分 类 号 学号

学校代码 10487 密级



硕士学位论文

基于循环分段残差网络的对抗关系抽取研究与实现

学位申请人：赵恢强

学科专业：信息与通信工程

指导教师：王玉明 副教授

答辩日期：2019年5月25日

A Dissertation Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Engineering

Design and Implementation of Adversarial Relation Extraction Based on Recurrent Piecewise Residual Networks

Candidate： Huiqiang Zhao

Major： Information and Communication Engineering

Supervisor：Prof. Wang Yuming

**Huazhong University of Science and Technology**

**Wuhan, Hubei 430074, P. R. China**

**May, 2019**

**独创性声明**

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期： 年 月 日

**学位论文版权使用授权书**

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保 密□，在\_\_\_\_\_年解密后适用本授权书。

不保密□。

（请在以上方框内打“√”）

学位论文作者签名： 指导教师签名：

日期： 年 月 日 日期： 年 月 日

摘 要

作为信息抽取的核心任务，有效的关系抽取是信息爆炸时代的一大挑战。本文在进行关系抽取这一基础性研究时发现，现有的模型都无法有效的抽取长而复杂或无直接关系触发词的句子中的实体对关系，而这种句子在自然语言文本中是普遍存在的，这使得关系抽取的效果遇到了瓶颈。

本文深入调研并分析了国内外在关系抽取方面的研究。在融合更多更深次信息的思想启发下，为了丰富模型的表示，提出**实体类型嵌入(Entity Type Embedding, ETE)**，ETE和词嵌入、位置嵌入一起作为模型的联合嵌入层；为了更精细地编码模型的嵌入表示，精心设计了**循环分段残差网络(Recurrent Piecewise Residual Networks, RPRN)**结构，RPRN能有效获取句子上下文中的潜在表示。本文在进行关系抽取建模时还引入了对抗训练，从而进一步增强模型的泛化能力。本文提出的模型**(ETE-RPRN)**有效解决了现存模型所遇到的困境，并在纽约时报公开数据集上进行了验证，实验结果显示本文提出的胜过现存最先进的模型。

关系抽取不仅具有很高的理论研究价值，而且具有非常广泛的实际应用。本文将提出的模型应用到了科研文档中，并应用知识图谱可视化技术构建了一个语义检索系统。通过机器学习相关技术自动抽取自然语言文本语料中蕴含的关系事实，人们可以更加有效地获取文本中涉及的各主客体之间的语义关系，避免了以人工阅读的方式从不断生成的海量文本中提取关系事实等结构信息，这对文本语料的信息化建设具有重大的意义。同时抽取的关系事实是构建语义检索、问答系统等应用的重要基础。

**关键词：**关系抽取；实体类型嵌入；循环神经网络；残差网络

Abstract

As a core mission in information extraction, effective relation extraction is a great challenge in the information explosion era. In this paper, we found that existing models usually fail in extracting correct relation of entity pair effectively when the sentence is long and complex or there is not the firsthand trigger word of corresponding relation, whereas such sentences are ubiquitous in texts formed of natural language, which makes the effects of relation extraction hit a bottleneck.

In this paper, we penetrate into investigating and anlyzing the researches on relation extraction at home and abroad. Inspired by the idea of fusing more and deeper information, we propose **Entity Type Embedding(ETE)** to enrich the representation of the model, ETE, word embedding and position embedding serve as the joint embedding of our model; For encoding the representation of the embedding subtly, we devise an architecture of **Recurrent Piecewise Residual Networks(RPRN)** meticulously, which may acquire the latent representation underlying the context of a sentence. We introduce adversrial training when modeling relation extraction as well, which further enhance generalization capacity of our model. Our proposal(**ETE-RPRN**) effectively solve the dilemma of existing models, and is validated by experiments on public dataset of New York Times. Experiment results show that our proposal outperforms the existing state-of-the-art models.

Relation extraction is not only of great values for theory study, but also has extremely extensive application. We apply proposed model to documents of scientific research, and employ visuallization technology of knowledge graph to build a system of semantic search. Extracting the relation facts implied in the texts formed of natural language automatically by the technology of machine learning, pepole are able to acquire semantic relations between the subjects and objects involved in the text more effectively, and get rid of extracting structural information such as relation facts from the massive texts constantly generated by means of artificial reading, which is of great significance to information construction of text corpus. Simutaneously, the extracted relation facts are key foundation of building applications such as semantic search, question answering system and etc.

**Keywords**：Relation Extraction, Entity Type Embedding, Recurrent Neural Networks, Residual Networks

# 目 录

[摘 要 I](#_Toc2839935)

[Abstract II](#_Toc2839936)

[目 录 III](#_Toc2839937)

[略缩语 V](#_Toc2839938)

[1 绪论 1](#_Toc2839939)

[1.1 课题背景及来源 1](#_Toc2839940)

[1.2 课题现状及分析 2](#_Toc2839941)

[1.3 研究内容及意义 5](#_Toc2839942)

[1.4 本文的组织结构 7](#_Toc2839943)

[2 相关理论及技术概述 8](#_Toc2839944)

[2.1 远程监督 8](#_Toc2839945)

[2.2 循环神经网络 10](#_Toc2839946)

[2.3 残差网络 13](#_Toc2839947)

[2.4 注意力机制 17](#_Toc2839948)

[2.5 对抗训练 17](#_Toc2839949)

[2.6 StanfordNLP 18](#_Toc2839950)

[2.7 Tensorflow 19](#_Toc2839951)

[2.8 图形数据库 19](#_Toc2839952)

[2.9 数据可视化 20](#_Toc2839953)

[2.10 本章小结 20](#_Toc2839954)

[3 关系抽取分析与设计 21](#_Toc2839955)

[3.1 困境分析 21](#_Toc2839956)

[3.2 模型架构设计 22](#_Toc2839957)

[3.3 联合嵌入 23](#_Toc2839958)

[3.4 基于RPRN的编码器 25](#_Toc2839959)

[3.5 选择性注意力机制 28](#_Toc2839960)

[3.6 本章小结 28](#_Toc2839961)

[4 关系抽取算法实现 30](#_Toc2839962)

[4.1 开发环境搭建 30](#_Toc2839963)

[4.2 数据清洗模块实现 31](#_Toc2839964)

[4.3 协作网络建模实现 38](#_Toc2839965)

[4.4 协作推荐模型实现 42](#_Toc2839966)

[4.5 本章小结 52](#_Toc2839967)

[5 模型评估及案例分析 53](#_Toc2839968)

[5.1 系统测试及性能分析 53](#_Toc2839969)

[5.2 协作关系网络探究 58](#_Toc2839970)

[5.3 本章小结 64](#_Toc2839971)

[6 总结与展望 65](#_Toc2839972)

[6.1 论文总结 65](#_Toc2839973)

[6.2 论文展望 66](#_Toc2839974)

[致 谢 67](#_Toc2839975)

[参考文献 69](#_Toc2839976)

# 略缩语

|  |  |
| --- | --- |
| **ETE** – Entity Type Embedding | 实体类型嵌入 |
| **RPRN** – Recurrent Piecewise Residual Networks | 循环分段残差网络 |
| **NYT** – New York Times | 纽约时报 |
| **CNN** – Convolutional Neural Networks | 卷积神经网络 |
| **RNN** – Recurrent Neural Networks | 循环神经网络 |
| **LSTM –** Long Short Term Memory | 长短时记忆 |
| **GRU** – Gated Recurrent Unit | 门控循环单元 |
| **NLP** – Natural Language Processing | 自然语言处理 |
| **POS** – Part Of Speech tagging | 词性标注 |
| **SRL** – Sematic Role Labeling | 语义角色标注 |
| **NER –** Named Entity Recognition | 命名实体识别 |
| **DARPA -** Defense Advanced Research Projects Agency | 国防高级研究计划局 |
| **NIST -** National Institute of Standards and Technology | 国家标准与技术研究所 |
| **MUC –** Message Understanding Conference | 消息理解会议 |
| **ACE –** Automatic Content Extracting | 自动内容抽取 |
| **OCR –** Optical Character Recognition | 光学字符识别 |
| **CRF -** Conditional Random Field | 条件随机场 |
| **MIML –** Multiple Instance Multiple Label | 多实例多标签 |
| **KB –** Knowledge Base | 知识库 |
| **BP** –Backpropagation | 反向传播 |
| **BPTT –**Backpropagation Through Time | 通过时间反向传播 |
| **ResNet –** Residual Networks | 残差网络 |
| **VGGNet** –Visual Geometry Group Network | 视觉几何群网络 |
| **ILSVRC** – Imagenet Large Scale Visual Recognition Challenge | 大规模视觉识别挑战赛 |
| **COCO –** Common Objects in Context | 上下文公共对象 |
| **ReLU –** Rectified Linear Units | 整流线性单元(激活函数) |
| **CPUs –** Central Processing Units | 中央处理器 |
| **GPUs –** Graphics Processing Units | 图形处理器 |
| **TPUs –** Tensor Processing Units | 张量处理器 |
| **ASICs** **–** Application Specific Integrated Circuits | 专用集成电路 |
| **DNNs** **–** Deep Neural Networks | 深度神经网络 |
| **AI** **–** Artificial Intelligence | 人工智能 |
| **Json –** JavaScript Object Notiation | JavaScript对象符号(一种轻量级的数据交换格式) |

# 绪论

随着信息科技的不断发展，人们的工作和生活变得越来越便捷。然而面对信息爆炸的数据洪流，要想从中获得有用信息则变得举步维艰。世界上的许多知识是用自然语言文字记录的，但如何以这种形式有效地加以利用是一项重大挑战[1]。本课题研究的关系抽取正是信息抽取[1][2]的一项核心而又艰巨的任务，同时又是构建语义搜索[3]、问答系统[4]等知识图谱[5]相关应用的重要基础。

近年来，关系抽取的研究越来越受到国内外研究学者的关注，在这方面的研究也取得了显著的进展。但是由于关系抽取这个任务本身的复杂性，现有的模型依然存在许多的局限，关系抽取的效果差强人意。

本课题就是要研究并解决目前的关系抽取模型所面临的困境，如无法有效的抽取长而复杂或无直接关系触发词的句子中涉及的实体对关系。在本章中，首先，简要介绍了本课题的背景及来源；其次，概述了国内外在关系抽取方面的研究，并剖析了关系抽取当前所面临的困境及本课题要解决的重难点问题；然后，简明扼要的阐述了本课题研究的主要内容是什么，以及研究的意义在哪里；最后，说明了下本文的行文组织结构。

## 课题背景及来源

由于网络应用爆炸式增长，互联网中每时每刻都有海量的数据在不断地生成，这标志着大数据、云计算的时代已经来临。这一方面给人们的生活方式带来了极大的变革，另一方面，却也使得人们从这些海量数据中获取有效信息变得愈发困难。因此我们亟需一种能够从海量的自然语言文本中抽取有用信息的技术，从而更加充分发挥这些文本语料的价值，信息抽取正是应上述之需而诞生的。关系抽取是信息抽取中极为重要而又复杂的任务，然而当前已有的模型在关系抽取上的效果却不尽人意。因此，关系抽取的研究依然任重而道远。

本课题来源于教育部社科研究重大委托课题“科研管理信息化建设”，并受到教育部哲学社科繁荣计划建设项目（2017-2018）、华中科技大学双一流建设项目（2016-2018）、华中科技大学自主创新研究基金项目（No.2015AE027）的资助。本课题针对当前已有模型遇到的困境，以丰富模型的嵌入层表示及编码模型的潜在表示为切入点，提出新的嵌入模型，并设计更好的神经网络结构，来建模关系抽取。同时将本文提出的模型应用到社会科学研究文档中，并构建一个语义检索系统。构建的系统用于为高校社科管理人员提供服务，进而推进科研管理的信息化建设。

## 课题现状及分析

### 国内外研究概况

由美国国防高级研究计划局(DARPA)资助的最后一届消息理解会议(**MUC**)首次引入关系抽取任务[6]。美国国家标准技术研究院(NIST)组织的自动内容抽取(ACE)评测会议中提出的一项重要任务就是实体关系识别[7]。ACE评测会议中关系语料的来源包括新闻专线、新闻广播、通过OCR识别的报纸、新闻组、电视访谈、网络日志等等。与MUC相比，ACE的数据规模和语种数量都有了大幅度提升。MUC、ACE评测会议的实体关系抽取涉及的关系类型局限于命名实体( 包括人名、地名、组织、机构名等)之间的少数几种类型的实体关系，如雇佣关系、地理位置关系、人—社会组织关系等。SemEval[8]是继MUC、ACE后信息抽取领域又一重要评测会议，该会议吸引了大量的院校和研究机构参与测评。SemEval-2007[9]定义了普通名词或名词性短语之间的语义关系，但提供的语料库规模较小，SemEval-2010[10]任务8在SemEval-2007基础上进行了丰富和完善，引发了普通名词或名词性短语之间语义关系抽取的研究新高潮。

