

电子科技大学  
UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

# 专业学位硕士学位论文

MASTER THESIS FOR PROFESSIONAL DEGREE



论文题目      基于神经网络的小样本  
关系抽取研究与应用

专业学位类别      工程硕士

学      号      201922080602

作者姓名      王   敏

指导教师      傅彦      教   授

学      院      计算机科学与工程学院

分类号 TP391.1 密级 公开  
UDC <sup>注1</sup> 004.8

# 学 位 论 文

## 基于神经网络的小样本关系抽取研究与应用

(题名和副题名)

王 敏

(作者姓名)

指导教师 傅彦 教 授  
电子科技大学 成 都  
(姓名、职称、单位名称)

申请学位级别 硕士 专业学位类别 工程硕士  
专业学位领域 计算机技术  
提交论文日期 2022 年 3 月 17 日 论文答辩日期 2022 年 5 月 11 日  
学位授予单位和日期 电子科技大学 2022 年 6 月  
答辩委员会主席 \_\_\_\_\_  
评阅人 \_\_\_\_\_

注 1: 注明《国际十进分类法 UDC》的类号。

# **Research and Application of Few-shot Relation Extraction Based on Neural Networks**

A Master Thesis Submitted to  
University of Electronic Science and Technology of China

Discipline **Master of Engineering**

Student ID **201922080602**

Author **Wang Min**

Supervisor **Prof. Fu Yan**

School **School of Computer Science and Engineering**

## 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作及取得的研究成果。据我所知，除了文中特别加以标注和致谢的地方外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示谢意。

作者签名： 王敏

日期：2022年5月30日

## 论文使用授权

本学位论文作者完全了解电子科技大学有关保留、使用学位论文的规定，有权保留并向国家有关部门或机构送交论文的复印件和磁盘，允许论文被查阅和借阅。本人授权电子科技大学可以将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、汇编学位论文。

（保密的学位论文在解密后应遵守此规定）

作者签名： 王敏

导师签名： 傅强

日期：2022年5月30日

## 摘 要

随着大数据技术的快速发展，信息抽取通过神经网络模型将信息密度低的非结构化数据信息挖掘形成准确的结构化信息，对大数据技术研究具有重要意义。实体关系抽取属于信息抽取任务中必不可少的一环，近年来引起越来越多的自然语言处理研究人员的关注。关系抽取主要是对非结构化数据中的语义知识信息进行学习，再利用学习到的语义知识对海量的非结构化数据如文本信息进行实体关系抽取，将这些非结构化数据转变为结构化的关系数据，来支持知识库、问答系统、信息检索等实际应用。但是在许多实际应用场景中，并没有足够的数据进行关系抽取训练，且对于一些有足够样本的领域，也存在关系标注成本过高的问题，因此基于小样本学习的关系抽取研究具有重要意义。本文的研究内容如下：

(1) 对使用上下文相关的预训练模型 BERT 和静态预训练模型 Glove 作为词嵌入编码器的模型复杂度进行定量分析，从理论和实验上得出了上下文相关预训练模型的浮点计算量。在此基础上改进使用 Glove 作为词嵌入编码器的模型，提出可训练的数据增强网络层和上下文相关采样方式使得简单神经注意力元学习器 SNAIL 作为句子特征分类器在使用 Glove 作为词特征编码器相比使用 BERT 时损失少量准确率，在 FewRel 上达到 75.71%，但是大幅度提升了模型前向传播速度；说明了将小样本关系抽取应用到实际系统中还需要解决当支撑集中没有查询样本类型时的问题，对现有的简单神经注意力元学习器进行改进使其具有双向结构，并且实验表明本文提出的结构能够提高辅助标注的准确率。

(2) 将本文模型应用到了精确的知识图谱构建系统中，分析说明了该系统的主要应用场景，说明了系统的功能模块和设计实现。并且使用该系统进行关系抽取智能辅助标注实验，实验表明本系统能够辅助人工构建精确结构化知识库准确率从 92.2% 提升到 99.5%。

在实际应用中，本文得到的知识图谱实际上属于比较好的标注样本，但是本文还尚未对其作出充分的应用，未来的研究可以探讨如何将知识图谱输入到模型中，来达到进一步提升模型能力的目的。

**关键词：**关系抽取，小样本，预训练模型，元学习器

## ABSTRACT

With the rapid development of big data technology, information extraction is of more and more great significance. One important reason is that it can mine accurate structured information from unstructured data with sparse information through neural network model. Entity relation extraction, which has attracted more and more attention from natural language processing researchers in recent years, is an essential part of information extraction. Relationship extraction is mainly to study semantic knowledge information in unstructured data. And then it uses the knowledge learned to extract entity relation in the vast amounts of unstructured data such as text information. In other words, it translates the unstructured data into structured relational data to support knowledge base, question answering system, information retrieval, etc. However, in many practical application scenarios, there is not enough data for models to train. Even for some fields with enough samples, there is also the problem that the cost of relationship annotation is too high. Therefore, the study of relationship extraction based on few-shot learning is of great significance. The research content of this thesis is as follows:

(1) Quantitative analysis is made on the model complexity of word embeddings encoder using BERT and Glove to obtain the floating points operations of context-dependent pre-training model theoretically and experimentally. BERT is a context-dependent pre-training model while Glove is a static pre-training model. On this basis, the model of word embedding encoder using Glove is improved, and the trainable data-enhanced network layer and context-dependent sampling method are proposed to make the simple neural attentional meta-learner SNAIL as a sentence feature classifier lose a little accuracy when Glove is used as word feature encoder compared with BERT. The accuracy is 75.71 percents on FewRel. But the forward propagation speed of the model is highly improved. It shows that the application of small sample relation extraction to the actual system needs to solve the problem when there is no query sample type in the support set. The existing simple neural attentional element learner is improved to make it have a bidirectional structure, and the experiment shows that the structure proposed in this thesis can improve the accuracy of auxiliary annotation.

(2) This thesis applies the model to the accurate knowledge graph construction system, analyzes and explains the main application scenarios of the system, and explains

the function modules and design implementation of the system. The experimental results show that the system can assist manual construction of an accurate knowledge graph, the accuracy of which increases from 92.2% to 99.5%.

In practical application, the knowledge graph obtained in this thesis is actually a relatively good annotation sample, but it has not been fully applied in this thesis. Future research can explore how to input the knowledge graph into the model to further improve the model capability.

**Keywords:** Relation Extraction, Few-shot Learning, Pre-training Model, Meta-Learner

## 目 录

<b>第一章 绪 论</b>	1
1.1 研究工作的背景与意义	1
1.2 国内外研究现状	2
1.2.1 关系抽取的发展及主要方法	3
1.2.2 基于神经网络的小样本关系抽取方法	5
1.3 本文的主要工作	6
1.4 本文的结构安排	6
<b>第二章 相关理论基础</b>	8
2.1 神经网络理论基础	8
2.1.1 单层神经网络	8
2.1.2 全连接层	9
2.1.3 注意力机制	9
2.1.4 激活函数	10
2.1.5 神经网络归一化	13
2.1.6 反向传播	14
2.1.7 优化器	15
2.2 神经网络进行关系抽取理论基础	16
2.2.1 嵌入表示	17
2.2.2 神经网络进行关系抽取	19
2.2.3 softmax 函数	21
2.3 小样本学习基础	21
2.3.1 小样本学习基本概念	21
2.3.2 基于小样本学习的关系抽取	22
2.4 相关数据集	24
2.5 本章小结	24
<b>第三章 小样本条件下关系抽取研究</b>	25
3.1 基于简单神经网络小样本关系抽取研究	25
3.1.1 引言	25
3.1.2 算法设计	26
3.1.3 实验设计与结果分析	37



3.2 基于双向简单神经注意力元学习器模型的关系抽取研究 .....	45
3.2.1 引言 .....	45
3.2.2 算法设计 .....	46
3.2.3 实验设计与结果分析 .....	48
3.3 本章小结 .....	51
<b>第四章 小样本条件下的关系抽取系统 .....</b>	<b>52</b>
4.1 引言 .....	52
4.2 系统设计 .....	53
4.2.1 流程分析 .....	53
4.2.2 用例分析 .....	55
4.2.3 系统功能设计 .....	56
4.3 系统实现 .....	56
4.3.1 系统环境和系统功能界面 .....	58
4.3.2 人工标注实验 .....	59
4.4 本章小结 .....	60
<b>第五章 全文总结与展望 .....</b>	<b>61</b>
5.1 全文总结 .....	61
5.2 后续工作展望 .....	61
<b>致 谢 .....</b>	<b>63</b>
<b>参考文献 .....</b>	<b>64</b>

## 第一章 绪论

关系抽取作为构建结构化知识库（如知识图谱）的重要环节，在自然语言处理领域尤为重要。但是在一些实际的应用场景下，比如医疗领域，材料领域等，收集到充足的样本来用于训练关系分类模型十分困难。因此，仅需要少量训练样本的小样本学习近年来逐渐出现在了关系抽取的研究中。本章节作为论文的绪论部分，首先介绍了小样本关系抽取的背景以及意义，接着从小样本学习的角度介绍了基于小样本学习的关系抽取研究现状。最后整理了论文涉及到的主要工作内容，并且对论文的结构安排进行大致说明。

### 1.1 研究工作的背景与意义

近年来大数据技术有着广泛研究与应用，但是互联网中的大部分信息通常是以半结构化的如百科数据或者非结构化的文本数据来呈现，通过文本分析模型将这些信息密度低的数据提炼抽取来获取高质量且精确的结构化信息对大数据应用具有重要意义。信息抽取技术在这种背景下得到了快速发展，信息抽取首先识别文本中的实体信息，进而使用关系抽取获取两个或者多个实体间的关系，因此关系抽取在信息抽取中不可或缺。通过对大量数据进行关系抽取，可以将无结构文本转化为有结构且格式统一的关系对，为下游任务如构建知识图谱、个性推荐系统、信息搜索等提供支持。其中知识图谱在关联查询，自然语言理解，自动客服系统，逻辑推理中都有广泛的应用。同时，关系抽取对文章阅读理解、文章主要内容生成等实际应用有着重要意义，具有相当的应用前景。但是在许多应用场景中，并没有足够的数据进行关系抽取训练，且对于一些有足够样本的领域，也存在关系标注成本过高的问题，因此如何对这些场景的非结构化数据进行关系抽取有着重要的研究意义。

当前应用在关系抽取任务的主流深度学习模型依赖大量监督数据作为驱动，使得模型适应新样本的能力，也就是泛化能力依赖监督数据的质量以及数量。Wang 等提出<sup>[1]</sup>，当标注数据不足时，简单的对模型加正则项不能提高模型的泛化能力。虽然对模型加正则项可以降低模型的过拟合程度，但是正则项并不能为模型提供更多的标注数据，也就不能提供有效的额外监督信息。

最开始，Mintz 给出了远程监督学习方法来一定程度上缓解了标注数据量不足的问题<sup>[2]</sup>。Mintz 提出假设：“如果一对实体在知识库中存在某种关系，那么包含这对实体的语句大概率也包含这种关系”。但是，其实这是一种将文本中的实体与

知识库中有结构数据中实体对齐来对数据进行自动标注的方法，达到一种自动标注数据的目的，这个方法仍然有些不足。Wang 等<sup>[3]</sup>指出，一些领域数据中实体对和关系存在长尾现象，即部分实体出现的句子资料很少，即便使用远程监督标注之后实体对也不足。Han 等<sup>[4]</sup>指出该方法虽然能够有效的利用更多的数据，但是不可避免地引入了噪声数据。如图1-1所示，句子 3 并不能得到”iPhone is a product of Apple Inc.”这一关系，句子 3 即是远程监督标注数据时引入的噪声数据。

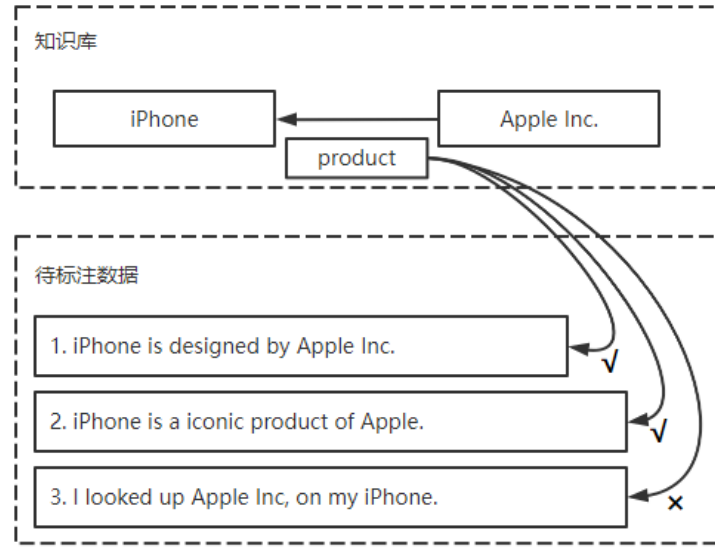


图 1-1 远程监督模型引入噪声

相比较远程监督这种辅助标注的思想，人类擅长通过极少量的样本判断一种新的事务，比如给小孩子两张照片，一张是金丝猴，一张是熊猫，即使这个小孩子以前从来没有看见过这两种动物，当这个小孩子看见熊猫时，也能够识别出该动物是熊猫。人类这种快速学习的能力启发了相关科研人员，希望神经网络模型在学习了某些类别的有标注样本后，能够在这种学习过程中学习到“学习的能力”，进而对于新的分类问题，只需要少量的样本就能快速训练，在新类别的样本上分类也有很好的效果，这就是小样本学习（Few-shot Learning）的思想。

综上所述，对于基于神经网络的小样本关系抽取研究对于实际的应用场景，尤其是数据稀缺领域有着重要的意义。

## 1.2 国内外研究现状

如何能够将深度学习应用到样本数量不够的场景下有着重要研究意义，国内外的研究人员也都提出过各种解决方案。本节主要介绍了关系抽取的发展，以及研究者们如何在样本数量不足的情况下进行关系抽取，最后引出并且介绍了小样

本关系抽取的研究发展与现状。

### 1.2.1 关系抽取的发展及主要方法

关系抽取任务第一次是在 1998 年的消息理解会议（Message Understanding Conference, MUC）被美国国防高级研究计划局单独作为评测任务提出。该任务的目的是从纽约时报部分新闻中提取实体关系，包含 Location\_of、Employee\_of 和 Product\_of 三种类型<sup>[5]</sup>。2007 年语义评估（Semantic Evaluation, SemEval）会议在评测任务中设置了 7 种实体关系类型。在 2010 年 SemEval 会议的第 8 个评测任务中，实体关系类型增加到了 10 种<sup>[6]</sup>。虽然以上会议促进了实体关系抽取的研究与进步，但是它们所使用的评测语料具有非常依赖手工标注的缺点。这样标注虽然准确率很高，但是无法提供大规模材料，并且当时训练得到的模型泛化能力比较差。后来，维基百科和在线数据库 Freebase 等规模巨大的百科知识库的出现为标注数据提供大量的语料支持，开放领域的关系抽取研究逐步发展，并且这些知识库具有比较优秀的跨领域性和规模性。

按照关系抽取的发展历程，关系抽取的方法主要有三种，或者分为三个阶段。早期的主要是基于规则和字典匹配的方法，后来随着发展又有基于机器学习的方法以及近年来基于神经网络模型的方法。

基于规则的方法主要是通过人工构建语法和语义规则。这些已定义规则主要由单词、单词的词性和简单语义规则组成的模式集组成。对预处理的语句片段进行模式匹配，完成关系分类。基于字典的方法利用字典，依靠字符串的匹配与否来识别文本中的实体，依靠字典中动词之间结构和关系区分关系类型。Miller 等人对实体信息词汇化，利用概率分布上下文无关的语法解析器生成规则用于关系抽取<sup>[7]</sup>。吴明智等人在医学领域相关的任务中使用模式匹配方法来进行关系抽取<sup>[8]</sup>。基于规则和字典匹配的关系抽取方法在构建规则和字典时要求相关人员对于相关领域有深入的了解。在特定领域和数据量比较小的情况下，基于字典匹配和规则的关系抽取方法取得了一定的成功，但是存在有召回率较低、不能移植到其它领域和标注语料成本高等问题。

