



Rock_y

码龄2年

暂无认证

46

原创

500

积分



私信

Q

热门文章

论文阅读：A Survey on Deep Learning for Named Entity Recognition

1079

动手学数据分析：简单数据重构

666

论文阅读：Aspect-based Sentiment Classification with Aspect-specific Graph Convolutional Networks

474

pip安装第三方库之清华镜像源安装方法

411

CLUENER 细粒度命名实体识别baseline：BiLSTM-CRF

364

分类专栏

最新评论

CLUENER 细粒度命名实体识别baseline...
 qdj6679: _forward_alg这个方法中： for i in range(feats.shape[1]): emit_score = fea ...
 pip本地安装第三方库方法
 Rock_y: 是的，不同的python版本，不同的操作系统都是不一样的
 pip本地安装第三方库方法
 学而时习之!: 原来是不同的python版本,对应的包不一样
 pip本地安装第三方库方法
 Rock_y: 有用呀，是实践成功的，你是哪里出了问题呢
 pip本地安装第三方库方法
 学而时习之!: 没有用

最新文章

推荐系统之GBDT+LR
 推荐系统之Wide&Deep
 推荐系统之矩阵分解和FM
 2020

论文阅读：A Survey on Deep Learning for Named Entity Recognition

原创

Rock_y

2020-09-11 09:54:11

1157

收藏 2

版权

文章标签：

神经网络

自然语言处理

深度学习

这是一篇2020年发的命名实体识别的综述性论文，从NER的语料库，定义，评估指标，到深度学习中的NER的技术都有涉及到。

A Survey on Deep Learning for Named Entity Recognition

1 INTRODUCTION

NER的演变

开展这项调查的动机

本次调查的贡献

2 BACKGROUND

2.1 What is NER?

2.2 NER资源：数据集和工具

2.3 NER评估指标

2.3.1 Exact-match Evaluation(精确匹配评估)

2.3.2 Relaxed-match Evaluation （宽松匹配评估）

2.4 NER的传统方法

2.4.1基于规则的方法

2.4.2无监督学习方法

2.4.3基于特征的监督学习方法

3 NER的3种深度学习技术

3.1为什么要深度学习NER?

3.2输入的分布式表示

3.2.1单词级表示

3.2.2字符级表示

3.2.3混合表示

3.3上下文编码器架构

3.3.1卷积神经网络

3.3.2循环神经网络

3.3.3递归神经网络

3.3.4神经语言模型

3.3.5 Deep Transformer

3.4标签解码器架构

3.4.1 Multi-Layer Perceptron + Softmax （多层感知器+ Softmax）

3.4.2 Conditional Random Fields （条件随机场）

3.4.3 Recurrent Neural Networks

3.4.4 Pointer Networks （指针网络）

3.5 Summary of DL-based NER

4 NER应用深度学习

4.1 Deep Multi-task Learning for NER （NER的深度多任务学习）

4.2 Deep Transfer Learning for NER （NER的深度转移学习）

4.3 Deep Active Learning for NER （对NER的深入主动学习）

4.4 Deep Reinforcement Learning for NER （NER的深层强化学习）



CSDN官方插件，现在体验可获得永久免费去广告特权！
.NET5零基础实战教程合集
QC活动小组的工具之关联图
网站漏扫工具的另类思路进行渗透测试
EBS 查找弹性域
2020年Arcgis在线调用天地图 、影像注记 、矢量地图、矢量注记。
webshell分类助手 微粒动态url分类助手 v1.0
迪兰恒进rx460酷能4G 原版BIOS 给需要的朋友
综合套餐
Redis入门到精通教程 （2020最新录制）
【LeetCode】312. Burst Balloons 戳气球 （Hard）（JAVA）
HClP-Transmission 华为认证传送网高级工程师 2.2 站点类型
电影院订票选座微信小程序（后台javaweb）使用教程
旅行商问题（TSP）三种解决算法 基于C++的编程
线上故障如何排查
CentOS7.5 安装MySQL的教程
OAuth 2.0实战课 09 笔记

10月	09月	08月	07月
14篇	16篇	7篇	7篇

4.5 Deep Adversarial Learning for NER （NER的深度对抗学习）

4.6 Neural Attention for NER （NER的神经注意力）

5挑战和未来方向

5.1 Challenges

5.2未来方向

6 CONCLUSION

命名实体识别（NER）是从 属于预先定义的语义类型（如人、位置、组织等）的文本中识别刚性指示符的任务。**NER**始终是许多自然语言应用程序（如问答、文本摘要和机器翻译）的基础。早期的**NER**系统在设计领域特定的特性和规则方面取得了巨大的成功，并以人工工程的成本实现了良好的性能。近年来，通过非线性处理，利用连续实值向量表示和语义合成的方法，在**NER**系统中进行了深入学习，取得了良好的性能。本文对国内外现有的网络学习技术进行了综述。我们首先介绍**NER**资源，包括标记的**NER**语料库和现成的**NER**工具。然后，我们基于三类分类法对现有作品进行系统分类：输入的分布式表示、上下文编码器和标记解码器。接下来，我们将调查最具代表性的方法，以了解新的**NER**问题设置和应用中最新的深度学习应用技术。最后，我们向读者介绍**NER**系统所面临的挑战，并概述这一领域的未来方向。

1 INTRODUCTION

命名实体识别（NERD）旨在从属于预定义语义类型（例如人，位置，组织等）的文本中识别对刚性指示符的提及[1]。****NER**不仅充当信息提取（IE）的独立工具，而且还在各种自然语言处理（NLP）应用程序中发挥重要作用，******例如文本理解[2]，[3]，信息检索[4]，[5]，自动文本摘要[6]，问题解答[7]，机器翻译[8]和知识库构建[9]等。

NER的演变

第六次邮件理解会议（MUC-6）[10]首次使用“命名实体”（NE），其任务是识别文本中的组织，人员和地理位置的名称，以及货币，时间和百分比表达式。自从MUC6以来，人们对NER的兴趣越来越高，各种科学事件（例如CoNLL03 [11]，ACE [12]，IREX [13]和TREC Entity Track [14]）为此主题付出了很多努力。

关于问题定义，**Petasis**等人[15]限制了命名实体的定义：“**ne**是专有名词，用作某物或某人的名称”。这种限制是由语料库中大量的专有名词来证明的。**Nadeau**和**Sekine**[1]声称，“命名”一词将任务仅限于一个或多个刚性指示符所代表的实体。刚性指示符，在[16]中定义，包括正确的名称和自然类术语，如生物物种和物质。尽管有不同的**NEs**定义，但研究人员已经就要识别的**NEs**类型达成共识。我们通常将**NEs**分为两类：一般**NEs**（例如，人和位置）和特定于域的**NEs**（例如，蛋白质、酶和基因）。

****** 在本文中，我们主要关注英文中的一般**NEs**。我们不主张本文是详尽的，也不主张所有**NER**作品都代表所有语言。

****** 关于**NER**中应用的技术，主要有四个方面：

- 1) 基于规则的方法，它们不需要注释数据，因为它们依赖手工编写的规则；
- 2) 无监督的学习方法，它依赖无监督的算法，而无需手动标记的培训示例；
- 3) 基于特征的监督学习方法，它依赖于监督的学习方法。具有仔细的特征工程的学习算法；
- 4) 基于深度学习的方法，它可以自动发现以端到端方式从原始输入进行分类和/或检测的表示。

我们将详细介绍1)、2)和3)，并回顾4)。

开展这项调查的动机

近年来，深度学习（DL，又称深度神经网络）在各个领域的成功应用引起了人们的广泛关注。从Collobert等人开始。**[17]**基于**DL**的具有最小特征工程的**NER**系统已经蓬勃发展。在过去的几年中，大量的研究已经将深度学习应用于NER，并逐步提高了最先进的性能[17]–[21]。这一趋势促使我们进行一项调查，报告NER研究中深度学习技术的现状。通过比较DL体系结构的选择，我们旨在找出影响NER性能的因素以及问题和挑战。

另一方面，尽管近几十年来NER研究一直在蓬勃发展，但据我们所知，迄今为止，在这一领域的评论很少。可以说，最成熟的一本书是由纳多和塞金于2007年出版的。本调查概述了从手工规则到机器学习的技术趋势。**Marerro**等人[22]总结了2013年NER从谬误、挑战和机遇的角度开展的工作。随后，**Patawar**和**Potey**[23]在2015年进行了简短的审查。最近的两次简短调查分别涉及新领域[24]和复杂实体[25]。总之，现有的调查主要涉及基于特征的机器学习模型，而不是现代基于DL的NER系统。与这项工作更密切的是2018年的两项最新调查[26]、[27]。**Goyal**等人[27]调查了NER的发展和进展。然而，他们没有包括最近的深度学习技术的进展。**Yadav**和**Bethard**[26]根据句子中的单词表示，对NER的最新进展进行了简短的调查。这项调查更多地关注输入的分布式表示（例如字符和字级嵌入），而不回顾上下文编码器和标记解码器。NER任务的应用深度学习（如多任务学习、转移学习、强化学习和对抗性学习）的最新趋势也不尽如人意。

本次调查的贡献

我们将重点回顾NER中深度学习技术的应用，以启发和指导这一领域的研究者和实践者。具体来说，我们将**NER**语料库，即现成的**NER**系统（来自学术界和工业界）合并为表格形式，为**NER**研究界提供有用的资源。然后，我们对**NER**的深度学习技术进行了全面的

vmware无法连接mks

虚拟机启动出现异常！

【JavaScript学习总结】——JS常用内置对象

snake贪吃蛇

c++笔记6：union的一种用法

平板表面燃烧ASTM E84测试

小白Java入门系列：入门第一课【你的第一个Java程序】

MySQL学习第四篇

零基础Java入门课程-IO流

交通事故应急救援预案

尹成JAVA快速入门

ARM微控制器C语言扩展官方文档

零基础Java入门课程-多线程

零基础Java入门课程-类和对象

python 使用h5py 中遇到的问题

Linux入门系列（11）： UBoot编译和使用

轻松掌握深度学习原理与实践

MySQL&8内幕讲解第一季

JMeter性能测试套餐

轻松掌握机器学习原理与实践

SpringBoot+MongoDB实现物流订单系统的代码

SAP FICO 项目实战之 银企直联

python爬虫入门

调查。为此，我们提出了一种新的分类法，它系统地沿着三个轴组织基于**DL**的方法：分布式输入表示法、上下文编码器（用于捕获标记解码器的上下文依赖项）和标记解码器（用于预测给定序列中的单词标签）。此外，我们还调查了最有代表性的方法，在新的**NER**问题设置和应用中最新的应用深度学习技术。最后，我们向读者介绍**NER**系统所面临的挑战，并概述这一领域的未来方向。