关系抽取的研究越来越受到国内外研究学者的关注，研究的方法也是层出不穷，概括起来主要包含基于传统有监督的方法，基于远程监督的方法，以及基于深度学习的学习方法三大类。

1. 基于传统有监督的方法

在关系抽取中，目前有监督的方法能抽取更有效的特征，其准确率和召回率都更高，因此有监督的方法受到了国内外研究学者更多的关注。

有监督的关系抽取方法需要标注好的训练语料，从标注语料中自动学习关系对应的抽取模式。典型的方法有：Miller, Scott等人[11]的基于句法解析增强的方法, Kambhatla, Nanda等人[12]的基于逻辑回归的方法,基于核的方法[13],包括Zhao and Grishman等人[14]的解析树核、Bunescu and Mooney等人[15]的子序列核等, Culotta, Aron等人[16]的基于条件随机场(CRF)的方法。然而在关系抽取中,阻碍这些方法的主要问题在于人工标注训练语料耗时耗力。

1. 基于远程监督的方法

针对传统有监督方法存在的上述局限性，Mintz et al[17],Min et al [18],Angeli et al [19]等人将远程监督的思想应用到了关系抽取的建模过程中。他们将纽约时报(NYT)新闻语料与大规模知识库FreeBase[20](包含7300多个关系和超过9亿个实体)进行实体对齐。远程监督假设为：一个包含两个实体的句子蕴含了该实体对在知识库当中的语义关系，然后将这个句子作为该实体对在知识库中所蕴含关系的训练正例。作者使用远程监督方法标注训练语料，然后抽取语料特征并训练关系抽取学习器，有效解决了关系抽取的标注训练语料规模的问题。后续众多研究学者从不同角度对远程监督技术进行了改进。例如，Takamatsu等人[21]提出了实体对齐技术的改进方案，缓解了噪声数据带来的性能下降问题；Yao等人[22]提出了基于无向图模型的实体关系抽取方法；Riedel等人[23]则增强了远程监督的假设，与 Mintz的相比错误率减少了31%，提高了关系抽取的整体效果。

上述基于远程监督的假设均是一个实体对只对应一种关系。然而实际中许多实体对具有多重关系。如“Steve Jobs founded Apple”和“Steve Jobs is the CEO of Apple”就表达了实体对之间的多重关系。为解决在很多实际场景中基于上述假设的不合理性，Hoffmann等人[24]提出了多实例多标签(MIML)方法来建模关系抽取，考虑了一个实体对存在多重关系的场景。类似地，Surdeanu等人[25]也提出了多实例多标签+贝叶斯网络(MIML+BN)来抽取自然语言文本语料中实体之间的语义关系。关于生成负样本的这一严重缺陷，Bonan等人[26]提出了在实体对级别上仅仅从正例与未标记样例中学习的算法，Angeli等人[27]提出了一种利用少量精心选取的实例对远程监督关系抽取器进行局部监督的方法。

1. 基于深度学习的方法

应用传统机器学习的方法尽管已经取得了很不错的效果，但这些方法过于依赖词性标注(POS)[28]、语义角色标注(SRL)[29]等自然语言处理(NLP)[30]任务。这些任务在经过NLP工具处理后存在大量的错误，而这些错误在关系抽取中被不断传播放大，进而影响关系抽取的整体性能。近年来，由于深度学习的迅速崛起，尤其是其在计算机视觉领域的革命性突破，越来越多的研究学者倾向于用深度学习的方法来建模关系抽取。值得关注的一点是，深度学习允许由多个处理层组成的计算模型学习具有多个抽象级别的数据表示[31]。

Socher等人[32]提出采用递归神经网络来建模实体关系抽取，该方法考虑了句子的句法结构信息，但没有考虑两个实体在句子中的位置及语义信息。Zeng et al[33], Nguyen et al[34],Xu et al[35]等人将卷积神经网络(CNN)运用于实体关系抽取任务中，该方法同时考虑实例的词向量和词的位置向量，在关系抽取中较好地考虑到了句子中的实体信息。之后，Santos等人[36]提出了一种新的卷积神经网络应用于实体关系抽取，并使用了新的损失函数，更加有效地区分了不同的关系类别。

深度学习方法在人工标注的数据集上取得了巨大成功，但是与基于特征的关系抽取方法类似，基于深度学习方法的关系抽取模型也面临着人工标注语料耗时耗力的问题。为此，Zeng等人[37]提出通过CNN与远程监督结合来进行关系抽取，该方法在一定程度上解决了神经网络关系抽取模型在远程监督数据上的应用，在NYT数据集上取得了远远高于基于特征的关系抽取模型的预测效果。Miwa等人[38]提出端到端的神经网络，该网络用双向LSTM[39]来建模关系抽取。Peng et al[40]、Song et al[41]等人探究了一种基于图LSTM的通用关系抽取框架，该框架很容易扩展到跨句子的n元关系抽取。为了有效缓解噪声数据带来的性能下降问题，Lin等人[42]将选择性注意力机制应用到了关系抽取中，该方法可以根据指定关系为袋子中的每个实例分配权重，通过训练学习从而使有效实例的权重不断提高，而带噪声的实例权重不断下降。在其他相同情况下，与不带注意力机制的模型相比，该方法的整体效果提升较大。类似地，Wang et al[43]、Du et al[44]等人提出一种新的多级别的注意力机制用于关系抽取。Wu et al[45]、Qin et al[46]等人将对抗模型引入到关系抽取中，从而进一步增强了模型的泛化能力。此外，也有改善子句模式的自训练[47]、全局关系嵌入[48]等各种其他的关系抽取方法。

### 重点难点问题

本课题对关系抽取这一基础性任务进行了研究，为提高关系抽取的效率，提出和设计了新的关系抽取模型，并将提出的模型应用到科研文档中，以构建一个语义检索系统。本课题在进行关系抽取的研究与实现过程中，重点问题主要为以下几点：

1. 大规模语料的标注

语料标注在关系抽取建模过程中是一个非常重要的问题，尤其是对于大规模语料，人工标注耗时耗力，这就亟需一种自动标注语料的技术。本课题是在关系抽取研究中广泛采用的NYT数据集上进行实验的，该数据集规模庞大，本文采用了远程监督[17]技术来解决语料标注的问题。

1. 噪声实例的影响

应用远程监督技术来自动标注自然语言文本语料在解决了大规模语料的标注问题的同时，也给关系抽取模型带来了很多噪声实例，从而影响关系抽取的效果。为了缓解噪声实例带来的性能下降问题，本文采用了选择性注意力机制[42]。

1. 关系图谱的可视化

本课题重点是针对已有关系抽取模型存在的局限性，提出新的模型以提升关系抽取的效果。此外还将提出的模型应用到了科研文档中，为科研管理者提供一个语义检索系统。关系抽取模型的结果是三元组的形式，在用户检索后，这种关系事实以什么样的形式呈现就显得比较关键。本文采用的是D3js技术来对用户检索的关系图谱结果进行可视化。

本课题深入探索并分析了国内外研究学者在关系抽取方面的研究工作，发现目前已有的研究方法存在很多的局限性，这些正是本文要攻克的难点所在，主要分为以下两点：

1. 无直接关系触发词的句子的关系抽取

在自然语言文本语料中，普遍存在类似图1-1第一个句子的情况，即句子中不存在要抽取关系的直接触发词，而已有的关系抽取模型在这种情景下往往是无法抽取出有效关系的。

1. 长而复杂的句子的关系抽取

对于图1-1第二个句子这样长而复杂的句子，由于两个实体之间的距离隔得比较远，现有的方法对实体对上下文语义关系的学习比较困难，也常常不能提取出这类句子中实体之间的正确关系。



图1-1 关系抽取模型例子

## 研究内容及意义

### 研究内容

本课题研究的关系抽取是信息抽取的一项核心任务和重要环节。信息抽取旨在从大规模非结构化或半结构化的自然语言文本中抽取结构化信息，其主要任务是识别句子中的实体、抽取实体之间的语义关系以及其中涉及的事件[49]。关系抽取主要是从自然语言文本语料中抽取实体对的语义关系，然后表示成三元组的形式以构成关系事实，如：<Jobs, founder, Apple>表示乔布斯是苹果公司的创始人这样一个关系事实。这些关系事实是构建其他上层应用如语义检索、问答系统等的重要基础[50]。本课题研究内容的应用范围并不局限于科研文本，对于包括新闻文本、百科文本等在内的文本数据也同样是适用的。

对于关系抽取的基础性研究，为了方便与其他研究学者论文中的方法进行比较，本文采用的是NYT公开数据集。本课题的研究内容主要有以下几点：

1. 解决无直接关系触发词的关系抽取的困境

现有的关系抽取方法通常采用词嵌入[51][52]及位置嵌入[53]作为模型的分布式表示层，而这种表示存在1.2.2小节所述的第一个难点，即无法有效的抽取无直接关系触发词的句子中的关系。为了解决这一困境，本文提出实体类型嵌入(ETE)。ETE和词嵌入、位置嵌入一起作为模型的联合嵌入层。

1. 处理长而复杂的句子的关系抽取的问题

在应对长而复杂的句子时，现有的模型也常常不能提取出正确的关系，如图1的第2个例子。我们认为，以上抽取失败是由于缺失深层语境下的隐藏信息导致的。为解决这一问题，我们在提出的ETE的基础上，精心设计了循环分段残差网络(RPRN)结构。RPRN网络结构深而复杂，能够有效编码句子上下文中的潜在信息。

1. 构建语义检索应用系统

本课题重点是进行关系抽取的基础性研究，以解决当前已有方法所面临的困境。此外，也将提出的模型应用到了科研文档中，构建一个语义检索系统。检索的结果以关系图谱的形式呈现，这主要用到D3js可视化技术。

### 研究意义

本课题深入探索并分析了国内外研究学者在关系抽取方面的研究，提出ETE，并精心设计了RPRN网络结构。本文提出的模型有效解决了当前关系抽取研究长期以来面临的困境，如无法很好地抽取无直接关系触发词或长而复杂的句子中的实体对关系。本课题开展的这一基础性研究，极大地提高了关系抽取的整体性能，这对关系抽取的研究与发展具有重要的意义。

通过机器学习相关技术自动抽取自然语言文本语料中蕴含的关系事实，人们可以更加有效地获取文本中涉及的各主客体之间的语义关系，有效地避免了以人工阅读的方式从不断生成的海量文本中提取关系事实等结构信息，这对自然语言文本语料的信息化建设具有重大的意义。同时抽取的关系事实是构建语义检索、问答系统等上层应用的重要基础。

本课题提出的关系抽取模型，不仅对本文要构建的语义检索系统提供了基础，而且在一定程度上为今后关系抽取方面的研究指明了方向，本课题的研究具有广泛的实际应用和较高的理论研究价值。

## 本文的组织结构

本文共计包含六个章节，各章节的内容简述如下：

第一章：绪论。在本章节中，首先，简要介绍了本课题的背景及来源；其次，概述了国内外研究学者在关系抽取方面的研究，并剖析了关系抽取当前所面临的困境及本课题要解决的重难点问题；然后，简明扼要的阐述了本课题研究的主要内容以及研究的意义，最后描述了下本文的组织结构。

第二章：相关理论及技术概述。本章主要介绍了在进行关系抽取研究中涉及的关键理论，以及在建模过程中用到的核心技术。包括远程监督、循环神经网络、残差网络、选择性注意力机制等重要理论，还有深度学习框架tensorflow、知识图谱可视化框架D3js等相关技术。

第三章：关系抽取分析与设计。本章深入分析了关系抽取研究中遇到的重点及难点问题，并针对现有方法遇到的困境，提出ETE并设计RPRN网络结构。然后概述模型的总体架构，最后详述模型的整个设计过程。

第四章：关系抽取实现。本章节首先针对前一章节分析和设计的模型给出了具体的实现细节，包括关系抽取模型的算法实现过程及语义检索系统的关键实现。

第五章：模型评估及案例分析。本章首先简单介绍本文所设计的模型的超参数的设置及模型评估的相关指标，然后，以表格和曲线图的形式对我们提出的模型和现有的模型在各项评估指标上进行比较，最后，对模型进行关系抽取的不同结果做一个具体的案例分析，以佐证本文所提出的模型的有效性。

第六章：总结与展望。本章节简要总结了本课题开展的关系抽取的研究工作，并对其未来的发展和研究做了进一步的探讨和展望。

# 相关理论及技术概述

本课题开展的关系抽取这一基础性研究，主要基于机器学习相关理论与算法来建模。本文是在NYT大规模公开数据集上进行的实验，首先为了解决人工标注自然语言文本语料耗时耗力的问题，本文应用了远程监督。利用循环神经网络(RNN)[54]进行序列学习的优势，以及残差网络[55]深度编码模型表示的能力，本文设计了一种新的神经网络结构RPRN作为关系抽取编码器。为进一步增强模型在测试数据集上的泛化能力，在关系抽取建模过程中还引入了对抗训练[45]。