基于机器学习的关系抽取方法主要包括有监督学习，无监督学习以及介于二者之间的半监督学习。在监督学习方法中，关系抽取问题被看作一个多分类问题，需要知道语料库中所有可能的关系类型作为标签，并利用手工标注数据来构建训练集，用标注数据训练得到分类器预测和判断新的标注实体及其关系。甘立新等人整合了词汇、实体、句法和语义等特征来补充实体间关系特征<sup>[9]</sup>，对比“基本特征”和“基本特征 + 句法语义特征”两种特征提取方法实验表明，后者在准确率

和召回率，F1 值均比前者有所提高。监督学习存在标注关系成本高的问题，后续就有学者研究利用少量的标注语料库或数据库进行关系抽取，这实际上也是本文要改善的要点。基于半监督学习的关系抽取主要是为了降低人工成本，根据关系类型，通过人工标注初始化质量高的少量样本作为训练语料，再对模式进行迭代来生成关系数据集，期间还需要人工辅助校验，常见算法有有放回的多次抽样 Bootstrapping 算法<sup>[10]</sup>、标签传播算法<sup>[11]</sup>等。Brin 首先使用了基于 Bootstrapping 的方法建立 Dirpe 系统实现对实体关系的抽取<sup>[12]</sup>。张立邦等运用 Bootstrapping 在中文电子病历的实体关系挖掘任务中进行实体关系抽取来获得数量比较多的医疗领域知识<sup>[13]</sup>。无监督学习的实体关系抽取则不需要提前准备标注样本，首先使用聚类算法将实体对及其上下文特征向量距离较近的聚合成一簇，然后对每簇进行分类，即是聚类之后再进行分类，然后选择有代表性的标签来标记聚类的簇，常见算法有 K-mean 算法。Hasegawa 等人首次基于无监督学习进行关系抽取，但是存在相似性阈值定义困难、简单按频率选择关系特征词忽略了噪声影响等问题<sup>[14-15]</sup>。

传统的自然语言处理过程依赖标注工具标注数据，得到的数据集即使存在少量标注错误，这些错误在后续迭代中会被一层一层放大，使得算法的准确性下降。神经网络模型可以通过学习提取数据更有价值的特征来降低输入数据的维度、将输入变为连续的词向量组合形成更抽象的高维向量表示语句信息。基于深度学习的标注流程通常是先通过人工标注获得有标签的文本语料，然后对标注语料使用如词向量、位置向量、word2vec 模型得到的词嵌入（embedding）表示词的语义。将特征向量作为神经网络模型的输入。通过特征提取后经 softmax 将特征映射到对应标签的概率，最终概率最大的即是该实体对关系。使用深度学习进行关系抽取的方法能够按照流程不同分为两种类型，分别是流水线学习方法和联合学习方法<sup>[16]</sup>。其中，流水线学习方法是先标注实体，然后对实体对进行关系分类；联合学习方法是在实体识别的同时，实体对的关系类型就确定下来。使用到的深度学习算法主要是基于现有的模型如卷积神经网络（Convolution Neural Network, CNN）、循环神经网络（Recurrent Neural Network, RNN）、长短期记忆模型（Long Short Term Memory, LSTM）、Transformer 等改进输入特征或某些神经网络层结构，比如增加不同特征、结合不同类别注意力机制算法、结合概率模型条件随机场（Conditional Random Field, CRF）和引入依存树来挖掘更深层次语义信息来提升模型的性能。也有学者考虑到图在处理异构数据具有其特别的优势，结合图卷积神经网络（Graph Neural Network, GNN）进行关系抽取。Socher 等人首次将矩阵递归神经网络模型 (MV-RNN) 应用到自然语言处理领域<sup>[17]</sup>。Sundermeyer 等人提出在 LSTM 中通过构建特殊的记忆单元来存储重要的信息来判断句子中距离较远

的单词之间的关系，对于关系抽取任务中句子内实体间隔远的问题有所改善<sup>[18]</sup>。图卷积神经网络（Graph Convolutional Network, GCN）的提出实现了在语法树上的卷积操作，也为处理具有图结构的数据提供了新的研究方向。Zhang 等提出了一种利用 GCN 修剪包含句子中词语间的依存关系的句法依存树<sup>[19]</sup>，实验表明，该方法可以得到更深层次的语义信息。

### 1.2.2 基于神经网络的小样本关系抽取方法

相比于传统的机器学习和神经网络模型，人类有较强的快速学习新概念的能力，研究人员希望创建一种模型具有人类快速学习新概念的能力。Li 等首先提出单样本学习（One-Shot Learning），利用 Bayes 模型实现每个样本只有一个样例时学习。

元网络（Meta Networks, MetaNet）通过处理比神经网络模型参数信息更高阶的神经网络模型元信息来快速参数化基础神经网络以进行快速概括<sup>[20]</sup>，从而灵活地根据任务产生不同的神经网络模型，MetaNet 包括两个学习模块分别是元学习器和基学习器。元学习器通过跨任务进行操作来生成快速权重，基学习器捕获任务目标执行每个任务，生成的快速权重再被集成到两个学习器中来矫正学习器的误差。该模型在小样本学习任务 Omniglot 和 Mini-ImageNet 中可以超过人类的水平，Omniglot 是一个由五十多种字符表创建的手写字符识别数据集<sup>[21]</sup>，Mini-ImageNet 包含 100 种图片，每类图片有 600 个样本。

Satorras 等用 GNN 的节点间特征传递特性将标签信息从有标签的样本节点传播到无标签的节点上<sup>[22]</sup>。这种信息传播可以理解为对输入图像和标签确定的图形模型的后验推理。Satorras 等提出的模型在 Omniglot 和 Mini-ImageNet 上都达到了当时任务的最好效果。Misturras 等提出了简单神经注意力元学习器（SNAIL）<sup>[23]</sup>，使用时序卷积和注意力的组合，时序卷积主要收集上文信息，而注意力机制则用于筛选上文中的重要信息。二者结合可以大范围的筛选重要信息，在小样本学习任务中也取得了不错的效果。

小样本学习是近几年才应用到自然语言领域，Han 等在 2018 年首次利用小样本学习方法进行关系抽取<sup>[24]</sup>。类似 Omniglot 等在图像领域的小样本数据库，Han 等提出了 FewRel，专用于自然语言处理中关系抽取的小样本数据库。并且使用了上述小样本学习方法包括元网络，图神经网络，SNAIL 和原型网络（Prototypical Networks）<sup>[25]</sup> 到自然语言处理领域，随后小样本关系抽取模型的研究越来越多。

Soares 等受到 word2vec 中 Harris 分布式假设的启发<sup>[26]</sup>，即是“如果两个词的前后文相似，那么这两个词也是相似的”。想利用上下文将关系编码成一个长度为

常量的向量，从而将关系抽取问题等价为两个子问题，一是如何对包含实体关系的语句进行编码，二是合理输出一个定长向量。对于第一个任务，作者借用了预训练模型 BERT 对文本进行编码并且对比了几种输入方式的影响；对于第二个任务提出了利用“完形填空”的训练方法来预训练任务不可知的模型。

Gao 等在 Han 等的基础上选出表现比较好的 GNN 与 Prototypical Network 作为基准<sup>[27]</sup>，并且提出 BERT-PAIR 结构，该结构将需要分类的目标句子与样本句子一一组合成对，再成对输入到 BERT 中，利用 BERT 预训练模型中大量语言知识对每组数据相似度进行打分，BERT-PAIR 在 FewRel 数据集上取得最好的效果。随即根据实际应用，提出了在小样本关系抽取应用领域，还需要“以上都不是”的选项以便于从几十类关系中进行多轮次筛选出正确的标签和更好的跨领域的适用性。在最新的任务上，所有现有的小样本学习方法效果都有所下降。

### 1.3 本文的主要工作

本文主要研究基于神经网络的小样本关系抽取，说明简单神经网络的优势并且如何在使用简单神经网络模型的条件下能够有比较高的准确率，并且将简单神经网络应用到实际的关系抽取应用系统中。

(1) 分别对静态预训练模型代表 Glove 和上下文相关预训练模型代表 BERT 的模型复杂度浮点运算量进行定量理论分析，并且实验验证了分析结果。对简单注意力元学习器 (SNAIL) 进行改进，改变其采样方式和提出可训练的数据增强网络层算法。

(2) 受到自然语言处理模型中常用的双向结构启发，在简单神经网络注意力元学习器的基础上改进模型，设计了双向简单神经网络注意力元学习器。将小样本关系抽取应用到实际的关系抽取系统中，在实际应用系统中，小样本关系抽取并不是只从支撑集的几个类型关系中选择一个，而是从几十种关系类型中选择标签，所以需要分批次的进行识别，对于每一批次，还要判定关系属于是“以上都不是”。通过实验证明了本文设计的系统用来构造精确的结构化知识库，并且准确率可以达到 99%。

### 1.4 本文的结构安排

本文一共分为五个章节，具体的内容和结构安排如下：

第一章为绪论部分。主要介绍了关系抽取以及关系抽取在应用领域会遇见标注样本少以及跨领域适应性的问题，因此针对如何解决样本比较少的关系抽取研究具有重要意义。然后简略的介绍了研究人员是如何针对这些问题作出相应的研

究，在此基础上对本文的主要工作类容进行说明，并且介绍论文的结构安排。

第二章为相关理论基础。主要对神经网络相关基础和小样本学习相关基础进行了介绍。神经网络基础部分说明了最简单的单层神经网络，常用的全连接层，注意力机制基础，几种常见的激活函数，网络层归一化，反向传播和神经网络优化器，基本包括了神经网络训练过程中的必备知识。同时还说明了小样本学习的一些基本概念和小样本学习的几个重要模型包括图神经网络，原型神经网络。

第三章为小样本条件下关系抽取研究。首先介绍了如何使用不同的词嵌入解码器和使用 PCNN 对词特征解码提取句子级特征向量，并且使用简单神经注意力元学习器进行小样本关系抽取的算法。在此算法的基础上提出了可训练的数据增强网络层以及上下文相关的采样方式。然后说明本文受到了自然语言处理中常用的双向结构启发，将简单神经注意力元学习器优化为双向结构。本文在公开数据集上进行实验，说明了改进的效果以及影响。

第四章为小样本关系抽取模型的应用系统。梳理了实际应用中的工作流程，设计了系统功能模块和展示了一些系统界面，评估了系统与已有系统的改进，实验说明了系统能够完成精确结构化知识库的构建。

第五章为全文总结与展望。对本文第三章和第四章工作进行简单总结，分析了存在的局限性以及未来研究工作的展望。



## 第二章 相关理论基础

在研究基于神经网络的小样本关系抽取研究之前，本章首先对小样本关系抽取中所涉及到的一些基本理论和概念进行说明，包括神经网络的部分比较重要的理论基础，神经网络应用于关系抽取的相关理论基础和基于小样本学习的关系抽取理论基础。

### 2.1 神经网络理论基础

本小节主要介绍神经网络的基本概念，包括神经网络的表示方法，常用的网络层，激活函数的作用和常见激活函数，不同神经网络归一化的作用，网络的反向传播以及优化器相关。

#### 2.1.1 单层神经网络

Rosenblatt 等最早提出了只有一层神经元的单层神经网络<sup>[28]</sup>，被称为感知机，是第一种可以进行学习的人工神经网络。Rosenblatt 等定义的感知机示意图如图2-1(a)所示。其中  $f$  表示非线性函数，发展到后来  $f$  被其它更多的不同性质的激活函数所替代。

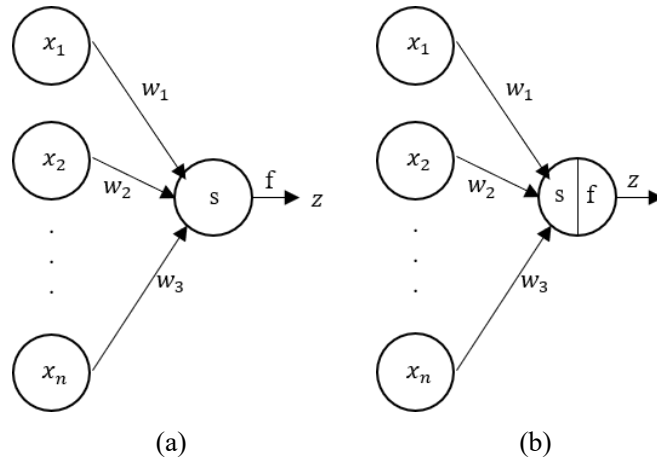


图 2-1 (a) 单层神经网络；(b) 简化表示神经元

神经元输出  $z$  如式 (2-1) 所示。其中  $w$  表示神经元的权重， $x$  表示输入。为了方便表示，神经网络的示意图通常将神经元中的激活函数收到神经元内部表示，如图2-1(b)所示。

$$z = f(w_1x_1 + w_2x_2 + w_3x_3) \quad (2-1)$$

### 2.1.2 全连接层

全连接层（Full Connection Layer, FC）指的是神经网络的层中的每个神经元都会连接它下一层的所有神经元，如图2-2所示的神经网络就是由两层全连接层组成。

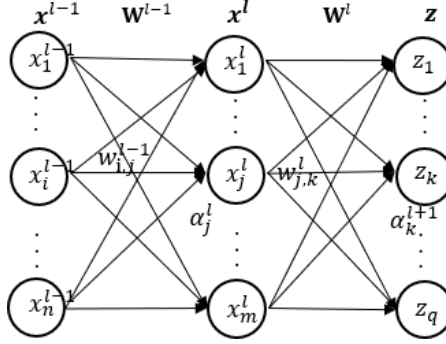


图 2-2 全连接层示意图

全连接层计算公式如式（2-2）所示。其中  $\mathbf{W} \in \mathbb{R}^{n \times m}$  表示层的权重， $\mathbf{x} \in \mathbb{R}^m$  表示每层的输入， $\mathbf{b} \in \mathbb{R}^n$  表示层的偏置。全连接层主要用于将特征融合并且映射到指定的维度，常用的就是神经网络最后输出要求特征映射到类型数目维度大小。比如需要对手写数字识别 0-9 总共 10 类标签，那么最后就映射到 10 维上。

$$\mathbf{x}^l = f(\mathbf{W}^l \mathbf{x}^{l-1} + \mathbf{b}^{l-1}) \quad (2-2)$$

### 2.1.3 注意力机制

注意力机制（Attention Mechanism）的思想主要是受到了人视觉具有快速扫描并且聚焦重点目标区域并且忽略其它部分的信息的能力的启发<sup>[29]</sup>，比如在人群中快速找到一个人的能力或者在一堆图像中快速查看目标图像。如图2-3所示，人类能够快速得到三角形数量，其中一个原因是因为在获取三角形数量时，只需要注意到图中阴影部分。

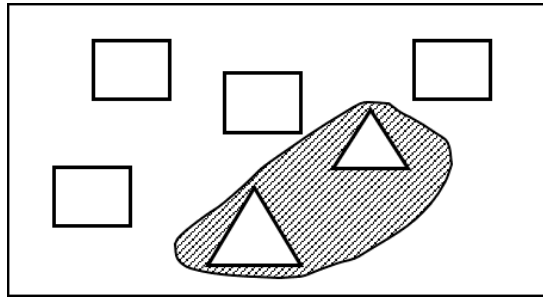


图 2-3 人类视觉的注意力

注意力函数可以定义为将查询（Query）和一组键值（Key-Value）对映射到输

出<sup>[30]</sup>，如图2-4所示。

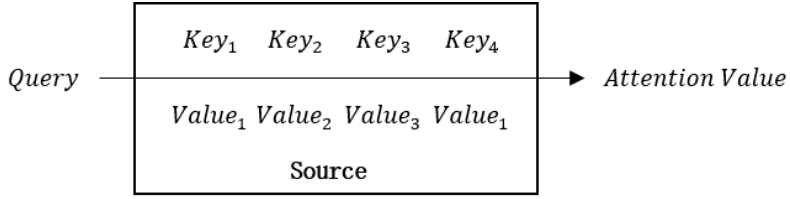


图 2-4 Query 到一组键值对的映射

其中注意力函数在 Encoder-Decoder 框架下可以表示为  $\langle \text{Source}, \text{Target} \rangle$  对，Encoder-Decoder 框架表示从源到目标的通用处理模型。其中 Query 可以看作 Target 中的单词，Source 可以看作由 Key-Value 组成。在自注意力机制（Self-Attention）中，Source 和 Target 都是从输入得到。注意力函数表达式如式（2-3）所示，其中 Q、K 和 V 分别是 Query、Key 和 Value， $l$  表示 Key-Value 对个数，S 表示相似性函数。

$$\text{Attention}(Q, \text{Source}) = \sum_{i=1}^l S(Q, K_i) V_i \quad (2-3)$$

其中相似性函数 S 可以是余弦距离、点积、链接等，如式（2-4）—（2-6），其中  $W$  为可训练参数。

$$S(Q, K_i) = Q^T K_i \quad (2-4)$$

$$S(Q, K_i) = Q^T W K_i \quad (2-5)$$

$$S(Q, K_i) = W[Q; K_i] \quad (2-6)$$

#### 2.1.4 激活函数

本小节先讨论激活函数在神经网络中的作用，再讨论几种常见的激活函数以及他们的优缺点。对于一个二维平面非线性分类问题，如图2-5中所示对图形按照形状分类，使用两层且每层只有单个神经元神经网络进行分类。

