Accuracy,Acc)、宏观平均 F1 值(Macro-averaging F1-Measure,Macro F1)以及微观平均 F1 值(Micro-averaging F1-Measure,Micro F1):

$$Acc = \frac{\sum_{m \in M} \sigma(Y = \hat{Y})}{M} \tag{12}$$

宏观平均 F1 值是宏观精确率(Macro Precision,P_{ma})和宏观召回率(Macro Recall,R_{ma})的平均值。

$$P_{\mathrm{ma}} = \frac{1}{|M|} \sum_{m \in M} \frac{|Y \cap \hat{Y}|}{\hat{Y}} \tag{13}$$

$$R_{\mathrm{ma}} = \frac{1}{|M|} \sum_{m \in M} \frac{|Y \cap \hat{Y}|}{Y} \tag{14}$$

微观平均 F1 值是微观精确率(Micro Precision,P_{mi})和微观召回率(Micro Recall,R_{mi})的平均值。

$$P_{\mathrm{m_i}} = \frac{\sum_{m \in M} |Y \cap \hat{Y}|}{\sum_{m \in M} \hat{Y}} \tag{15}$$

$$R_{\mathrm{m_i}} = \frac{\sum_{m \in M} |Y \cap \hat{Y}|}{\sum_{m \in M} Y} \tag{16}$$

其中,m 为实体指称,M 为其集合,Y 表示实体指称的真实类型标签,$\hat{Y}$ 表示实体指称的预测类型标签。

4.3 细粒度实体分类方法的对比研究

为比较以上细粒度实体分类方法的性能表现,本文在相同数据集上将各种方法的实验结果列出进行对比研究,如表 2 所示。选取的细粒度实体分类方法有以下几类:经典方法 FIGER[2]、HYENA[12];基于 RNN 的细粒度实体分类方法 Attentive[19];对于噪声处理方面,选取启发式剪枝噪声方法 CFGET[7],根据规则划分数据集方法 AFET[8] 和 AAA[24],优化铰链损失函数方法 FGET-RR[30],优化交叉熵损失函数方法 NFETC[31]、CLSC[32]、AR[33] 和 LME[34] 进行对比分析。

由表 2 可以看出,早期提出的经典细粒度实体方法(如 FIGER、HYENA)主要集中在将原始的粗粒度的命名实体类型扩展到细粒度的实体类型识别上,因此在 3 个数据集上的性能表现较差,特别是 HYENA 将所有实体类型划分为 9 层、共计 505 种的细粒度类别,难度大,因此最终的准确率、Macro F1 值和 Micro F1 值相对较低。引入神经网络模型后,Attentive 创新性地使用 LSTM 和注意力机制,使得模型的性能表现有大幅提高,在 FIGER 数据集上,准确率提高约 12%,Macro F1 值提高约 10%,Micro F1 值提高约 10%;在 OntoNotes 数据集上,准确率能够提升近 15%,Macro F1 值提高近 14%,Micro F1 值提高近 7%。在处理标签噪声方面,CFGET 采用剪枝训练集噪声的方法,但由于训练集规模的减小,在数据集上的表现较差,与 FIGER 实验结果相近。而 AFET 和 AAA 根据类型路径划分干净数据集和含噪声数据集,能够有效地提高实体分类的准确率、Macro F1 值和 Micro F1 值,尤其是 AAA 加入注意力机制,提取更为重要的特征信息,在 3 个数据集上表现良好,与 Attentive 相比,在 FIGER 数据集上,准确率提高约 6%,Macro F1 值提高约 2%,Micro F1 值提高约 2%;在 BBN 数据集上,准确率能够提升近 12%,Macro F1 值提高近 1%,Micro F1 值提高近 3%。FGET-RR 采用图卷积网络分析上下文信息,并对干净数据和含噪声数据分别设计损失函数,在 FIGER、BBN、OntoNotes 数据集上的性能能够得到显著的提升。CLSC、AR 都是在 NFETC 的基础上做出相应改进,实验结果表明 AR 对所有标签通过最小化预测标签与伪真标签之间的 KL 散度进行概率更新,最终在 FIGER 数据集上,较 NFETC 准确率提高

约 2%,Macro F1 值提高约 2%,Micro F1 值提高约 1%;在 BBN 数据集上,较 NFETC 准确率提高约 4%,Macro F1 值提高约 2%,Micro F1 值提高约 3%。LME 从语义角度,主要考虑了语言增强模型,未对预测分类模型做出改进,因此 LME 在 3 个数据集上的性能表现不如 NFETC。