2 BACKGROUND

我们首先给出**NER**问题的形式化表述。然后我们介绍了广泛使用的**NER**数据集和工具。接下来，我们详细描述了评估指标，并总结了**NER**的传统方法。

2.1 What is NER?

命名实体是一个词或短语，它从一组具有类似属性的其他项中清楚地标识一个项[28]。命名实体的例子有：一般域中的组织、人和位置名称；生物医学域中的基因、蛋白质、药物和疾病名称。**NER**是将文本中的命名实体定位并分类为预定义实体类别的过程。形式上，给定一系列标记 $\mathbf{s}=\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_n$ ，**ner**将输出一个元组列表，即，其中每一个都是 \mathbf{s} 中提到的命名实体。这里， $\mathbf{l}_s \in [1, n]$ 和 $\mathbf{l}_e \in [1, n]$ 是一个命名实体的开始和结束索引； \mathbf{t} 是来自一个预定义类别集的实体类型。图1显示了一个例子，其中一个**NER**系统识别给定句子中的三个命名实体。当**NER**在**MUC-6**[10]中首次定义时，任务是识别文本中的人员、组织、地点和时间、货币、百分比表达式的名称。请注意，该任务集中于一个小组粗糙的实体类型和每个命名实体一个类型。我们将这种**NER**任务称为粗粒度**NER**[10], [11]。最近，一些细粒度的**NER**任务[29]–[33]将重点放在一组更大的实体类型上，在这些类型中，可以指定多个细粒度的类型。

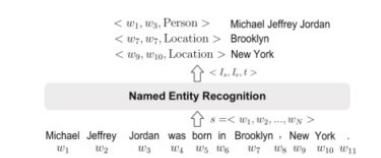


Fig. 1. An illustration of the named entity recognition task.

对于诸如信息检索、问答、机器翻译等各种下游应用程序，**NER**是一个重要的预处理步骤。这里，我们以语义搜索为例来说明**NER**在支持各种应用程序方面的重要性。语义搜索是一组技术，使搜索引擎能够理解用户查询背后的概念、含义和意图[34]。根据[4]，大约71%的搜索查询包含至少一个命名实体。在搜索查询中识别命名实体有助于我们更好地理解用户意图，从而提供更好的搜索结果。为了在搜索中合并命名实体，Raviv等人提出了基于实体的语言模型[34]，该模型考虑单个术语以及已注释为实体（文档和查询中）的术语序列。[35]。还有一些研究利用命名实体来增强用户体验，例如查询建议[36]、查询自动完成[37]、[38]和实体卡[39]、[40]。

2.2 NER资源：数据集和工具

高质量的注释对于模型学习和评估都至关重要。在下面，我们总结了广泛使用的数据集和现成的工具为英语**NER**。标记语料库是包含一个或多个实体类型注释的文档集合。表1列出了一些广泛使用的数据集及其数据源和实体类型（也称为标记类型）的数量。表1总结了2005年之前的数据集，主要是通过注释具有少量实体类型的新闻文章来开发的，适用于粗粒度的**NER**任务。之后，在各种文本源上开发了更多的数据集，包括维基百科文章、对话和用户生成的文本（例如，**tweets**和**youtube**评论以及**w-nut**中的**stackexchange**帖子）。标签的编号类型明显变大，例如**HYENA**中的505。我们还列出了一些特定于领域的数据集，特别是在**PubMed**和**Medline**文本上开发的数据集。实体类型的数量从**NCBI**疾病中的1个到**Genia**中的36个不等。

Corpus	Year	Text Source	#Tags	URL
MUC-6	1995	Wall Street Journal	7	https://catalog.ldc.upenn.edu/LDC2003T13
MUC-6 Plus	1995	Additional news to MUC-6	7	https://catalog.ldc.upenn.edu/LDC96T10
MUC-7	1997	New York Times news	7	https://catalog.ldc.upenn.edu/LDC2001T02
CoNLL03	2003	Reuters news	4	https://www.clips.uantwerpen.be/conll2003/ner/
ACE	2000 - 2008	Transcripts, news	7	https://www.ldc.upenn.edu/collaborations/past-projects/ace
OntoNotes	2007 - 2012	Magazine, news, web, etc.	18	https://catalog.ldc.upenn.edu/LDC2013T19
W-NUT	2015 - 2018	User-generated text	6/10	http://noisy-text.github.io
BBN	2005	Wall Street Journal	64	https://catalog.ldc.upenn.edu/LDC2005T33
WikiGold	2009	Wikipedia	4	https://figshare.com/articles/Learning_multilingual_named_entity_recognition_from_Wikipedia/5462500
WiNER	2012	Wikipedia	4	http://rali.iro.umontreal.ca/rali/en/winer-wikipedia-for-ner
WikiFiger	2012	Wikipedia	112	https://github.com/xiaoling/figer
HYENA	2012	Wikipedia	505	https://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/hyena
N ³	2014	News	3	http://aksw.org/Projects/N3NER/NEDNIF.html
Gillick	2016	Magazine, news, web, etc.	89	https://arxiv.org/e-print/1412.1820v2
FG-NER	2018	Various	200	https://fgner.aist.ai/
NNE	2019	News wire	114	https://github.com/nickyringland/nested_named_entities
GENIA	2004	Biology and clinical text	36	http://www.geniaproject.org/home
GENETAG	2005	MEDLINE	2	https://sourceforge.net/projects/bioc/files/
FSU-PRGE	2010	PubMed and MEDLINE	5	https://julielab.de/Resources/FSU_PRGE.html
NCBI-Disease	2014	PubMed	1	https://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/
BC5CDR	2015	PubMed	3	http://bioc.sourceforge.net/
DFKI	2018	Business news and social media	7	https://dfki-iti-re-group.bitbucket.io/product-corpus/

我们注意到，许多最近的NER工作报告了它们在conll03和ontoNotes数据集上的性能（见表3）。conll03包含路透社新闻的注释，有两种语言：英语和德语。英语数据集在体育新闻中占有很大一部分，并以四种实体类型（人、地点、组织和其他）进行注释[11]。本体注释项目的目标是对一个大型语料库进行注释，该语料库由各种类型（weblogs、news、talk show、broadcast、usenet新闻组和会话电话语音）组成，具有结构信息（语法和谓词参数结构）和浅语义（与本体和引用链接的词汇）。有5个版本，从1.0版到5.0版。文本注释有18种实体类型。我们还注意到两个Github存储库1承载了一些ner语料库。

网上有许多NER工具可用于预培训模型。表2按学术（顶部）和行业（底部）总结了英语NER的流行趋势。

TABLE 2

Off-the-shelf NER tools offered by academia and industry projects.

NER System	URL
StanfordCoreNLP	https://stanfordnlp.github.io/CoreNLP/
OSU Twitter NLP	https://github.com/aritter/twitter_nlp
Illinois NLP	http://cogcomp.org/page/software/
NeuroNER	http://neuroner.com/
NEksuite	http://nersuite.nlplab.org/
Polyglot	https://polyglot.readthedocs.io
Gimli	http://bioinformatics.ua.pt/gimli
spaCy	https://spacy.io/api/entityrecognizer
NLTK	https://www.nltk.org
OpenNLP	https://opennlp.apache.org/
LingPipe	http://alias-i.com/lingpipe-3.9.3/
AllenNLP	https://demo.allennlp.org/
IBM Watson	https://natural-language-understanding-demo.ng.bluemix.net
FG-NER	https://fgner.alt.ai/extractor/
Intellexer	http://demo.intellexer.com/
Reputate	https://reputate.com/named-entity-recognition-api-demo
AYLIEN	https://developer.aylien.com/text-api-demo
Dandelion API	https://dandelion.eu/semantic-text/entity-extraction-demo
displaCy	https://explosion.ai/demos/display-ent
ParallelDots	https://www.paralldots.com/named-entity-recognition
TextRazor	https://www.textrazor.com/named_entity_recognition

2.3 NER评估指标

NER系统通常通过比较其输出与人类注释来评估。比较可以通过精确匹配或轻松匹配来量化。

2.3.1 Exact-match Evaluation(精确匹配评估)

NER基本上包括两个子任务：边界检测和类型识别。在“精确匹配评估”[11]、[41]、[42]中，正确识别的实例需要一个系统同时正确识别其边界和类型。更具体地说，假阳性（fp）、假阴性（fn）和真阳性（tp）的数量用于计算Precision, Recall, and F-score。

•假阳性（fp）：由NER系统返回但不出现在基本事实中的实体。（实际不是但预测是）

•假阴性（FN）：NER系统未返回但出现在基本事实中的实体。（实际是但预测不是）

•真阳性（tp）：NER系统返回的实体，也出现在基本真理中。（实际是预测也是）

精度是指正确识别系统结果的百分比（看预测的）。召回是指系统正确识别的实体总数的百分比（看实际的）。

此外，宏观平均F分数和微平均F分数都考虑了多个实体类型的性能。宏平均f-得分独立计算不同实体类型的f-得分，然后取f-得分的平均值。微平均F-得分汇总了所有实体类型中的个别假阴性、假阳性和真阳性，然后应用它们获得统计数据。后者可能会受到语料库中大型类中实体的识别质量的严重影响。

2.3.2 Relaxed-match Evaluation（宽松匹配评估）

MUC-6[10]定义了一个宽松的匹配评估：只要为实体分配了正确的类型，而不论其边界如何，只要正确的类型与基本的真实边界重叠，就可以归为正确的类型；无论实体的类型分配如何，正确的边界都将归功于此。然后，ACE[12]提出了一个更复杂的评估程序。它解决了部分匹配和错误类型等一些问题，并考虑了命名实体的子类型。然而，这是有问题的，因为只有当参数固定时，最终分数才是可比的[1]、[22]、[23]。复杂的评价方法不直观，误差分析困难。因此，复杂的评价方法在最近的研究中并没有得到广泛的应用。