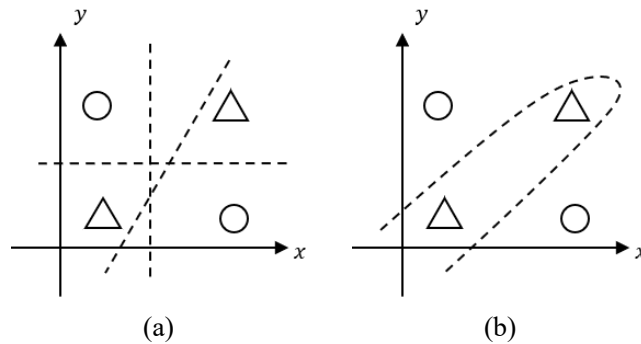


图 2-5 (a)2 维平面非线性分类问题；(b) 非线性函数才能正确分类

图2-5(a)表示没有激活函数时，该网络只能输出一个线性的曲线进行区分。此时无法完成分类任务。上述过程可以从式(2-2)推导，此时参数矩阵  $\mathbf{W}$  只有一个参数使用  $w$  表示。那么可以得到网络层第一层的表达式为式(2-7)。

$$x^1 = w^0 x^0 + b^0 \quad (2-7)$$

第二层表达式可以表示为式(2-8)。

$$y = w^1 x^1 + b^1 \quad (2-8)$$

联立式(2-7)和式(2-8)可以得到式(2-9)，解得输出  $y$  是一个线性函数，此时可以得出结论，如果没有非线性的激活函数，任意数量层数的神经网络都无法求出图2-5(a)中分类问题的解，解应该是个非线性表达式，解的示意图如图2-5(b)所示。激活函数能够将线性的表达转换为非线性函数，进而理论上可以拟合到任意的曲线，这就是激活函数在神经网络中存在的必要性。

$$\begin{aligned} y &= w^1 x^1 + b^1 \\ &= w^1 (w^0 x^0 + b^0) + b^1 \\ &= w x^0 + b \end{aligned} \quad (2-9)$$

下面讨论常见的激活函数，常用的激活函数包括 sigmoid 函数，其表达式如式(2-10)所示。

$$f(x) = \frac{1}{1 + e^x} \quad (2-10)$$

sigmoid 的导数如式(2-11)所示。

$$f'(x) = f(x)(1 - f(x)) \quad (2-11)$$

由 sigmoid 的导数式(2-11)可以看出来 sigmoid 函数的优点是可以很方便的求出其导数，缺点在于每次计算时需要进行幂运算，运算量比较大。sigmoid 和其导数如图2-6所示。由函数图像可得 sigmoid 函数可以将输入映射到  $(0, 1)$ ，缺点可以看出 sigmoid 的导数比较小，在比较深的神经网络中容易出现梯度消失的情况。

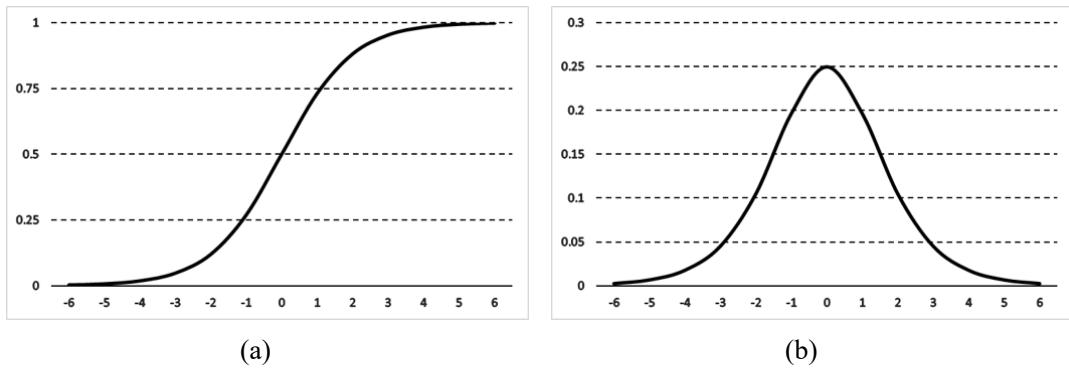


图 2-6 (a)sigmoid 曲线图；(b)sigmoid 导数曲线图

为了解决在深度神经网络学习过程中出现的梯度消失问题和运算复杂的问题，后来又提出如式（2-12）所示的线性整流函数（Rectified Linear Unit, ReLU）<sup>[31]</sup>。可以从式子中看出，相较于 sigmoid 函数，ReLU 函数计算简单，且大于 0 时导数等于 1，在深层神经网络中的链式求导中不容易造成梯度消失<sup>[32]</sup>。这两个优点使得 ReLU 更适合作为深度神经网络的激活函数。当小于 0 是导数为 0，也可用于深度学习去除样本中噪声。

$$f(x) = \max(0, x) \quad (2-12)$$

其导数是一个分段函数，如式（2-13）所示。

$$f'(x) = \begin{cases} 0 & x \leq 0 \\ 1 & x > 0 \end{cases} \quad (2-13)$$

ReLU 的函数凸显和导函数图像如图2-7所示。

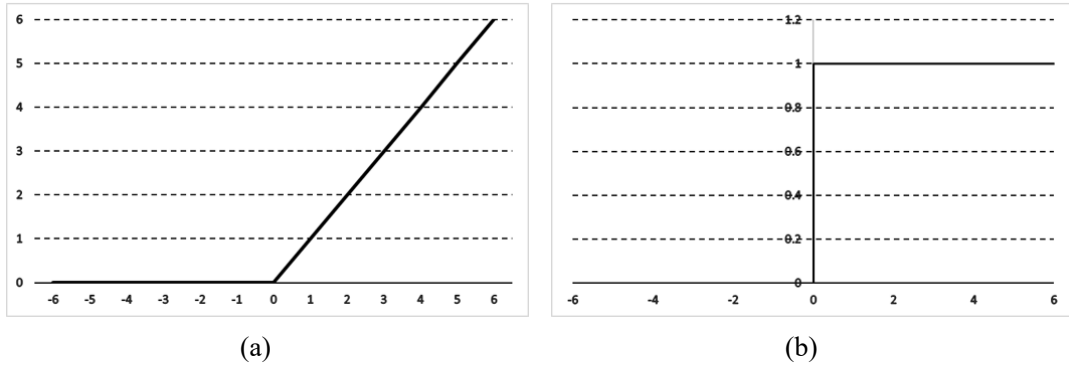


图 2-7 (a)ReLU 函数曲线图；(b)ReLU 函数的导数曲线图

带泄露线性整流函数 (Leaky-ReLU) 最早由 Mass 等提出<sup>[33]</sup>，主要目的是为了防止使用 ReLU 作为激活函数时，神经元在传播过程中出现“死亡”的问题<sup>[34]</sup>。所谓神经元“死亡”指的是当梯度为负时，激活函数 ReLU 的梯度为 0，该神经元将在后续训练中不再更新。Leaky-ReLU 的函数形式为式（2-14），对比式（2-13）可以设置  $\alpha$  为非零数来避免神经元“死亡”。

$$f(x) = \begin{cases} \alpha x & x \leq 0 \\ x & x > 0 \end{cases} \quad (2-14)$$

其中  $\alpha$  为一个比较小的常数。Leaky-ReLU 的导数为式（2-15）。

$$f'(x) = \begin{cases} \alpha & x \leq 0 \\ 1 & x > 0 \end{cases} \quad (2-15)$$

如图2-8所示为 Leaky-ReLU 函数的图像和导数图像，后来研究人员又提出了将  $\alpha$  作为模型的一个训练参数，即是所谓的参数线性修正单元 (Parametric Rectified

Linear Unit, PReLU)。

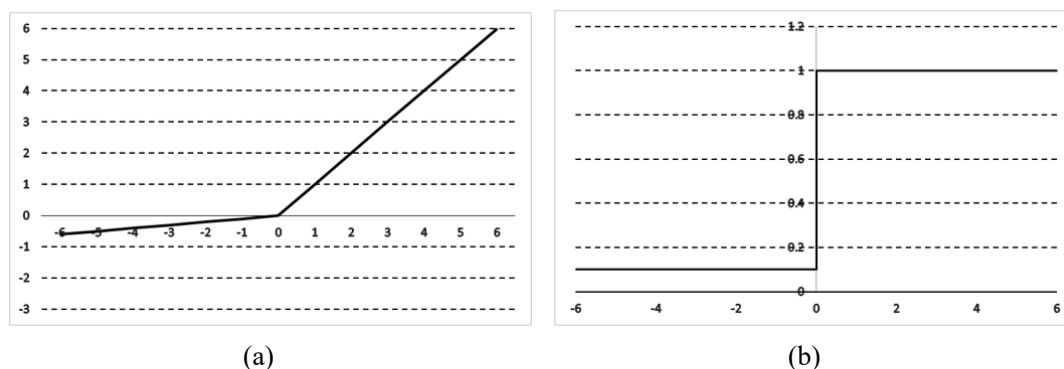


图 2-8 (a)Leaky-ReLU 函数曲线图; (b)Leaky-ReLU 函数的导数曲线图

### 2.1.5 神经网络归一化

归一化 (Normalization) 被广泛应用于神经网络之中, Ioffe 等提出批归一化 (Batch Normalization, BN) 使得各层分布相同<sup>[35]</sup>, 由第二章 sigmoid 的导函数图像可以看出, 当导数在中间位置时函数变化明显, BN 可以将偏离的数据分布拉回中心, 可以达到加速神经网络收敛, 并且增加激活函数效果的目的。Santurkar 等证明了 BN 层可以使损失函数的梯度变得更加平滑<sup>[36]</sup>, 从而对学习率更加不敏感并且加速收敛。图2-9可以比较好的表现这一点<sup>[37]</sup>。

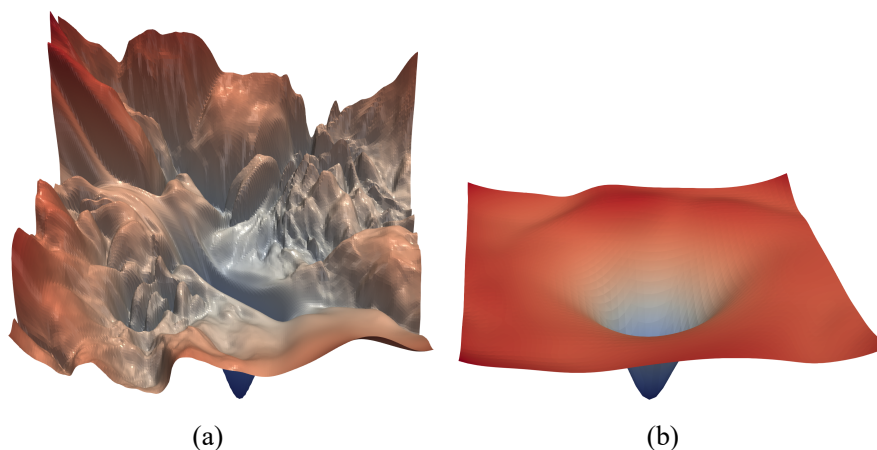


图 2-9 (a) 未使用归一化时 loss; (b) 使用归一化后 loss

BN 是根据当前批次样本的特征维度进行相当于从总体部分取批次大小的样本的归一化, 当硬件资源受到限制, 一个训练批次只能取少量样本, 每批次的样本规律不能反映总体样本规律时, BN 的效果并不好。且在循环神经网络中, 每个批次样本的长度不一致, BN 每次得到的统计结果会随着批次改变而改变。BN 的

算法如算法2-1所示。

<b>算法 2-1</b> 应用于一个最小批次的 Batch Normalizing	
<b>Input:</b>	每个最小批次的输入 $B = \{x_1, \dots, x_m\}$ , 可学习参数 $\gamma$ 和 $\beta$
<b>Output:</b>	$y_i$
<b>1 begin</b>	
	// 计算 prototype
<b>2</b>	$\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^m x_i$ $\sigma_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2$ // 计算方差
<b>3</b>	$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$ // 归一化
<b>4</b>	$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv BN_{\gamma, \beta}(x_i)$ // 放缩, 调整
<b>5 end</b>	

为了解决 BN 存在的问题, Ba 等人提出层归一化 (Layer Normalization, LN)<sup>[38]</sup>, 如图2-10所示是 BN 和 LN 的区别。BN 是对当前 batch 样本的同一纬度特征归一化, LN 是对某个样本的所有特征进行归一化。

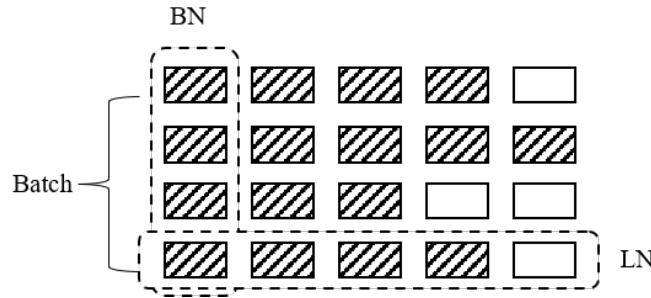


图 2-10 BN 和 LN 的区别

定义 LN 为一个函数  $LN: R^H \rightarrow R^H$ , 其中 H 是输出的特征维度。LN 如式 (2-16) 所示。

$$LN(\mathbf{z}; \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{(\mathbf{z} - \boldsymbol{\mu})}{\sigma} \odot \boldsymbol{\alpha} + \boldsymbol{\beta} \quad (2-16)$$

其中  $\boldsymbol{\alpha}$  和  $\boldsymbol{\beta}$  是可学习参数,  $\boldsymbol{\mu}$  是一个样本所有特征的均值,  $\sigma$  是一个样本所有特征的方差。

### 2.1.6 反向传播

损失函数 (loss function) 是用来评价神经网络输出的预测值与真实值不一样的程度, 反向传播是用来训练神经网络的常见方法。反向传播对神经网络中所有权重计算关于损失函数的梯度, 这个梯度用于更新神经网络中权值以最小化损失函数。如图2-2中,  $w_{ij}^{l-1}$  表示第  $l-1$  层第  $i$  个神经元链接第  $l$  层第  $j$  个神经元权重,

$\alpha_j$  表示输出为  $y_j^l$  的神经元的输入。式 (2-17) 表示对  $w_{ij}^{l-1}$  权重更新, 其中  $\eta$  表示学习率。

$$w_{ij}^{l-1} = w_{ij}^{l-1} - \eta \Delta w_{ij}^{l-1} \quad (2-17)$$

设损失函数为  $J$ , 那么  $J$  关于  $w_{j,k}^l$  的偏导数可以由链式法则得式 (2-18)。

$$\Delta w_{j,k}^l = \frac{\partial J(\theta)}{\partial w_{j,k}^l} = \frac{\partial L(\theta)}{\partial z_k} \frac{\partial z_k}{\partial \alpha_k^{l+1}} \frac{\partial \alpha_k^{l+1}}{\partial w_{j,k}^l} \quad (2-18)$$

