Computer Science and Application 计算机科学与应用, 2022, 12(12), 2712-2724
Published Online December 2022 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2022.1212275

Hans汉斯

基于多模融合的Java领域命名实体识别

李凯微[1]，王佳英[1,2]，单　菁[1,2]

[1]沈阳建筑大学，辽宁　沈阳
[2]沈阳工业大学，辽宁　沈阳

收稿日期：2022年11月1日；录用日期：2022年11月30日；发布日期：2022年12月7日

摘　要

命名实体识别是构建学科知识图谱的重要步骤。近年来，随着深度学习的发展，通用领域、医学等领域命名实体识别的性能得到了很大的提升。Java学科领域知识点繁杂，实体中英文掺杂，并且存在其特有的实体内部特征，因此通用模型在此领域实体识别准确率并不高、不能有效识别实体边界。提出改进的单模结构，在嵌入层融入词边界信息，引入了词性信息和Java领域实体识别的规则信息，以提高模型识别实体边界的准确率。编码层使用BiLSTM和IDCNN进行上下文信息提取，解码层使用CRF得到序列全局最优提取。其次，提出对多个异构单模结果进行融合互补的想法，以提高模型实体识别性能和模型的泛化能力。实验结果显示，基于自主构建的Java领域数据集，新的单模模型相比于主流模型实体识别F1值提高了约2个百分点。多模融合后的实体识别的性能也有明显的提升，表明模型在Java领域命名实体识别任务上有着更好的效果。

关键词

命名实体识别，多模融合，实体边界，BiLSTM，CRF

Named Entity Recognition in Java Domain Based on Multi-Mode Fusion

Kaiwei Li[1], Jiaying Wang[1,2], Jing Shan[1,2]

[1]Shenyang Jianzhu University, Shenyang Liaoning
[2]Shenyang University of Technology, Shenyang Liaoning

Received: Nov. 1st, 2022; accepted: Nov. 30th, 2022; published: Dec. 7th, 2022

Abstract

Named entity recognition is an important step in constructing disciplinary knowledge map. In

文章引用: 李凯微, 王佳英, 单菁. 基于多模融合的 Java 领域命名实体识别[J]. 计算机科学与应用, 2022, 12(12): 2712-2724. DOI: 10.12677/csa.2022.1212275

recent years, with the development of deep learning, the performance of named entity recognition in general field, medicine and other fields has been greatly improved. The knowledge of Java subject is complicated, the entities are mixed in Chinese and English, and there are unique internal characteristics of the entities. Therefore, the accuracy of entity recognition of the general model in this field is not high, and the entity boundary cannot be effectively identified. In order to improve the accuracy of entity boundary recognition, an improved single-mode structure is proposed, and word boundary information is incorporated into the embedding layer, part of speech information and Java domain entity recognition rule information are introduced. BiLSTM and IDCNN are used in encoding layer to extract context information, and CRF is used in decoding layer to obtain global optimal sequence extraction. Secondly, the idea of fusing and complementing multiple heterogeneous single-mode results is proposed to improve the entity recognition performance and generalization capability of the model. Experimental results show that, based on the self-constructed Java domain data set, the entity recognition F1 value of the new single-mode model is improved by about 2 percentage points compared with the mainstream model. The performance of entity recognition after multi-mode fusion is also significantly improved, indicating that the model has better performance in Java domain named entity recognition task.

Keywords

Named Entity Recognition, Multimode Fusion, Solid Boundary, BiLSTM, CRF

1. 引言

由于互联网，人工智能，在线视频等的发展，教育行业衍生出了的各种新生产物，现在的教育形式产生了翻天覆地的变化[1]。构建 Java 领域的知识图谱，能把零散的知识汇集成一个完整的知识网络体系，把 Java 相关的知识点转换成模块化、结构化、可视化的知识图谱[2]。命名实体识别又是构建 Java 领域知识图谱中，不可或缺的一部分。在面向通用领域中，识别人名、地名和机构名等方面的命名实体识别方法的性能有着较大幅度的提升[3]。并且近几年国内在军事文本、医疗、司法等领域实体识别方面也取得了较大的进展[4]。但对于 Java 学习领域的相关研究就很少，Java 相关知识点繁冗复杂[5]，相关扩展知识、不断更新发展的技术栈也很繁杂，实体往往中英文掺杂，并普遍存在简写缩写现象。因此针对此领域的命名实体识别的研究还处于起步阶段。单纯地基于规则的命名实体识别[6]，准确率很高，但是人工成本很高。随着统计机器学习算法的发展，隐马尔可夫模型[7] (Hidden Markov Model, HMM)和条件随机场[8] (Conditional Random Field, CRF)、支持向量机[9] (Support Vector Machines, SVM)渐渐被人们所关注。但其缺点在于需要构建大规模的标注语料库，费时费力。近几年。深度学习在各领域都取得了不错的成果，其优点在于可以深度学习语义知识并且缓解数据稀疏的问题[10]。