表 2　细粒度实体分类性能比较

Table 2　Comparison of fine-grained entity classification performance

分数方法	FIGER			BBN			Onto Notes		
	Acc	Macro F1	Micro F1	Acc	Micro F1	Micro F1	Acc	Micro F1	Micro F1
FIGER	47.4	69.2	65.5	46.7	67.2	61.2	36.9	57.8	51.6
HYENA	28.8	52.8	50.6	52.3	57.6	58.7	24.9	49.7	44.6
Attentive	59.7	79.0	75.4	48.4	73.2	72.4	51.7	71.0	64.9
CFGET	45.3	69.1	63.1	44.4	67.1	61.3	37.3	57.0	50.9
AFET	53.3	69.3	66.4	67.0	72.7	73.5	55.1	71.1	64.7
AAA	65.8	81.2	77.4	60.4	74.1	75.7	52.2	68.5	63.3
FGET-RR	71.0	84.7	80.5	70.3	81.9	82.3	57.7	74.3	68.5
NFETC	68.9	81.9	79.0	73.9	78.8	79.4	60.2	76.4	70.2
CLSC	—	—	—	71.9	79.8	79.5	62.8	77.8	72.0
AR	70.1	83.2	80.1	74.9	80.4	80.3	64.0	78.8	73.0
LME	62.9	80.6	77.0	60.7	74.3	76.0	52.9	72.4	65.2

因此,由上述分析可以看出,在细粒度实体分类领域中采用 BiLSTM 处理实体指称上下文,并通过注意力机制提取更为重要的特征,同时利用 ELMo、BERT 等大规模的预训练模型代替原有的词嵌入,有助于提高分类的准确率。另外,为规避远程监督产生的噪声问题,以无监督的方式,选取伪真标签中最大值的标签,也能显著改善分类效果。

5　研究展望

对现有的细粒度实体分类方法以及基于噪声标签处理的方法进行了详细介绍,下面对未来细粒度实体分类的发展趋势和研究热点进行探讨,主要包括以下两个方面。

(1) 目前,基于神经网络的细粒度实体分类大多数都是监督学习,少部分以无监督的方式也取得良好的实验结果。未来以半监督方式,通过训练有标注数据,在验证集上验证无标注数据以获得伪标签数据,将标签数据与伪标签数据结合再次进行训练或以无监督方式,不断优化相似类型标签之间的距离都是可研究的方向。

(2) 对于细粒度实体分类的噪声处理,大多利用远程监督的方法,使得模型关注于实体指称及其上下文,并采用词嵌入、BiLSTM 处理指称和上下文向量。LSTM 的变体 GRU 利用更新门和重置门控制输入值、记忆值和输出值,结构较 LSTM 更为简单,能够简化神经网络,因此利用 GRU 处理实体指称或上下文的实验

有待尝试。另外,利用大规模的预训练模型 ELMo、BERT 等增强原有处理上下文的 BiLSTM 方法。现在可挖掘其他大型语料库的信息作为原来只基于实体指称上下文方法的一种补充,提取更优价值的信息,有利于提高实体分类模型的准确率。

本文对细粒度实体分类方法进行了详细叙述,介绍了现有的基于不同神经网络的细粒度实体分类方法以及基于噪声处理的细粒度实体分类方法,并对常用的数据集、评价指标和细粒度实体分类方法的性能表现进行了整理归纳,同时分析了未来发展趋势和研究热点。

参考文献(References):

[1] CHINCHOR N, ROBINSON P. MUC-7 named entity task definition[C]//Proceedings of the 7th Conference on Message Understanding, 1997.

[2] LING X, WELD D. Fine-grained entity recognition [C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2012.

[3] LECUN Y, BOTTOU L, BENGIO Y, et al. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11): 2278—2324.

[4] PINEDA F J. Generalization of back-propagation to recurrent neural networks[J]. Physical Review Letters, 1987, 59(19):602—611.

[5] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735—1780.

[6] 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,2009: 1003—1011.

[7] GILLICK D, LAZIC N, GANCHEV K, et al. Context-dependent fine-grained entity type tagging[J]. Computer Science, 2014(12):54—60.

[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, 2016: 1369—1378.

[9] LEE C, HWANG Y G, OH H J, et al. Fine-grained named entity recognition using conditional random fields for question answering[C]//Asia Information Retrieval Symposium. Berlin: Springer, 2006: 581—587.

[10] EKBAL A, SOURJIKOVA E, FRANK A, et al. Assessing the challenge of fine-grained named entity recognition and classification[C]//Proceedings of the 2010 Named Entities Workshop,2010: 93—101.

[11] SEKINE S. Extended named entity ontology with attribute information[C]//Proceedings of the Sixth International Conference on Language Resources and Evaluation,2008.

[12] YOSEF M A, BAUER S, HOFFART J, et al. HYENA: hierarchical type classification for entity names[C]// Proceedings of COLING 2012,2012: 1361—1370.

[13] DEL CORRO L, ABUJABAL A, GEMULLA R, et al. FINET: context-aware fine-grained named entity typing [C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015: 868—878.

[14] 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,2014: 601—610.