关系抽取的建模依赖于命名实体识别(NER)[56]任务，本文采用的是StanfordNLP[57]工具来完成这一任务。在关系抽取的设计过程中，我们应用了深度学习的相关方法，具体采用的框架是google推出的Tensorflow[58]。此外，在应用本文提出的模型构建语义检索系统时，对抽取的三元组关系事实采用Neo4j[59]图形数据库来存储，语义检索结果利用D3js[60]框架来进行关系图谱的数据可视化。

## 远程监督

关系抽取就是要抽取句子中包含的实体之间的语义关系，然后表示成如<Jobs, founder, Apple>这样的三元组形式。在传统有监督的关系抽取中，首先要对训练语料进行人工标注，因为关系抽取器模型本身是无法给要训练的实体对关系打标签的。然而，人工标注语料存在耗时耗力的问题，这极大地限制了大规模语料在关系抽取方面的应用。因此，Mintz等人[17]提出应用远程监督来自动标注语料。

### 远程知识库

应用远程监督来自动标注文本语料需要一个远程的知识库(KB)[61]。知识库是一种用于存储被计算机系统使用的复杂结构化或非结构化信息的技术[62]，包括专家系统所使用的规则集，文本中蕴含的关系事实、涉及的事件等等。本课题研究的关系抽取采用的远程监督知识库为FreeBase[20]，如下表2-1所示：

表2-1 FreeBase知识库-关系事实

|  |  |  |
| --- | --- | --- |
| **relation** | **subject\_entity** | **object\_entity** |
| /people/person/nationality | Sanath Jayasuriya | Sri Lanka |
| /location/location/contains | Russia | Arkhangelsk |
| **…** | **…** | **…** |

### 命名实体识别

命名实体识别(NER)是自然语言处理中一项非常重要的任务[63]，也是本课题要开展的关系抽取研究的重要基础。NER的主要目的是识别自然语言文本中的人名、地名、机构名、日期时间、专有名词等具有特定意义的实体。在关系抽取中要应用远程监督，首先就要识别出文本语料中的实体对，然后再结合知识库来自动标注。NER识别句子中的实体对如下所示：

**Steve Jobs** was the co-founder and CEO of **Apple** and formerly **Pixar**.

在上述实例中，使用NER可以识别出三个实体，这三个实体在该句子的上下文中可以组成两对实体，接下来就是自动标注实体对在对应实例中的语义关系。

### 实体链接

识别出了自然语言文本语料中涉及的实体后，就要采用远程监督来自动标注训练语料。在自动标注语料时，需要应用实体链接[64] ，如图2-1所示，即将训练语料中所提及的实体对链接到大规模远程KB中。

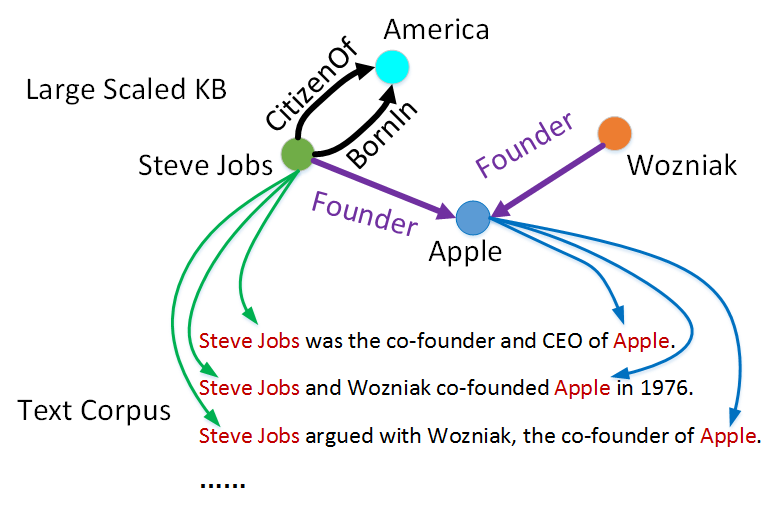


图2-1 远程监督实例

远程监督假设为：一个句子包含的实体对蕴含了该实体对在知识库中对应的关系，然后在模型训练时，将该句子作为其实体对对应关系的训练正例。最初的远程监督假设一个实体对只对应一种关系，但在自然语言文本语料中，大量存在一个实体对拥有多种关系的场景。因此，后续有很多的研究学者从不同角度对远程监督进行了改进，以适应其在大规模语料上的应用。

## 循环神经网络

循环神经网络(RNN)是一种处理序列数据的网络结构[65]，例如处理可变长度的自然语言文本序列。相较于全连接的前馈神经网络只有相邻层之间的神经元有连接而言，RNN的网络结构(如图2-2)的每一个隐藏层中的神经元也有连接，这使得RNN在几个时间步内可以共享相同的参数，有效避免了在不同位置上重复学习相同的特征规则。



图2-2 多层循环神经网络

对于任何一个图灵可计算的函数而言，都可以通过图2-2这样一个有限维的RNN结构来模拟计算[65]，从这一角度来讲，RNN可以说是万能的。在RNN的设计过程中，主要包含以下几个重要的设计模式：

1. RNN模型每一个时间步都有输出，并且隐藏层神经元之间有如图3所示的循环连接；
2. RNN模型每个时间步都会产生一个输出，只有从一个时间步的输出到下一个时间步的隐藏层神经元有循环连接；
3. RNN模型隐藏层神经元之间存在循环连接，其读取整个序列后产生单个输出。

### 通过时间反向传播

反向传播(BP)算法自20世纪70年代被引入以来，作为一种神经网络模型的优化算法，一直是深度学习的主力[66]。而RNN是一种具有长期记忆能力的序列学习网络模型，普通的BP算法并不适用于这种模型的优化。在RNN模型中，采用的是通过时间反向传播(BPTT)[67]的神经网络优化算法，这是一种更有效的关于在RNN中传递参数的方法。在RNN的前向传播中，以特定的初始状态为起点，对于每一个从t=1到t=T的时间步，应用如下方程式来进行更新：

(2-1)

(2-2)

(2-3)

(2-4)

其中对于每一个时间步t，表示输入，表示隐藏层，表示输出，为模型输出向量，W、U、V表示权重矩阵，b、c为偏置向量。模型的损失函数为：

(2-5)

为在给定到t时刻的序列及t时刻的目标下的负对数似然，模型的总损失是所有时间步损失的累积之和。通过BPTT来计算RNN的梯度如以下公式所示：

(2-6)

(2-7)

(2-8)

(2-9)

(2-10)

(2-11)

为对角矩阵，是关于t时刻与隐藏层神经元i关联的Jacobian，指的是包含这个元素。

### RNN变体

在普通RNN模型的训练过程中，存在一个众所周知的问题，及梯度弥散和梯度爆炸[68]。梯度弥散和梯度爆炸本质上可以归结为梯度不稳定问题，其主要原因是神经网络在优化时进行反向传播的过程中，同一矩阵经过了多次相乘，梯度倾向于消失或超过某个阈值，导致后续的神经元无法更新参数，神经网络的优化无法正常进行。

为了解决RNN在训练过程中遇到的上述问题，一些学者提出了RNN相关的一些变体结构。如非常经典的LSTM，其结构原理如图2-3所示。



图2-3 LSTM结构原理图

LSTM结构设计的巧妙之处在于：除了RNN拥有的外部循环外，内部结构还有自循环，这种结构具有学习长期依赖的能力。其主要由3个门来控制，分别为输入门，遗忘门，输出门。结构内部各状态的输出如下公式所示：

(2-12)

(2-13)

(2-14)

(2-15)

(2-16)

(2-17)

其中、、为输入门的权重矩阵，、、为遗忘门的权重矩阵，、、为输出门的权重矩阵，、、分别为3个门的偏置向量，为t时刻隐藏状态的输出向量。

LSTM的结构看起来比较复杂，那么该结构的所有构成都是必须的吗？有没有一种机制可以动态控制模型的遗忘功能呢？相较于LSTM，GRU[69]是一种更加简化的门控循环神经网络结构， 其结构原理如图2-4所示。



图2-4 GRU结构原理图

从图2-4与图2-3的对比中可以看出，GRU在原LSTM结构的基础上将门控单元简化成了2个，即更新门和重置门，其公式分别表示为：

(2-18)

(2-19)

其中，、表示更新门的权重矩阵，、表示重置门的权重矩阵，和分别表示两个门的偏置向量，隐藏层状态向量为，由以下公式表示：

(2-20)

(2-21)

GRU结构已经不再显示包含记忆单元，但它至少和LSTM一样有效，并且收敛速度更快。与LSTM的最大不同在于，GRU中的单个门控单元可以同时控制遗忘功能和状态更新。

LSTM、GRU等门控机制的RNN模型都能够有效解决梯度弥散的问题，但处理不了梯度爆炸的问题。梯度截断(Gradient clipping)是一种非常流行的处理梯度爆炸问题的方法[70]，该方法通过截断梯度来使得传递的梯度总是小于给定的阈值。

## 残差网络

深度学习冠之以“深度”一词，以区别于早期浅层的神经网络模型。浅层网络模型最大的缺陷在于其很难学出具有不同抽象级别的数据表示，而深层模型则通过堆叠多层次网络结构有效地解决了这一难题。然而，越深的网络模型越难以训练和优化，甚至有时候训练出的效果还不如浅层模型。

为了解决深层模型训练及优化难的问题，Kaiming He等人[55]提出了残差网络(ResNet)模型架构。作者使用了深度达152层的ResNet在ImageNet数据集[71]进行评估，虽然该网络结构的深度是VGGNet-19模型的8倍，但其复杂度却比VGGNet-19低，并且在ImageNet测试集上的效果达到了3.57%的错误率，从而赢得了ILSVRC和COCO 2015比赛[[1]](#footnote-1)中多项任务的第一名。

ResNet的基本单元是卷积池化模块，然后以此为基础进行残差学习，进行残差学习的网络结构由恒等映射块来构造，块中的单元通过捷径来连接。

### 深度卷积网络

深度卷积网络在图像分类方面取得了一系列的突破[72]，其主要由一系列的卷积和池化两种操作构成。卷积的特性是输入表示的潜在模式比整个输入小且出现在不同区域，池化的则是通过再次抽样来获取主要特征而不失去原始输入的表征。

卷积操作的原理示意图如图2-5所示，每一层的卷积操作都会对应一组过滤器，然后用该组过滤器中的每一个分别与上层输出进行点乘操作。通过不同层次的卷积操作，输入表示被编码成了具有不同抽象级别的局部潜在特征。

卷积操作之后会进行池化操作，即将一层神经元簇的输出组合成下一层的某个神经元。池化操作的类型有多种，如最大池化、平均池化等。顾名思义，最大池化就是使用前一层每一个簇的最大值作为该层当前神经元的值，其操作原理如图2-6所示；以此类推，平均池化即使用前一层每一个簇的平均值作为该层当前神经元的值。当然，卷积和池化操作后还会经过激活函数处理，如sigmoid、ReLU[73]等激活函数。



图2-5 卷积操作原理



图2-6 最大池化操作原理

### 残差学习

假定将作为一个由几个堆叠的层(不一定是整个网络)匹配的潜在映射，其中是这些堆叠层的第一个的输入。假设多个非线性层可以渐进近似复杂的函数，那么等价于假设其可以渐进近似残差函数，即：

(2-22)

在以上公式中假定输入和输出具有相同的维度。并不是要堆叠层去近似，而是显示地让这些层近似，原始函数因此而变成了。尽管两种形式都要能够渐进近似期望的函数，但学习的难度却有很大的不同。残差学习不仅训练优化的复杂度低，而且效果极好。残差函数是残差构造块的基础。

### 通过捷径的恒等映射

上述提法可以通过拥有快捷连接的前馈神经网络来实现。快捷连接，即在构建块中跳过一或多个层。在残差学习中，快捷连接就是执行简单的恒等映射，然后将该层的输出添加到下一堆叠层的输出中。残差网络的残差学习构建块可以定义为以下形式：

(2-23)

其中和分别表示所考虑层的输入和输出向量，表示要学习的残差映射，对于图2-7构建块中的2层，，为ReLU激活函数，为简化表示偏置项被忽略了。操作通过快捷连接和元素级相加来执行。公式2-23既没有引入额外的参数，也没有增加计算的复杂性，这不仅在实践中极具吸引力，而且在残差网络与普通网络的比较中也相当重要。公式2-23中的和必须具有相同的维度，如果不是这种情况的话，我们可以通过快捷连接执行一个线性投影来匹配维度：

(2-24)

当然，在公式2-23中也可以使用方阵，但对于处理退化问题来说恒等映射是足够的，故一般只用于维度匹配上。

### ResNet架构

ResNet基于普通的深度卷积网络，由堆叠的残差构造块组成，并在其中加入了快捷连接。ResNet架构与其他深度神经网络模型的对比如图2-7所示(左：VGG-19，中间：34层的普通深度卷积网络，右：34层的ResNet)。值得注意的是，和VGGNet相比，ResNet有更少的过滤器和更低的复杂度，且其在众多的机器学习比赛项目中取得了更好的效果。