尽管近几年提出了许多新的模型来解决实体识别相关问题，但在 Java 学科领域的实体识别仍然面临着很多的挑战，具体体现在：

1) 此领域没有公开数据集，由于缺乏有标注的文本资源，且自主构建数据集费事费力，因此导致此领域研究发展滞后。

2) Java 学科领域的知识点繁杂，中英文混杂，并且实体含有特有的内部特征。现有的实体识别方法

难以对其进行准确地识别。

3) 现有大多数命名实体识别模型是基于字的划分粒度的，因此不能充分利用词相关信息，丢失了词边界，不能准确区分出词边界。导致识别较长实体名时，准确率并不高。

针对以上问题，本文首先构建了 Java 学科领域数据集，数据来源于计算机权威学习网站菜鸟教程和 C 语言中文网，知识点齐全，构建的数据集共包含 46,316 句文本信息。此外，本文针对 Java 实体专有特点，融入了词性信息和相关规则，以提高识别词边界信息、长实体名的准确率。并且提出了多模融合的方法提高模型的泛化性和实体识别的准确性。

2. 相关工作

2013 年，Marrero 等人[11]从应用方面，提出了命名实体识别的标准。1997 年，Hochreiter 等人[12]提出来 LSTM 模型，解决了长距离依赖问题。2003 年，Hammerton 等人[13]首次把长短期记忆网络(Long Short Term, LSTM)使用在命名实体领域。2014 年，Bahdanau 等人[14]首次将注意力机制引入自然语言处理领域，有效提高了命名实体识别模型性能。2016 年，Peng [15]在分词的基础上，使用了双向长短词记忆模型 (Bi-directional Long Short Term Memory, BiLSTM)-CRF 模型。2017 年，Strubell 等人[16]提出了空洞卷积 (Iterated Dilated Convolutional Neural Network, IDCNN)-CRF 模型，加快了模型的训练速度。2018 年，Zhang 等人[17]提出了不需要分词的 Lattice LSTM 模型，引入外部词典以充分利用词序信息，解决了分词错误带来的误差传递问题。2018 年，谷歌公司提出了 BERT 模型[18]，在多个领域中取得了不多的结果。2020 年，Yiming Cui 等人[19]考虑到中文分词，提出来 BERT-www 模型。在不同领域的命名识别任务中，2020 年，Li 等人[20]在医学领域中，使用未标记的数据，使用 BERT 对电子简历进行预训练，并引入了字典和部首特征以提高模型识别准确率。2020 年，Wang 等人[21]在司法领域中，利用注意力机制提取句子信息，提出来了 Attention-BiLSTM-CRF 模型。2021 年，Liu 等人[22]在历史领域中引入了 BERT-BiLSTM-CRF 模型，从非结构数据中提取实体信息。

综合上述情况，本文得到启发提出了基于 Java 领域的命名实体识别方案。

本文主要的贡献如下：

1) 本文自主构建并标注 Java 专有领域中文数据集。

2) 提出了针对 Java 专有领域的实体识别单模模型，融入了词性信息和 Java 领域实体识别规则，以提高识别实体边界的准确率。

3) 提出对多个异构单模的结果进行融合互补的方法，以提高最终预测结果的准确性和模型的泛化能力。

3. 模型

本文提出的融入词性信息和 Java 专有领域实体识别规则的单模模型，以更好地识别实体边界，单模结构分为三层：嵌入层，编码层，解码层。单模验证成功后，对多个异构单模预测结果进行加权融合，输出最终的预测结果。单模结构如图 1 所示。

3.1. 嵌入层

嵌入层，主要作用是对文本信息的特征进行提取。本层模型在 BERT 的基础上进行了改进，将嵌入层增添至四部分，boundary embeeding，token embeeding，segment embeeding，position embeeding。将输入信息映射为字向量，词边界信息映射为边界向量，随后字向量和边界向量进行了融合，再加入句向量和位置向量，四者融合作为了最终的输出向量。

本文研究的 Java 领域命名实体识别任务，字与字之间有相互作用的关系，并且实体名称长度较长。且中文与英文不同，中文没有天然的词边界信息[23]，若仅对字编码就缺少了词的边界信息，因此本文提出了边界嵌入。