其中  $\theta$  表示所有参数,  $\alpha_k^{l+1}$  表示输出为  $z$  网络层的输入, 如式 (2-19) 所示。

$$\alpha_k^{l+1} = \sum_{i=1}^m w_{i,k}^l \times x_i^l + b_i^l \quad (2-19)$$

$J$  关于  $w_{ij}^{l-1}$  的偏导数可以由链式法则得式 (2-20)。

$$\Delta w_{ij}^{l-1} = \frac{\partial J(\theta)}{\partial w_{ij}^{l-1}} = \frac{\partial J(\theta)}{\partial z_k} \frac{\partial z_k}{\partial \alpha_k^{l+1}} \frac{\partial \alpha_k^{l+1}}{\partial x_j^l} \frac{\partial x_j^l}{\partial \alpha_j^l} \frac{\partial \alpha_j^l}{\partial w_{ij}^{l-1}} \quad (2-20)$$

对比式 (2-18) 和 (2-20) 可以得到共同部分式 (2-21) 表示的  $\delta$ 。

$$\delta = \frac{\partial J(\theta)}{\partial z_k} \frac{\partial z_k}{\partial \alpha_k^{l+1}} \quad (2-21)$$

从上述的过程可以看出, 在计算式 (2-20) 时, 需要式 (2-18) 的部分结果, 且两者都需要得到最终结果  $J$  之后才能计算, 这种由后往前的计算被形象的被称为反向传播。

### 2.1.7 优化器

参数更新最终目的是使得损失函数能够达到全局最小, 如式 (2-17) 表示对于神经网络中的神经元权重  $w_{ij}^{l-1}$  更新来使得式 (2-18) 中损失函数  $J$  全局最小, 优化器的主要作用一是加速模型训练来更快的得到最优参数, 二是避免最终结果陷入局部最优解<sup>[39]</sup>。本小节从加快训练速度和避免陷入局部最优解角度来分别说明几种常用优化算法。

梯度下降 (Gradient Descent, GD) 算法利用基本的数学原理来计算最小损失, 但是计算量将会非常大。如果对于损失函数  $J$  可以写作期望或者连加的形式, 如式 (2-22) 所示。

$$J(w_1, \dots, w_l) = \frac{1}{n} \sum_{i=1}^n F_i(w_1, \dots, w_l) \quad (2-22)$$

此时梯度下降法参数更新公式如式子所示 (2-23), 可以看出每轮更新参数需要用到所有样本, 计算量较高。

$$w_i = w_i - \eta \sum_{i=1}^n \nabla_{w_i} F_i(w_1, \dots, w_n) \quad (2-23)$$

随机梯度下降法 (stochastic gradient descent, SGD) 每次从集合  $\{1, 2, \dots, n\}$  中随机取一个数记作  $j$ , 随机梯度下降法的参数更新公式如式 (2-24) 所示, 对比



式2-23的 GD 更新方式降低了计算量，能够加快训练速度。

$$w_i = w_i - \eta \nabla_{w_i} F_j(w_1, \dots, w_n) \quad (2-24)$$

但是一个样本并不总能反映总体的分布，小批量随机梯度下降法（mini-batch SGD）作为二者的折中，兼顾了二者的优点，即提升了训练速度，也保证了最终结果的准确性，小批量随机梯度下降法如图2-11所示。

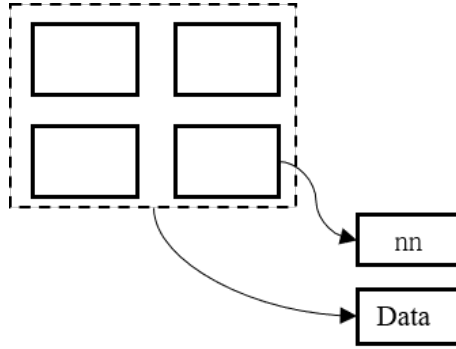


图 2-11 小批量随机梯度下降法示意图

带动量的梯度下降法（Momentum）会记录前一次梯度的影响，如式（2-25）中的  $\varepsilon \Delta w_i^{k-1}$  即是记录的上一次参数更新时的梯度。

$$\Delta w_i^k = -\eta \nabla_{w_i^k} F_j(w_1, \dots, w_n) + \varepsilon \Delta w_i^{k-1} \quad (2-25)$$

当遇到局部最优点时，积累的“动量”能够跳出局部最优，如示意图2-12所示，累计的动量有助于跳出局部最优点 A。

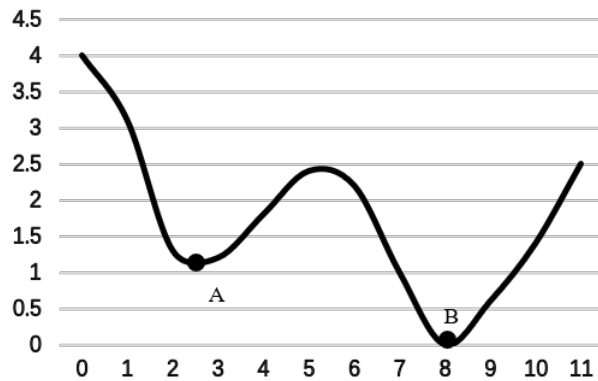


图 2-12 带动量的梯度下降法有助于跳出局部最优

## 2.2 神经网络进行关系抽取理论基础

以神经网络进行关系抽取整体的框架流程如图2-13所示。本小节的结构和图中层次基本顺序一致，先介绍了常用的词嵌入方式，再介绍常用的神经网络，然

后介绍 softmax 函数作用与性质。其中常用的数据集和数据集规模在本章最后一节也进行了说明。

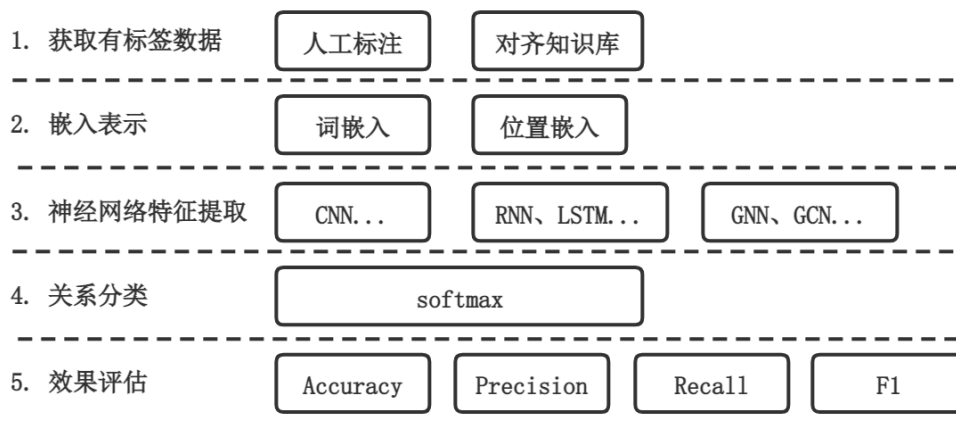


图 2-13 使用神经网络进行关系抽取框架

### 2.2.1 嵌入表示

将文本输入到神经网络中需要将词转为嵌入向量，下面介绍嵌入表示的常用方法。早期，自然语言处理常使用独热编码（one-hot embedding），独热编码用来表示离散的特征，而文本是一个一个词组成，将每个词看作一个特征，这正和独热编码的功能契合。在独热编码中，定义字典为  $D$ ，字典大小为  $N$ ，字典中的第  $i$  个词为  $w_i$ ，单词  $w$  的 one-hot 编码表示为  $C(w)$ ，使用  $N$  个比特位表示特征如式 (2-26) 所示，表示单词和字典中相等时该位为 1，其它位为 0。

$$C(w) = \{1_{w=w_i}\}_{i \in [0, N)} \quad (2-26)$$

其中  $N$  为词典的大小。假设  $N = 10$ ，词典为：

{ warsaw, modlin, airport, is, located, there, new, york, -, . }

那么独热编码使用词袋模型来提取句子特征，可以转化为嵌入表示：

1, 1, 1, 1, 1, 1, 0, 0, 1, 1

当  $N = 10000$  时，该表示中有 9992 个 0，当  $N$  更大时，特征表示过于稀疏，且词袋模型不能表示句子在词语中的位置信息。为了解决词袋模型表示引发的维度灾难问题，Mikolov 等提出 word2vec 模型<sup>[40]</sup>，word2vec 能够将词语通过低维向量表示，该想能同时包含了单词的上下文之间的信息。word2vec 只考虑到了词的局部信息，无法考虑到当前词与远距离词之间的联系。随后 Pennington 等提出 GloVe

模型表示词的特征<sup>[41]</sup>，Glove 模型中利用了共现矩阵，兼顾局部和整体的信息。GloVe 训练词向量分两步，一是统计共现矩阵，二是训练词向量。由于篇幅限制，本小节主要说明 GloVe 模型的损失函数的数学推导。

定义  $\mathbf{X}$  为共现矩阵， $\mathbf{X}_{ij}$  表示单词  $j$  在单词  $i$  上下文中出现次数。则  $\mathbf{X}_i$  是字典中所有单词在  $i$  出现的次数，如式 (2-27) 所示。 $P_{ij}$  表示单词  $j$  在单词  $i$  上下文中出现频率，如式 (2-28) 所示。定义  $P_{ij}$  和  $P_{ji}$  的比值为  $F$  如式 (2-29) 所示。

$$\mathbf{X}_i = \sum_k \mathbf{X}_{ik} \quad (2-27)$$

$$P_{ij} = P(j|i) = \mathbf{X}_{ij} / \mathbf{X}_i \quad (2-28)$$

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}} \quad (2-29)$$

经过统计，Pennington 等发现如下规律，以下简称单词  $i$ ，单词  $j$  和单词  $k$  为  $w_i$ ， $w_j$  和  $w_k$ 。当  $w_i$  与  $w_k$  相关且  $w_j$  与  $w_k$  相关，或者当  $w_i$  与  $w_k$  不相关且  $w_j$  与  $w_k$  不相关时  $F$  接近 1。当  $w_i$  与  $w_k$  相关但是  $w_j$  与  $w_k$  不相关， $F$  很大。当  $w_i$  与  $w_k$  不相关但是  $w_j$  与  $w_k$  相关， $F$  很小。如表 2-1<sup>[41]</sup> 所示。可以从经验得出 ice 和 water 相关，water 和 stream 相关， $P(\text{water}|\text{ice})/P(\text{water}|\text{stream})$  接近 1。

表 2-1 单词相关规律

概率与 F 值	k = solid	k = gas	k = water	k = fashion
$P(k \text{ice})$	$1.9 \times 10^{-4}$	$6.6 \times 10^{-5}$	$3.0 \times 10^{-3}$	$1.7 \times 10^{-5}$
$P(k \text{steam})$	$2.2 \times 10^{-5}$	$7.8 \times 10^{-4}$	$2.2 \times 10^{-3}$	$1.8 \times 10^{-5}$
$P(k \text{ice})/P(k \text{steam})$	8.9	$8.5 \times 10^{-2}$	1.36	0.96

为了拟合这种规律，作者设计了损失函数，该损失函数应该具有模板，形式如同公式 (2-30)。

$$J = \sum_{i,j,k}^N \left( \frac{P_{i,k}}{P_{j,k}} - g(\mathbf{w}_i, \mathbf{w}_j, \tilde{\mathbf{w}}_k) \right) \quad (2-30)$$

考虑到生成词向量的线性性质，自然联想到使用差值，可以得到：

$$F(\mathbf{w}_i - \mathbf{w}_j, \tilde{\mathbf{w}}_k) = \frac{P_{ik}}{P_{jk}} \quad (2-31)$$

考虑到右边值为一个比值是标量，可以得到：

$$F((\mathbf{w}_i - \mathbf{w}_j)^T \tilde{\mathbf{w}}_k) = \frac{P_{ik}}{P_{jk}} \quad (2-32)$$

考虑到左边是差右边是商，希望  $F$  函数满足性质如式 (2-33) 所示：

$$F((\mathbf{w}_i - \mathbf{w}_j)^T \tilde{\mathbf{w}}_k) = \frac{F(\mathbf{w}_i^T \tilde{\mathbf{w}}_k)}{F(\mathbf{w}_j^T \tilde{\mathbf{w}}_k)} \quad (2-33)$$

联合式 (2-29) 可得式 (2-34)，通过  $\exp$  函数将左右关联起来。

$$F(\mathbf{w}_i^T, \tilde{\mathbf{w}}_k) = P_{ik} = \frac{\mathbf{X}_{ik}}{\mathbf{X}_i} \quad (2-34)$$

容易想到满足式 (2-33) 性质的是  $\exp$  函数，即：

$$\mathbf{w}_i^T \tilde{\mathbf{w}}_k = \log(P_{ik}) = \log(\mathbf{X}_{ik}) - \log(\mathbf{X}_i) \quad (2-35)$$

考虑到式子左边具有对称性，但是右边没有对称性，为  $\mathbf{w}_i$  和  $\tilde{\mathbf{w}}_k$  各增加一个偏置  $\mathbf{b}_i$  和  $\tilde{\mathbf{b}}_k$  项得到式 (2-36)。

$$\mathbf{w}_i^T \tilde{\mathbf{w}}_k + \mathbf{b}_i + \tilde{\mathbf{b}}_k = \log(\mathbf{X}_{ik}) \quad (2-36)$$

由式 (2-30) 和 (2-36) 可得最后代价方程为式 (2-37) 所示，其中  $f(\mathbf{X}_{ij})$  为和单词在矩阵位置相关函数。

$$J = \sum_{i,j=1}^N f(\mathbf{X}_{ij})(\mathbf{w}_i^T \tilde{\mathbf{w}}_j + \mathbf{b}_i + \tilde{\mathbf{b}}_j - \log(\mathbf{X}_{ij})) \quad (2-37)$$

现在常用上下文相关预训练模型作为句子编码器，上下文相关的预训练模型可以从大量语料中获取知识，并且根据上下文动态的生成词向量。其中 BERT 作为常用的上下文相关编码器，自提出后被引用到各种自然语言处理任务。

## 2.2.2 神经网络进行关系抽取

早期，Liu 等利用 CNN 进行关系抽取<sup>[42]</sup>，Liu 等仅仅优化了独热编码表示方式，使用近义词表进行单词表示，没有充分利用词嵌入的优势<sup>[43]</sup>。Zeng 等结合了词嵌入表示和卷积神经网络<sup>[44]</sup>，本节主要介绍 Zeng 等构建的模型结构。

设句子为  $S$ ， $\mathbf{x}_i$  表示第  $i$  个单词的词嵌入， $n$  为句子的单词数量，则  $S = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ 。为了能够利用到句子中每个单词的上下文信息，定义一个滑动窗口，设窗口大小为  $w$ ，当  $w = 3$  时，句子的词特征 (Word Features, WF) 可以表示为式 (2-38)

$$WF = \{[\mathbf{x}_s, \mathbf{x}_0, \mathbf{x}_1], [\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2], \dots, [\mathbf{x}_{n-1}, \mathbf{x}_n, \mathbf{x}_e]\} \quad (2-38)$$

为了得到句子中两个实体的位置特征 (Position Embedding, PE)，将位置信息用当前单词与标注单词的位置的欧氏距离作为特征， $\mathbf{d}_1, \mathbf{d}_2$  表示当前单词与标注实体的距离，随机初始化作为训练参数。

$$PF = \{\mathbf{d}_1, \mathbf{d}_2\} \quad (2-39)$$

将前面得到的 WF 和 PF 合并得到卷积层的输入  $X^{n_0 \times t}$ ，其中  $n_0 = w \times n$ ， $w$  为窗口大小， $n$  为参数表示特征维度， $t$  为输入的 token 数量。在神经网络中，卷积是一种常用来融合特征的操作，定义卷积层操作如下所示。首先处理前两步得到的