当输入和输出具有相同的维度的时候可以直接使用公式2-23，当维度在增加的时候，需要考虑以下两点：

1. 捷径依然执行恒等映射，再附加额外的零项填充以增加维度，这并不会引入额外的参数；
2. 公式2-24中的投影捷径用于匹配维度。

对于这两点而言，当快捷连接跨了两个大小的特征映射时，它们所执行的步幅都是2。

ResNet受深度所蕴含的重大意义的驱动，通过残差学习、恒等映射、快捷连接等手段构造基本块，再由基本块以堆叠的方式来构造网络结构，使得学习更好的网络就像堆叠更多的层一样那么容易，网络的训练与优化也变得不是那么的困难，这对各种机器学习任务在大规模数据集的广泛应用具有重要意义。



图2-7 残差网络与其他深度神经网络结构对比

## 注意力机制

在关系抽取的建模过程中，通常采用远程监督的方法来解决标注训练语料耗时耗力的问题，但远程监督会出现错误标记的问题，错误标记的实例称为噪声，噪声实例会使关系抽取的整体性能恶化，从而影响模型效果的提升。因此要有一种机制能从实体对对应的袋子中挑选有效的句子作为特定关系与该实体对的训练正例。最初的远程监督假设是将袋子中的所有句子都作为训练正例，即对袋子中实例做平均处理，经选择器后的输出向量表示为:

(2-25)

其中b表示相应袋子的实例个数。这种方式存在大量噪声实例参与训练过程，使得关系抽取的性能难以提升。针对这个问题，Riedel等人[23]提出从实体对对应袋子中选出最有效的一个作为相应关系的训练正例，用该方法选择的输出向量V为：

(2-26)

其中B为选择函数，参数为袋子中实例的索引，用于选择标记矩阵对应列标记向量，R为关系矩阵，bias为偏置向量，表示第i个袋子的标记。该方法选择了最有效的实例作为训练正例，极大地缓解了噪声实例带来的性能下降的影响，但同时也损失了大量有效的句子，进而限制了关系抽取模型的整体效果。近些年很多学者在采用深度学习方法解决各种问题时应用了注意力机制，Yankai Lin等人[42]也将这种机制引入到了关系抽取的建模过程中。

## 对抗训练

尽管深度神经网络在各种不同的领域取得了巨大的成功，但其对输入数据中的噪声倾向于太过自信。一个神经网络模型往往在训练集上可能表现出很好的效果，但在测试集上的预测却不尽人意，这通常是由于过拟合造成的，这类问题很多情况下都可以采用正则化的策略来处理。对抗训练(Adversarial Training)就是一种通过生成对抗噪声到训练数据中的正则化分类算法[45]，以增强模型在测试集上的泛化能力。使用对抗训练编码实例的计算图如图2-8所示。

其中表示第i个训练实例，表示关于的对抗扰动。对于一个实体对，关系的概率及损失函数表示如下：

(2-27)

(2-28)



图2-8 使用对抗训练编码实例的计算图

其中为投影矩阵，为偏置向量，表示关系个数，为给定的参数集合。对抗训练会在训练过程产生持续的微小扰动，然后将产生的对抗噪声加入到词嵌入层。引入对抗训练后模型的损失函数表示如公式2-29所示，表示当前值的一个固定副本，因为从计算的角度来看，公式2-30是比较难处理的。

(2-29)

(2-30)

## StanfordNLP

StanfordNLP是斯坦福NLP组研发的一套NLP软件，它为主要的计算语言学问题提供了统计NLP、深度学习NLP和基于规则的NLP，这些工具将人类语言技术的需求结合到了应用中，并在工业界、学术界和政府中被广泛使用[57]。

本课题使用StanfordNLP来进行关系抽取模型训练前的数据预处理相关工作，主要包括用其中的POS Tagger工具对原始自然语言文本语料进行分词，然后用NER工具来识别语料中的命名实体，其示意如图2-9所示。



图2-9 命名实体识别实例

## Tensorflow

由google大脑团队研发的深度学习框架Tensorflow，是一种大规模、异构环境下运行的机器学习系统[74]，不仅大量用于机器学习方面的研究，而且在计算机视觉、语音识别、自然语言处理等计算领域被广泛应用。

Tensorflow使用数据流图来表示计算、共享状态以及改变该状态的操作。数据流图是由节点和边构成的有向图，节点代表计算操作，也可表示输入数据的起点或输出数据的终点，边表示两节点之间的输入输出关系。从Tensorflow的名字可以看出其最重要的两个概念：张量(Tensor)、流(Flow)。张量可以简单的理解为多维数组，流即为张量在计算图各节点上流动并通过节点上的计算而相互转化的过程。图2-10是Tensorflow中一个训练管道的数据流图的原理图，其中包含了用于读取输入数据、预处理、训练和检查点状态的子图。Tensorflow将数据流图的各节点映射到集群中多台机器上，其中一台机器跨多个计算设备，包括多核CPUs、通用GPUs和定制的被称为TPUs的ASICs，被分配到不同计算设备的各节点可以并行异步的进行计算。这种架构为应用程序开发人员提供了很大的灵活性：在其之前的“参数服务器”的设计中，系统内置了共享状态管理，使得开发者可以尝试新的优化和训练算法。



图2-10 Tensorflow中一个训练管道的数据流图

本课题开展的关系抽取研究利用深度神经网络来建模，至于具体的建模工具则采用目前最流行的Tensorflow。Tensorflow的计算图模型为本课题的研究及所提出的模型算法的具体实现提供了极大的便利和灵活性。

## 图形数据库

我们生活在一个彼此联系的世界，我们周围没有孤立的信息，只有丰富的、互相关联的领域。只有原生就包含关系的数据库才能有效地存储、处理、查询连接操作。在其他数据库通过代价很大的连接操作在查询时计算关系的情况下，图数据库将连接和模型中的数据一起存储。

简单的说，图形数据库就是一种将数据之间的关系和数据本身视为同等重要的数据库。其目的在不限制数据在预定义的模式下保存数据，相反，数据的存储方式与我们最初描绘它的方式是一样的，即显示单个实体是如何与其他实体进行连接或相互关联的。访问本地图形数据库中的节点和关系是一种高效、固定时间的操作，允许每秒钟快速地遍历每个核中的数百万个连接。

在进行关系抽取这一基础性研究的同时，本课题还将提出的模型应用到了科研文档的信息化建设中，构建了一个简单的语义检索系统。模型提取的关系事实是存储在图形数据库中，本文采用的是广泛使用的Neo4j图形数据库。

## 数据可视化

我们正处在一个信息洪流的大数据时代，那么数据以什么样的形式展现给用户就显得非常重要且意义重大。其中重点要考虑的是数据本身具有什么样的特点，展示的时候什么样的形式更加地简洁、清晰，以及怎样在用户获取到有价值信息的同时还会有一种美的享受。

考虑到本课题进行关系抽取的结果是三元组的关系事实，这些关系事实会构成一个庞大的实体关系网络，故本文在数据可视化技术选型时采用的是当前非常流行的D3js，然后再结合前端的HTML、CSS、JavaScript等技术，构建了一个以关系图谱的形式呈现的语义检索系统，呈现时可以与用户进行动态的交互。

## 本章小结

本章主要概述了在进行关系抽取的研究和实现过程中所涉及的关键理论和相关技术。首先，介绍了在关系抽取建模过程中基于的重要理论，包括用于自动标注大规模语料的远程监督，进行序列学习的循环神经网络，学习深层信息的残差网络，以及提升在测试集上泛化能力的对抗训练；其次，简述了用于关系建模的机器学习相关框架，包括用于自然语言处理的StanfordNLP，实现深度神经网络的Tensorflow框架；最后，简介了在进行关系抽取应用时所需要的相关技术，包括用于存储关系数据的图形数据库，进行数据可视化的D3js等前端技术。

# 关系抽取研究与算法设计

本课题充分调研并分析了国内外研究学者在关系抽取方面的研究，发现了现有方法都普遍存在的问题，这些问题严重阻碍了关系抽取效果的提升。针对已有方法所遇到的困境，本文提出**实体类型嵌入(ETE)**，ETE将和词嵌入、位置嵌入一起作为模型的联合嵌入层，此外，还精心设计了**循环分段残差网络(RPRN)**，已解决现有模型学习语料深层信息的困难。

本章首先会对已有方法所遇到的困境进行深入分析，包括出现的困境是什么、为什么会有这样的困境以及解决的思路；然后，介绍下模型的整体架构设计；最后，分别阐述模型设计过程中的核心构成要素。

## 困境分析

关系抽取作为信息抽取的核心任务和重要环节，目的是为了抽取自然语言文本语料中涉及的实体对关系以构成如<Steve Jobs, founder, Apple>这样的关系事实，然后便可以利用这些关系事实去构建语义检索、问答系统等知识图谱相关的应用。现有的方法在应对如下这类句子(其中红色加粗词的是利用NER技术识别出的命名实体，而紫色加粗的斜体词为关系触发词)时抽取出来的关系效果良好。

**Steve Jobs** was the ***co-founder*** and ***CEO*** of **Apple** and formerly **Pixar**.

通过观察上述句子可以发现，首先这个句子比较简短，其次在这个句子中出现了要抽取的实体对关系对应的触发词，这就为该类句子中涉及的实体对关系的正确抽取提供了重要依据。然而，基于我们在NYT公开数据集上所进行的实验，现有的模型往往无法在以下具有挑战性的情况下抽取出有效的关系。

1. 句子中没有对应关系的直接触发词，如1.2.2小节的图1-1中第一个句子所示的情况；
2. 句子长而复杂并且两个实体之间的距离相隔比较远，如1.2.2小节的图1-1中第二个句子所示的情况。

我们认为现有模型在以上情况下抽取失败是由于缺失深层语境中隐藏的潜在信息造成的。为解决这一困境，我们提出一种基于联合嵌入和新的神经网络结构的方法。更具体地说，提出**实体类型嵌入(ETE)**，并将其和词嵌入、位置嵌入一起作为模型的联合嵌入层；此外，设计了**循环分段残差网络(RPRN)**的结构，和现有关系抽取模型所用的神经网络结构相比，RPRN大大增加了模型的深度，且其内部结构也是精心设计的，这样使其能有效编码句子上下文中深层次的潜在信息。

## 模型架构设计

受到融合更多更深次信息的思想启发，本文将提出的ETE与词嵌入、位置嵌入一起作为模型的联合嵌入层，以丰富模型的嵌入表示，从而有助于抽取句子所涉及实体对中潜在的隐晦关系；而设计的RPRN结构则弥补了现有关系抽取浅层模型存在的缺陷，即学出的特征太粗粒度以致于无法获取句子中丰富的潜在语义。根据以上提出的两点，本文设计的模型的整体设计架构如图2-1所示。

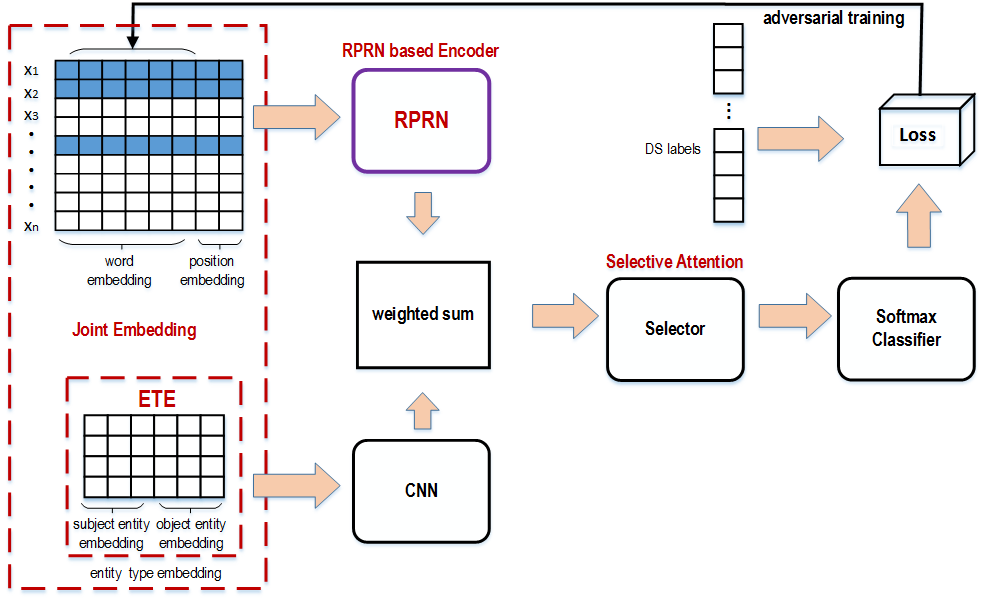


图3-1 提出的关系抽取模型整体设计架构

本文设计的如图3-1所示的模型架构主要可以分为以下3个模块：

1. 嵌入层：由提出的ETE与词嵌入、位置嵌入一起作为模型的联合嵌入层，以丰富模型的分布式表示；
2. 编码层：对于词嵌入和位置嵌入，使用我们设计的RPRN进行编码，而对于提出的ETE则使用普通的CNN来编码；
3. 注意力层：为缓解噪声实例带来的性能下降问题，我们在模型训练过程中应用了选择性注意力机制。