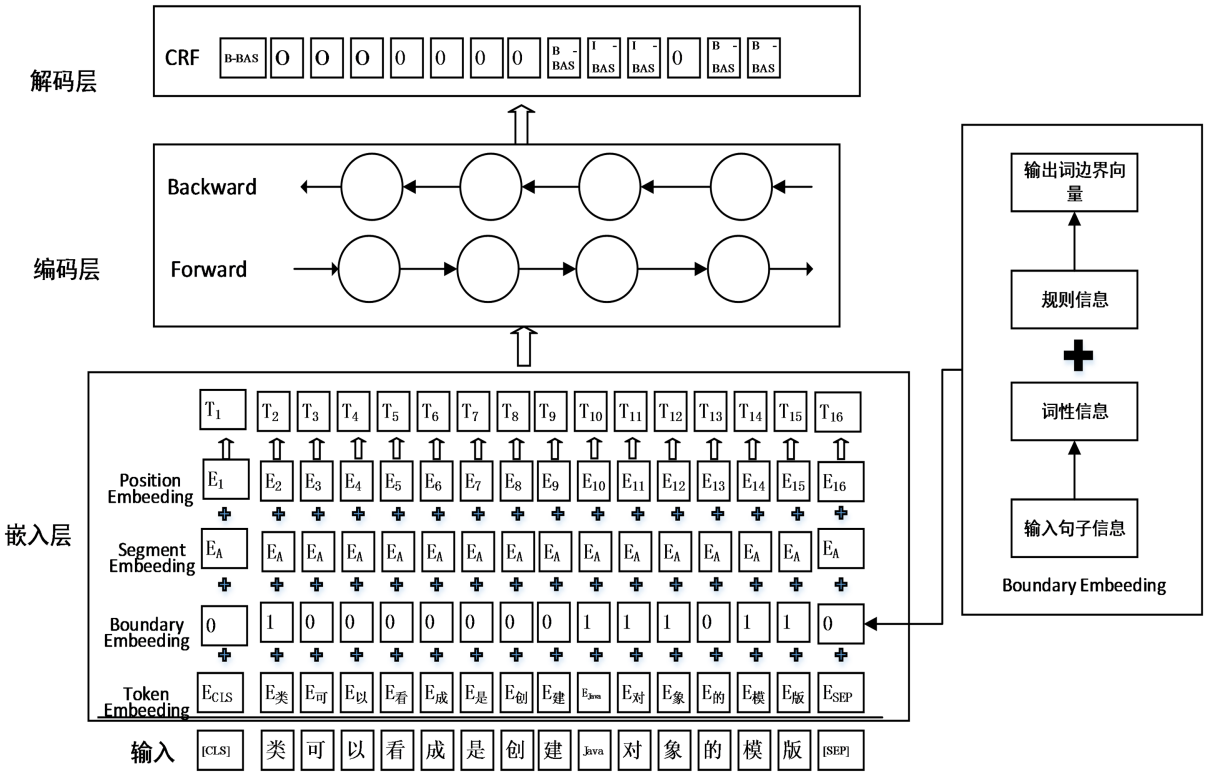

Figure 1. Single mode structure
图 1. 单模结构图

(一) boundary embeeding，边界嵌入，若缺少词的边界信息，实体识别的准确率就会降低，因此融合边界信息，以更准确地识别词边界。边界嵌入主要由两个模块组成，词性信息和规则设置。

词性信息，用来辨别字所属的词性，为名词，动词等。实体一般由名词组成，且后续针对 Java 专有领域实体规则也需要利用词性信息，因此选择融入词性信息，为下一步制定规则做准备，用来更准确地判断的词的边界信息。

制定规则。针对 Java 专有领域实体特点制定特有的识别规则。Java 领域主要有以下两种特有表示结构：

1) Java 专有名词特点为：英文字符 + 名词的结构。例如：“Java 虚拟机”、“Date 类”、“final 变量”等实体，常被识别为[Java、虚拟机]，[Date、类]，[final、变量]。

2) Java 专有名词特点为：名词+名词的结构。例如：“条件运算符”、“时间模式字符串”、“时间复杂度”等实体，常被现有模型识别为[条件，运算符]，[时间模式、字符串]，[时间、复杂度]。此规则的作用就是选择长度更长的实体作为标注的候选答案。

针对上述特点，融入词性信息，并且制定出适合 Java 领域的特有实体识别规则，可以更好的解决实体边界问题，从而增加实体识别的准确率。

(二) token embeeding，字的嵌入信息，是通过训练得到的。一个句子的开头用 CLS，结束用 SEP 表示。此嵌入层只负责融入字的信息。

(三) segment embeeding，句子嵌入，是通过训练得到的，是为了区分不同句子的向量。若输入的是两个句子，则用 EA 和 EB 区分这个字属于句子 A 还是句子 B。若只输入一个句子，就使用 EA。

(四) position embeeding，位置嵌入，为了区分字在句子中的位置信息。相同的字出现在文本的不同位置，所拥有的语义信息存在差异。因此对不同位置的字标记一个不同的向量，进行区分。

如公式(1)所示，将四个向量相加作为编码层的输出

$$E_{word} = E_{token} + E_{seg} + E_{bound} + E_{pos} \tag{1}$$

E_{token} 表示字嵌入向量，E_{seg} 为句子嵌入向量，E_{bound} 为位置嵌入的边界信息向量，E_{pos} 为位置嵌入向量。E_{word} 为最终融合输出的向量。

上述 4 个分量都可以用其独热码与嵌入系数矩阵 W 相乘的形式，如公式(2)所示：

$$E_{word} = O_{token}W_{token}^{|V|\times H} + O_{seg}W_{seg}^{|S|\times H} + O_{bound}W_{bound}^{|B|\times H} + O_{pos}W_{pos}^{|P|\times H} \tag{2}$$

O_{token} 是根据输入的当前字符在字典中位置下标构造的独热码表示。O_{seg} 是根据输入字符所属句子下标构造的独热码表示，O_{bound} 是根据词性信息和规则信息，判断的词边界信息编码。O_{pos} 是根据输入字符在整个句子中位置信息，构造的独热码表示。H 是嵌入维度，$|V|$ 是序列个数，$|B|$ 是词性边界数量，$|P|$ 是最大位置数。