2.4 NER的传统方法

传统的NER方法大致可分为三类：基于规则的，无监督学习和基于特征的监督学习方法[1]，[26]。

2.4.1基于规则的方法

基于规则的NER系统依赖手工制定的规则。规则可以根据特定领域的研究者[9]、[43]和句法词汇模式[44]来设计。Kim[45]建议使用Brill规则推理方法进行语音输入。该系统根据Brill的部分语音标记自动生成规则。在生物医学领域，Hanisch等人[46]提议的ProMiner，它利用一个预先处理的同义词词典来识别在生物医学文本中提到的蛋白质和潜在基因。Quimbaya等人[47]提出了一种基于词典的电子健康记录NER方法。实验结果表明，该方法提高了召回率，但对精度的影响有限。其他一些著名的基于规则的NER系统包括LASIE II[48]、Netowl[49]、Facile[50]、SAR[51]、Fastus[52]和LTG[53]系统。这些系统主要基

于手工制作的语义和句法规则来识别实体。基于规则的系统在词汇穷尽时工作得很好。由于特定领域的规则和不完整的词典，这些系统往往具有较高的精度和较低的召回率，无法将系统转移到其他领域。

2.4.2无监督学习方法

无监督学习的一个典型方法是聚类[1]。基于聚类的**NER**系统基于上下文相似性从聚类组中提取命名实体。关键思想是，可以使用大型语料库上计算的词汇资源、词汇模式和统计信息来推断命名实体的提及情况。Collins等人[54]观察到，使用未标记的数据将监管要求降低到7条简单的“种子”规则。然后，作者提出了两种无监督的命名实体分类算法。类似地，knowitall[9]利用一组谓词名称作为输入，并从一小组通用提取模式中引导其识别过程。Nadeau等人[55]提出了一个无监督的系统，用于地名词典建设和命名实体歧义解决。该系统结合了基于简单而高效的启发式方法的实体提取和消歧。此外，张和Elhadad[44]提出了一种从生物医学文本中提取命名实体的无监督方法。他们的模型没有监督，而是使用术语、语料库统计（例如文档频率和上下文向量相反）和浅层的句法知识（例如名词短语分块）。对两个主流生物医学数据集的实验证明了其无监督方法的有效性和可推广性。

2.4.3基于特征的监督学习方法

应用监督学习，将**NER**转换为多类分类或序列标记任务。给定带注释的数据示例，特征被精心设计来表示每个培训示例。然后利用机器学习算法学习一个模型，从未看到的数据中识别相似的模式。特征工程在有监督的NER系统中至关重要。特征向量表示是对文本的抽象，其中一个或多个布尔值，数值或标称值[1]，[56]代表一个单词。单词级特征（例如，大小写，词法和词性标记）[57]-[59]，列表查找功能（例如，Wikipedia地名索引和DBpedia地名索引）[60]-[63]，以及文档和语料库特征（例如，本地语法和多次出现）[64]-[67]已广泛用于各种有监督的NER系统。[1]，[28]，[68]中讨论了更多特征设计。基于这些特征，许多机器学习算法已在有监督的NER中应用，包括隐马尔可夫模型（HMM）[69]，决策树[70]，最大熵模型[71]，支持向量机（SVM）[72]。和条件随机场（CRF）[73]。Bikel等。[74] [75]提出了第一个基于HMM的NER系统，名为Identifer，用于识别和分类名称，日期，时间表达式和数字量。此外，Szarvas等。[76]通过使用C4.5决策树和AdaBoostM1学习算法开发了一种多语言NER系统。一个主要优点是它提供了机会通过特征的不同子集训练几个独立的决策树分类器，然后通过多数表决方案组合其决策。Borthwick等。[77]通过应用最大熵理论提出了“最大熵命名实体”（MENE）。MENE能够在做出标记决策时利用各种知识资源。其他使用最大熵的系统可以在[78] – [80]中找到。McNamee和Mayfield[81]使用1000种语言相关和258种拼写和标点符号功能来训练SVM分类器。每个分类器都进行二元决策，决定当前令牌是否属于八个类中的一个，即人员、组织、位置和MIS标记的B-（开始）、I-（内部）。SVM在预测实体标签时不考虑“相邻”字。CRF考虑了上下文。McCallum和Li[82]提出了一种NER中CRF的特征归纳方法。实验在conll03上进行，英语成绩达到84.04%。Krishnan和Manning[67]提出了一种基于两个耦合CRF分类器的两阶段方法。第二个CRF利用来自第一个CRF输出的潜在表示。我们注意到，基于CRF的NER已经广泛应用于各个领域的文本，包括生物医学文本[58]、[83]、tweets[84]、[85]和化学文本[86]。

3 NER的3种深度学习技术

近年来，基于DL的NER模型成为主流，并取得了最先进的效果。与基于特征的方法相比，深度学习有利于自动发现隐藏的特征。接下来，我们首先简要介绍了什么是深度学习，以及为什么要对NER进行深度学习。然后我们调查基于DL的NER方法。

3.1为什么要深度学习NER？

深度学习是一个机器学习领域，它由多个处理层组成，用于学习具有多个抽象级别的数据表示[87]。典型的层是由前通和后通组成的人工神经网络。正向传递计算前一层输入的加权和，并通过非线性函数传递结果。后向传递是通过导数链规则计算目标函数相对于多层模块堆栈权重的梯度。深度学习的主要优点是表示学习的能力以及向量表示和神经处理所赋予的语义组合。这使得机器能够获得原始数据，并自动发现分类或检测所需的潜在表示和处理[87]。将深度学习技术应用于NER有三个核心优势。首先，NER受益于非线性变换，该变换生成从输入到输出的非线性映射。与线性模型（例如对数线性HMM和线性链CRF）相比，基于DL的模型能够通过非线性激活函数从数据中学习复杂而复杂的特征。其次，深度学习为设计NER功能节省了大量精力。传统的基于功能的方法需要大量的工程技术和领域专业知识。另一方面，基于DL的模型可有效地从原始数据中自动学习有用的表示和潜在因素。第三，可以通过梯度下降在端到端范式中训练深度神经NER模型。此属性使我们能够设计可能复杂的NER系统。

为什么在此调查中使用新的分类法？现有的分类法[26]，[88]基于字符级编码器，单词级编码器和标签解码器。我们认为“单词级编码器”的描述不准确，因为在基于DL的典型NER模型中两次使用了单词级信息：1）单词级表示用作原始特征，以及2）单词级表示（以及字符级表示形式）用于捕获上下文相关性以进行标签解码。在这项调查中，我们总结了NER的最新进展，其总体架构如图2所示。输入的分布式表示形式考虑了单词和字符级别的嵌入，以及结合了对特征有效的POS标签和地名词典等其他特征。上下文编码器将使用CNN，RNN或其他网络捕获上下文相关性。Tag解码器预测输入序列中令牌的标签。例如，在图2中，每个令牌都使用具有其类型的命名实体的B-（开始），I-（内部），E-（结束），S-（单个），或O-（外部）的命名实体。请注意，还有其他标记方案或标记符号，例如BIO。还可以训练标签解码器以检测实体边界，然后将检测到的文本跨度分类为实体类型

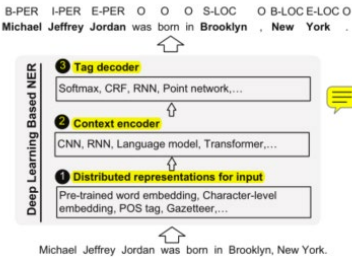


Fig. 2. The taxonomy of DL-based NER. From input sequence to predicted tags, a DL-based NER model consists of distributed representations for input, context encoder, and tag decoder.

3.2输入的分布式表示

表示单词的一个直接选择是**ont-hot**表示。在一个**onehot**空间中，两个不同的词具有完全不同的表示形式并且是正交的。分布式表示形式代表低维实值密集向量中的单词，其中每个维代表一个潜在特征。分布式表示法是自动从文本中学习的，它捕获单词的语义和句法属性，这些属性在**NER**的输入中未明确显示。接下来，我们回顾**NER**模型中使用的三种分布式表示形式：单词级，字符级和混合表示。

3.2.1单词级表示

一些研究[89] – [91]使用单词级表示，通常通过无监督算法（例如连续词袋（**CBOW**）和连续跳过语法模型[92]）对大量文本进行预训练。最近的研究[88]，[93]显示了这种预训练词嵌入的重要性。作为输入，可以在**NER**模型训练期间固定或进一步微调预训练的词嵌入。常用的词嵌入包括**Google Word2Vec**，斯坦福**GloVe**，**Facebook fastText**和**SENNA**。姚等。[94]提出了Bio-**NER**，一种基于深度神经网络架构的生物医学**NER**模型。使用跳过语法模型在**PubMed**数据库上训练Bio-**NER**中的单词表示形式。该词典在600维向量中包含205,924个单词。Nguyen等。[89]使用word2vec工具包从Gigaword语料库学习英语的词嵌入，并补充了来自BOLT（广泛使用语言技术）的新闻组数据。翟等。[95]设计了一个用于序列分块的神经模型，它由两个子任务组成：分段和标记。可以用**SENNA**嵌入或随机初始化的嵌入来馈送神经模型。郑等。[90]使用单个模型联合提取实体和关系。这个端到端模型使用word2vec在NYT语料库上学习到的词嵌入。Strubell等。[91]提出了一种基于迭代扩张卷积神经网络（**ID-CNN**）的标记方案。他们的模型中的查找表通过在**SENNA**语料库上通过skip-n-gram训练的100维嵌入来初始化。在他们提出的用于提取实体及其关系的神经模型中，Zhou等人。[96]使用了来自Google的预训练的300维单词向量。另外，GloV e [97]，[98]和fastText [99]也广泛用于**NER**任务。

3.2.2字符级表示

几项研究[100]，[101]并没有将单词级表示作为基本输入，而是结合了从端到端神经模型中学到的基于字符的单词表示。已经发现字符级表示对于利用显式子单词级信息（例如前缀和后缀）很有用。字符级表示的另一个优点是，它自然可以处理语音提示。因此，基于字符的模型能够推断出看不见的单词的表示形式，并共享词素级规则性的信息。有两种广泛使用的用于提取字符级表示的体系结构：基于**CNN**和基于**RNN**的模型。图3（a）和3（b）说明了两种架构。

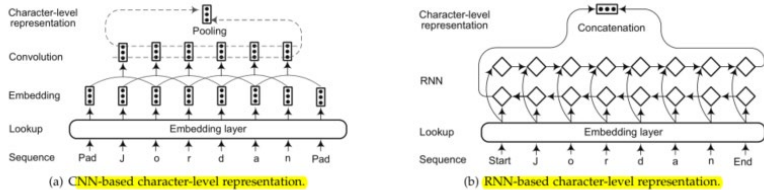


Fig. 3. CNN-based and RNN-based models for extracting character-level representation for a word.