以图3-1中的实例作为例子，最上面的表表示词嵌入与位置嵌入的联合，其中词嵌入是预训练好的，而位置嵌入是随机初始化的，下面的表表示我们提出的ETE，这也是随机初始化的，并将和词嵌入、位置嵌入一起作为模型的联合嵌入层。词嵌入与位置嵌入的联合表示将经过设计的RPRN结构进行编码，并将和ETE的编码输出进行加权连接。编码之后，拥有相同实体对的实例会被放到同一袋子中，然后，基于注意力机制的选择器将会尽可能从袋子中选择训练关系的相应正例。最后，会得到一个表示训练关系潜在模式的向量表示，最后一层的分类器将会基于这个向量表示输出最可能的关系。

## 联合嵌入

本文设计的关系抽取模型的嵌入层由词嵌入、位置嵌入及所提出的ETE一起联合构成，这是模型的一种分布式向量表示，其表示是否充分直接影响了关系抽取模型最后的整体效果。

### 词嵌入

词嵌入是将词从高维的词空间映射到低维的向量空间的一种分布式表示，语义相似的词在这个向量空间中相对也更加地靠近。由此可以看出，词嵌入可以表达词之间的语义关系。在我们提出的关系抽取模型中，使用的是skip-gram[52]预训练以得到词向量，但是预训练得到的词向量仅仅作为词嵌入的初始化值，之后再模型训练过程中会动态更新。

### 位置嵌入

为了有效利用句子中每个词的位置信息，位置嵌入[53]被应用到关系抽取的建模过程中。上面提到的位置信息指的是句子中的每一个词与相应的实体对之间的相对距离，如图3-2所示，然后将每一个词映射到一个向量空间中。此外，每一个词对应的两个向量都是随机初始化的，然后随着模型的继续训练而不断地动态更新。



图3-2 句子中每个词和相应实体的相对位置

### 实体类型嵌入

由3.1小节分析的困境可知，现有方法在所述情景中失败的主要原因是缺少深层语境中隐藏的潜在信息造成的，那有没有什么信息可以弥补这一缺陷呢？我们认为，实体类型信息在关系抽取中至关重要。假如<Steve Jobs, founder, Apple>这一关系事实在不同的实例中都有体现，现有模型通常只能从简单明了的实例中抽取出这一有效的关系，而在困境所述情景中会失败，如果有一种机制可以学出实例中实体对对应的实体类型，从直观上来讲，该类型信息可以有效避免错判和漏判的情况。

基于以上想法，我们提出ETE。但ETE的结构与词嵌入、位置嵌入的大相径庭，故不能直接将ETE和词嵌入、位置嵌入一起简单地连接起来作为联合嵌入层。我们采取的方案是将ETE作为表示层的一部分单独拿出来并使用CNN进行编码，如图3-1最下面的表所示。

给定某一实体对应的实体类型序列，训练时要最小化以下目标函数来学习ETE表示：

(3-1)

(3-2)

(3-3)

其中，为目标实体类型，表示大小为c的窗口内的上下文实体类型，和是分别作为目标和上下文时实体类型E的嵌入，负实体类型样本来自平滑的unigram[75]概率分布，即噪声分布：，表示实体类型在训练数据中出现的频率，为平滑参数()。

为了自适应学习率，目标函数应用AdaGrad来优化，给定目标上下文的实体类型对(和)以及k个来自于噪声分布的负样本()，梯度约为：

(3-4)

梯度被计算后，梯度下降就会执行更新、，以及。在训练过程执行参数更新后，我们就可以得到实体类型嵌入，即提出的ETE。ETE核心算法的伪代码如算法1所示。

|  |
| --- |
| 算法1：ETE嵌入表示训练 |
| for i = 1, … , n do      for j = -c, … , -1, 1, … , c do  从 分布中抽取k个负样本  使用AdaGrad执行更新、，以及  end for  end for |

在我们的关系抽取模型中，词嵌入的维度为，位置嵌入的维度为，然后它们连接在一起将一个训练实例转变成了一个矩阵，表示语料中句子的最大长度，表示联合词嵌入与位置嵌入的表示层的维度，矩阵将会喂给RPRN编码器。ETE的表示维度为，ETE表示为，表示语料中一个实体最多的类型个数。

## RPRN编码器

由于ETE被引入到联合嵌入层并被独立编码，我们的关系抽取模型的编码器就包含了两个部分：一个是用于编码词嵌入、位置嵌入的联合嵌入表示的基于RPRN的编码器，另一个是用于编码ETE的基于CNN的编码器。RPRN的网络结构设计原理如图3-3所示。



图3-3 RPRN的结构原理

由3.1节分析的困境可知，对于复杂的句子，关系抽取模型要想获取深层语境下隐藏的潜在表示是很困难的。受抽象分层思想的启发，本文设计了RPRN的网络结构，该结构主要包含三个部分：进行序列学习的循环结构模块，用于对序列特征表示进行深度分级抽象的残差模块，以及采样特征以获取主特征的分段池化模块。下面几节将详细阐述各模块的设计与原理。

### 序列学习

对于自然语言文本、语音等可变长度的序列，如何共享不同位置的参数信息，以及如何处理不同距离的依赖，是极为重要而又较难解决的问题。RNN网络模型的出现有效解决了序列学习的困难，然而对于长期依赖的问题，普通RNN处理的效果依然不尽如人意。为解决这一问题，有研究学者就提出了各种RNN的变体，如经典的LSTM、GRU等门控机制的循环网络单元。

本文在对关系抽取进行建模时，利用双向RNN来学习句子的上下文信息，其中使用GRU作为循环单元。GRU与LSTM都能较好地解决长期依赖问题，但GRU由于比LSTM少一个门控单元而更加简化，优化收敛的速度都更快，且GRU在我们的关系抽取模型中确实取得了更好的效果。GRU主要由更新门和重置门来控制，在我们的模型中，其表示分别如公式3-5和3-6所示。

(3-5)

(3-6)

其中，为句子中第个位置的词所对应的词嵌入与位置嵌入的联合表示，、表示更新门的权重矩阵，、表示重置门的权重矩阵，和分别表示两个门的偏置向量，当前位置隐藏层状态向量为，这是由上一时刻更新和保留的信息共同决定的，公式3-7与3-8为其推导过程。

(3-7)

(3-8)

在本模型中，GRU作为双向RNN的基本循环单元，通过横向随着序列延伸并纵向堆叠的方式来进行序列学习，如图3-1右下角部分所示。

### 残差结构

深度神经网络(DNNs)是极其强大的机器学习模型，它在一些诸如语音识别、视觉对象识别等困难的问题上依然有着非常优秀的表现。通常来说，网络结构越深，模型的表示能力就越强，但是随着网络层数的加深，模型也将变得越难以优化，甚至有些情况的表现还不如浅层模型。因此，ResNet[55]神经网络模型被提出，从而有效解决了深度网络模型优化难的问题。

为了学出深层语境下隐藏的潜在表示，本文在进行关系抽取建模时，设计的RPRN网络结构借鉴了ResNet的优秀思想，但RPRN中的残差结构与ResNet中的有所不同，我们模型的残差基本构造块的原理如图3-4所示。在RPRN中，由残差基本构造块得出的残差函数如公式3-9所示。

(3-9)



图3-4 残差基本构造块的原理

其中，为1维卷积函数，为激活函数，在恒等映射中，基本构造块的大小为，基本构造块的个数为，在残差结构之前有一个卷积层，残差块后就会执行分段池化，以在各段上采样主要特征。

### 分段池化

为了有效抽取实例中实体对的语义关系，我们利用分段池化来学习相应上下文中的结构化信息，这是一种极为有效的结构化学习[76]的方法。在分段池化中，我们引入隐码信息，表示为，其嵌入表示为，其中，m表示每一个批量的大小，n表示语料中句子的最大长度。执行分段池化操作后得到的结果为，其推导如公式3-10所示，其中表示模型编码器的隐藏层的大小。

(3-10)

关于公式3-10中的参数做以下说明：表示调整张量形状的函数，是执行分段池化操作的函数，表示扩展张量中的某一维的函数，且其值为1以保持该张量的整体维度不变，为查找隐码嵌入的函数，表示能使分段池化操作能有效执行而定义的一个常量，是残差模块的输出张量。图3-1中残差结构后的分段池化操作的实现原理如图3-5所示。



图3-5 分段池化操作的原理

作为RPRN结构的一部分，分段池化层是为了从之前层表示的特征编码中采样出主要特征。分段池化，在我们的关系抽取的上下文中，表示根据实体对在句子中的位置而将当前实例分成三段，然后每一段分别执行最大池化操作。

至于ETE，编码之后使用的最大池化操作，其隐藏层的大小为。最大池化的结果会与分段池化的进行连接作为关系抽取器的编码器，其输出将会喂给下一阶段的实例选择器。

## 注意力选择器

在应对大规模自然语言文本语料时，远程监督方法通常被应用到关系抽取的建模中来，以解决手动标注语料耗时耗力的问题。然而，该方法同时也伴随着错误标记的问题，错误标记的实例被称为噪声。在关系抽取模型训练时，具有相同实体对的实例会被放到同一个袋子中。

远程监督最初的假设是将一个袋子中的所有实例都作为正例，这使得在模型训练过程中存在大量的噪声。为解决这一问题，Riedel等人[23]采取的是将袋子中最有可能的一个实例作为训练正例，这极大地缓解了噪声实例带的性能下降问题，但同时也丢弃了大量有效的实例，从而阻碍了关系抽取性能的进一步提升。因此，Lin等人[42]将选择性注意力机制应用到关系抽取中。

本文在进行关系抽取的研究时，也是用选择性注意力机制来解决噪声标注的问题，其原理如图3-6所示，其中，表示同一袋子中的第i个实例，表示第i个实例对应的特征潜在编码，表示由注意力机制学出的权重参数。经选择器得到的输出向量推导如公式3-11所示。

(3-11)

其中，表示挤压函数，用于去掉一个张量中所有所有维度值为1的维度，表示矩阵相乘操作的函数，表示扩展张量中的某一维的函数，且其值为1以保持该张量的整体维度不变，表示编码器的输出，表示关系矩阵，为语料的真实标记，表示偏置向量，表示袋子的大小。

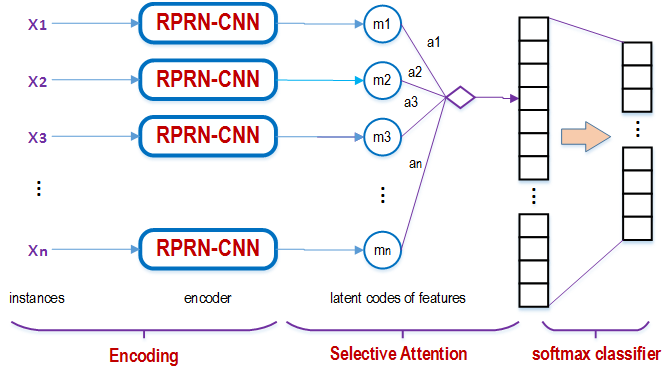


图3-6 选择性注意力机制

利用上述的选择性注意力机制，那些确实表达了其实体对在知识库中对应关系的句子就能够被挑出来作为正例喂给模型训练，而那些没有表达对应关系的则被当成噪声。更具体的说，这种选择性注意力机制保证了有效实例的权重会上升，而噪声实例的权重会下降，从而极大缓解了噪声带来的性能下降的问题。

我们模型的最后一层采用softmax来分类输出最可能的关系，损失函数使用的是交叉熵损失，其表示及其梯度分别如公式3-12和3-13所示，公式3-14则表示在给定实例下输出第k个关系的概率，其中为实例个数，为总共训练的关系数，和则分别表示实例和相应标记。

(3-12)

(3-13)

(3-14)

此外，在模型训练优化的过程中，会不断地给嵌入层中的词嵌入表示一个反馈扰动，即进行对抗训练，其原理及推导已在第2章2.5节描述，此处不再详述。实验表明，这种方式可以增强模型在测试集上的泛化能力。

至此，我们整个模型从嵌入表示到最后的分类输出总共构成了12层的深度神经网络，模型的核心训练流程如算法2的伪代码所示。

|  |
| --- |
| 算法2：ETE-RPRN模型训练核心算法 |
| 输入：训练语料，超参数，预训练的词向量  使用*算法1*训练  初始化参数，以及  ，， ，，    for to do      for to do  在上应用公式3-5到3-8  在上应用公式3-9  在上应用公式3-10        foreach do  尽最大可能从袋子中挑选出正例来  在上应用公式3-11  end for    更新参数    end for  end for  输出：抽取的关系 |