[15] JIA Y, XU W, QIN P, et al. Fine-grained entity typing for knowledge base completion[C]//IEEE International Conference on Network Infrastructure & Digital Content. Piscataway: IEEE, 2016:361—365.

[16] KIM Y. Convolutional neural networks for sentence classification[C]//Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014:1746—1751.

[17] CHEN Y, XU L, KANG L, et al. Event extraction via dynamic multi-pooling convolutional neural networks [C]//The 53rd Annual Meeting of the Association for Computational Linguistics,2015:167—176.

[18] 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),2018: 97—109.

[19] SHIMAOKA S, STENETORP P, INUI K, et al. An attentive neural architecture for fine-grained entity type classification[C]//Proceedings of the 5th Workshop on Automated Knowledge Base Construction,2016:69—74.

[20] SHIMAOKA S, STENETORP P, INUI K, et al. Neural architectures for fine-grained entity type classification [C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics,2017: 1271—1280.

[21] 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, 2018, 32(1):5997—6004.

[22] CHOI E, LEVY O, CHOI Y, et al. Ultra-fine entity typing[C]//Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018: 87—96.

[23] ZHANG D, LI M, CAI P, et al. Path-based attention neural model for fine-grained entity typing [C]// Proceedings of the AAAI Conference on Artificial Intelligence, 2018, 32(1):8179—8180.

[24] ABHISHEK A, ANAND A, AWEKAR A. Fine-grained entity type classification by jointly learning representations and label embeddings[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics,2017: 797—807.

[25] 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,2015: 291—296.

[26] WESTON J, BENGIO S, USUNIER N. WSABIE: scaling up to large vocabulary image annotation [C]// Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Three,2011: 2764—2770.

[27] DAI H, DU D, LI X, et al. Improving fine-grained entity typing with entity linking[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing,2019: 6211—6216.

[28] HUANG L, MAY J, PAN X, et al. Building a fine-grained entity typing system overnight for a new x (x = language, domain, genre) [J]. arXiv preprint arXiv: 1603.03112, 2016.

[29] ZHOU B, KHASHABI D, TSAI C T, et al. Zero-shot open entity typing as type-compatible grounding[C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,2018: 2065—2076.

[30] ALI M A, SUN Y, LI B, et al. Fine-grained named entity typing over distantly supervised data based on refined representations [C]//Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(5): 7391—7398.

[31] 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, 2018: 16—25.

[32] CHEN B, GU X, HU Y, et al. Improving distantly-supervised entity typing with compact latent space clustering[C]//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019: 2862—2872.

[33] ZHANG H, LONG D, XU G, et al. Learning with noise: improving distantly-supervised fine-grained entity typing via automatic relabeling [C]//29th International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence,2020: 3808-3815.

[34] XIN J, ZHU H, HAN X, et al. Put it back: entity typing with language model enhancement[C]// Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,2018: 993—998.

[35] 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),2019: 6198—6203.

[36] PETERS M, NEUMANN M, IYYER M, et al. Deep contextualized word representations[C]//Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,2018: 2227—2237.

[37] WEISCHEDEL R, BRUNSTEIN A. BBN pronoun coreference and entity type corpus[J]. Linguistic Data Consortium, Philadelphia, 2005, 112(4):71—76.

A Comparative Study Based on Fine-grained Entity Classification

ZHOU Qi, TAO Wan

(School of Computer and Information, Anhui Polytechnic University, Anhui Wuhu 241000, China)

Abstract: Fine-grained entity typing is a multi-class and multi-label task, which can help a wide range of downstream tasks (relationship extraction, co-reference resolution, question answering system, etc.) to enhance productivity and optimize accuracy. It has become a research hotspot in natural language processing field. In view of the difficulty and low accuracy of the traditional fine-grained entity typing method to annotate large corpus, researchers proposed the fine-grained entity typing method based on neural network, which can not only solve the time-consuming and laborious problem of manual annotation, but also improve the accuracy of classification. However, most of the existing neural network models require the participation of distant supervision, which will introduce noise labels and other problems in the process. The noise labels processing method can effectively suppress the impact of noise labels on the classification results and further improve the classification performance. Under the same evaluation datasets, we compared the performance of various fine-grained entity typing methods according to the same evaluation metrics. It can be found that in the field of fine-grained entity typing, using BiLSTM to process the context of entity mention and extracting more important features through the attention mechanism are helpful to improve the accuracy, Macro F1 value and Micro F1 value of fine-grained entity typing method.

Key words: fine-grained entity typing; neural network; distant supervision; noise processing

责任编辑:罗姗姗

引用本文/Cite this paper:

周祺,陶皖. 基于细粒度实体分类的对比研究[J]. 重庆工商大学学报(自然科学版),2022,39(4):9—18.

ZHOU Qi, TAO Wan. A comparative study based on fine-grained entity classification[J]. Journal of Chongqing Technology and Business University (Natural Science Edition),2022,39(4):9—18.