Ma等。[97]利用CNN提取单词的字符级表示。然后，在将字符表示向量与单词嵌入进行串联之前，将其馈入RNN上下文编码器。同样，李等人。[98]应用了一系列的卷积和高速公路层来生成单词的字符级表示。单词的最终嵌入被馈送到双向递归网络中。杨等。[102]提出了一种用于**NER**的神经重排模型，其中在字符嵌入层的顶部使用了具有固定窗口大小的卷积层。最近，彼得斯等。[103]提出了ELMo单词表示，它是在具有字符卷积的两层双向语言模型的基础上计算的。对于基于**RNN**的模型，长短期记忆（**LSTM**）和门控循环单元（**GRU**）是基本单元的两个典型选择。库鲁等。[100]提出了Char**NER**，一种独立于语言的**NER**的字符级标记器。Char**NER**将句子视为字符序列，并利用**LSTM**提取字符级表示形式。它为每个字符而不是每个单词输出标签分布。然后从字符级标签获得单词级标签。他们的结果表明，以字符为主要代表优于以单词为基本输入单位。Lample等。

[19]利用双向LSTM提取单词的字符级表示。类似于[97]，字符级表示与单词查找表中的预训练单词级嵌入连接在一起。Gridach [104]在识别生物医学命名实体中研究了词嵌入和字符级表示。Rei等。[105]使用门控机制将字符级表示与单词嵌入相结合。通过这种方式，Rei的模型可以动态地确定要从字符或单词级组件中使用多少信息。Tran等。[101]引入了具有堆栈残差LSTM和可训练偏差解码的神经网络NER模型，其中从词嵌入和字符级RNN中提取词特征。杨等。[106]开发了一个模型，以统一的方式处理跨语言和多任务联合训练。他们采用了一个深层的双向GRU来从单词的字符序列中学习信息丰富的形态表示。然后，将字符级表示和单词嵌入连接起来，以生成单词的最终表示。

使用循环神经网络进行语言建模的最新进展使得将语言建模为字符分布成为可能。Akbik等人的上下文字符串嵌入。[107]，使用字符级神经语言模型为句子上下文中的字符串生成上下文嵌入。一个重要的特性是，嵌入内容由其周围的文本进行上下文文化，这意味着同一单词根据其上下文用途而具有不同的嵌入内容。图4说明了在句子上下文中提取单词“Washington”的上下文字符串嵌入的体系结构。

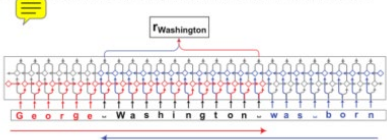


Fig. 4. Extraction of a contextual string embedding for word "Washington" in a sentential context [107]. From the forward language model (shown in red), the model extracts the output hidden state after the last character in the word. From the backward language model (shown in blue), the model extracts the output hidden state before the first character in the word. Both output hidden states are concatenated to form the final embedding of a word.

https://blog.csdn.net/Rock_y

3.2.3混合表示

除了单词级和字符级的表示法外，一些研究还将其他信息（例如，地名词典[18]，[108]，词汇相似性[109]，语言依赖性[110]和视觉特征[111]）纳入最终的表示法中。在输入上下文编码层之前换句话说，基于DL的表示以混合方式与基于特征的方法结合在一起。添加额外的信息可能会导致NER性能的提高，但代价是会损害这些系统的通用性。

[17]首次提出了将神经模型用于NER的方法，其中提出了基于单词序列上的时间卷积神经网络的体系结构。当合并常见的先验知识（例如，地名词典和POS）时，仅使用词级表示形式，生成的系统就会优于基线。在Huang等人的BiLSTM-CRF模型中。[18]，用于NER任务的特征有四种：拼写特征，上下文特征，词嵌入和地名词典特征。他们的实验结果表明，这些额外功能（即地名词典）可提高标记的准确性。Chiu和Nichols [20]的BiLSTM-CNN模型结合了双向LSTM和字符级CNN。除了单词嵌入以外，该模型还使用其他单词级别的功能（大写，词典）和字符级别的功能（表示字符类型的4维向量：大写，小写，标点符号等）。Wei等。[112]提出了一种基于CRF的神经系统，用于识别和标准化疾病名称。这个系统除了嵌入单词外，还使用了丰富的特征，包括单词、pos标签、分块和单词形状功能（例如字典和形态学功能）。Strubell等人[91]将100维嵌入与一个5维字形向量连接起来（例如，全部大写、未大写、首字母大写或包含大写字母）。Lin等人[113]连接字符级表示、字级表示和语法字表示（即位置标记、依赖角色、字位置、头位置）以形成一个综合的字表示。Aguilar等人提出了NER的多任务方法。[114]。这种方法利用CNN在字符级别捕获拼写特征和字形。对于单词级别的语法和上下文信息，例如POS和单词嵌入，该模型实现了LSTM体系结构。Jansson和Liu [115]提出将潜在狄利克雷分配（LDA）与字符级和单词级嵌入的深度学习相结合。

徐等。[116]提出了一种基于固定大小的普通遗忘编码（FOFE）的NER局部检测方法[117]，FOFE探索了每个片段及其上下文的字符级和单词级表示。在Moon等人的多模式NER系统中。[118]，对于嘈杂的用户生成的数据（如推文和Snapchat标题），单词嵌入，字符嵌入和视觉功能会与模式关注结合在一起。Ghaddar和Langlais [109]发现，词法特征在神经NER系统中被大部分丢弃是不公平的。他们提出了一种替代的词汇表示形式，可以离线训练，并且可以将其添加到任何神经NER系统中。使用120维向量为每个单词计算词法表示，其中每个元素编码该单词与实体类型的相似性。最近，Devlin等人。[119]提出了一种称为BERT的新语言表示模型，即来自变压器的双向编码器表示。BERT使用屏蔽语言模型来启用预训练的深度双向表示。对于给定的令牌，其输入表示形式是通过将相应的位置，片段和令牌嵌入相加而构成的。请注意，经过预先训练的语言模型嵌入通常需要大规模的语料库进行训练，并且本质上并入了辅助嵌入（例如位置和片段嵌入）。因此，在本次调查中，我们将这些语境化的语言模型嵌入归类为混合表示

3.3上下文编码器架构

在这里，我们现在回顾广泛使用的上下文编码器体系结构：卷积神经网络，递归神经网络，循环神经网络和深度transformer。

3.3.1卷积神经网络

Collobert等。[17]提出了一种句子逼近网络，其中考虑到整个句子对单词进行标记，如图5所示。在输入表示阶段之后，将输入序列中的每个单词嵌入到N维向量中。然后使用卷积层在每个单词周围产生局部特征，并且卷积层的输出大小取决于句子中单词的数量。通过组合由卷积层提取的局部特征向量来构造全局特征向量。全局特征向量的维数是固定的，与句子的长度无关，以便应用后续的标准仿射图层。广泛使用两种方法来提取全局特征：对句子中的位置（即“时间”步长）进行最大值或平均运算。最后，将这些固定大小的全局特征馈入标签解码器，以计算网络输入中单词的所有可能标签的分配分数。继Collobert的工作之后，Yao等人。[94]提出将Bio-NER用于生物医学NER。Wu等。[120]利用卷积层来生成由许多全局隐藏节点表示的全局特征。然后将局部特征和全局特征都输入到标准仿射网络中，

以识别临床文本中的命名实体。

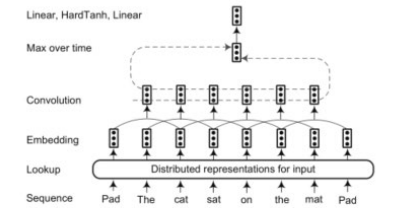


Fig. 5. Sentence approach network based on CNN [17]. The convolution layer extracts features from the whole sentence, treating it as a sequence with global structure.

周等。[96]观察到后一个词比前一个词更能影响最后一句的表达。然而，重要的词可能出现在句子的任何地方。在他们提出的名为BLSTMRE的模型中，blstm用于捕获长期依赖关系并获得输入序列的完整表示。然后CNN被用来学习一个高级的表示，然后将其输入到一个sigmoid分类器中。最后，将整个句子表示（由blstm生成）和关系表示（由sigmoid分类器生成）输入另一个lstm中，以预测实体。

传统上，长度为N的序列的LSTM的时间复杂度以并行方式为O（N）。Strubell等。[91]提出的ID-CNN，称为迭代膨胀卷积神经网络，由于能够处理更大的上下文和结构化预测，因此计算效率更高。图6显示了一个扩张的CNN块的架构，其中四个堆叠的宽度为3的扩张的卷积产生令牌表示。实验结果表明，与Bi-LSTM-CRF相比，ID-CNN的测试时间加快了14-20倍，同时保持了相当的准确性

3.3.2循环神经网络

黄等人的工作。[18]率先使用双向LSTM CRF架构对标记任务（POS，分块和NER）进行排序。继[18]之后，大量的作品[19]，[20]，[89]，[90]，[95]–[97]，[101]，[105]，[112]，[113]应用了BiLSTM作为编码序列上下文信息的基本架构。杨等。[106]在字符和单词级别都采用了深层GRU来编码形态和上下文信息。他们通过共享体系结构和参数，将模型进一步扩展到跨语言和多任务联合训练。Gregoric等。[121]在同一输入上采用了多个独立的双向LSTM单元。他们的模型通过使用模型间正则化术语来促进LSTM单元之间的多样性。通过将计算分布在多个较小的LSTM中，他们发现参数总数减少了。最近，一些研究[122]，[123]设计了基于LSTM的神经网络，用于嵌套命名实体识别。Katiyar和Cardie [122]提出了对基于标准LSTM的序列标签模型的修改，以处理嵌套的命名实体识别。Ju等。[123]提出了一种神经模型，通过动态堆叠平坦的NER层直到没有外部实体被提取来识别嵌套实体。每个平坦的NER层都使用双向LSTM来捕获顺序上下文。该模型将LSTM层的输出合并到当前平面NER层中，以构造检测到的实体的新表示形式，然后将其输入到下一个平面NER层中。

3.3.3递归神经网络

递归神经网络是非线性自适应模型，能够通过以拓扑顺序遍历给定的结构来学习深度的结构化信息。命名实体与语言成分（例如名词短语）高度相关[98]。但是，典型的顺序标注方法很少考虑句子的短语结构。为此，李等人。[98]提出对NER的选区结构中的每个节点进行分类。该模型递归地计算每个节点的隐藏状态向量，并通过这些隐藏向量对每个节点进行分类。图8显示了如何为每个节点递归计算两个隐藏状态特征。自下而上的方向计算每个节点的子树的语义组成，自上而下的对应对象将包含该子树的语言结构传播到该节点。给定每个节点的隐藏矢量，网络将计算实体类型加上特殊的非实体类型的概率分布。

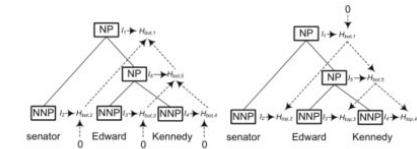


Fig. 8. Bidirectional recursive neural networks for NER [98]. The computations are done recursively in two directions. The bottom-up direction computes the semantic composition of the subtree of each node, and the top-down counterpart propagates to that node the linguistic structures which contain the subtree.