本文预训练模型基于 BERT 模型的基础上，根据 Java 领域实体独有的特点，加入了边界嵌入信息，引入了词性信息和相关提取规则信息。

3.2. 编码层

本层的目的是更好的让模型理解文本的上下文关系，所用模型为 BiLSTM 和 IDCNN。

BiLSTM [24]：由于序列过长，RNN 等模型存在梯度消失的问题。LSTM 模型由于其特殊的门结构能很好的保存上下文信息，从而解决这一问题。BiLSTM 由前向 LSTM 和后向 LSTM 组成，捕捉两个方向的上下文信息。由嵌入层输出的向量序列信息作为 BiLSTM 的输入，让每个序列通过一个前向 LSTM 和一个后向 LSTM，通过计算，得到两个不同的向量表示，将这两个向量进行拼接作为输出，如公式(3~5)所示：

$$\vec{h}_i = \mathrm{LSTM}\left(\vec{h}_{i-1}, x_i\right) \tag{3}$$

$$\overleftarrow{h}_i = \mathrm{LSTM}\left(\overleftarrow{h}_{i-1}, x_i\right) \tag{4}$$

$$h_i = \vec{h}_i \oplus \overleftarrow{h}_i \tag{5}$$

$\vec{h}_i$ 和 $\overleftarrow{h}_i$ 表示 i 位置的前向 LSTM 和后向 LSTM 输出表示，x_i 表示前一时刻隐藏层输入状态。$\oplus$ 表示整合信息。进过双向 LSTM 编码处理得到的向量序列，再使用 softmax 预测出每个字对应的标注概率。

双向 LSTM 结构如图 2 所示。

为了缓解过拟合问题，本文在模型训练时使用了 dropout，其思想是按一定概率，随机选取神经层的一些神经元进行隐藏，下次训练时，又隐藏另外选取的神经元。Dropout 的值对训练结果影响较大。

IDCNN [16]：作用于 BiLSTM 类似，用来捕捉长序列文本的上下文信息。IDCNN 是对卷积神经网络(CNN)的改进，在局部信息丢失的情况下，利用空洞来捕捉文本上下文特征。IDCNN 优势就是时间复杂度低，可以提高模型的运行效率，即使在并行情况下，处理长度为 N 的句子，处理的时间复杂度只需 O(N)，可以缩短模型预测时间。

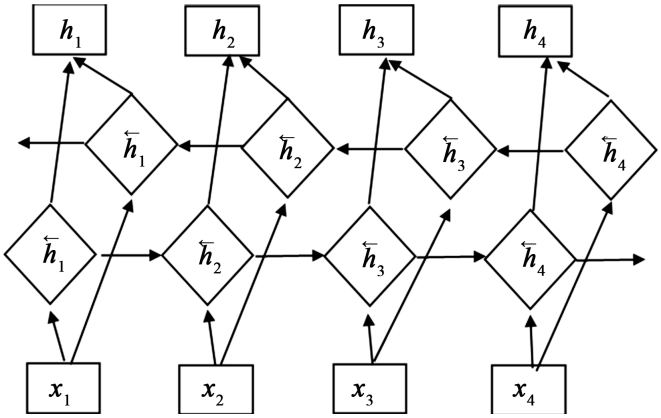

Figure 2. Bidirectional LSTM structure
图 2. 双向 LSTM 结构

3.3. 解码层

本层主要学习标签之间的约束关系，以提高标签预测的准确率，进行全局最优提取。本文主要识别三类实体：Java 基础类(BAS)、Java 扩展类(EXT)、实例类(SPE)。文本采用的是 BIO 标注方法，例如：“修饰符”的标注应为[B-BAS, I-BAS, I-BAS]，但模型预测结果若为[B-BAS, I-EXT, I-BAS]，明显不符合常理。因此解码层的作用，主要是为了修正上述标签之间约束关系问题。此模型解码层使用的是 CRF。

CRF 模型本身就是一个非常优秀的序列标注模型，在多项任务中表现出色[25]。CRF 模型适用于整个句子，而不是单个字的位置。假设输入为 $\boldsymbol{X}$，BiLSTM 预测的标签序列为 $\boldsymbol{y}=\{y_1,y_2,\cdots,y_n\}$ (n 为句子长度)，y_i 就表示每个字的标签。计算序列概率如公式(6)：

$$P_{X(y_1,y_2,\cdots,y_n)}=P_{\boldsymbol{X}(y_1)}\cdot P_{\boldsymbol{X}(y_2|y_1)}\cdots P_{\boldsymbol{X}(y_n|y_{n-1})} \tag{6}$$

若 f 为打分函数，$g(y_{n-1},y_n)$ 代表相邻位置的转移矩阵，代表标签从 $n-1$ 位置转移到 n 位置的可能性。h 是 CRF 层从 BiLSTM 处得到的预测每个字的标签概率的发射矩阵。计算打分方法如公式(7)：

$$f_{(y_1,y_2,\cdots,y_n|X)}=h_{(y_1|X)}+g_{(y_1,y_2)}+\cdots+h_{(y_2|X)}+\cdots+g_{(y_{n-1},y_n)}+h_{(y_n|X)} \tag{7}$$