关于算法2中涉及的符号，我们接下来将做一些简要说明。作为训练语料，主要包含词、位置及实体类型信息，即分别表示为、及，表示模型在整个训练过程中的参数，表示位置向量，这是随机初始化的，表示实体类型向量，这个由算法1先预训练，再随着模型的继续训练而不断被更新，在算法2中直接用到的超参数包括最大轮数和批量大小。此外，表示隐藏层的输出，表示最后一层的输出，为整个模型的损失函数。

## 本章小结

本章主要对关系抽取这一基础性研究进行了分析并精心设计了我们自己提出的模型。首先，详细剖析了国内外研究学者在关系抽取方面所遇到的困境，包括困境是什么，为什么会有这样困境，以及解决的思路是什么；然后从整体上阐述了模型的架构设计及各主要构成要素；接下来，详细论述了模型的几个主要构成模块的设计，具体的来说，第一，联合嵌入层的设计，包括词嵌入、位置嵌入以及我们提出的ETE，并给出了ETE设计的详细推导及训练算法的伪代码，第二，我们设计的RPRN网络结构，其中包括进行序列学习的循环单元GRU，残差结构的设计和原理说明，以及分段池化的作用及原理阐释，第三，应用的注意力选择器，主要对其作用、推导及原理做了详细的阐述；最后，简单描述了对抗训练在本模型的作用及运作机制，以及给出了我们整个模型训练的核心算法的伪代码和相关说明。

# 关系抽取模型实现

前一章剖析了目前在关系抽取方面的研究所遇到的困境，针对这些困境及解决的思路，设计了整个模型的架构，并详细论述了我们提出的模型中主要模块的设计及核心算法，这为本章要阐述的关系抽取模型的实现提供了重要的理论基础。本章将介绍所提出的关系抽取模型的实现细节，及模型应用的语义检索系统的构建。

## 模型构建环境

### 依赖环境及组件

本文开展的关系抽取的研究主要分为两个阶段，一是对前一章我们所设计的模型及模型的应用的实现，二是对实现的模型进行训练，这两个阶段所需的环境如表4-1所示，其中最后两个是模型应用环境。

表4-1 关系抽取模型依赖环境

|  |  |
| --- | --- |
| 依赖环境 | 软件名称/版本 |
| 模型开发IDE | PyCharm 2018.1.4 |
| 模型开发语言 | Python 3.6 |
| 模型开发服务器 | Ubuntu 18.04.1 LTS |
| 模型训练服务器 | NVIDIA GPU(Centos 7.5) |
| 部署容器 | Docker 18.09.1 |
| 应用运行服务器 | Centos 7.6 |
| 图形数据库 | Neo4j 3.3.4 |

设计的关系抽取模型实现时所依赖的重要组件如表4-2所示，其中最后三项为模型应用的语义检索系统在构建时所依赖的组件。

表4-2 关系抽取模型实现所依赖组件

|  |  |
| --- | --- |
| 依赖组件 | 版本 |
| Tensorflow | 1.11.0 |
| StanfordNLP | 3.9.2 |
| Scikit-learn | 0.20.0 |
| Numpy | 1.15.2 |
| Scipy | 1.1.0 |
| Matplotlib | 3.0.0 |
| Neo4j | 1.7.1 |
| Flask | 1.0.2 |
| D3js | 4.12.0 |

### 模型训练环境配置

从前一章所设计的关系抽取模型可以看出，本文提出的模型是基于深度神经网络结构的，这类模型在训练的时候涉及到大量的矩阵运算，而GPU在这方面具有天然的优势。因此，为了加快模型的训练速度，本模型是在实验室配备的用于AI研究的GPU服务器上进行训练的，该GPU的配置如表4-3所示。

表4-3 GPU服务器的配置

|  |  |  |
| --- | --- | --- |
| NVIDIA-SMI 396.37 Driver Version: 396.37 | | |
| GPU Name Persistence-M  Fan Temp Perf Pwr:Usage/Cap | Bus-Id Disp.A  Memory-Usage | Volatile Uncorr. ECC |
| 0 TITAN V Off  28% 41C P8 28W / 250W | 00000000:1A:00.0 Off  11MiB / 12066MiB | N/A  0% Default |
| 1 TITAN V Off  29% 43C P8 27W / 250W | 00000000:1B:00.0 Off  11MiB / 12066MiB | N/A  0% Default |
| 2 TITAN V Off  28% 42C P8 28W / 250W | 00000000:3D:00.0 Off  11MiB / 12066MiB | N/A  0% Default |
| 3 TITAN V Off  30% 45C P8 28W / 250W | 00000000:3E:00.0 Off  11MiB / 12066MiB | N/A  0% Default |
| 4 TITAN V Off  28% 37C P8 25W / 250W | 00000000:88:00.0 Off  11MiB / 12066MiB | N/A  0% Default |
| 5 TITAN V Off  28% 41C P8 28W / 250W | 00000000:89:00.0 Off  11MiB / 12066MiB | N/A  0% Default |
| 6 TITAN V Off  28% 41C P8 28W / 250W | 00000000:B1:00.0 Off  11MiB / 12066MiB | N/A  0% Default |
| 7 TITAN V Off  28% 41C P8 26W / 250W | 00000000:B2:00.0 Off  11MiB / 12066MiB | N/A  0% Default |

本课题开展的关系抽取研究是在NYT公开数据集上进行的，该数据集规模较大，在CPU上运行极为耗时。故本模型的训练是在GPU服务器上进行的，训练前可以通过参数指定利用的GPU核数，本模型指定的是8个核，大大节省了训练的时间。

## 数据预处理模块实现

对原始的自然语言文本语料进行建模，第一步就是要对数据进行预处理，即对原始数据做清理、标注和变换等工作，使其满足模型训练所需的格式规范。为方便与其他研究学者的模型进行比较，本模型是在NYT公开数据集上进行实验和验证的，但该数据预处理流程和数据格式规范对不同的数据集、不同的关系抽取模型都是有效的。以下两个小节将主要介绍数据预处理的大体流程，以及处理的数据格式规范。

### 数据预处理流程

本课题开展的关系抽取研究是在自然语言文本语料上进行的，在模型训练之前，进行数据预处理的流程大致如图4-1所示。

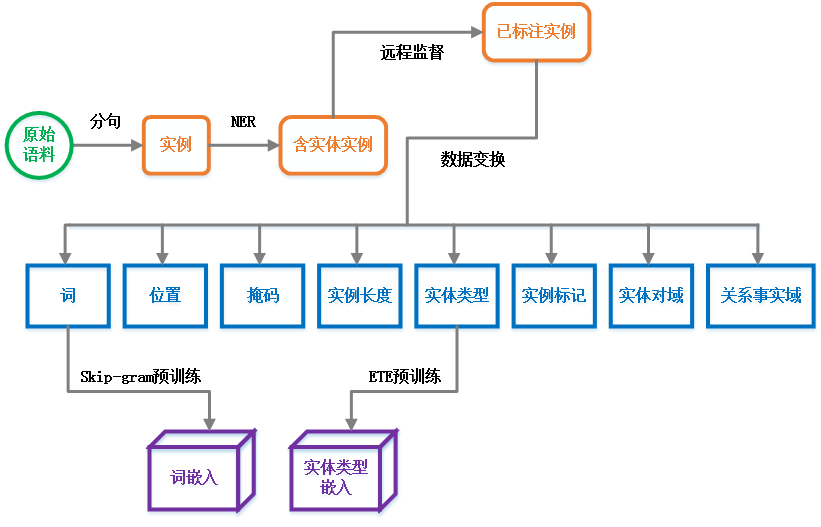


图4-1 数据预处理流程

从最原始的语料开始，首先经过分句处理会得到很多的实例，然后对每一个实例使用StanfordNLP工具来识别其中的命名实体，接下来，使用远程监督来自动标注每一个实例，最后，对已标注的实例进行数据的变换，以得到模型训练所需的如词、位置、掩码等各种形式的信息。除此之外，我们模型训练时要用的词嵌入及位置嵌入是先分别经过Skip-gram预训练和ETE预训练后再喂给模型的。

### 数据格式规范

原语料经过数据预处理得到的数据必须满足模型训练所需的格式规范，且对不同的数据集都使用同样的规范，从而使数据预处理流程具有较强的通用性。

首先，原始语料经过分句后得到如表4-4所示的多个实例，然后使用StanfordNLP识别出每个实例的实体对，紧接着利用远程监督自动标注实例得到如表4-5所示的Json格式的数据，并以中间形式的训练语料保存到文件中；最后，读取保存的如表4-5所示的Json格式的训练语料，并经过数据变换得到训练需要的各种形式的数据，这些数据都是通过numpy以数字形式的多维数组的格式保存。

表4-4 原语料分句后得到的多实例

|  |
| --- |
| "She also oversaw the refinancing of the state Superfund law ; the creation of a plan for decontaminating heavily polluted Onondaga Lake , near Syracuse ; the acquisition of hundreds of thousands of acres of Adirondack woodlands ; and the imposition of tough new acid rain rules .",  "More than 2.5 million cubic yards of contaminated mud will be dredged from Onondaga Lake , near Syracuse , under a consent decree between the state and Honeywell International that was announced yesterday .",  "The Onondaga Nation has about 1,500 members , and its federally recognized territory is an 11-square-mile parcel in a valley south of Syracuse and about eight miles from the southern end of Onondaga Lake .", |

表4-4 经远程监督处理得到的已标注实例

|  |
| --- |
| {  "head": {  "id": "/guid/9202a8c04000641f8000000000038574",  "type": "/location/dated\_location,/location/place\_with\_neighborhoods,/user/brendan/  default\_domain/top\_architectural\_city,/location/citytown,/location/location,  /user/nitromaster101/default\_domain/abh\_city,/sports/sports\_team\_location,  /common/topic,/location/statistical\_region",  "word": "Syracuse"  },  "relation": "/location/location/contains",  "sentence": "She also oversaw the refinancing of the state Superfund law ; the creation of a plan for decontaminating heavily polluted Onondaga Lake , near Syracuse ; the acquisition of hundreds of thousands of acres of Adirondack woodlands ; and the imposition of tough new acid rain rules .",  "tail": {  "id": "/guid/9202a8c04000641f80000000002fe8e4",  "type": "/common/topic,/geography/geographical\_feature,/geography/lake,/geography/  body\_of\_water,/location/location",  "word": "Onondaga Lake"  }  } |

其中，词、实体类型、实例标记的关系中的token都会转换成数字id，掩码是根据每一个词与实体对位置关系为实例设置的掩码，每一个词有一个掩码数字，主要用于模型的分段池化，实体对域和关系事实域分别表示同一实体对和同一关系事实各自关联的袋子所对应的实例范围，能够这样表示是因为处理的中间形式的语料是按照关系事实、实体对来排序的。词嵌入、实体类型嵌入预训练后的每一个词、每一个实体类型都各自对应一个向量，且也是采用numpy并以二维数组的格式来保存的。

## 关系抽取建模实现

基于上一章的模型与算法的设计，我们使用Python语言、Tensorflow深度学习框架等相关工具具体实现了我们提出的关系抽取模型，模型实现中各核心模块之间的依赖关系如图4-2所示。

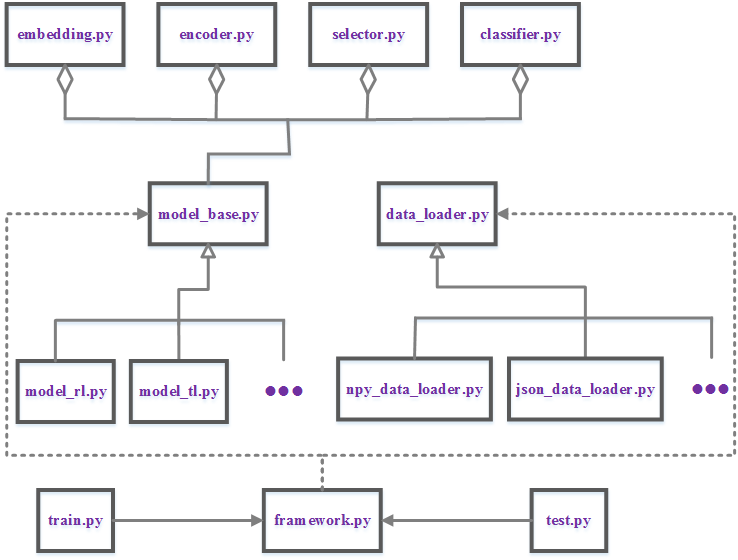


图4-2 关系抽取模型各核心模块之间的依赖

在实现前一章所设计的关系抽取模型的过程中，我们遵循最佳实践，制定了以下关系抽取模型实现的编程规范，以达到更优雅、更通用等目的。

1. 训练与测试过程均直接与框架关联，框架依赖模型和数据加载器；
2. 基模型实现了基本的初始化与整体架构，聚集了模型架构的核心构成模块：联合嵌入层、编码器、选择器和最后一层的分类器，并给出了基于神经网络结构的默认实现，其子类可以有不同的具体实现，如利用强化学习、迁移学习等技术来实现；
3. 基数据加载器完成了与格式无关的数据处理与加载操作，其子类实现特定格式相关的数据操作，如Numpy、Json等格式。