3.3.4神经语言模型

语言模型是描述序列生成的一系列模型。Rei [124]提出了一个具有次要目标的框架-学习预测数据集中每个单词的周围单词。图9展示了带有NER任务的简短句子的体系结构。在每个时间步长（即令牌位置），网络都经过优化以预测序列中的前一个令牌，当前标签和下一个令牌。添加的语言建模目标鼓励系统学习更丰富的功能表示，然后将这些功能表示重新用于序列标记。彼得斯等。[21]提出了TagLM，一种语言模型增强序列标记器。该标记器为输入序列中的每个标记考虑预训练的词嵌入和双向语言模型嵌入，以进行序列标记任务。图10显示了LM-LSTM-CRF模型[125]，[126]的体系结构。语言模型和序列标记模型以多任务学习的方式共享

相同的字符级层。来自字符级嵌入，预训练的单词嵌入和语言模型表示的向量被串联起来，并输入到单词级LSTM中。实验结果表明，多任务学习是指导语言模型学习任务特定知识的有效方法。

图4显示了使用Akblk等人的神经字符级语言建模进行上下文字符串嵌入。[107]。他们利用了前向递归神经网络的隐藏状态来创建上下文化的词嵌入。该模型的主要优点是字符级语言模型独立于标记化和固定的词汇表。彼得斯等。[103]提出了ELMo表示形式，该表示形式是在具有字符卷积的两层双向语言模型的基础上计算的。这种新型的深层上下文化词表示法能够对词用法的复杂特征（例如语义和语法）以及跨语言上下文的用法变化（例如多义）进行建模。

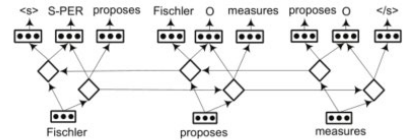


Fig. 9. A sequence labeling model with an additional language modeling objective [124], performing NER on the sentence "Fischer proposes measures". At each token position (e.g., "proposes"), the network is optimised to predict the previous word ("Fischer"), the current label ("O"), and the next word ("measures") in the sequence.

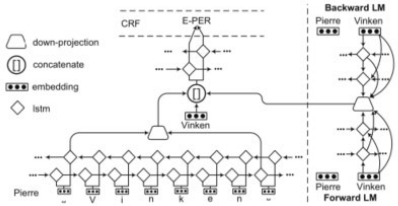


Fig. 10. Sequence labeling architecture with contextualized representations [125]. Character-level representation, pre-trained word embedding and contextualized representation from bidirectional language models are concatenated and further fed into context encoder.

3.3.5 Deep Transformer

使用Transformer预训练的这些语言模型嵌入正在成为NER的新范例。首先，这些嵌入是上下文相关的，可以用来代替传统的嵌入，例如Google Word2vec和Stanford GloVe。一些研究[108]，[110]，[133] – [136]通过利用传统嵌入和语言模型嵌入的组合获得了令人鼓舞的性能。其次，这些语言模型的嵌入可以通过一个附加的输出层进行进一步的微调，以完成包括NER和组块在内的各种任务。特别是，李等人。[137]，[138]将NER任务定义为机器阅读理解（MRC）问题，可以通过微调BERT模型来解决。

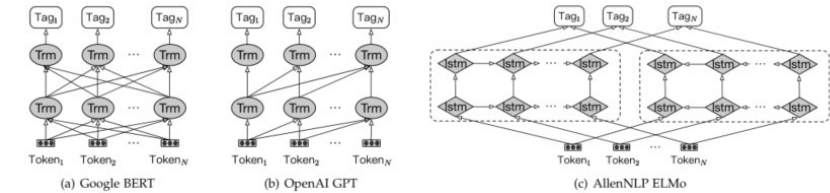


Fig. 11. Differences in pre-training model architectures [119]. Google BERT uses a bidirectional Transformer (abbreviated as "Trm"). OpenAI GPT uses a left-to-right Transformer. AllenNLP ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks.

3.4标签解码器架构

标签解码器是NER模型的最后阶段。它以上下文相关的表示形式作为输入，并生成与输入序列相对应的标签序列。图12总结了标签解码器的四种体系结构：MLP + softmax层，条件随机场（CRF），循环神经网络和指针网络。

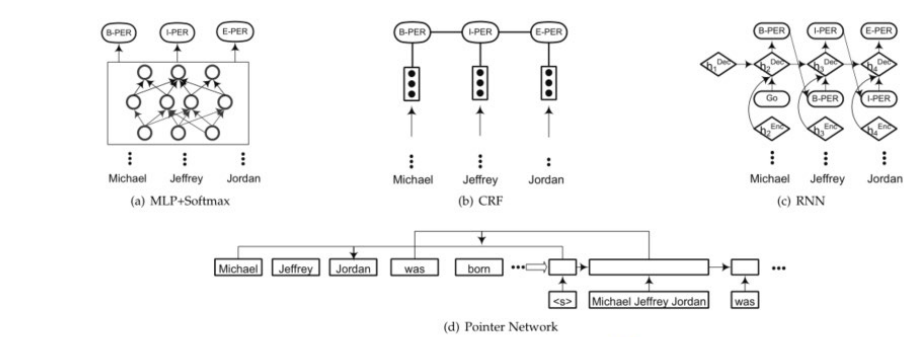


Fig. 12. Differences in four tag decoders: MLP+Softmax, CRF, RNN, and Pointer network.

https://blog.csdn.net/Rock_y

3.4.1 Multi-Layer Perceptron + Softmax（多层感知器+ Softmax）

NER通常被表述为序列标记问题。使用多层**Perceptron + Softmax**层作为标签解码器层，序列标记任务将转换为多类分类问题。每个单词的标签是根据上下文相关表示独立预测的，无需考虑其邻居。较早引入的许多**NER**模型[91]，[98]，[116]，[119]，[139]使用**MLP + Softmax**作为标签解码器。作为域特定的NER任务，Tomori等人。[140]使用**softmax**作为标签解码器来预测日本象棋游戏的游戏状态。他们的模型既接受来自文本的输入，也接受来自国际象棋棋盘的输入（9×9正方形，其中有40种14种不同类型的棋子），并预测该游戏特有的21个命名实体。文本表示和游戏状态嵌入都被馈送到**softmax**层，以使用**BIO**标签方案预测命名实体。

3.4.2 Conditional Random Fields（条件随机场）

条件随机场（**CRF**）是一个以观察序列为整体条件的随机场[73]。**CRF**已广泛用于基于特征的监督学习方法中（请参见第2.4.3节）。许多基于深度学习的**NER**模型使用**CRF**层作为标签解码器，例如，在双向**LSTM**层[18]，[90]，[103]，[141]顶部和**CNN**层[17]顶部，[91]，[94]。表3列出了**CRF**是标签解码器的最常见选择，[107]使用**CRF**标签解码器可以实现**CoNLL03**和**OntoNotes5.0**的最新性能。但是，**CRF**无法充分利用段级别的信息，因为段的内部属性无法用单词级别的表示形式进行完全编码。卓等。[142]然后提出了门控递归半马尔可夫**CRF**，它直接对片段而不是单词进行建模，并通过门控递归卷积神经网络自动提取片段级特征。最近，Ye和Ling [143]提出了用于神经序列标记的混合半马尔可夫**CRF**。这种方法采用段而不是单词作为特征提取和过渡建模的基本单位。单词级标签用于推导片段分数。因此，这种方法能够利用词级信息和词段级信息进行片段得分计算。

3.4.3 Recurrent Neural Networks

一些研究[88]–[90]，[96]，[144]探索了**RNN**来解码标签。沉等。[88]报道，当实体类型的数量很大时，**RNN**标签解码器的性能优于**CRF**，并且训练速度更快。图12（c）说明了基于**RNN**的标签解码器的工作流程，该流程用语言模型以贪婪地生成标签序列。第一步的[GO]符号作为 y_1 提供给**RNN**解码器。随后，在每个时间步 i ，**RNN**解码器根据先前步骤标签 y_i ，先前步骤解码器隐藏状态 h_{Dec}^i 和当前步骤编码器隐藏状态 $h_{Enc}^i + 1$ 来计算当前解码器隐藏状态 h_{Dec}^{i+1} ；当前输出标签 y_{i+1} 通过使用**softmax**损失函数进行解码，并进一步作为输入提供给下一个时间步长。最后，我们获得所有时间步长的标签序列。

3.4.4 Pointer Networks（指针网络）

指针网络应用**RNN**来学习输出序列的条件概率，其中元素是与输入序列中的位置相对应的离散标记[145]，[146]。它使用**SoftMax**概率分布作为“指针”来表示可变长度字典。翟等。[95]首先应用指针网络生成序列标记。如图12（d）所示，指针网络首先标识一个块（或一个段），然后标记它。重复此操作，直到处理输入序列中的所有字。在图12（d）中，给定起始标记“”，首先标识段“**michael jeffery jordan**”，然后标记为“**person**”。在指针网络中，可以通过两个独立的神经网络进行分割和标记。接下来，“迈克尔·杰弗里·乔丹”作为输入，输入指针网络。因此，段“**was**”被识别并标记为“**o**”。