$z(x)$ 为归一化因子，若我们考虑 g 和 X 无关，序列概率如公式(8)：

$$P_{(y_1,y_2,\cdots,y_n|X)}=\frac{1}{z(x)}\exp\left(h_{(y_1|X)}+\sum_{i=1}^{n-1}\left[g_{(y_i,y_{i+1})}+h_{(y_{i+1}|X)}\right]\right) \tag{8}$$

BiLSTM-CRF 模型，通过 BiLSTM 层学习上下文信息，使用 CRF 层学习标签于标签间的相关性，得到标注信息，计算得分进行反向传播，最后得到最优的标注信息。

3.4. 多模融合

由于单个模型的局限性，其命名实体识别的准确性受限。本文提出基于集成学习的对多个异构单模的结果进行融合互补的思路，以提高最终识别结果的准确性和泛化能力。

主要思路：

1) 针对解决同一个问题，构建出多个异构单模。

2) 设置融合模型个数，设置阈值过滤概率得分极低的模型，防止成为噪音。因为模型相对于其他模型效果太差时，该模型会成为噪音。根据模型的评估得分来设置模型加权系数。

3) 对多个单模进行软投票，输出预测准确率与预测标签。

整个模型的处理流程如图 3 所示：

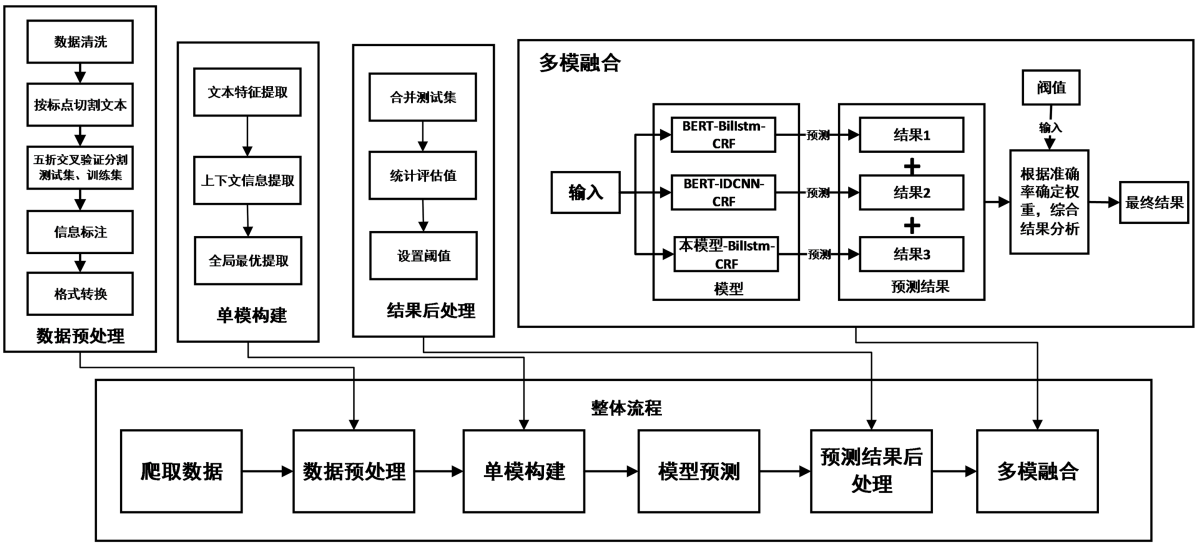

Figure 3. Overall flow chart
图 3. 整体流程图

4. 实验

4.1. 数据集

本文所使用的数据集是自主构建的，文本数据来源为两部分：Java 基础知识部分来自菜鸟教程网站[1]。Java 扩展知识部分来自 C 语言中文网[2]，包括 Java 设计模型、Spring、SpringBoot、Maven 等方面。这些数据资料知识点全面、内容丰富，并且网页数据编排合理，在计算机领域影响力较强。数据集全文共包含 46316 句文本信息，划分为 80%的训练集和 20%测试集，并对训练数据集进行五折交叉验证分割。按照 BIO 规则进行信息标注，B 代表实体的开始，I 代表实体的中间及边界，O 代表其他，即非实体的部分。根据 Java 知识体系结构，很容易对实体进行分类，本文识别任务分为三类：Java 基础类(BAS)、Java 扩展类(EXT)、实例类(SPE)，三类实体和一个非实体标签共计 7 个标签，即 B-BAS，I-BAS，B-EXT，I-EXT，B-SPE，I-SPE，O。标签具体设置如表 1 所示：

Table 1. Label settings
表 1. 标签设置

类别	标签说明	实体举例
Java 基础类(BAS)	包括 Java 基础知识体系所涉及的知识点，包括 Java 基础语法、Java 条件语句、异常处理、Java 面向对象、Java 集合框架等方面。	非访问控制符、关键字、逻辑运算、变量、数组、默认值、栈、静态存储区
Java 扩展类(EXT)	Java 扩展知识体系所涉及的知识点，随着时间发展，不断推出的 Java 框架等，包括 MyBatis、Spring、SpringMVC、SpringBoot、Nodejs、Redis 等方面。	控制反转、xml 解析、对象工厂、依赖注入、p 名称、级联赋值、增强类、面向切面编程、环绕通知、注解开发、中间件