输出。

$$\mathbf{Z} = \mathbf{W}_1 \mathbf{X} \quad (2-40)$$

其中  $\mathbf{W}_1 \in R^{n_1 \times n_0}$ ,  $n_1$  表示隐藏层的维度。为了提取句子中最有用的特征以及将句子长度对齐, 对  $\mathbf{Z}$  定义  $\max$  操作。

$$m_i = \max \mathbf{Z}(i, \cdot) \quad 0 \leq i \leq n_1 \quad (2-41)$$

其中  $\mathbf{Z}(i, \cdot)$  表示  $\mathbf{Z}$  的第  $i$  行。最终得到输出  $\mathbf{m} = \{m_1, m_2, \dots, m\}$ , 此时句子特征不再依赖句子长度。为了学习到更复杂的特征使用  $\tanh$  作为激活函数, 如式2-42所示,  $\tanh$  的倒数可以由前向传播时结果直接得到, 是非常优良的特征。

$$\mathbf{g} = \tanh(\mathbf{W}_2 \mathbf{m}) \quad (2-42)$$

对比上一层的特征  $\mathbf{m}, \mathbf{g}$  可以看做更抽象的特征, 句子级别的特征。定义  $\mathbf{f} = [\mathbf{l}, \mathbf{g}]$ , 其中  $\mathbf{l}$  为词法层面的特征, 由人工标注提取。

$$\mathbf{z} = \mathbf{W}_3 \mathbf{f} \quad (2-43)$$

其中  $\mathbf{z} \in R^{n_4 \times l}$ ,  $l$  为关系类别数目。整体体结构如图2-14所示。

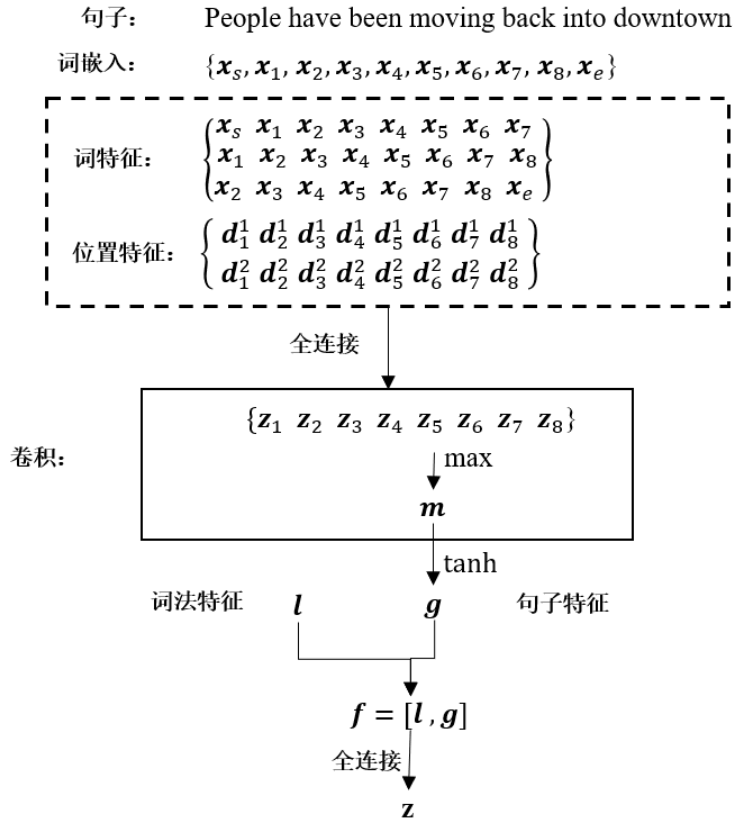


图 2-14 卷积神经网络进行关系抽取

### 2.2.3 softmax 函数

softmax 将多个神经元的输出，映射到  $(0, 1)$  区间，可以表示成某个类别的概率，从而来进行分类。设神经网络最后输出为式 (2-44)。

$$\mathbf{z} = (z_1, z_2, \dots, z_k)^T \quad (2-44)$$

那么对  $z_i$  使用 softmax 函数可以得到第  $i$  个类别的概率  $p_i$  为式 (2-45) 所示。

$$p_i = \frac{e^{z_i}}{\sum_k e^{z_k}} \quad (2-45)$$

设损失函数为交叉熵损失函数：

$$L = - \sum_i y_i \ln a_i \quad (2-46)$$

则可以对神经元的第  $i$  个输出  $z_i$  求偏导：

$$\begin{aligned} \frac{\partial L}{\partial z_i} &= \frac{\partial L}{\partial a_j} \frac{\partial a_j}{\partial z_i} \\ &= \frac{- \sum_i y_j \ln a_j}{\partial a_j} \frac{\partial a_j}{\partial z_i} \\ &= -y_i + a_i \sum_j y_j \end{aligned} \quad (2-47)$$

可以看到偏导数由前向传播中间结果组成，而在 pytorch 中每个 Variable 会存储前向传播值和偏导数值，所以反向传播计算可以利用前向传播作简单的加减可以得到偏导数，降低计算难度。

## 2.3 小样本学习基础

与图2-14所示传统深度学习进行关系抽取不同，小样本学习只需要少量样本甚至一个样本便可以得到不错的结果，因此网络结构也有所不同。本小节主要介绍小样本学习的基本概念，以及已经应用到关系抽取中的小样本模型。

### 2.3.1 小样本学习基本概念

小样本学习是监督学习的一种比较特殊的监督学习<sup>[45]</sup>，对比常见的增加正则项来降低模型的过拟合程度，小样本学习通过减少训练样本，学习“学习的能力”来增加泛化能力，定义小样本学习任务为  $T$ ，如式 (2-48) 所示。

$$T = \{S, Q, J\} \quad (2-48)$$

$S$  如式 (2-49)，称为支撑集，包含  $n$  类样本，每个样本有  $k$  个实例。

$$S = \{(x_1^1, y_1), \dots, (x_1^k, y_1), \dots, (x_n^1, y_n), \dots, (x_n^k, y_n)\} \quad (2-49)$$

$Q$  如式 (2-50)，称为查询集，包含和支撑集  $S$  同样类型的  $n$  类样本，每个样

本有  $q$  个实例。

$$Q = \{x_1^1, \dots, x_1^q, \dots, x_n^1, \dots, x_n^q\} \quad (2-50)$$

小样本关系抽取模型的目的是通过支撑集  $S$  与损失函数  $J$  学习到从  $x$  映射到  $y$  的最优解，且希望该解具有良好的泛化能力，也就是  $Q$  和  $S$  在不是同一批数据的情况下，也能有比较好的结果。

### 2.3.2 基于小样本学习的关系抽取

小样本学习是近几年才应用到自然语言领域，Han 等在 2018 年首次利用小样本学习方法进行关系抽取<sup>[24]</sup>，使用了常见的小样本学习方法包括元网络（Meta Network）<sup>[20]</sup>，GNN<sup>[22]</sup>，SNAIL<sup>[23]</sup> 和原型神经网络（Prototypical Networks）<sup>[25]</sup>。Gao 等在 Han 等的基础上选出表现比较好的 GNN 与原型网络作为基准<sup>[27]</sup>，并且提出 BERT-PAIR 结构。本节主要说明 GNN 和原型网络的算法思想。

#### 2.3.2.1 基于图神经网络的小样本学习

设输入  $T = \{S, Q\}$ ，对比常规的小样本查询集定义，其中  $Q = \{\tilde{x}\}$ ，此处查询集只包含一个样本。使用 GNN 进行小样本学习的思想是将有标签的样本的标签信息传播到无标签的样本上。使用 GNN 作小样本关系抽取步骤如下。第一步将输入转换位节点特征（node features），作为图神经网络的输入  $x^0$ ，如式（2-51）。

$$x_i^0 = (\varphi(x_i^j), h(y_j)), \tilde{x}^0 = (\varphi(\tilde{x}), \frac{1}{k} \cdot 1_k) \quad (2-51)$$

其中  $\varphi$  是解码器，将句子嵌入表示， $h(y_j) \in \mathbb{R}^k$  表示将标签转换为 one-hot 编码的编码器，对于待分类样本  $\tilde{x}$ ，将标签维度用  $K$  维的均匀分布补充。GNN 的迭代如式（2-52）所示。其中  $\mathcal{A}$  表示图自身的线性运算集合。 $Gc(\cdot)$  表示 GNN 层，输入为  $x^{(k)} \in \mathbb{R}^{V \times d_k}$ ，输出  $x^{(k+1)} \in \mathbb{R}^{V \times d_{k+1}}$ 。 $\Theta = \{\theta_1^{(k)}, \dots, \theta_{|\mathcal{A}|}^{(k)}\}_k$ ， $\theta_B^{(k)} \in \mathbb{R}^{d_k \times d_{k+1}}$  表示可训练参函数， $\sigma$  表示非线性函数，Han 等选用 *leaky-ReLU*<sup>[46]</sup>。

$$x_l^{(k+1)} = Gc(x^{(k)}) = \sigma\left(\sum_{B \in \mathcal{A}} Bx^{(k)}\theta_{B,l}^{(k)}\right), l = d_1 \dots d_{k+1} \quad (2-52)$$

GNN 的小样本关系抽取如图2-15所示意。

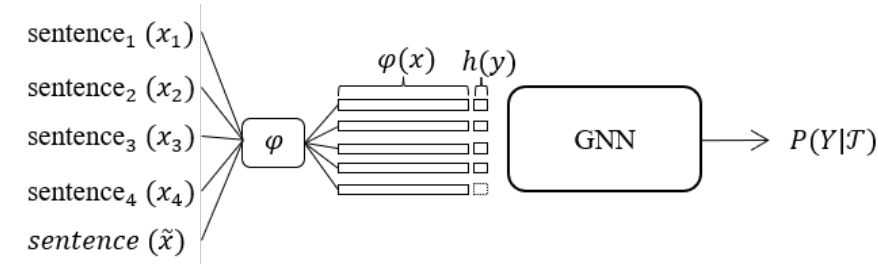


图 2-15 图神经网络小样本学习关系抽取示意

## 2.3.2.2 基于原型网络的小样本学习

原型网络目的是计算一个  $M$  维的表示  $\mathbf{c}_k \in \mathbb{R}^M$ ，或者称为 **prototype**，如式 (2-53)。对于每个类型，学习到映射  $f_\varphi: \mathbb{R}^D \rightarrow \mathbb{R}^M$ ，其中  $\varphi$  是可学习的参数。每个 **prototype** 是支撑集的嵌入向量的平均值。

$$\mathbf{c}_k = \frac{1}{|S_k|} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\varphi(\mathbf{x}_i) \quad (2-53)$$

定义距离函数为  $d: \mathbb{R}^M \times \mathbb{R}^M \rightarrow [0, +\infty)$ ，原型网络通过计算查询样本的嵌入表示到 **prototype** 的距离来得到概率分布。式2-54中分子表示查询样本到第  $k$  类样本的 **prototype** 距离，可以理解为查询样本到那一类样本的 **prototyoe** 近，属于哪一类的概率越高。

$$p_\varphi(y = k|\mathbf{x}) = \frac{e^{-d(f_\varphi(\mathbf{x}), \mathbf{c}_k)}}{\sum_{k'} e^{-d(f_\varphi(\mathbf{x}), \mathbf{c}_{k'})}} \quad (2-54)$$

原型网络的训练过程如算法2-2所示。

**算法 2-2** 训练集 loss 计算，其中  $N$  是训练集大小， $K$  是类型数量， $N_C \leq K$  是每个 episode 的训练类型数， $N_S$  是每个类型的样例数目， $N_Q$  是每个类查询的数目。 $rand(S, N)$  表示从  $S$  中随机取  $N$  个元素。

```

Input: 训练集  $D = (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)$ 
Output: loss 值  $J$ 

1 begin
2    $V \leftarrow rand(\{1 \dots K\}, N_C)$ 
3   for  $k$  in  $\{1, \dots, N_C\}$  do
4      $S_k \leftarrow rand(D_{V_k}, N_S)$ 
5      $Q_k \leftarrow rand(D_{V_k} \setminus S_k, N_Q)$ 
6      $\mathbf{c}_k = \frac{1}{N_C} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\varphi(\mathbf{x}_i)$  // 计算 prototype
7   end
8    $J \leftarrow 0$ 
9   for  $k$  in  $\{1, \dots, N_C\}$  do
10    for  $(\mathbf{x}, y)$  in  $Q_k$  do
11       $J \leftarrow J + 1 / (N_Q N_C) \times [d(f_\varphi(\mathbf{x}), \mathbf{c}_k) + \log \sum_{k'} e^{-d(f_\varphi(\mathbf{x}), \mathbf{c}_{k'})}]$ 
12    end
13  end
14 end

```



## 2.4 相关数据集

王传栋等人整理了常见实体关系抽取数据集<sup>[15]</sup>，如表2-2所示。其中 NYT 是一条句子中可能有多个关系，且是远程监督获取的数据库。

表 2-2 常用的关系实体关系抽取数据集

数据集	总样本数	类型数
ACE04	16771	24
SemEval-2010 Task 8	6674	9
TACRED	21784	42
FewRel	70000	100
NYT	143391	57

Han 等人对比了其所整理的小样本关系数据库 FewRel 和其它小样本学习常用数据库<sup>[24]</sup>，如表2-3所示。Ominglot 与 miniImageNet 都是图像识别领域，其中 FewRel 不仅仅是关系抽取数据集，同时也是对比两个小样本学习领域的数据集，也是最大的小样本学习数据集，本文的研究也主要使用 FewRel 数据集。

表 2-3 常用的小样本关系数据集

数据集	类型数	类型/样本	总样本数
Omniglot	1623	20	32460
miniImageNet	100	600	60000
FewRel	100	700	70000

## 2.5 本章小结

本章首先通过嵌入表示，神经网络基础，softmax 函数以及模型评价方式介绍了神经网络进行关系抽取的理论基础，再介绍了两种常见的小样本学习模型以及这些模型如何应用于关系抽取，最后介绍了相关的数据集。

## 第三章 小样本条件下关系抽取研究

目前基于预训练模型的简单的小样本关系抽取准确率已经可以达到人类的分类水平，但是常见的预训练模型具有大量的参数以及难解释性，且存在使用硬件要求高的缺点。本章第一部分说明使用简单的神经网络模型可以更快速的训练以及快速的得出结果，但是简单的网络模型难以获取复杂的自然语言语义，在小样本学习任务中表现还有提升空间。因此第一部分主要研究如何提高简单的网络模型在小样本学习下的表现。本章第二部分说明在实际应用中还需要有“以上都不是”选项，对简单神经注意力元学习器进行进一步改进，使得其能够捕获前后文信息，并且在公开数据集上验证改动的有效性。

### 3.1 基于简单神经网络小样本关系抽取研究

#### 3.1.1 引言

岳增营等将预训练模型（Pre-trained Models, PTMs）分为基于静态词嵌入技术与上下文相关嵌入技术<sup>[47]</sup>，BERT（Bidirectional Encoder Representations from Transformers）等上下文相关嵌入技术得到预训练模型自提出之时在各项自然语言处理任务中都取得了很好的效果<sup>[48]</sup>，Han 等利用 BERT 作为词嵌入编码器来进行小样本学习也取得了不错的效果，如表3-1所示。

表 3-1 原型网络使用不同编码器下小样本关系抽取结果

	5-way 1-shot	5-way 5-shot	10-way 1-shot	10-way 5-shot
Proto (CNN)	69.2	84.79	56.44	75.55
Proto (BERT)	80.68	89.60	71.48	82.89
HUMAN	92.22	-	85.88	-

PTMs 从是否结合上下文信息上可以分为早期的 word2vec 和 Glove 等以及现代的 GPT 和 BERT 等<sup>[49]</sup>。早期的 PTMs 利用语料训练后会获得一个单词到向量的映射矩阵  $M$ ，此时获取单词到词向量相当于利用矩阵查询单词向量属于矩阵某一行，并不能根据上下文信息对同一个单词产生不同的嵌入表示。现代的 PTMs 则可以融入上下文信息，最直观的解释就是相同的单词在不同的上下文下有不同的词向量，如图3-1所示。现代 PTMs 通过超大语料的训练，使其获得更多的语法信息，这使得现代 PTMs 有更好的表现。但是现代 PTMs 由于引入了更多的参数，使得模型更加复杂且可解释性越来越差<sup>[49]</sup>。