### 模型网络架构

为了在NYT公开数据集上更好地与其他研究学者提出的关系抽取模型进行比较，本文除了实现所提出的模型外，也复现了很多其他论文的模型，得益于以上制定的编程规范，不同模型使用的是同一套代码框架，在训练的时候只需通过传递不同参数来加以区别。本文实现的模型网络架构的核心代码如下：

|  |
| --- |
| **def** \_network(self):  *# embedding* self.\_embedding()  **with** tf.variable\_scope((**'et\_' if** self.et **else ''**) + *# entity\_type* FLAGS.en + **"\_"** + FLAGS.se +  ((**'\_'** + FLAGS.cl) **if** FLAGS.cl != **'softmax' else ''**) + *# classifier* ((**'\_'** + FLAGS.ac) **if** FLAGS.ac != **'relu' else ''**) + *# activation* ((**'\_'** + FLAGS.op) **if** FLAGS.op != **'sgd' else ''**) + *# optimizer* **'\_ad\_'** + (**'y' if** FLAGS.ad != 0 **else 'n'**), *# adversarial* reuse=tf.AUTO\_REUSE):  *# encoder selector classifier* self.\_encoder\_selector\_classifier(reuse=**False if** FLAGS.ad **else True**)  *# adversarial\_training* self.\_adversarial()  **def** \_encoder\_selector\_classifier(self, reuse=**True**):  **with** tf.variable\_scope(FLAGS.en + **"\_"** + FLAGS.se +  ((**'\_'** + FLAGS.cl) **if** FLAGS.cl != **'softmax' else ''**), reuse=reuse):  self.\_encoder() *# encoder* self.\_selector() *# selector* self.\_classifier() *# classifier* |

在上述实现的网络模型架构中，主要由嵌入层、编码器、选择器和分类器组成，此外，还可以选择在模型中是否引入对抗训练过程。这是在基模型中给出的模型架构核心实现，对于不同的实现子类可以复写这些核心模块以达到定制化的效果。其中，编码器、选择器和分类器三个模块是可以多次被复用的，如对抗训练在对词嵌入进行反馈扰动的时候，故单独抽出来了。

本文提出的模型的核心是ETE和RPRN，与其设计原理对应的算法实现伪代码分别由前一章的算法1、以及算法2中的部分给出，此处便不再赘述。

### 数据定义规范

在模型训练之前，要给模型准备好训练时要用的各种数据，数据的格式要满足模型所需的要求，因此，在模型初始化时给出了数据定义的规范，如下代码所示。本文给出的关系抽取模型的实现是基于Tensorflow的计算图模型，数据都以张量的形式来定义，具体采用占位符来定义，模型在训练时会将对应的具体数据以批量的形式喂给这些占位符，其中，每一个批量的各种数据与对应的占位符在维度上是一致的。

|  |
| --- |
| **def** \_\_init\_\_(self, data\_loader, et=**False**, activation=tf.nn.relu, is\_training=**True**):  self.rel\_tot = data\_loader.rel\_tot  self.enttype\_tot = data\_loader.enttype\_tot  self.word\_vec = data\_loader.word\_vec  self.activation = activation  self.is\_training = is\_training  self.keep\_prob = 0.5 **if** is\_training **else** 1.0  self.et = et  batch\_size = FLAGS.batch\_size // FLAGS.gn **if** is\_training **else** FLAGS.batch\_size  *# 此处省略数据约束的定义，以及下面定义的数据规范所需满足的约束*  self.word = tf.placeholder(dtype=tf.int32, shape=[**None**, FLAGS.max\_length], name=**'word'**)  self.pos1 = tf.placeholder(dtype=tf.int32, shape=[**None**, FLAGS.max\_length], name=**'pos1'**)  self.pos2 = tf.placeholder(dtype=tf.int32, shape=[**None**, FLAGS.max\_length], name=**'pos2'**)  self.mask = tf.placeholder(dtype=tf.int32, shape=[**None**, FLAGS.max\_length], name=**"mask"**) self.length = tf.placeholder(dtype=tf.int32, shape=[**None**], name=**'length'**) self.label = tf.placeholder(dtype=tf.int32, shape=[batch\_size], name=**'label'**) self.instance\_label = tf.placeholder(dtype=tf.int32, shape=[**None**], name=**'instance\_label'**) self.entity\_pos = tf.placeholder(dtype=tf.int32, shape=[**None**, 2], name=**'entity\_pos'**) self.scope = tf.placeholder(dtype=tf.int32, shape=[batch\_size + 1], name=**'scope'**) self.weights = tf.placeholder(dtype=tf.float32, shape=[batch\_size], name=**'weights'**) self.head\_enttype = tf.placeholder(dtype=tf.int32, shape=[**None**, FLAGS.et\_max\_length], 　 　 | name=**"head\_enttype"**) self.tail\_enttype = tf.placeholder(dtype=tf.int32, shape=[**None**, FLAGS.et\_max\_length], | name=**"tail\_enttype"**) self.enttype\_length = tf.placeholder(dtype=tf.int32, shape=[**None**, 2], name=**"enttype\_length"**) self.enttype\_mask = tf.placeholder(dtype=tf.int32, shape=[**None**, 2 \* FLAGS.et\_max\_length], | name=**"enttype\_mask"**) |

以上定义了模型所需的关键数据的格式规范，其中rel\_tot、enttype\_tot分别表示要训练关系总数以及通过远程监督标注的语料中涉及的实体类型总数，其他定义的数据含义与其名字是一致的。在上述定义的数据规范中，省略了数据约束的定义以及所定义的数据规范所需满足的约束条件，这些主要包括什么样的数据在什么条件下可以使用等约束。

### 训练过程

在训练本文实现的关系抽取模型时，可以在终端给出命令python3 train.py help来获取训练时可指定的参数，帮助信息大体如下所示格式：

|  |
| --- |
| Usage: python3 train.py [--dn dataset\_name] [--et: ent\_type] [--en encoder] [--se selector] **……**  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*args details\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  --dn: (dataset\_name)[nyt(New York Times dataset)...], put it in origin\_data dir  --et: (ent\_type)whether to add entity type info(default 0), 0(no), 1(yes)  --en: (encoder)[cnn pcnn rnn birnn rnn\_lstm birnn\_lstm rnn\_gru birnn\_gru] **…**  --se: (selector)[instance att one ave cross\_max att\_rl one\_rl ave\_rl cross\_max\_rl]  **……** |

为了加快模型训练的速度，本文实现的模型支持在GPU服务器上训练，训练时可指定并发运行的GPU核数，整个训练的核心实现流程如下代码所示：

|  |
| --- |
| **def** train(self, model):  print(**"Start training..."**)  *# 定义会话和模型参数等变量* ***……***  **for** gpu\_id **in** range(FLAGS.gn):  **with** tf.device(**"/gpu:%d"** % gpu\_id):  **with** tf.name\_scope(**"gpu\_%d"** % gpu\_id):  *# 定义模型，计算损失、梯度等操作* ***……***  **for** epoch **in** range(FLAGS.max\_epoch):  print(**'###### epoch '** + str(epoch) + **' ######'**)  *# 初始化评估统计等信息* ***……***  **while True**:  **try**:  *# 判定约束并定义训练所需参数*  ***……***  iter\_loss, iter\_output, iter\_label = self.\_one\_step\_multi\_models(models,  [op, lo, out])[1:]  **except** StopIteration:  **break**  *# 计算训练时间、评估统计等信息* ***……***  **if** self.acc\_not\_na.total > 0:  *# 输出训练过程、评估统计信息到控制台* ***……***  self.step += 1  print(**"\nAverage iteration time: %f"** % (time\_sum / self.step))  **if** (epoch + 1) % FLAGS.save\_epoch == 0:  metric = self.test(model)  **if** metric > best\_metric:  *# 保存最佳模型* ***……***  **else**:  not\_best\_count += 1 |

在上述所示的代码实现中，整个语料库会分成多个批次在多个GPU核上并行训练，语料整个训练完一次称为一轮(epoch)，并和上一轮保存的最佳模型进行比较，如果更优，则更新保存的模型，否则会根据设定的阈值判定是否终止训练。

## 语义检索实现

本文实现的关系抽取模型适用于不同类型的数据集，因此将提出的模型应用在了社科文档上，并做了一个简单的轻量级的语义检索系统，其实现流程大致如下：

1. 利用提出的模型去抽取社科文档中涉及的实体对关系，并将所有抽取的关系事实存到Neo4j数据库中以构成关系网络；
2. 使用Flask微框架实现后端检索功能，检索的实体相关的关系数据采用D3js来实现关系图谱的可视化；
3. 本应用采用微服务的方式来部署，应用运行在docker容器中。

## 本章小结

基于前一章的关于关系抽取的分析研究与算法设计，本章给出了模型核心部分的具体实现细节。首先，介绍了模型所需的基本环境，包括模型开发、训练等的依赖环境和重要组件，以及模型训练环境GPU的配置；然后，介绍了数据预处理模块的详细实现流程，还有模型训练所需的数据格式规范；接下来，详细阐述了关系抽取模型的实现，包括较通用的模型网络架构的核心实现，数据定义的规范，以及训练过程的关键实现流程；最后，基于本文所提出的关系抽取模型，简单地介绍了下在社科文档上所构建的语义检索系统的实现流程。

# 模型评估及案例分析

第三章和第四章详细介绍了推荐系统的设计和实现，本章将从不同的角度对推荐系统进行测试和分析，主要包括系统的性能测试及分析和协作关系网络数据分析两大部分。系统测试及性能分析主要是从系统的基本功能、响应时间和推荐质量来进行分析，协作网络数据分析主要是探索和分析协作关系网络呈现的一系列规律。为了便于描述，本章将采用图片和表格来进行展现和分析测试结果及相关数据。

## 模型评估

## 案例分析

## 本章小结

本章主要对推荐系统的基本功能进行了测试，并对测试结果和推荐质量进行分析，从对测试结果和推荐质量分析上可知，推荐系统达到了设计要求和预期效果，满足了为科研合作者推荐潜在合作者的服务需求。同时本章尝试对协作关系网络呈现的一系列现象和规律进行探索和分析，为进一步研究协作关系网络夯实基础。

# 总结与展望

## 论文总结

本文是对提高科研管理数据资源利用率和解决广大科研工作者寻求合作者效率和质量低下问题的一个初步尝试，试图通过先进的计算机技术和理论，实现对科研管理数据资源的利用，并在一定程度上提高广大科研人员寻找合作者效率和质量。促进科研人员之间的交流，一定程度上可以起到增强科研成果和协作质量的效果。以下是本文的主要内容：

1）确定了课题需要解决的问题。通过对科研管理研究数据现状和广大科研工作者在寻求高质量合作者过程中遇到的问题进行分析，确定了本课题要解决的实际问题，并对该问题进行了拆分。

2）对相关理论、技术及架构的学习和选型。通过对传统推荐系统的相关理论和技术的学习，并结合本课题所要解决问题的实际情况，确定了本推荐系统所采用的理论原型——基于关系网络的推荐。调研并学习了基于图的相关理论、基于SSH的Web开发框架技术、基于面向架构的Thrift通信技术、基于Python数据处理的Networkx框架、基于前端可视化的Highcharts和D3技术等。

3）系统设计和实现。通过分析传统推荐系统原型在解决本课题实际问题时表现的不足，设计并提出了一套适用于该课题的推荐模型。包括数据清洗的设计、协作网络模型的设计、协作关系相似度和研究领域相似度的设计等。同时基于前期对相关理论和技术的调研、学习和研究，实现了推荐模型的基础框架，并在此框架上对外提供推荐服务和相关的数据分析。

4）系统测试及数据分析。对推荐系统进行了测试并对结果进行了简要的分析，同时针对在系统设计与实现过程中协作关系网络呈现的一些现象和规律做了简要的探究和分析。

综上所述，本文从系统的需求分析、技术及架构选型、系统设计、编码实现、系统测试以及协作关系网络探究等方面介绍了基于项目协作关系的合作者推荐系统的设计与实现过程。该系统借助于挖掘科研管理研究数据中的项目及人员信息，提高了数据的利用率，同时为科研工作者快速高效的寻找合作者提供了服务，能够促进科研人员之间的有效合作和有效交流，进而提高协作质量，在一定程度上为促进学科发展和增强科研质量提供了新的思路和方向，进一步推进了高校信息化平台的建设。

## 论文展望

到目前为止，本课题的研究已经告一阶段，通过基于项目协作关系的合作者推荐系统可以实现对给定作者推荐其潜在合作者，但是由于本人知识水平和时间等因素的限制，仍然还存在一些需要完善和值得探索的地方，现简述如下：

1. 用户反馈收集

目前实现的合作者推荐系统不包含用户对推荐结果的反馈功能。一方面可以通过分析用户对推荐结果的反馈来表征推荐系统的推荐质量；另一方面可以通过分析用户对推荐结果的反馈来完善推荐模型和调节参数，使得推荐系统更好的为用户服务。下一步可以考虑加入收集用户推荐结果反馈和反馈结果分析功能，为进一步优化模型和提高推荐质量服务。