[1] 菜鸟教程网址：https://www.runoob.com/。
[2] C 语言中文网址：http://c.biancheng.net/sitemap/。

Continued

实例类(SPE)	知识点下具体的实例名，例如标识符的名称，具体的方法名称等。	==、!=、>、<<、&= (属于具体的操作符) Boolean、Byte、Short、Float、Character (属于具体的包装类)

4.2. 实验环境

硬件环境配置如表 2：

Table 2. Hardware environment configuration
表 2. 硬件环境配置

名称	配置信息
操作系统	Win11
CPU	R7-5800H
内存	16G
GPU	RTX3050

软件环境配置如表 3：

Table 3. Software environment configuration
表 3. 软件环境配置

名称	配置信息
python	3.7
Tensorflow-gpu	1.15
cuda	11.2
cudnn	8.1

4.3. 参数设置

模型参数设置如表 4：

Table 4. Model parameters
表 4. 模型参数

模型	参数名称	参数
预训练模型	初始学习率	5e−5
	句子序列长度	256
	batch-size	128
Billstm	隐藏层维度	150
	优化函数	Adam
	Dropout 率	0.5
	batch-size	128
	Epoch	120

Continued

IDCNN	窗口大小	3
	滤波器个数	150
	Dropout 率	0.5
	batch-size	128
	Epoch	120

4.4. 评估标准

本文采用三种评估方式[26]，分别为准确率 P (Precison)、召回率 R (Recall)和 F1，具体如公式(9~11)：

$$P = \frac{T_P}{T_P + F_P} \times 100\% \tag{9}$$

$$R = \frac{T_P}{T_P + F_n} \times 100\% \tag{10}$$

$$\mathrm{F1} = \frac{2PR}{P + R} \times 100\% \tag{11}$$

其中，T_P 表示识别正确的实体数量，F_P 表示识别错误的实体数量，F_n 表示未正确识别出的实体数量。P 代表在所有实体中准确识别出来的百分比，其也被称为查准率。R 代表在所有的样本中，被正确识别出的实体的占比，也被称为查全率。但 P、R 两个指标有时会发生矛盾的情况，因此引入了 F1 值，其是综合 P 和 R 两者的评估指标，用来反映整体的综合情况。

4.5. 实验结果和分析

改进的模型和其他主流模型进行对比，实验结果如表 5：

Table 5. Single mode comparison experiment
表 5. 单模对比实验

Model	Type	Precison	Recall	F1
BiLSTM + CRF (模型 1)	BAS	73.63	71.74	72.67
	EXT	70.44	69.85	70.14
	SPE	74.31	72.92	73.60
IDCNN + CRF (模型 2)	BAS	74.28	72.67	73.47
	EXT	71.45	72.24	71.84
	SPE	73.16	73.52	73.40
BERT + BiLSTM + CRF (模型 3)	BAS	78.95	78.31	78.63
	EXT	76.61	77.29	76.95
	SPE	82.30	83.54	82.92
BERT + IDCNN + CRF (模型 4)	BAS	79.28	80.62	79.94
	EXT	77.36	76.57	76.96
	SPE	81.75	84.11	82.91

Continued

本模型 + BiLSTM + CRF (模型 5)	BAS	81.61	80.32	81.96
	EXT	79.46	78.27	78.86
	SPE	83.48	84.52	83.99

单模不同类别的 F1 值如图 4 所示：

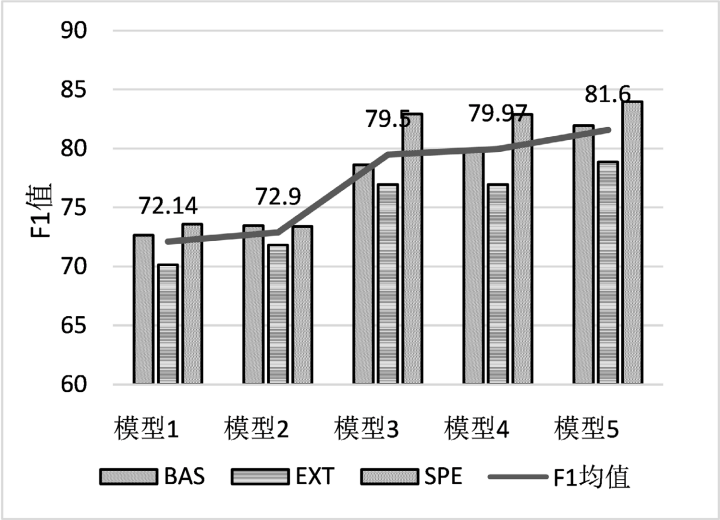

Figure 4. Comparison results of single-mode F1
图 4. 单模 F1 对比结果

通过实验可以看出来，模型 1 和模型 2 的结果不太理想，加入预训练模型 Bert 后，效果得到显著的提升。从编码层模型的选择来看，BiLSTM 和 IDCNN 区别也不是很明显。加上融入边界信息的本模型，单模效果比现有主流模型 3 和 4 的 F1 值提升了约两个百分点，从实验结果看效果还是可以的。