3.5 Summary of DL-based NER

Architecture Summary:

表3根据神经网络的体系结构选择总结了最近的研究成果。**BILSTM CRF**是使用深度学习的**NER**最常见的架构。方法[132]以完形的方式预先训练双向变压器模型，在**conll03**上实现了最先进的性能（93.5%）。Bert和Dice Loss[138]的作品在**OnNoteS5.0**上实现了最先进的性能（92.07%）。**NER**系统的成功很大程度上依赖于它的输入表示。整合或微调预先训练的语言模型嵌入正在成为神经网络的一种新范式。在利用这些语言模型嵌入时，性能有了显著的提高[103]、[107]、[108]、[132]–[138]。表3的最后一列列出了一些基准数据集在F-score中报告的性能。虽然正式文件（如**conll03**和**ontotes5.0**）中报告了高F分数，但噪声数据（如**w-nut17**）的**NER**仍然具有挑战性。

TABLE 3
Summary of recent works on neural NER. LSTM: long short-term memory; CNN: convolutional neural network; GRU: gated recurrent unit; LM: language model; ID-CNN: iterated dilated convolutional neural network; BRNN: bidirectional recursive neural network; MLP: multi-layer perceptron; CRF: conditional random field; Semi-CRF: Semi-markov conditional random field; F2F2: fixed-size ordinaly tagging encoding.

Work	Character	Input representation	Context encoder	Tag decoder	Performance (F-score)	
[94]	-	Trained on PubMed	POS	CNN	CRF	GENIA 71.01%
[95]	-	Trained on Gigaword	-	GRU	GRU	ACE 2005 90.00%
[96]	-	Random	-	LSTM	CRF	ATIS 96.56%
[97]	-	Trained on NYT	-	LSTM	CRF	NTT 49.50%
[91]	-	SENNA	Word shape	ID-CNN	CRF	CoNLL03 90.65%
[98]	-	Google word2vec	-	LSTM	LSTM	OntoNotes0.9 86.84%
[100]	LSTM	GloVe	-	LSTM	CRF	CoNLL04 75.0%
[97]	CNN	GloVe	-	LSTM	CRF	CoNLL03 84.52%
[105]	LSTM	Google word2vec	-	LSTM	CRF	CoNLL03 91.21%
[106]	LSTM	SENNA	-	LSTM	CRF	CoNLL03 84.09%
[109]	CNN	SENNA	-	GRU	CRF	CoNLL03 90.94%
[104]	GRU	SENNA	POS	GRU	CRF	CoNLL03 90.94%
[99]	LSTM-LM	-	-	BRNN	Softmax	OntoNotes0.9 87.21%
[107]	-	-	-	LSTM	CRF	CoNLL03 93.09%
[103]	CNN-LSTM-LM	-	-	LSTM	CRF	OntoNotes0.9 89.71%
[17]	-	Random	POS	CNN	CRF	CoNLL03 92.22%
[18]	-	SENNA	Spelling, n-gram, gazetteer	LSTM	CRF	CoNLL03 90.05%
[20]	CNN	SENNA	capitalization, lexicons	LSTM	CRF	CoNLL03 91.62%
[116]	-	-	-	MLP	CRF	OntoNotes0.9 86.34%
[101]	LSTM	GloVe	-	CRF	CRF	CoNLL03 91.17%
[113]	LSTM	GloVe	Syntactic	LSTM	CRF	CoNLL03 91.07%
[102]	CNN	SENNA	-	LSTM	CRF	W-NUT17 40.42%
[114]	CNN	Twitter Word2vec	POS	LSTM	Reformer	CoNLL03 91.62%
[115]	LSTM	GloVe	POS, topics	LSTM	CRF	W-NUT17 41.80%
[118]	LSTM	GloVe	Images	LSTM	CRF	W-NUT17 41.81%
[109]	LSTM	SSkip	Lexical	LSTM	CRF	ImageCaption 52.4%
[119]	LSTM	WordPiece	Segment, position	Transformer	Softmax	CoNLL03 92.0%
[121]	LSTM	SENNA	-	LSTM	Softmax	OntoNotes0.9 87.99%
[124]	LSTM	Google Word2vec	-	LSTM	CRF	CoNLL03 92.8%
[21]	GRU	SENNA	LM	GRU	CRF	CoNLL03 91.48%
[126]	LSTM	GloVe	-	LSTM	CRF	CoNLL03 86.26%
[142]	LSTM	SENNA	POS, gazetteers	CNN	Semi-CRF	CoNLL03 91.09%
[143]	LSTM	GloVe	-	Semi-CRF	Semi-CRF	CoNLL03 91.71%
[98]	CNN	Trained on Gigaword	-	LSTM	LSTM	CoNLL03 90.69%
[110]	-	GloVe	ELMo, dependency	LSTM	CRF	CoNLL03 91.38%
[108]	CNN	GloVe	ELMo, gazetteers	LSTM	Semi-CRF	OntoNotes0.9 86.13%
[133]	LSTM	GloVe	ELMo, POS	LSTM	CRF	CoNLL03 92.4%
[137]	-	BERT	-	Softmax	Softmax	CoNLL03 93.04%
[138]	-	-	-	Softmax +Dice Loss	-	CoNLL03 92.28%
[134]	LSTM	GloVe	BERT, document-level embeddings	LSTM	CRF	CoNLL03 93.37%
[135]	CNN	GloVe	BERT, global embeddings	GRU	GRU	OntoNotes0.9 90.3%
[132]	CNN	-	Character-level LM embeddings	LSTM	CRF	CoNLL03 93.47%
[136]	-	GloVe	Predefined contextual embeddings	RNN	CRF	CoNLL03 93.47%

Architecture Comparison：

我们从三个角度讨论优缺点：输入、编码器和解码器。

首先，对于外部知识是否应该或如何整合到基于DL的NER模型中，还没有达成共识。一些研究[108]、[110]、[133]、[142]表明，外部知识可以提高NER绩效。然而，缺点也很明显：
 1）获取外部知识是劳动密集型的（例如，观察者）或计算成本高（例如，依赖性）；
 2）整合外部知识对端到端学习产生不利影响，损害了基于DL的系统的通用性。

其次，在大型语料库上对变压器进行预训练时，变压器编码器比LSTM更有效。如果变压器没有经过预先培训，并且培训数据有限，则变压器无法完成NER任务[147]，[148]。另一方面，当序列n的长度小于表示d的维数时，变压器编码器比递归层更快（复杂度：**self-attention** $O(n^2 \cdot d)$ 和**recurrent** $O(n \cdot d^2)$ ）[128]。

第三，RNN和指针网络译码器的一个主要缺点是译码过于贪婪，这意味着当前步骤的输入需要上一步骤的输出。这种机制可能会对速度产生重大影响，并且是并行化的障碍。CRF是标签解码器最常用的选择。CRF在采用非语言模型（即非上下文文化）嵌入（如Word2vec和Glove）时，能够捕获标签转换依赖性。然而，当实体类型的数量很大时，CRF的计算代价可能很高。更重要的是，在采用上下文文化语言模型嵌入（如Bert和Elmo[137]、[139]）时，与SoftMax分类相比，CRF并不总是能够获得更好的性能。对于最终用户，要选择的体系结构取决于数据和领域任务。如果数据丰富，可以考虑从零开始使用RNN和微调上下文文化语言模型的培训模型。如果数据稀缺，采用传输策略可能是更好的选择。对于Newswires领域，有许多预先培训过的现成模型可用。对于特定领域（如医疗和社会媒体），使用特定领域的数据对通用上下文文化语言模型进行微调通常是一种有效的方法。

NER for Different Languages：

在本次调查中，我们主要关注英语和一般领域的NER。除了英语之外，还有许多关于其他语言或跨语言环境的研究。Wu等人[120]和Wang等人[149]在中文临床文本中调查了NER。张和杨[150]提出了一个格型结构的汉语NER LSTM模型，该模型对输入字符序列以及所有与词汇匹配的潜在单词进行编码。除汉语外，许多其他语言的研究都在进行。例如蒙古语[151]、捷克语[152]、阿拉伯语[153]、乌尔都语[154]、越南语[155]、印度尼西亚语[156]和日语[157]。每种语言都有其自身的特点，可以理解NER任务对该语言的基本原理。还有许多研究[106]、[158]–[160]旨在通过将知识从源语言转移到目标语言（很少或没有标签）来解决跨语言环境中的NER问题。

4 NER应用深度学习

第3.2、3.3和3.4节概述了NER的典型网络架构。在本节中，我们调查了NER正在探索的最新应用的深度学习技术。

4.1 Deep Multi-task Learning for NER（NER的深度多任务学习）

多任务学习[161]是一种共同学习一组相关任务的方法。通过考虑不同任务之间的关系，期望多任务学习算法比单独学习每个任务的算法取得更好的效果。

Collobert等人[17]培训了一个窗口/句子方法网络，共同执行POS、Chunk、NER和SRL任务。这种多任务机制允许训练算法发现对所有感兴趣的任务有用的内部表示。Yang等人[106]提出了一个多任务联合模型，学习特定于语言的规则，为pos、chunk和ner任务联合训练。REI[124]发现，通过在培训过程中包含一个无监督的语言建模目标，序列标记模型实现了一致的性能改进。Lin等人[160]针对低资源环境，提出了一种多语言多任务体系结构，可以有效地传递不同类型的知识，以改进主模型。

除了考虑**NER**和其他序列标记任务外，多任务学习框架可用于实体和关系的联合提取**[90]**、**[96]**，或将**NER**建模为两个相关子任务：实体分割和实体类别预测**[114]**、**[162]**。在生物医学领域，由于不同数据集的差异，每个数据集上的**NER**被认为是多任务设置中的一项任务**[163]**，**[164]**。这里的一个主要假设是不同的数据集共享相同的字符级和字级信息。然后采用多任务学习的方法来提高数据的使用效率，并鼓励模型学习更多的广义表示。