1. 协作关系网络探究

目前仅对协作关系网络进行了简要的探究和应用，后续通过深入学习网络科学理论，可以进一步分析协作关系网络的生长、演变状况以及群组的生成发展机制；通过对网络社团结构的分析和检测，可以分析网络社团的构成情况；通过分析节点度的分布、节点强度等因素，可以进一步推荐核心作者和检测挂名作者等。除了提及到的探索方向之外，还有众多的未知的方向值得探索，但是以上任何一个方向都具有进一步分析和探究的意义。

1. 应用的深化

目前合作者推荐系统仅仅只是对给定的作者进行潜在合作者推荐，我们可以深化应用，通过推荐系统把该平台打造成科研合作者的交友平台，使得科研人员之间除了项目协作交流之外还可以进行在线交流，极大的促进了科研人员之间的交流，有助于新的科研成果、科研创新和研究方向的产生。

# 致 谢

弹指一挥间，时光如白驹过隙，转眼间充实而又忙碌的三年研究生生涯即将结束。在这三年的学习生活中，有过喜悦、也有过悲伤、有过成功、也有过挫折、有过付出、也有过回报。研究生期间不仅积累了专业知识，还学会了对人生的思考，不仅开拓了视野，还学会了为人处世，这些都将成为宝贵的财富，陪伴着我。在此，我衷心的感谢这三年来默默支持、关心和鼓励我的人。

首先，要感谢我的导师王玉明副教授，本论文在是王老师悉心指导下完成的。王老师治学严谨、待人真诚、学识渊博，不仅在学习上给予指导，在生活中也给予我莫大帮助和支持。王老师教会了我如何思考，如何做事，如何沟通，这些让我终生受益。在此，向王老师表示由衷的感谢。

感谢曾经帮助过我的老师和同学，在我最迷茫的时候，有你们的支持和鼓励，使得我能够不断前行。感谢实验室和宿舍的小伙伴们，他们是冯哲奇、彭亮、张楠、刘甲、林晓佳、冯子健、黄骏、姚雨涛、欧霄、钟远维、汪贝贝、黄钊伟、涂云、黄山、佘健夫等。你们在生活上给予我莫大的包容，在学习上给予我很大的帮助和支持，是你们使我这三年的研究生生活丰富多彩，祝愿大家在以后的日子一帆风顺。

感谢一直支持我的朋友们，他们是汤显才、熊兵兵、刘浩、于国星、张弛、徐江伟、张陶盛、张毛等，虽有大家身处天南地北，但是在我遇到困难和挫折的时候，你们的鼓励和支持让我有了不断前行的动力。

最后，要感谢我的家人，你们给了我最为无私的爱和无条件的支持。感谢你们在二十多年里对我的关怀、照顾和包容，亲情是至高无上的，值得我一辈子珍惜，因为你们，我才有更大的勇气和信心不断前行。希望你们身体健健康康，快乐幸福每一天。

# 参考文献

1. Grishman, Ralph. “Information Extraction.” IEEE Intelligent Systems 30 (2015): 8-15.
2. Cui, Lei et al. “Neural Open Information Extraction.” ACL (2018).
3. Hass, Richard W.. “Semantic search during divergent thinking.” Cognition 166 (2017): 344-357.
4. Hoffmann, Michael et al. “Benchmarking question answering systems.” Semantic Web 10 (2019): 293-304.
5. Shi, Baoxu and Tim Weninger. “Open-World Knowledge Graph Completion.” AAAI (2018).
6. Chinchor, Nancy and Elaine Marsh. “Appendix D: MUC-7 Information Extraction Task Definition (version 5.1).” MUC (1998).
7. Doddington, George R. et al. “The Automatic Content Extraction (ACE) Program - Tasks, Data, and Evaluation.” LREC (2004).
8. Gábor, Kata et al. “SemEval-2018 Task 7: Semantic Relation Extraction and Classification in Scientific Papers.” SemEval@NAACL-HLT (2018).
9. Verhagen, Marc et al. “SemEval-2007 Task 15: TempEval Temporal Relation Identification.” SemEval@ACL (2007).
10. Hendrickx, Iris et al. “SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations between Pairs of Nominals.” SemEval@ACL (2010).
11. Miller, Scott et al. “A Novel Use of Statistical Parsing to Extract Information from Text.” ANLP (2000).
12. Kambhatla, Nanda. “Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Information Extraction.” ACL (2004).
13. Moncecchi, Guillermo et al. “A survey of kernel methods for relation extraction.” (2010).
14. Zhao, Shubin and Ralph Grishman. “Extracting Relations with Integrated Information Using Kernel Methods.” ACL (2005).
15. Bunescu, Razvan C. and Raymond J. Mooney. “Subsequence Kernels for Relation Extraction.” NIPS (2005).
16. Culotta, Aron et al. “Integrating Probabilistic Extraction Models and Data Mining to Discover Relations and Patterns in Text.” HLT-NAACL (2006).
17. Mintz, Mike et al. “Distant supervision for relation extraction without labeled data.” ACL/IJCNLP (2009).
18. Min, Bonan et al. “Distant Supervision for Relation Extraction with an Incomplete Knowledge Base.” HLT-NAACL (2013).
19. Angeli, Gabor et al. “Combining Distant and Partial Supervision for Relation Extraction.” EMNLP (2014).
20. Färber, Michael et al. “Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO.” Semantic Web 9 (2018): 77-129.
21. Takamatsu, Shingo et al. “Reducing Wrong Labels in Distant Supervision for Relation Extraction.” ACL (2012).
22. Yao, Limin et al. “Collective Cross-Document Relation Extraction Without Labelled Data.” EMNLP (2010).
23. Riedel, Sebastian et al. “Modeling Relations and Their Mentions without Labeled Text.” ECML/PKDD (2010).
24. Hoffmann, Raphael et al. “Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations.” ACL (2011).
25. Surdeanu, Mihai et al. “Multi-instance Multi-label Learning for Relation Extraction.” EMNLP-CoNLL (2012).
26. Min, Bonan et al. “Distant Supervision for Relation Extraction with an Incomplete Knowledge Base.” HLT-NAACL (2013).
27. Angeli, Gabor et al. “Combining Distant and Partial Supervision for Relation Extraction.” EMNLP (2014).
28. Goldwater, Sharon. “Part of Speech Tagging.” Encyclopedia of Machine Learning and Data Mining (2017).
29. Tan, Zhixing et al. “Deep Semantic Role Labeling With Self-Attention.” AAAI (2018).
30. Collobert, Ronan et al. “Natural Language Processing (almost) from Scratch.” Journal of Machine Learning Research 12 (2011): 2493-2537.
31. LeCun, Yann et al. “Deep Learning.” Nature 521 (2015): 436-444.
32. Socher, Richard et al. “Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.” EMNLP (2013).
33. Zeng, Daojian et al. “Relation Classification via Convolutional Deep Neural Network.” COLING (2014).
34. Nguyen, Thien Huu and Ralph Grishman. “Relation Extraction: Perspective from Convolutional Neural Networks.” VS@HLT-NAACL (2015).
35. Xu, Kun et al. “Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling.” EMNLP (2015).
36. Santos, Cícero Nogueira dos et al. “Classifying Relations by Ranking with Convolutional Neural Networks.” ACL (2015).
37. Zeng, Daojian et al. “Distant Supervision for Relation Extraction via Piecewise Convolutional Neural Networks.” EMNLP (2015).
38. Miwa, Makoto and Mohit Bansal. “End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures.” CoRR abs/1601.00770 (2016): n. pag.
39. Yao, Kaisheng et al. “Depth-Gated LSTM.” CoRR abs/1508.03790 (2015): n. pag.
40. Peng, Nanyun et al. “Cross-Sentence N-ary Relation Extraction with Graph LSTMs.” Transactions of the Association for Computational Linguistics 5 (2017): 101-115.
41. Song, Linfeng et al. “N-ary Relation Extraction using Graph-State LSTM.” EMNLP (2018).
42. Lin, Yankai et al. “Neural Relation Extraction with Selective Attention over Instances.” ACL (2016).
43. Wang, Linlin et al. “Relation Classification via Multi-Level Attention CNNs.” ACL (2016).
44. Du, Jinhua et al. “Multi-Level Structured Self-Attentions for Distantly Supervised Relation Extraction.” EMNLP (2018).
45. Wu, Yi et al. “Adversarial Training for Relation Extraction.” EMNLP (2017).
46. Qin, Pengda et al. “DSGAN: Generative Adversarial Training for Distant Supervision Relation Extraction.” ACL (2018).
47. Vo, Duc-Thuan and Ebrahim Bagheri. “Self-training on refined clause patterns for relation extraction.” Inf. Process. Manage. 54 (2018): 686-706.
48. Su, Yu et al. “Global Relation Embedding for Relation Extraction.” NAACL-HLT (2018).
49. Zhang, Qianqian et al. “A Review on Entity Relation Extraction.” 2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE) (2017): 178-183.
50. Kumar, Shantanu. “A Survey of Deep Learning Methods for Relation Extraction.” CoRR abs/1705.03645 (2017): n. pag.
51. Mikolov, Tomas et al. “Distributed Representations of Words and Phrases and their Compositionality.” NIPS (2013).
52. Leimeister, Matthias and Benjamin J. Wilson. “Skip-gram word embeddings in hyperbolic space.” CoRR abs/1809.01498 (2018): n. pag.
53. Shi, Weiwei and Sheng Gao. “Relation Extraction via Position-Enhanced Convolutional Neural Network.” 2017 International Conference on Intelligent Environments (IE) (2017): 142-148.
54. Wang, Jianyong et al. “Recurrent Neural Networks With Auxiliary Memory Units.” IEEE Transactions on Neural Networks and Learning Systems 29 (2018): 1652-1661.
55. He, Kaiming et al. “Deep Residual Learning for Image Recognition.” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 770-778.
56. Röder, Michael et al. “GERBIL - Benchmarking Named Entity Recognition and Linking consistently.” Semantic Web 9 (2018): 605-625.
57. “Software - The Stanford Natural Language Processing Group.” Stanford.Edu, 1 Jan. 2014, nlp.stanford.edu/software/.
58. “TensorFlow.” TensorFlow, 1 Jan. 2019, www.tensorflow.org/.
59. “Neo4j Graph Database Platform.” Neo4j Graph Database Platform, 19 Oct. 2017, neo4j.com/.
60. Bostock, Mike. “D3.Js - Data-Driven Documents.” D3js.Org, 1 Jan. 2013, d3js.org/.
61. Ding, Boyang et al. “Leveraging Text and Knowledge Bases for Triple Scoring: An Ensemble Approach - The BOKCHOY Triple Scorer at WSDM Cup 2017.” CoRR abs/1712.08356 (2017): n. pag.
62. Wikipedia Contributors. “Knowledge Base.” Wikipedia, Wikimedia Foundation, 17 Feb. 2019, en.wikipedia.org/wiki/Knowledge\_base.
63. Zhang, Qi et al. “Adaptive Co-attention Network for Named Entity Recognition in Tweets.” AAAI (2018).
64. Sil, Avirup et al. “Neural Cross-Lingual Entity Linking.” AAAI (2018).
65. Goodfellow, Ian et al. “Deep Learning.” MITPress, 2016, www.deeplearningbook.org.
66. Ranganathan, Varun and S. Natarajan. “A New Backpropagation Algorithm without Gradient Descent.” CoRR abs/1802.00027 (2018): n. pag.
67. Gruslys, Audrunas et al. “Memory-Efficient Backpropagation Through Time.” NIPS (2016).
68. Pascanu, Razvan et al. “On the difficulty of training recurrent neural networks.” ICML (2013).
69. Dey, Rahul and Fathi M. Salem. “Gate-variants of Gated Recurrent Unit (GRU) neural networks.” 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS) (2017): 1597-1600.
70. Kanai, Sekitoshi et al. “Preventing Gradient Explosions in Gated Recurrent Units.” NIPS (2017).
71. Russakovsky, Olga et al. “ImageNet Large Scale Visual Recognition Challenge.” International Journal of Computer Vision 115 (2015): 211-252.
72. Krizhevsky, Alex et al. “ImageNet Classification with Deep Convolutional Neural Networks.” Commun. ACM 60 (2012): 84-90.
73. Agarap, Abien Fred. “Deep Learning using Rectified Linear Units (ReLU).” CoRR abs/1803.08375 (2018): n. pag.
74. Abadi, Martín et al. “TensorFlow: A System for Large-Scale Machine Learning.” OSDI (2016).
75. Alatawi, Abdulrahman et al. “Bayesian Unigram-Based Inference for Expanding Abbreviations in Source Code.” 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI) (2017): 543-550.
76. Rohekar, Raanan Y. Yehezkel et al. “Bayesian Structure Learning by Recursive Bootstrap.” NIPS (2018).

1. <http://image-net.org/challenges/LSVRC/2015/> and <http://cocodataset.org/#detections-challenge2015> [↑](#footnote-ref-1)