从实体识别的类别来看，实例名预测的准确率最高，其原因可能在于相比另外两种实体，实例名特征最为明显，容易被识别出来。Java 扩展实体类预测的准确率偏低，其原因可能在于 Java 的扩展领域知识点冗杂，并且普遍存在简写缩写名称，模型难以进行准确地预测。融入边界信息后的模型对扩展类识别准确率提升很大。

多个异构单模的多模融合结果如表 6 所示：

Table 6. Multi-mode comparison experiment
表 6. 多模对比实验

Model	Type	Precison	Recall	F1
多模融合 (模型 1 + 模型 2)	BAS	76.22	73.47	74.82
	EXT	72.64	72.52	72.58
	SPE	75.72	75.43	75.57
多模融合 (模型 3 + 模型 4)	BAS	79.85	77.15	78.48
	EXT	78.24	78.66	78.45
	SPE	83.97	85.13	84.55

Continued

多模融合 (模型 1 + 模型 2 + 模型 5)	BAS	81.43	82.97	82.19
	EXT	79.81	80.21	80.01
	SPE	84.62	83.93	84.27
多模融合 (模型 3 + 模型 4 + 模型 5)	BAS	83.13	85.26	84.18
	EXT	81.02	82.64	81.82
	SPE	87.24	85.49	86.36
多模融合 (模型 1 + 模型 3 + 模型 4 + 模型 5)	BAS	82.72	84.33	83.56
	EXT	80.53	81.21	80.86
	SPE	85.26	84.34	84.80

多模不同类别的 F1 值如图 5 所示：

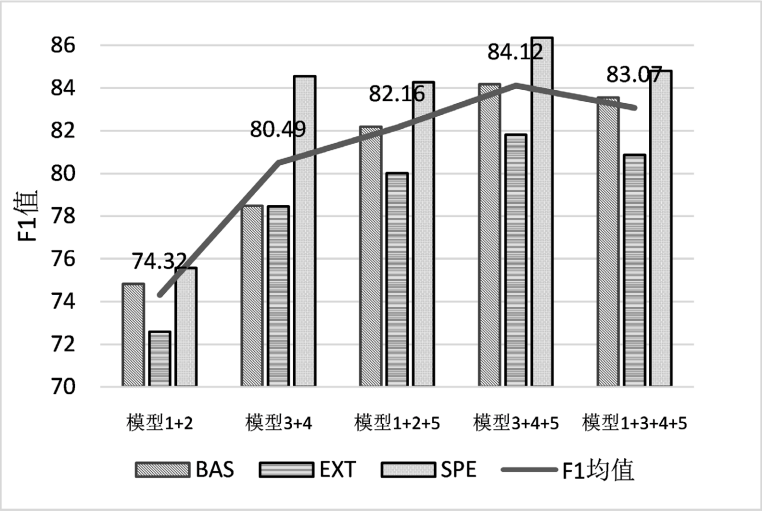

Figure 5. Comparison results of multi-mode F1
图 5. 多模 F1 对比结果

通过图 5 的实验结果可以看出来，并不是融合的模型个数越多效果就越好，模型 1 + 3 + 4 + 5 这 4 个模型相融合的结果还不如模型 3 + 4 + 5 这 3 个模型的融合结果好，原因可能在于模型 1 的预测结果比较差，拉低了整个模型的准确率。

从实体识别的类别来看，多模融合在扩展类提升并不是很大，原因可能在于本来单模效识别效果不是很好，两者识别是错误的，那么融合结果可能也是错误的。但融合之后在实例类、基础类实体识别的准确率有所提升，单模准确率高，融合之后识别结果会更接近正确预测。

5. 结论

由于 Java 领域缺少开放的数据集，本文进行了相关数据爬取，人工标注了 4 万多句标签信息。本文针对 Java 领域实体识别独有特点，提出了融入词性信息和相关规则的模型，以提高 Java 领域实体边界识别的准确率，通过实验结果证明了改进后模型的有效性。

考虑到单个模型具有局限性，本文将多个异构单模结果进行加权融合，以提高模型的泛化能力和识

别准确性。从整体的实验结果中可以看出 Java 领域的实体识别准确率还有提升的空间，未来我们可以考虑加上图神经网络融入更多的文本信息，并且对模型进行剪枝、蒸馏等操作，进一步降低模型的时间复杂度和空间复杂度。

基金项目

辽宁省教育厅基金(lnqn202015)，面向生物大数据的局部近似匹配技术研究。

参考文献

[1] 余蕾. 互联网背景下教学模式探究[J]. 当代教育实践与教学研究, 2019(23): 8-9.

[2] 李艳茹, 周子力, 倪睿康, 等. 基于知识图谱的学科知识构建[J]. 计算机时代, 2021(4): 65-68.

[3] 赵山, 罗睿, 蔡志平. 中文命名实体识别综述[J]. 计算机科学与探索, 2022, 16(2): 296-304.