4.2 Deep Transfer Learning for NER（NER的深度转移学习）

转移学习的目的是利用从源域学到的知识，在目标域上执行机器学习任务**[165]**。在**NLP**中，转移学习也称为域适应。对于**NER**任务，传统方法是通过引导算法**[166]**–**[168]**。最近，一些方法**[127]**、**[169]**–**[173]**已被提出，用于低资源和跨领域的**NER**，使用深度神经网络。Pan等人**[169]**提出了一种跨域**NER**的转移联合嵌入（**TJE**）方法。Tje采用标签嵌入技术将多类分类转化为低维潜空间中的回归。Qu等人**[174]**观察到相关的命名实体类型通常共享词汇和上下文特性。他们的方法使用两层神经网络学习源和目标命名实体类型之间的相关性。它们的方法适用于源域具有与目标域相似（但不相同）的命名实体类型的设置。Peng和Dredze**[162]**探讨了多任务学习环境下的转移学习，在多任务学习环境中，他们考虑了两个领域：新闻和社交媒体，用于两个任务：分词和**NER**。在传递学习的设置中，不同的神经模型通常在源任务和目标任务之间共享不同的模型参数部分。Yang等人**[175]**首先研究了不同表示层的可转移性。然后，他们为跨域、跨语言和跨应用场景提供了三种不同的参数共享架构。如果两个任务有可映射的标签集，则有一个共享的**CRF**层，否则，每个任务学习一个单独的**CRF**层。实验结果表明，在低资源条件下（即可用注释较少），各种数据集都有显著的改进。PIUS和Mark**[176]**扩展了杨的方法，允许对非正式语料库进行联合培训（如Wnut 2017），并将句子级特征表示纳入其中。他们的方法在Wnut 2017年的共同任务**NER**中获得第二名，获得了40.78%的F1分数。Zhao等人**[177]**提出了一个域自适应的多任务模型，其中完全连接层适应不同的数据集，并分别计算**CRF**特征。赵的模型的一个主要优点是，在数据选择过程中，过滤出具有不同分布和不对齐注释准则的实例。与这些参数共享架构不同，Lee等人**[170]**通过在源任务上训练模型，并使用在目标任务上训练的模型进行微调，在**NER**中应用转移学习。近年来，林和鲁**[171]**还通过引入三个神经适应层（单词适应层、句子适应层和输出适应层），提出了**NER**的微调方法。Beryozkin等人**[178]**提出了一个异类标记集**NER**设置的标记层次模型，在推理过程中使用层次结构将细粒度标记映射到目标标记集。此外，一些研究**[164]**、**[179]**、**[180]**探索了生物医学**NER**中的转移学习，以减少所需的标记数据量。

4.3 Deep Active Learning for NER（对NER的深入主动学习）

主动学习背后的关键思想是，主动学习背后的关键思想是，如果允许从机器学习算法中选择学习的数据，则机器学习算法的性能将大大降低。深度学习通常需要大量的培训数据，而这些数据的获取成本很高。因此，将深度学习与主动学习相结合，可以减少数据注释的工作量。主动学习的培训分多轮进行。然而，传统的主动学习方案对于深度学习来说代价高昂，因为每一轮之后，它们都需要用新注释的样本对分类器进行完整的再培训。因为从零开始的再培训对于深度学习来说是不实际的，Shen等人**[88]**建议对每批新标签进行**NER**增量培训。他们将新注释的样本与现有的样本混合，并在新一轮查询标签之前，更新少量时期的神经网络权重。具体地说，在每一轮的开始，主动学习算法选择要注释的句子，达到预定的预算。在接收到所选注释后，通过对增强数据集的培训来更新模型参数。主动学习算法在选择要注释的句子时采用不确定性抽样策略**[182]**。实验结果表明，主动学习算法仅使用**24.9%**的英语数据集训练数据和**30.1%**的中文数据集，就可以达到**99%**的全数据训练的最佳深度学习模型的性能。此外，**12.0%**和**16.9%**的训练数据足以使深度主动学习模型优于在完整训练数据上学习的浅层模型**[183]**。

4.4 Deep Reinforcement Learning for NER（NER的深层强化学习）

强化学习（**RL**）是受行为主义心理学启发而产生的机器学习的一个分支，它涉及软件代理如何在环境中采取行动，从而最大化一些累积奖励**[184]**，**[185]**。其理念是，一个代理将通过与环境交互并获得执行行为的奖励来从环境中学习。具体地说，**RL**问题可以表述为**[186]**：环境被建模为一个具有输入（来自代理的动作）和输出（对代理的观察和奖励）的随机有限状态机。它由三个关键部分组成：（i）状态转换函数；（ii）观察（即输出）函数；（iii）奖励函数。代理也被建模为一个随机有限状态机，具有输入（环境的观察/回报）和输出（环境的动作）。它由两部分组成：（i）状态转换函数和（ii）策略/输出函数。代理的最终目标是通过尝试最大化累积奖励来学习良好的状态更新功能和策略。Narasimhan等。[187]将信息提取的任务建模为马尔可夫决策过程（**MDP**），该过程动态地合并了实体预测，并提供了从一组自动生成的替代方案中选择下一个搜索查询的灵活性。该过程包括发出搜索查询，从新来源中提取以及对提取的值进行核对，然后重复该过程，直到获得足够的证据为止。为了学习代理的良好策略，他们利用深层**Q**网络**[188]**作为函数逼近器，其中，通过使用深层神经网络来近似状态作用值函数（即**Q**函数）。最近，Yang等。[189]利用远距离监督产生的数据在新领域中进行了新型的实体识别。实例选择器基于强化学习，并从**NE**标记器获得反馈奖励，旨在选择肯定的句子以减少嘈杂注释的影响。

4.5 Deep Adversarial Learning for NER（NER的深度对抗学习）

对抗学习**[190]**是明确训练对抗实例模型的过程。目的是使模型对攻击更健壮或减少其在纯输入上的测试错误。对抗网络学习通过2人游戏从训练分布中产生：一个网络生成候选（生成网络），另一个评估候选（区分网络）。通常，生成网络学习从潜在空间映射到感兴趣的特定数据分布，而区分网络则在生成器生成的候选对象与真实数据分布的实例之间进行区分**[191]**。对于**NER**，通常以两种方式产生对抗性示例。一些研究**[192]**–**[194]**将源域中的实例视为目标域的对抗示例，反之亦然。例如，Li等。[193]和曹等人。[194]都结合了来自其他领域的对抗性例子，以鼓励跨域**NER**的领域不变特征。另一种选择是通过添加带有扰动的原始样

本来准备对抗样本。例如，在 [104] 中提出的双重对抗传输网络（D²TN）旨在解决资源匮乏的对抗性问题。对抗性样本是通过添加原始样本而产生的，该样本的扰动受一个小范数 Θ 限制，以最大化损失函数，如下所示： $\eta x = \arg \max_{\eta} \|\eta x\|_2 \leq \Theta; x + \eta$ ，其中 Θ 是当前模型参数可以根据验证集确定。通过 $x_{adv} = x + \eta x$ 构造一个对抗性示例。对分类器进行原始示例和对抗示例混合训练，以提高通用性。

4.6 Neural Attention for NER（NER的神经注意力）

注意力机制大致基于人类发现的视觉注意力机制[196]。例如，人们通常以“高分辨率”关注图像的某个区域，而以“低分辨率”感知周围的区域。神经注意力机制使神经网络能够专注于其输入的子集。通过应用注意力机制，**NER**模型可以捕获输入中信息最多的元素。特别是，第**3.3.5**节中介绍的**Transformer**体系结构完全依赖于注意力机制来绘制输入和输出之间的全局依存关系。在NER任务中还有许多其他方法可以应用注意力机制。Rei等。[105]应用了一种注意机制来动态地决定要从端到端**NER**模型中的字符或单词级别组件使用多少信息。Zukov-Gregoric等。[197]探索了NER中的自我注意机制，其中权重取决于单个序列（而不是取决于两个序列之间的关系）。徐等。[198]提出了一种基于注意力的神经**NER**体系结构，以利用文档级的全局信息。特别地，从具有神经注意力的预训练双向语言模型表示的文档获得文档级信息。张等。[199]在推文中对NER使用了自适应共同注意网络。该自适应共同注意网络是使用共同注意过程的多模式模型。共同注意包括视觉注意和文本注意，以捕获不同形式之间的语义交互。

5挑战和未来方向

在第3.5节中讨论，选择标记解码器与输入表示和上下文编码器的选择不同。从**google word2vec**到最新的**bert**模型，基于**dl**的**ner**显著受益于在建模语言中预先训练的嵌入技术。不需要复杂的功能工程，我们现在有机会重新审视**NER**任务的挑战和潜在的未来方向。

5.1 Challenges

数据注释：有监督的**NER**系统，包括基于DL-based的NER，在培训中需要大的注释数据。然而，**数据注释仍然是耗时和昂贵的。对于许多缺乏资源的语言和特定领域来说，这是一个巨大的挑战，**因为需要领域专家来执行注释任务。由于语言的歧义，注释的质量和一致性都是主要的关注点。例如，同一个命名实体可以使用不同的类型进行注释。例如，“巴尔的摩”在“巴尔的摩击败洋基”一句中，被标记为MUC-7的位置和CONLL03的组织。“帝国大厦”和“帝国大厦”都被标记为conll03和ace数据集中的位置，导致实体边界混乱。由于数据注释中的不一致性，在一个数据集上训练的模型在另一个数据集上可能无法很好地工作，即使两个数据集的文档来自同一个域。为了使数据注释更加复杂，Katiyar和Cardie[122]报告说嵌套实体相当常见：genia语料库中17%的实体嵌入到另一个实体中；在ace语料库中，30%的句子包含嵌套实体。需要开发适用于嵌套实体和细粒度实体的通用注释方案，其中一个命名实体可以被分配多个类型。

非正式的文本和未看到的实体：在表3中列出，在具有正式文档（例如新闻报道）的数据集上报告了不错的结果。但是，在用户生成的文本（例如**WUT17**数据集）上，最佳**F**得分略高于**40%**。由于简短和嘈杂，非正式文本（例如，推文，评论，用户论坛）的**NER**比正式文本更具挑战性。许多用户生成的文本也是特定于域的。在许多应用场景中，**NER**系统必须处理用户生成的文本，例如电子商务和银行业务中的客户支持。评估**NER**系统的健壮性和有效性的另一个有趣的方面是它在新出现的讨论中识别不寻常，以前未见过的实体的能力。对于**WUT-17**数据集[200]的这一研究方向，存在一个共同的**任务2**。