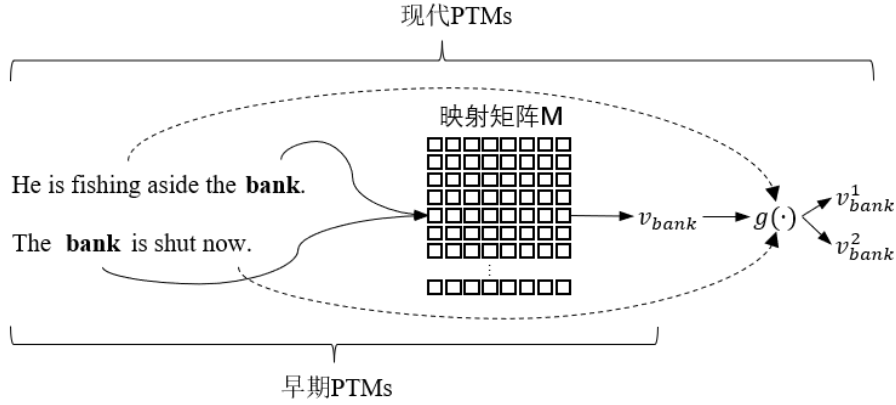


图 3-1 相比早期 PTMs，现代 PTMs 结合了上下文

本节主要从理论和实验上对比了简单神经网络利用预训练模型的计算复杂度，并且提出了可训练的改变语序的数据增强的训练方法以及通过上下文相关的采样方式来弥补使用静态词嵌入的问题。实验表明，在损失了一定的准确率的情况下，本文提出的模型预测响应时间比使用上下文相关的模型响应速度快数十倍。

### 3.1.2 算法设计

本节使用两类提取词嵌入的模型进行实验对比，从理论和实验说明静态词嵌入模型在计算复杂度上具有相当优势，但是准确率方面略差于上下文相关词嵌入模型。本节还提出了分段卷积网络中上下文相关的采样方式，来提高简单神经网络模型在小样本关系抽取上的准确率。

#### 3.1.2.1 预训练模型提取词嵌入

本节将分别对比静态词嵌入代表模型 Glove 和上下文相关词嵌入模型代表模型 BERT。其中静态词嵌入模型的计算复杂度低，仅仅是通过嵌入矩阵查询即可，基于静态词嵌入模型的神经网络模型本章称为简单神经网络模型。Glove 模型的具体算法在第二章已经详细阐述，本小节不作赘述。BERT 自提出后被应用到各种自然语言处理任务，下面从 BERT 的输入，网络结构以及预训练任务来说明使用 BERT 进行词嵌入提取的原理。

BERT 的输入由三种不同的嵌入信息求和，包括 wordpiece embedding, position embedding 和 segment embedding。其中 wordpiece 把词看做一些公共字符段组成，position embedding 表示位置嵌入，通过初始化值然后学习，segment embedding 用来区分多段输入。BERT 的输入如图3-2所示意，其中“playing”被拆分成“play”与“#ing”，其它带有“#ing”的单词也可以这样拆分。

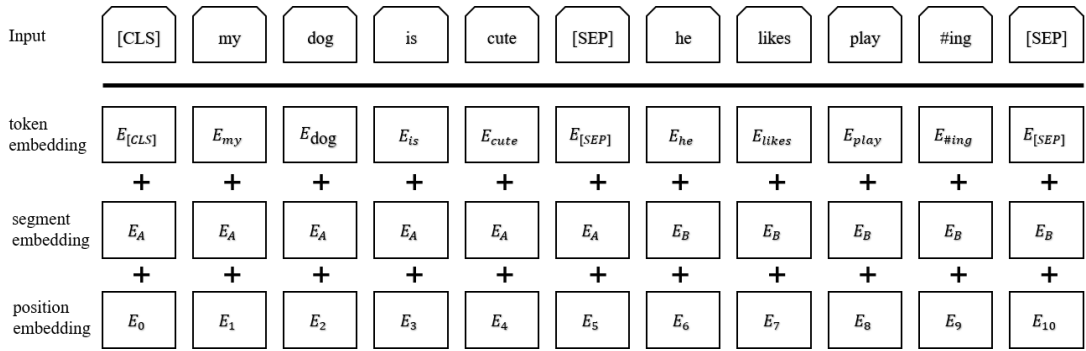


图 3-2 BERT 的输入

BERT 的网络结构是由 Transformer 的编码器连接在一起的，基本大小的 BERT 也就是 BERT<sub>BASE</sub> 由 12 层编码器连接组成<sup>[48]</sup>。Transformer 的编码器结构如图3-3所示，包含两个子层，第一个子层是多头注意力机制层（Multi-Head Attention），第二个子层是简单的前向传播网络（Feed-Forward Net），比如可以用残差网络 (residual connection)<sup>[50]</sup>。

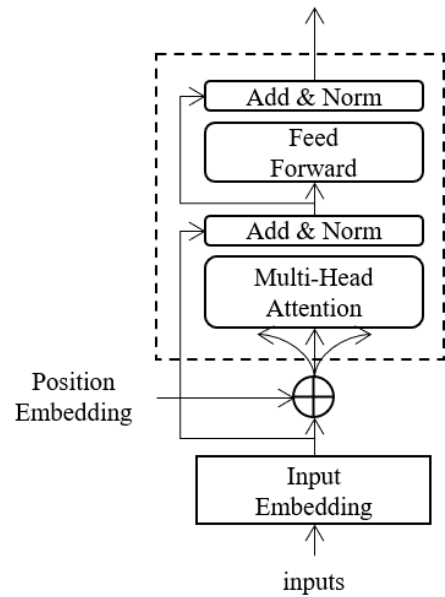


图 3-3 Transformer 的编码器结构

其中多头注意力由自注意力机制（self-attention）组成，此处自注意力机制由 Vaswani 等提出的点积自注意机制（Scaled Dot-Product Attention）实现<sup>[30]</sup>，结构如图3-4所示。自注意力机制的目的是为了解决传统神经网络无法很好地获取输入向量与向量之间的关系，注意力机制有效的解决这个问题，在自然语言处理模型中取得了广泛的应用。

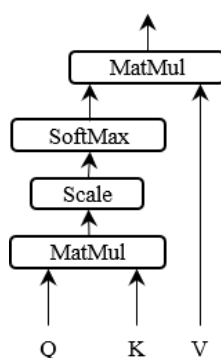


图 3-4 点积自注意力机制结构图

多头注意力是将多个自注意机制结果合并而得到，如图3-5所示。

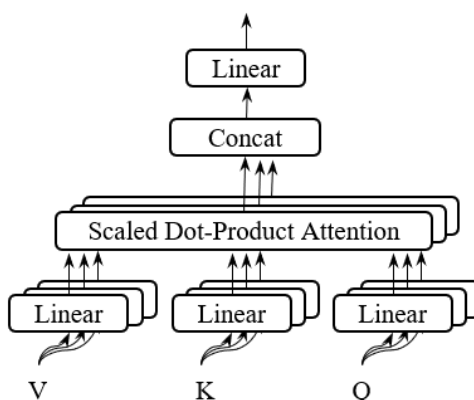


图 3-5 多头注意力机制结构图

BERT 由两个训练任务来从大规模语料中获取知识。任务一称为掩码语言模型，随机掩盖一些单词。其中 15% 的词段会随机被掩盖，在这 15% 的词段中，有 10% 的概率随机替换一个词段，10% 的概率保持原来的单词不变，80% 使用词段 “[MASK]” 替换，任务一主要目的是为了捕捉上下文关系。任务二称为下一句预测，在训练语料中有 50% 的概率下一句被替换，作为负样本。任务二是为了 BERT 模型能够获得捕捉连续长序列特征的能力。

BERT<sub>BASE</sub> 的训练使用了 4 个 TPU 集群，每个集群 16 个 TPU，也就是总共 64 个 TPU 训练了 4 天时间<sup>[48]</sup>，所以根据自己训练预训练模型难度太高，本章 BERT 的预训练模型使用 Google 训练得到的预训练模型 BERT<sub>BASE</sub>，Glove 的预训练模型使用斯坦福大学训练得到的预训练模型 Glove.6B.200d，其中 6B 指的是训练语料的规模达到 6 百亿，200d 是指得到的向量维度为 200。作为对比，本文也取 BERT<sub>BASE</sub> 的嵌入层对比效果，表示为 BERT<sub>BASE</sub><sup>eb</sup>。

## 3.1.2.2 可训练数据增强网络层

本小节根据人类在阅读文本时，文本中字符顺序在乱序情况下有时也不会影响人类对文本的理解的启发，再由普通交换输入顺序的数据增强，推广到设计出可训练数据增强网络层 (Trainable Data Enhancement layer, TDE layer)，即是根据损失函数调整词嵌入的顺序。全连接层可以表示为输入和一个权重矩阵相乘，初始化全连接层为单位矩阵，算法3-1为反向传播时参数更新算法，即是根据 loss 来调整该单位矩阵得到初等矩阵，并且在权重更新时保持该网络层为初等矩阵。

**算法 3-1** TDE layer 权重更新算法， $row(\cdot)$  计算矩阵行数， $argmax(\cdot)$  计算向量最大值的下标， $sort(\cdot)$  表示对向量排序且返回排序后下标位置

**Input:** TDE layer 初始化权重  $W$ , TDEL 层反向传播梯度  $D$   
**Output:** 更新后 TDE layer 权重  $\tilde{W}$

```

1  $C \leftarrow \emptyset$ 
2  $l \leftarrow row(W)$ 
3 for  $i$  in  $\{0, \dots, l-1\}$  do
4    $m = argmax(W_i)$  // 计算  $W$  第  $i$  行值为 1 下标
5    $ID_i \leftarrow sort(D_i)$  // 对梯度第  $i$  行排序
6   if  $m \in C \parallel D_{i,m} < 0$  then
7      $W_{i,m} \leftarrow 0$  // 如果该位置被占用，或者梯度小于 0
8     for  $j$  in  $\{0, \dots, l-1\}$  do
9        $sortj \leftarrow ID_i$ 
10      if  $sortj \notin C$  then
11         $C \leftarrow C \cup \{sortj\}$ 
12         $W_{i,sortj} \leftarrow 1$ 
13        break
14      end
15    end
16  else
17     $C \leftarrow C \cup \{m\}$ 
18  end
19 end
20  $\tilde{W} \leftarrow W$ 

```

由矩阵的基本知识可知，左乘一个位置变化的初等矩阵可以将行顺序改变，以此来达到调整词特征在句子中的位置，如式（3-1）所示。算法3-1的目的就是保证该全连接层是初等矩阵，达到改变输入顺序进行数据增强的目的。

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \begin{pmatrix} w_2 \\ w_3 \\ w_1 \end{pmatrix} \quad (3-1)$$

### 3.1.2.3 分段卷积神经网络提取句子特征

受到 Chen 等的启发，本文使用分段卷积神经网络（Piecewise Convolutional Neural Networks, PCNN）提取句子特征<sup>[51-52]</sup>。PCNN 主要过程为对得到的词向量特征进行一维卷积，然后将得到的特征向量按照实体位置进行分段然后进行最大采样，PCNN 示意图如图3-6所示。

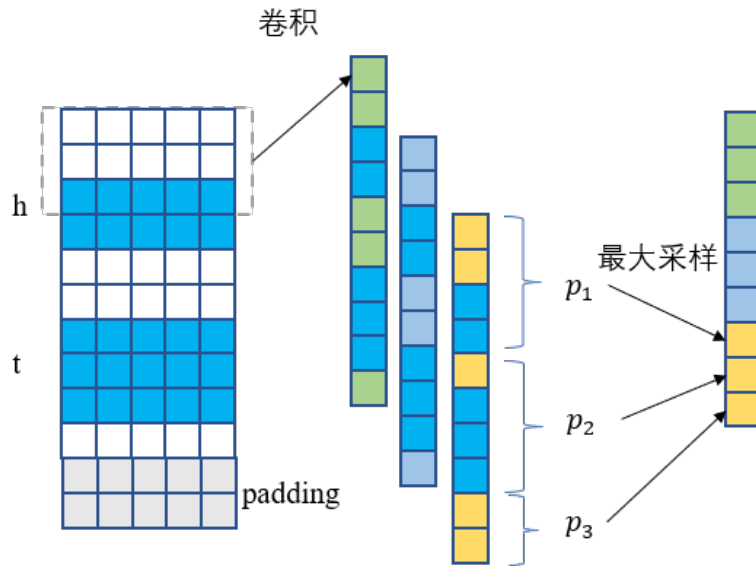


图 3-6 PCNN 提取句子特征示意图

本文认为完全的分段池化会弱化实体与上下文的关系，且本章使用的静态词嵌入模型也没有考虑上下文关系获取词向量。因此本文提出了实体与上下文交互更加密切的交互分段采样（Cross-Piece Pooling, CPP），如图3-7所示。其中图3-7(a)表示头实体只和前文划分为一段，尾实体只和后文划分为一段，记为稀疏交互分段采样（Sparse CPP）。其中图3-7(b)表示头尾实体和前后上下文段划分为一段，记为稠密交互分段采样（Dense CPP）。考虑到最大采样容易采样到相同值，使用平均采样方式。

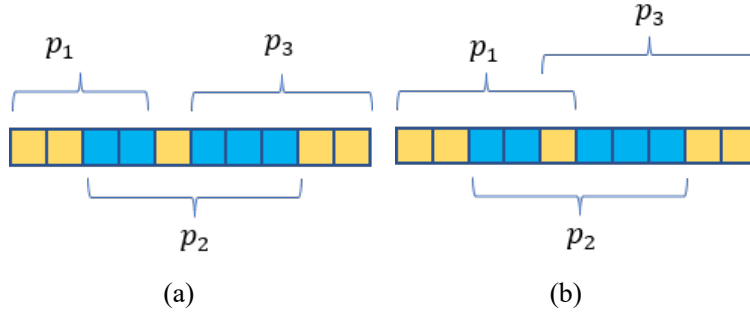


图 3-7 (a) 稀疏交互分段采样 (b) 稠密交互分段采样

### 3.1.2.4 特征数据处理

小样本学习实际是求解查询集中的样本属于训练集中的那一类样本，对于 N-way K-shot 的小样本学习，训练集由多组包含支撑集和查询集的训练样本组成。FewRel 数据集的训练集包含的关系类型集合为  $\mathcal{R} = \{r_1, \dots, r_{48}\}$ ，其中第  $i$  类所含的样本集合为  $E_i = \{e_i^1, \dots, e_i^{700}\}$ 。

设训练集为  $T$ ，大小为  $n$ ，则  $T$  可以表示为式 (3-2) 所示，表示  $T$  由  $n$  组支撑集和查询集组成。其中第  $i$  组的支撑集  $S_i$  包含  $N$  类样本，每类样本  $K$  个实例。 $Q_i$  包含  $N$  类样本，每个样本  $q$  个实例，用来进行正样本的训练，比如  $Q_i$  中第 1 个查询样本的标签从 0 开始计算即是“0”。

$$T = \{\{S_1, Q_1\}, \dots, \{S_n, Q_n\}\} \quad (3-2)$$

式 (3-3) 表示训练集  $T$  中的第  $i$  组数据  $\{S_i, Q_i\}$  的构造是先从 FewRel 训练集中随机取  $N$  个类型的关系，代表 N-way K-shot 中的 N-way。其中表达式  $rand\_choice(A, a)$  表示从集合  $A$  中随机取  $a$  个元素。

$$\mathcal{R}_i = rand\_choice(\mathcal{R}, N) = \{r_{i,1}, \dots, r_{i,N}\} \quad (3-3)$$

再分别从这  $N$  类关系的样本集合  $E$  中随机取出  $K + q$  个样本，其中  $K$  个放入支撑集，如式 (3-4) 所示。

$$S_i = \{e_{r_{i,1}}^1, \dots, e_{r_{i,1}}^K, \dots, e_{r_{i,N}}^1, \dots, e_{r_{i,N}}^K\} \quad (3-4)$$

其中  $q$  个放入查询集，如式 (3-5) 所示。这样构造的查询集  $Q$  保证了每组训练集合对于每个类型都有  $q$  次查询训练。

$$Q_i = \{e_{r_{i,1}}^{K+1}, \dots, e_{r_{i,1}}^{K+q}, \dots, e_{r_{i,N}}^{K+1}, \dots, e_{r_{i,N}}^{K+q}\} \quad (3-5)$$