[4] 邓依依, 邬昌兴, 魏永丰, 等. 基于深度学习的命名实体识别综述[J]. 中文信息学报, 2021, 35(9): 30-45.

[5] 姚雅峰. Java 技术的发展趋势与应用研究[J]. 无线互联科技, 2021, 18(6): 81-82.

[6] Li, Y., Chiticariu, L., Reiss, F., *et al*. (2010) Domain Adaptation of Rule-Based Annotators for Named-Entity Recognition Tasks.

[7] Morwal, S. (2012) Named Entity Recognition using Hidden Markov Model (HMM). *International Journal on Natural Language Computing*, **1**, 15-23. https://doi.org/10.5121/ijnlc.2012.1402

[8] Song, S., Nan, Z. and Huang, H. (2017) Named Entity Recognition Based on Conditional Random Fields. *Cluster Computing*, **22**, 5195-5206. https://doi.org/10.1007/s10586-017-1146-3

[9] Ju, Z., Wang, J. and Zhu, F. (2011) Named Entity Recognition from Biomedical Text Using SVM. 2011 5*th International Conference on Bioinformatics and Biomedical Engineering*, Wuhan, 10-12 May 2011, 1-4. https://doi.org/10.1109/icbbe.2011.5779984

[10] 何玉洁, 杜方, 史英杰, 宋丽娟. 基于深度学习的命名实体识别研究综述[J]. 计算机工程与应用, 2021, 57(11): 21-36.

[11] Marrero, M., Urbano, J., Sánchez-Cuadrado, S., *et al*. (2013) Named Entity Recognition: Fallacies, Challenges and Opportunities. *Computer Standards & Interfaces*, **35**, 482-489. https://doi.org/10.1016/j.csi.2012.09.004

[12] Hochreiter, S., *et al*. (1997) Long Short-Term Memory. *Neural Computation*, **9**, 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

[13] Hammerton, J. (2003) Named Entity Recognition with Long Short-Term Memory. *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL* 2003, Edmonton, 31 May-1 June 2003, 172-175. https://doi.org/10.3115/1119176.1119202

[14] Bahdanau, D., Cho, K. and Bengio, Y. (2014) Neural Machine Translation by Jointly Learning to Align and Translate. 3*rd International Conference on Learning Representations*, *ICLR* 2015, San Diego, 7-9 May 2015.

[15] Peng, N. and Dredze, M. (2016) Improving Named Entity Recognition for Chinese Social Media with Word Segmentation Representation Learning. *Proceedings of the* 54*th Annual Meeting of the Association for Computational Linguistics*, Volume 2, 149-155. https://doi.org/10.18653/v1/P16-2025

[16] Strubell, E., Verga, P., Belanger, D., *et al*. (2017) Fast and Accurate Entity Recognition with Iterated Dilated Convolutions. *Proceedings of the* 2017 *Conference on Empirical Methods in Natural Language Processing*, Copenhagen, September 2017, 2670-2680. https://doi.org/10.18653/v1/D17-1283

[17] Zhang, Y. and Yang, J. (2018) Chinese NER Using Lattice LSTM. *Proceedings of the* 56*th Annual Meeting of the Association for Computational Linguistics*, Volume 1, 1554-1564. https://doi.org/10.18653/v1/P18-1144

[18] Devlin, J., Chang, M.W., Lee, K., *et al*. (2018) BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.

[19] Cui, Y., Che, W., Liu, T., *et al*. (2020) Revisiting Pre-Trained Models for Chinese Natural Language Processing. *Findings of the Association for Computational Linguistics*: *EMNLP*, November 2020, 657-668. https://doi.org/10.18653/v1/2020.findings-emnlp.58

[20] Li, X., Zhang, H. and Zhou, X.H. (2020) Chinese Clinical Named Entity Recognition with Variant Neural Structures Based on BERT Methods. *Journal of Biomedical Informatics*, **107**, Article ID: 103422. https://doi.org/10.1016/j.jbi.2020.103422

[21] Wang, C., Li, B. and Zhang, W. (2020) Attention-BLSTM-CRF Based Method for Named Entity Recognition in Judi-

cial Domain. *Journal of Physics Conference Series*, **1616**, Article ID: 012108. https://doi.org/10.1088/1742-6596/1616/1/012108

[22] Liu, S., Yang, H., Li, J., *et al*. (2021) Chinese Named Entity Recognition Method in History and Culture Field Based on BERT.

[23] 王佳楠, 梁永全. 中文分词研究综述[J]. 软件导刊, 2021, 20(4): 247-252.

[24] Graves, A. and Schmidhuber, J. (2005) Framewise Phoneme Classification with Bidirectional LSTM Networks. *IEEE International Joint Conference on Neural Networks*, Vol. 4, 2047-2052. https://doi.org/10.1016/j.neunet.2005.06.042

[25] Konkol, M. and Konopík, M. (2013) CRF-Based Czech Named Entity Recognizer and Consolidation of Czech NER Research. Springer, Berlin. https://doi.org/10.1007/978-3-642-40585-3_20

[26] 周玉新. 命名实体识别研究发展综述[J]. 科技风, 2016(16): 99.