5.2未来方向

随着建模语言的发展和现实应用的需求，我们期望NER能受到更多研究者的关注。另一方面，NER通常被认为是下游应用程序的预处理组件。这意味着特定的NER任务是由下游应用程序的需求定义的，例如命名实体的类型以及是否需要检测嵌套实体[201]。基于本次调查的研究，我们列出了NER研究的进一步探索方向。

细粒度NER和边界检测：尽管许多现有研究[19]，[97]，[109]都关注普通域中的粗粒度NER，但我们希望在特定领域中对细粒度NER进行更多研究，以支持各种实词应用[202]。细粒度的NER面临的挑战是NE类型的显着增加以及允许命名实体具有多种NE类型所带来的复杂性。这要求重新访问通用NER方法，其中例如通过使用BI-ES-S-（实体类型）和O作为解码标签来同时检测实体边界和类型。值得考虑将命名实体边界检测定义为专用的任务，以在不考虑NE类型的情况下检测NE边界。边界检测和NE类型分类的解耦实现了可以在不同域之间共享的边界检测的通用且强大的解决方案，以及用于NE类型分类的专用于域的专用方法。正确的实体边界还可以有效减轻链接到知识库的实体中的错误传播。已经有一些研究[95]，[203]将实体边界检测视为NER中的中间步骤（即子任务）。据我们所知，没有现有的工作单独关注实体边界检测以提供可靠的识别器。我们希望将来在这一研究方向上有所突破。

NER和实体联合链接：实体链接（EL）[204]也被称为命名实体规范化或消除歧义，旨在根据知识库（如通用领域的维基百科和生物医学领域的统一医疗语言系统）为文本中提及的实体分配唯一的身份。大多数现有工程在管道设置中单独解决了NER和EL两个独立的任务。我们认为，成功链接的实体（例如，通过知识库中的相关实体）所携带的语义显著丰富[66]，[205]。也就是说，链接实体有助于成功检测实体边界和正确分类实体类型。值得探索的方法是联合执行NER和EL，甚至是实体边界检测、实体类型分类和实体链接，从而使每个子任务受益于其他子任务

的部分输出，并减轻在管道设置中不可避免的错误传播。

基于DL的带有辅助资源的非正式文本NER：

如5.1节所述，基于**DL的NER**在非正式文本或用户生成的内容上的性能仍然很低。这要求在这一领域进行更多研究。特别是，我们注意到**NER**的性能显着受益于辅助资源的可用性[206]-[208]，例如用户语言中的位置名称词典。 尽管表3并没有提供强有力的证据证明地名词典的存在，因为附加特征会导致**NER**的性能在通用范围内提高，但我们认为通常需要辅助资源来更好地理解用户生成内容。 问题是如何为用户生成的内容或特定于域的文本的**NER**任务获取匹配的辅助资源，以及如何有效地将辅助资源整合到**DL**中基于**NER**。

基于DL的NER的可伸缩性：

使神经**NER**模型更具可扩展性仍然是一个挑战。此外，仍然需要一种解决方案，以在数据大小增长时优化参数的指数增长[209]。一些基于DL的NER模型已经以大量计算能力为代价获得了良好的性能。例如，ELMo表示使用3×1024维矢量表示每个单词，并在32个GPU上对模型进行了5周的训练[107]。Google BERT表示已在64个云TPU上进行了培训。但是，如果最终用户无法访问强大的计算资源，则无法微调这些模型。开发平衡模型复杂性和可伸缩性的方法将是一个有希望的方向。另一方面，模型压缩和修剪技术也是减少模型学习所需空间和计算时间的选择。

NER的深层转移学习：

许多针对实体的应用程序都使用现成的**NER**系统来识别命名实体。但是，由于语言特征的不同以及注释的不同，在一个数据集上训练的模型可能无法在其他文本上很好地工作。尽管有一些将深度迁移学习应用于NER的研究（请参阅第4.2节），但尚未充分探讨此问题。通过探索以下研究问题，应致力于未来的更多努力：如何有效地将知识从一个领域转移到另一个领域： （a）开发一个健壮的识别器，该识别器能够在不同领域中正常工作； （b）在**NER**任务中探索零散，单发和少发学习； （c）提供解决领域不匹配和跨域设置中标签零匹配的解决方案。

针对基于DL的NER的易于使用的工具包：

最近，Röder等人。[210]开发了**GERBIL**，它为研究人员，最终用户和开发人员提供了易于使用的界面，用于对实体注释工具进行基准测试，目的是确保可重复和可存档的实验。但是，它不涉及最新的基于DL的技术。Ott [211]提出了**FAIRSEQ**，这是一种快速，可扩展的工具包，用于序列建模，尤其是机器翻译和文本污名化。Dernoncourt等。[212]实现了一个名为**NeuroNER**的框架，该框架仅依赖于递归神经网络的变体。近年来，许多深度学习框架（例如TensorFlow，PyTorch和Keras）已被设计为通过高级编程接口提供用于设计，训练和验证深度神经网络的构建块。为了重新实现架构在表3中，开发人员可以使用现有的深度学习框架从头开始编写代码。我们设想，一个易于使用的**NER**工具包可以指导开发人员使用一些标准化的模块来完成它：数据处理，输入表示，上下文编码器，标签解码器和有效性度量。我们相信专家和非专家都可以从此类工具包中受益。

6 CONCLUSION

这项调查旨在回顾基于深度学习的NER解决方案的最新研究，以帮助新的研究人员对该领域进行全面的了解。在本次调查中，我们包括了NER研究的背景，传统方法的简要介绍，当前的最新技术，挑战和未来的研究方向。首先，我们合并了可用的**NER**资源，包括带标签的**NER**语料库和现成的**NER**系统，重点是一般领域的**NER**和英语的**NER**。我们以表格形式显示这些资源，并提供指向它们的链接以便于访问。其次，我们介绍了诸如**NER**任务的定义，评估指标，**NER**的传统方法以及深度学习的基本概念之类的预备知识。第三，我们回顾了基于各种深度学习模型的文献，并根据新的分类法对这些研究进行了映射。我们进一步调查了在新的问题设置和应用中最近应用的深度学习技术的最具代表性的方法。最后，我们总结了**NER**的应用，并向读者介绍**NER**以及未来方向的挑战。我们希望这项调查可以为设计基于**DL**的**NER**模型提供良好的参考。

参考：https://arxiv.org/abs/1812.09449

点赞
 评论
 分享
 收藏 2
 打赏
 举报
 关注
 一键三连

排雷——数据分析师最容易跳进去的5个大坑！就在身边离你很近！
 12-11
 对数据的解读能力、问正确问题的能力以及解答问题时的灵活性，是衡量一位数据分析师是否称职的关键。

A Survey on Deep Learning for Named Entity Recognition.pdf
 10-30
 NAMED Entity Recognition (NER) aims to recognizementions of rigid designators from text belonging to



第一篇关于深度学习在命名实体识别（NER）上应用的综述文章
 qq_33290002的博客
 505
 《A Survey on Recent Advances in Named Entity Recognition from Deep Learning models》这是第一篇关于深度学习在NER上应用的综述文章 这篇文...

命名实体识别（Named Entity Recognition，NER）
 幸福驿站
 8024
 命名实体识别（Named Entity Recognition，NER）命名实体识别（NER）其目的是识别语料中人名、地名、组织机构名等命名实体，识别文本中具有特...

论文笔记：A Survey on Deep Learning for Named Entity Recognition
 Wangpei9799的博客
 1157

前言： 因为实习接触到了命名实体识别的问题，因此打算调研一下相关方法。即有了这篇综述的论文《A Survey on Deep Learning for Named Entity Rec...		
A Survey on Deep Learning for Named Entity Recognition;论文笔记	qq_41657873的博客	442
前言：研究课题定为特定领域的命名实体识别，所以先阅读一篇综述，在此简单记录阅读过程。摘要 在文章中，首先介绍现有的NER资源，包括标记的N...		
神经网络结构在命名实体识别（NER）中的应用	qq_27009517的博客	1072
from: http://www.cnblogs.com/robert-dlut/p/6847401.html 近年来，基于神经网络的深度学习方法在自然语言处理领域已经取得了不少进展。作为NLP领...		
机器学习项目（二） 人工智能辅助信息抽取（十一）	qq_33357094的博客	242
信息抽取深度学习综述 Survey A Survey on Deep Learning for Name Entity Recognition Flair Embedding Contextual String Embeddings for Sequence La...		
A Survey on Deep Learning for Named Entity Recognition （2020）	XB_please的博客	287
本文介绍Deep Learning 在命名实体识别上的应用，主要介绍三部分：输入的分布式表示、上下文编码器（用于捕获标签解码器的上下文）和标签解码器...		
论文阅读：（2020版）A Survey on Deep Learning for Named Entity Recognition 命名实体识别中的深度学习方法	南有芙蕖	3082
A Survey on Deep Learning for Named Entity Recognition 命名实体识别中的深度学习方法 目录A Survey on Deep Learning for Named Entity Recognition...		
[ICLR2018]Deep Active Learning for Named Entity Recognition	Deep Learning and NLP Farm	2748
目前神经网络在各领域很多任务上都取得了瞩目的成就，但是最大的缺陷就是需要大量的标注语料，成本较高，于是本文采用active learning 挑选一部分...		
©2020 CSDN 皮肤主题: 游动-白 设计师: 上身试试 返回首页		

<div> <div>关于我们</div> <div>招贤纳士</div> <div>广告服务</div> <div>开发助手</div> <div>400-660-0108</div> <div>kefu@csdn.net</div> <div>在线客服</div> <div>工作时间 8:30-22:00</div> </div>
<div> <div>公安备案号11010502030143</div> <div>京ICP备19004658号</div> <div>京网文〔2020〕1039-165号</div> <div>经营性网站备案信息</div> <div>北京互联网违法和不良信息举报中心</div> <div>网络110报警服务</div> <div>中国互联网举报中心</div> <div>家长监护</div> <div>Chrome商店下载</div> <div>©1999-2022北京创新乐知网络技术有限公司</div> <div>版权与免责声明</div> <div>版权申诉</div> </div>