由于神经网络的模型计算量大，难以一次性输入所有数据到模型进行计算，并且 Goyal 等实验认为每个批次过大会影响模型泛化能力<sup>[53]</sup>，所以需要对数据进行分批输入。下面的过程主要是说明如何将原始训练数据变为模型每次输入训练



的张量数据。

设每批同时训练的样本数为  $B$ ，表达式  $encoder(A)$  表示将集合  $A$  中的句子编码得到句子特征。式 (3-6) 表示将一批数据中支撑集中的所有句子解码得到句子特征的张量。

$$s = encoder(\{S_1, \dots, S_B\}) \quad (3-6)$$

得到张量  $s$  的形状为  $(B, N, K, H)$ ，其中  $H$  表示隐藏层维度。由于查询集中每类关系都有 1 个查询样本训练，需要将  $s$  在维度 2 上开辟一个维度并且对高维进行复制，相当于对查询集建立  $N$  个副本，以达到每个查询样本能够有属于该样本自身的支撑集进行训练。

表达式  $unsqueeze(v, d)$  表示对张量  $v$  在  $d$  维度上开辟一个维度，表达式  $expand(v, d, n)$  表示对张量  $v$  在  $d$  维度以上进行  $n$  次复制。式 (3-7) 表示将张量  $s$  的 2-4 维度数据复制  $n$  次，最后  $s$  的维度为  $(B, N, N, K, H)$ 。

$$s = expand(unsqueeze(s, 2, N), 2, B) \quad (3-7)$$

由于每类查询集合都有一个样本实例，随后维度扩张的数据可以看作是新的训练数据批次的扩充，每个批次相当于大小从  $B$  变为  $B \cdot N$ 。 $view(v, (\cdot))$  表示改变张量  $v$  的形状。式 (3-8) 表示将  $s$  变形，此变形是为了后续操作符合时序卷积网络的输入，时序卷积网络输入要求有三个维度，一维可以视为每批次样本数量，二维表示每次网络输入有多少条句特征，三维表示句特征的维度。可以得出扩充后一维变为  $B \cdot N$ ，二维表示总的样本数量即是样本类型数量和每类样本数量乘积，即是  $N \cdot K$ 。

$$s = view(s, (B \cdot N, N \cdot K, H)) \quad (3-8)$$

每个查询集里有  $N$  个类，可以分别标注为  $(0, \dots, N-1)$ ，将这些标签进行 one-hot 编码，得到标签数据的张量  $l$ 。参考式 (3-6) 可得查询集句子特征  $q$ ，将  $q$ ， $l$  复制之后与  $s$  合并可以得到神经网络输入数据  $x$ ，如式 (3-9)。

$$x = cat(s, q, l) \quad (3-9)$$

图3-8表示上述所描述的批大小为 1 的 5-way, 1-shot 的训练集构造过程示意图，其中  $encoder$  表示提取句特征的算法。

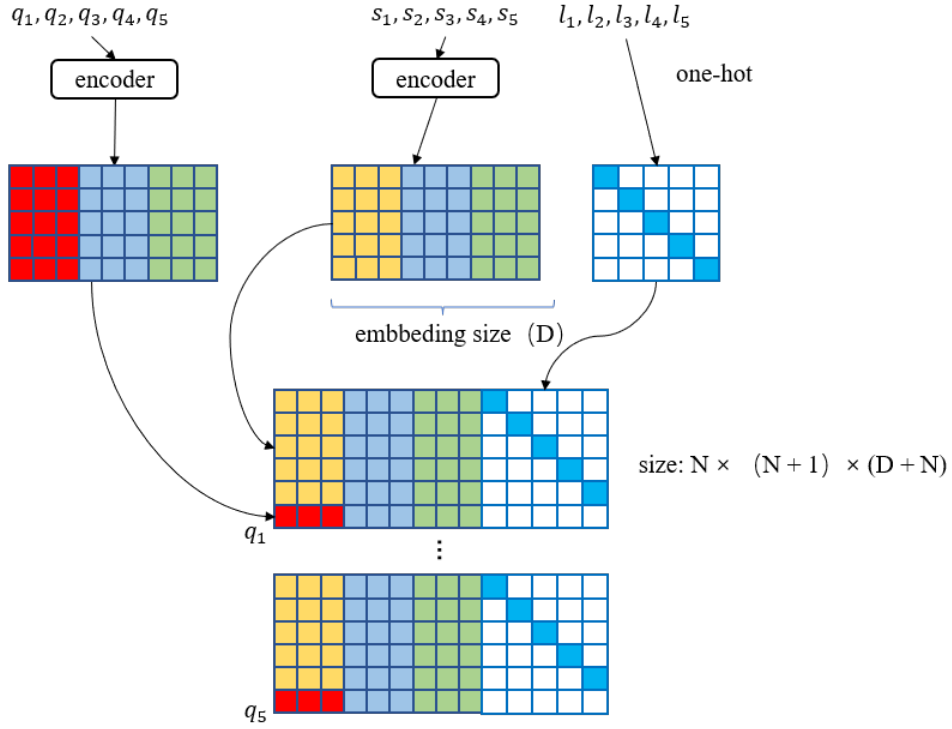


图 3-8 5-way 1-shot 训练集构造示意图

本节对一批数据的处理过程表示为式 (3-10)，其中  $BS = \{S_1, \dots, S_B\}$ ，表示一批样本的集合。

$$x = \text{feature\_process}(BS, BQ, BL) \quad (3-10)$$

### 3.1.2.5 元学习器模型结构

在 Gao 等的小关系抽取实验中，简单神经注意力元学习器（Simple Neural Attentive Meta-Learner, SNAIL）表现并不是最好<sup>[23]</sup>，本文认为 SNAIL 具有更大提升空间来验证本文改进的有效性，且 SNAIL 的网络模型结构相对简单，本文选为关系抽取的分类网络模型。

SNAIL 的主要由时序卷积块（Temporal Convolutions Block, TC Block）和注意力机制交错构成，结构如图3-9所示。图3-9中橙色表示时序卷积块，时序卷积块由指数增长的感受野，当感受野超过序列长度时便停止增长，绿色表示自注意力机制。其输入样例如图3-8所示，由有标签支撑集和无标签的查询数据处理得到句子特征和标签信息拼接得到。从图中可以看出，对于输入层其感受野为 1，当第二层时序卷积时感受野为 2，第三次时序卷积感受野为 4，这即是感受野呈指数增长的含义。

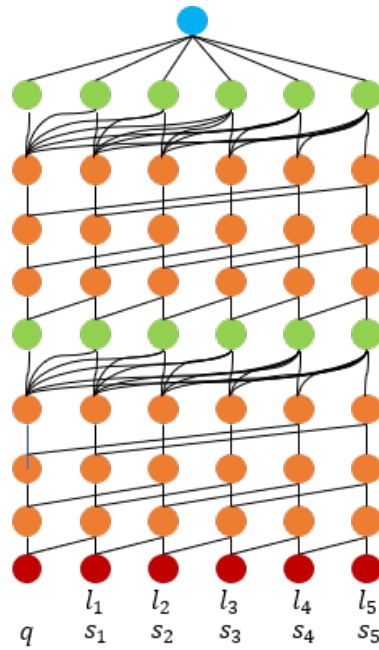


图 3-9 SANIL 的结构

时序卷积可以提供范围比较大的上下文信息，但是当时序卷积的层数过多时，这些信息会呈现指数增长而缺乏实际意义。注意力机制可以筛选出上下文中比较重要信息，但是对于有序的文本来讲，其缺乏时序关系。SNAIL 将时序卷积和注意力机制结合在一起，目的是使得两者优势互补，使模型能够大范围的筛选重要信息。其中时序卷积块由密集块（Dense Block）组成，密集块的结构如图3-10所示，将输入经过滤波器数量为  $D$ ，卷积核大小为 2，膨胀率为  $R$  的一维卷积。密集块的感受野呈现指数增长，直到感受野超过序列长度。注意机制层则采用 Vanswani 等提出的自注意力机制模型<sup>[30]</sup>，如图3-4所示。

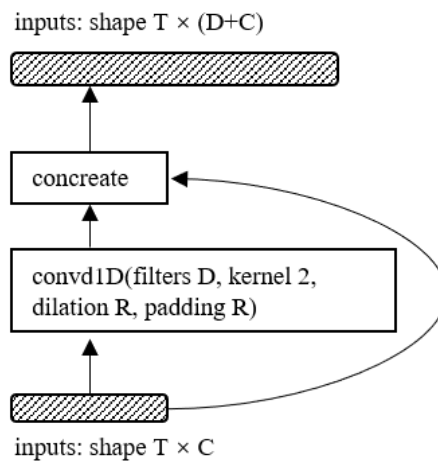


图 3-10 密集块的结构

小样本关系抽取模型如果在训练集中过拟合，会出现训练集准确率过高，不再收敛，但是模型在测试集中的准确率确差强人意。

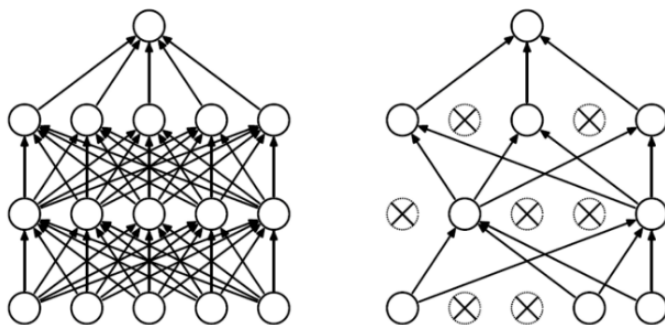


图 3-11 dropout 示意图

Hinton 等提出在每次训练的时候，随机让一定的神经元停止更新，这样可以提高网络的泛化能力，该过程又被称为 dropout<sup>[54]</sup>，如图3-11所示意。本章也讨论使用不同 dropout 值对实验结果的影响。

### 3.1.2.6 模型整体架构

本节提到模型整体架构如图3-12所示。

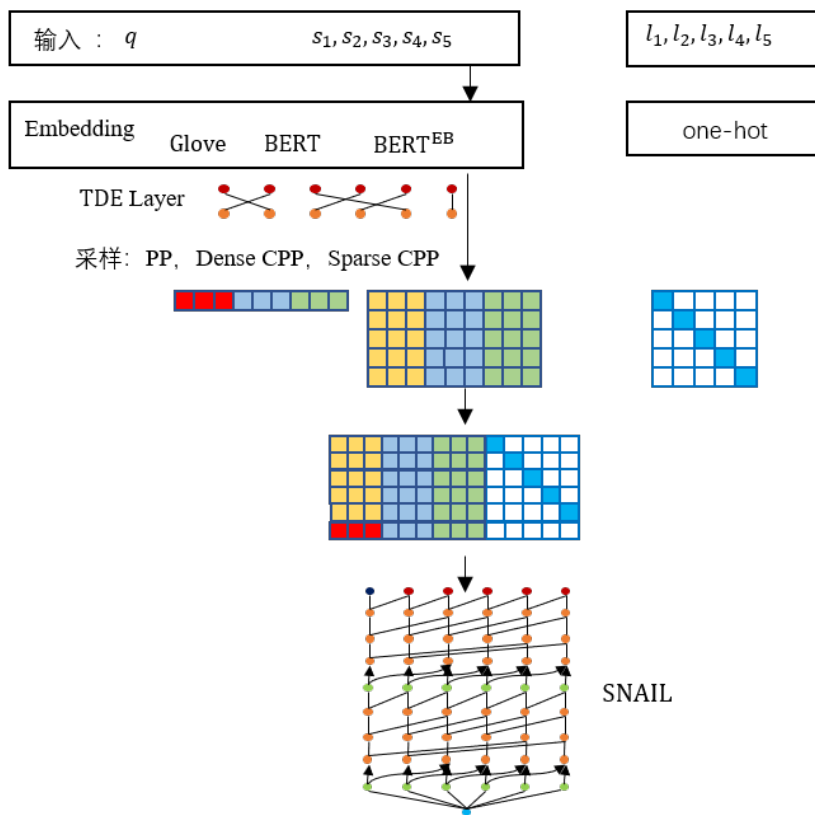


图 3-12 模型整体架构

图3-12中表示模型预测 5-way 1-shot 的过程，首先对 1 条查询样本语句  $q$  和 5 条不同类型的支撑集语句  $s_1 - s_5$  使用词嵌入技术获取词嵌入，再使用 PCNN 的采样得到句子特征；同时使用 one-hot 获取样本嵌入。将得到的句子嵌入和标签嵌入合并经过本节提出的 TDE Layer 输入到 SNAIL 模型中，最后 SNAIL 输出的就是查询样本  $q$  为各个标签的概率分布。

### 3.1.2.7 模型训练

本小节主要介绍如何对模型进行训练。小样本学习可以看做一个多分类问题，即判断查询样本属于支撑集中的哪一种类型，模型训练算法如算法3-2所示。

**算法 3-2** 模型训练算法， $rand\_choice(A, a)$  表示从集合  $A$  随机取  $a$  个元素

**Input:** 支撑集类型数  $N$ ，每类样本数  $K$ ，关系类型集合  $\mathcal{R}$ ，其中第  $i$  类所含的样本句子集合为  $S_i$ ，训练集  $T$ ，大小为  $t$ 。

**Output:** 更新后全连接层权重  $\tilde{W}$

```

1  begin
2       $T \leftarrow \emptyset$ 
3      for  $i$  in  $\{0, \dots, t-1\}$  do
4           $C_i = rand\_choice(\mathcal{R}, N)$ 
5           $S \leftarrow \emptyset, Q \leftarrow \emptyset, L \leftarrow \emptyset$ 
6          for  $c$  in  $C_i, l$  in  $\{1, \dots, N\}$  do
7               $S \cup rand\_choice(E_c, K), Q \cup rand\_choice(E_c, 1)$ 
8               $L \cup \{l\}$ 
9          end
10          $T \cup \{(S, Q, L)\}$ 
11     end
12     BT = batch(T) // 循环迭代器
13     for epoch in  $\{0, \dots, train\_iter\}$  do
14          $(BS, BQ, BL) \leftarrow BT$ 
15          $x = feature\_process(BS, BQ, BL)$  // 特征处理
16          $logits \leftarrow model(x)$ 
17          $ls = loss(logits, BL)$ 
18          $ls.backward()$ 
19     end
20 end
    
```

算法中  $\mathbf{x}$  表示神经网络最后一层全连接层输出向量，向量长度为样本类型数目，可以认为是对每个类别的评分（logits）， $l$  表示查询集样本标签，易得取值范围为  $(0, \dots, N-1)$ 。可以看出当对正确的标签  $l$  预测值评分越高， $-\mathbf{x}_l$  越小，模型损失越小。

分类问题常常使用交叉熵损失函数<sup>[55]</sup>，交叉熵损失函数的优点在于可以使神经网络最后一层权重梯度只跟输出值和真实值的差值成正比，此时收敛较快。交叉熵损失函数如式（3-11）所示。可以看出交叉熵损失函数应用到了 softmax 函数。

$$\text{loss}(\mathbf{x}, l) = -\log \left( \frac{\exp(\mathbf{x}_l)}{\sum_j \exp(\mathbf{x}_j)} \right) = -\mathbf{x}_l + \log \left( \sum_j \exp(\mathbf{x}_j) \right) \quad (3-11)$$

### 3.1.3 实验设计与结果分析

本节主要说明了实验环境，对使用数据集进行简单说明，分析了实验结果并且说明本章改进点的效果。

#### 3.1.3.1 实验环境

实验设计主要采用 PyTorch 深度学习框架<sup>[56]</sup>，PyTorch 提供 Python 的编程方式，支持代码模型化且易于调试，同时可以快速实现 GPU 加速。其中 BERT 的实现使用 Wolf 等人提出的 Transformers<sup>[57]</sup>，Transformers 提供了上千种预训练模型，支持上百种语言的自然语言处理任务，Transformers 有基于 Jax、PyTorch 和 TensorFlow 的实现，本文采取基于 PyTorch 的实现。

FLOPs 的统计是基于 thop 工具，thop 的原理是基于 PyTorch 提供的“注册前向传播钩子函数”。钩子函数是指在执行原有函数前执行一些操作，比如在执行每个全连接操作前对输入向量的形状统计并且计算 FLOPs，那么只要对网络中所有最小子层前向传播前注册既可以计算出模型总的 FLOPs。实验环境如表3-2所示。

表 3-2 实验环境

组件名称	参数
处理器	AMD Ryzen 5 3600 6-Core
主频	3.59 GHz
显卡	RTX 2070S
内存	16GB
硬盘	500GB
操作系统	Windows 10 专业版
开发语言	Python
集成开发环境	PyCharm

### 3.1.3.2 数据集

本节使用 FewRel 作为数据集，FewRel 数据集合标注形式如表3-3所示，每个数据集包含四个部分的主要信息，分别是 tokens 句子切分，h（head）头实体，t（tail）尾实体和关系标注。

表 3-3 数据样例示意

标注名称	列表内容
tokens	'Warsaw'、'-'、'Modlin'、'Airport'、'is'、'located'、'there'、'.'
h(head)	'modlin airport'、'Q1401995'、[2,3]
t(tail)	'warsaw'、'Q270'、[0]
relation	P931

表3-3中头实体和尾实体都被 uncased 处理，也就是将所有字母转为小写字母，包含了被标注词语、被标注词语 id 和标注词语在句子中的位置。表中“Q1401995”和“P931”分别表示实体和关系在维基数据库中的唯一键值，其详细解释可以通过维基数据库查询，如图表示维基数据库中对关系 P931 的详细解释，其含义为“机场服务地区”。

place served by transport hub (P931)

---

territorial entity or entities served by this transport hub (airport, train station, etc.)  
serves city | city served | train station serves

[In more languages](#)  
[Configure](#)

Language	Label	Description	Also known as
English	place served by transport hub	territorial entity or entities served by this transport hub (airport, train station, etc.)	serves city city served train station serves
Chinese	机场服务地区	该机场服务的城市或地区	
Spanish	ciudad asociada	La ciudad o región a la que sirve este centro de transporte (aeropuerto, estación de tren, etc.)	
Traditional Chinese	機場服務地區	該機場服務的城市或地區	

[All entered languages](#)

图 3-13 维基数据库中 P931 解释

为了确定模型处理句子长度的值，对 FewRel 数据集的训练集进行 token 数量统计如图3-14所示。

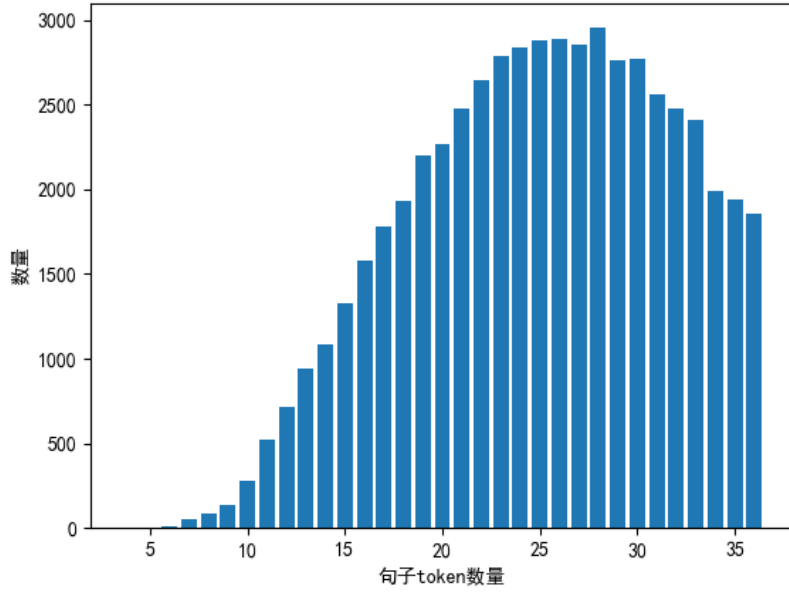


图 3-14 FewRel 数据集句子长度统计 L

从图3-14中可以看出 FewRel 数据集中大部分句子长度在 10 到 36 之间，句子长度最长不超过 40。

### 3.1.3.3 词嵌入层浮点计算量理论分析

本小节主要从理论分析和实验说明研究传统预训练模型和现代预训练模型的浮点运算量以及效果。

神经网络的复杂度通常通过两个参数来评估，一是前向传播时的计算量，称为浮点运算数量（Floating Point Operations, FLOPs），FLOPs 并没有确切的定义，通常一次乘加运算算作一次浮点运算，二是参数个数。本小节主要从浮点运算量分析词嵌入层的复杂程度，进而说明基于现代 PTMs 的神经网络模型时间复杂度远远高于基于传统 PTMs 的神经网络模型。

设  $b$  为神经网络每个批次大小， $l$  为句子最长长度， $e$  为句子中每个单词的嵌入特征维度。设  $F(b, l, e, \cdot)$  为  $\cdot$  网络层的浮点运算量。

Embedding 层是一层映射矩阵，用于映射单词在词典中的 id 到嵌入向量，不需要进行运算，设  $F(b, l, e, EM)$  是 Embedding 层 FLOPs 的数量，可得式 (3-12)。

$$F(b, l, e, EM) = 0 \quad (3-12)$$

LayerNormalization (LN) 层是将一层所有特征进行归一化，目的是为了缓解梯度消失来加速训练，其计算公式为式 (2-16)，可以得到对于每增加一个特征



$x_i$ , 在求标准差  $\sigma$  时增加了 1 次乘法运算和 1 次除法运算。除法比乘法更为复杂, Facebook 开发的开源工具 `fvcore` 中对此给出估计值平均每个输入的每个特征有 5 次浮点运算, 设  $F(b, l, e, LN)$  为 LayerNormalization 的浮点运算次数, 可以得到式 (3-13)。

$$F(b, l, e, LN) = 5 \cdot b \cdot l \cdot e \quad (3-13)$$

设全连接层 FC 的 FLOPs 为  $F(b, l, e, FC(i, o))$ , 其中  $i$  为全连接层输入维度,  $o$  为全连接层输出维度。通过第二章全连接层示意图可以得到式 (3-14)。

$$F(b, l, e, FC(i, o)) = b \cdot l \cdot i \cdot o \quad (3-14)$$

通过图3-3和图3-5中的 BERT 的网络模型结构可以推导 BERT 的浮点运算量。设 BERT 的 FLOPs 为  $F(b, l, r, B(h, m, a, n))$ , 其中  $h$  为隐藏层维度,  $m$  为 Encoder 中前馈神经网络的中间层大小,  $n$  为 Encoder 层层数。在 BERT<sub>BASE</sub> 中, 取  $h = 768$ ,  $m = 3072$ ,  $n = 12$ 。

$$\begin{aligned} F(b, l, e, B(h, m, n)) &= F(b, l, e, BEb) + F(b, l, e, BP(h)) \\ &\quad + nF(b, l, e, BEc(h, m)) \end{aligned} \quad (3-15)$$

其中  $BEb$  表示 BERT 嵌入层, 其中由一个 Embedding 层和 LayerNormalization 层组成, 由 (3-12) 可以得到式 (3-16),  $BP$  层表示采样层, 由一个全连接层构成, 则  $BEb$  的 FLOPs 如式 (3-16) 所示。

$$F(b, l, e, BEb(h)) = F(b, l, e, LN) \quad (3-16)$$

$BEc$  层是表示 BERT 中的 Encoder 层, 如图3-3, 由多头注意力机制层和前馈神经网络层组成。多头注意力机制使用的是自注意力机制加上一个全连接层和 LN 层用以增加网络的非线性性。在计算 Self-Attention 时, 需要将输入通过全连接层映射到 Q, K, V 共 3 个空间, 如图3-5所示。前馈神经网络在 BERT 中是将特征映射到  $m$  维空间标准化之后再映射回  $h$  维空间归一化后激活, 故包含两个全连接层, 这样是为了增加网络参数的非线性, 根据以上分析可以得到式子 (3-17)。

$$\begin{aligned} F(b, l, e, BEc(h, m)) &= 3F(b, l, e, FC(h, h)) + F(b, l, e, FC(h, h)) \\ &\quad + F(b, l, e, LN) + F(b, l, e, FC(h, m)) \\ &\quad + F(b, l, e, FC(m, h)) + F(b, l, e, LN) \end{aligned} \quad (3-17)$$

BP 层是池化层 (pool layer), 也叫作采样层, 使用全连接层实现, 则 BP 的 FLOPs 为式 (3-18)。

$$F(b, l, e, BP(h)) = F(b, l, e, FC(h, h)) \quad (3-18)$$

联立式 (3-13) - (3-18) 可以得到 BERT 模型 FLOPs。取  $b = 20$ ,  $l = 128$ ,

$$h = 768, m = 3072$$

$$\begin{aligned} F(b, l, e, B(h, m, n)) &= 5blh + n \cdot (4blh^2 + 10blh + 2blhm) + blh^2 \\ &= 3072000 + 68021452800 + 471859200 \\ &= 68496384000 \end{aligned} \quad (3-19)$$

综上从理论上分析  $BERT_{BASE}$  的 FLOPs 为 68496384000, Glove 和  $BERT_{BASE}^{eb}$  为 0。

#### 3.1.3.4 词嵌入层浮点计算量实验结果

本实验在算法3-2中对 *feature\_process* 使用不同的预训练模型, 其中有  $BERT_{BASE}$  和 Glove, 为了获取上下文相关模型的静态层实验效果, 同时对比了  $BERT_{BASE}$  中的 Embedding 层, 记为  $BERT_{BASE}^{eb}$ 。考虑到取批大小为  $B$ , 句子最大长度为  $l$ , 支撑集中有  $N$  类, 结合式 (3-8) 在数据输入网络模型时, 会将批数据扩充为  $B * N$ , 且考虑到查询集中每个类一个样本句子故一批数据中要编码的句子实际为  $(2 * B * N)$ 。在测试预训练模型复杂度时, 随机取句子  $(2 * B * N)$  到词嵌入层解码。使用 *thop* 工具统计模型计算时浮点运算次数并且通过 10 次实验记录词嵌入层解码时间。

实验参数设置如表3-4所示。

表 3-4 模型 FLOPs 实验参数设置

参数	参数含义	参数值
B	批大小	4
L	句子最大长度	40
H	隐藏层大小	768
q	查群集中每类样本数量	1

实验结果的 FLOPs 如表3-5所示, 理论与实验的少量差距主要原因是 *thop* 工具对于某些计算过程没有统计。

表 3-5 模型 FLOPs 实验结果对比

	$BERT_{BASE}$	Glove	$BERT_{BASE}^{eb}$
理论 FLOPs	68496384000	0	0
实验 FLOPs	67951411200	0	0

分别使用上述模型作为词嵌入解码器以及分别使用 GPU 和 CPU 条件下, 解

码十次获取每次解码的时间，结果如表3-6所示。

表 3-6 使用不同方式解码句子花费时间，单位：毫秒。

	Glove gpu	Glove	BERT <sub>BASE</sub> gpu	BERT <sub>BASE</sub>	BERT <sub>BASE</sub> <sup>cb</sup> gpu	BERT <sub>BASE</sub> <sup>cb</sup>
1	1006.58	7.33	604.97	5156.11	1017.95	7.62
2	5.04	7.93	99.26	5320.99	4.96	7.15
3	4.35	7.42	50.43	5324.94	5.25	8.03
4	5.61	6.83	74.26	5387.76	5.48	7.32
5	4.93	6.77	74.01	5361.15	4.66	7.23
6	4.53	6.98	75.39	5397.85	4.64	7.20
7	4.48	6.55	74.97	5351.78	5.10	6.80
8	4.45	6.23	75.72	5312.47	4.73	7.10
9	4.26	7.40	74.80	5313.91	4.75	7.02
10	4.43	7.62	77.53	5340.42	4.73	7.21
去尾平均	4.73	7.11	78.24	5339.18	4.96	7.23

表中使用 GPU 的第一次耗时比较长，是因为 GPU 启动时需要预热，故在统计时使用去尾平均值，即是去掉最大值与最小值再平均。从表中可以看出 Glove 与 BERT<sub>BASE</sub><sup>cb</sup> 在 gpu 和 cpu 上耗时都相近。而 BERT<sub>BASE</sub> 在 GPU 上耗时要比 Glove 多二十多倍，在 CPU 上运行时则耗时多千倍。且由于 BERT<sub>BASE</sub> 模型比较复杂，批大小最多不能超过 4，否则普通的单个 GPU 无法将数据存入显存。本节从理论和实验说明了上下文相关预训练模型复杂度远高于静态预训练模型，下一节将说明在我们的改进下，简单预训练模型的能力有所提升，并且简单预训练模型也能更好的应用于实际系统中。

### 3.1.3.5 关系抽取结果评价指标

关系抽取的结果类似于多分类问题，一般通过计算对应的准确率（Accuracy）、精确率（Precision）、召回率（Recall）和 F1 值来评价。对于给定的测试数据集，所有预测正确与所有样本之比为准确率；正确分类为正类的样本数与全部正类样本数之比为精确率；预测正确的正类与所有正类数据的比值为召回率；F1 是准确率和召回率的调和平均值，用作模型的性能的综合性评价。计算如公式（3-20）—（3-23）所示。

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (3-20)$$

$$Precision = \frac{TP}{TP + FP} \quad (3-21)$$

$$Recall = \frac{TP}{TP + FN} \quad (3-22)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3-23)$$

其中 TP (True Positive) 表达含义是原本是正类, 同时预测结果为正类。FP (False Positive) 表达含义是原本是负类, 但是预测结果为正类。TN (True Negative) 表达含义原本是负类, 同时预测结果负类。FN (False Negative) 表达含义是原本是正类, 但是预测结果为负类。本节主要采用准确率作为评价指标。本章主要采用准确率作为评价指标。

### 3.1.3.6 实验结果

对比使用 TDE layer 和未使用 TDE layer, 使用 TDE layer 在训练时对于训练样本的 loss 值比不使用 TDE layer 时高, 降低了对于训练样本的过拟合现象, 如图3-15所示, 使得模型在测试集有更好的表现。

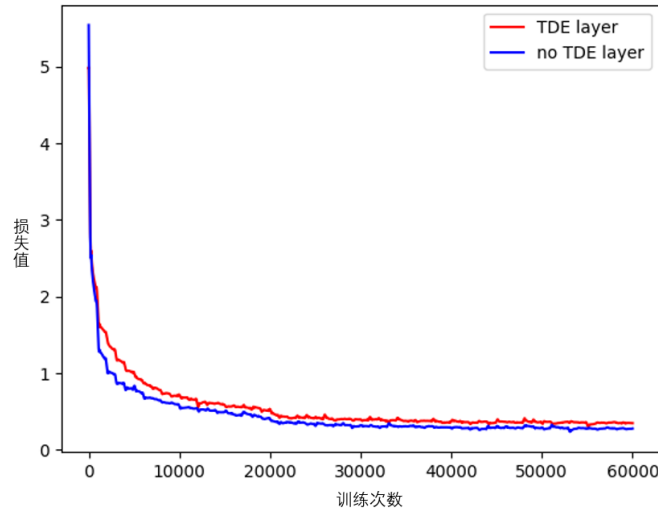


图 3-15 损失值随训练迭代次数变化图

本节对算法3-2中 *sentence\_encoder* 使用 Glove 作为词嵌入生成并使用三种不同的采样方式, 记为分段采样 (PP), 稀疏上下文分段最大采样 (Sparse CPP), 密集上下文分段采样 (Dense CPP)。通过网格搜索法确定模型中的参数, 模型使用参数范围如表3-7所示。

实验首先对比了 dropout 的不同取值对验证集准确率的影响, 结果如图3-16所示, 从图中可以看出当设置 dropout 为 0.5 时, 虽然模型的收敛变得更慢, 但是最

表 3-7 模型 FLOPs 实验参数范围

参数	参数含义	参数范围
B	批大小	[4, 8, 12]
L	句子最大长度	[30, 35, 40]
H	隐藏层大小	[128, 200, 256]
train_iter	最大训练次数	[30000, 40000, 50000]
lr	学习率	[0.01, 0.05, 0.1]
dropout	每次不更新神经元比例	[0.3, 0.5, 0.7]

后验证集准确率提升了。但是随着 dropout 的提升，模型中更多的神经元不更新参数，会导致结果准确率下降。

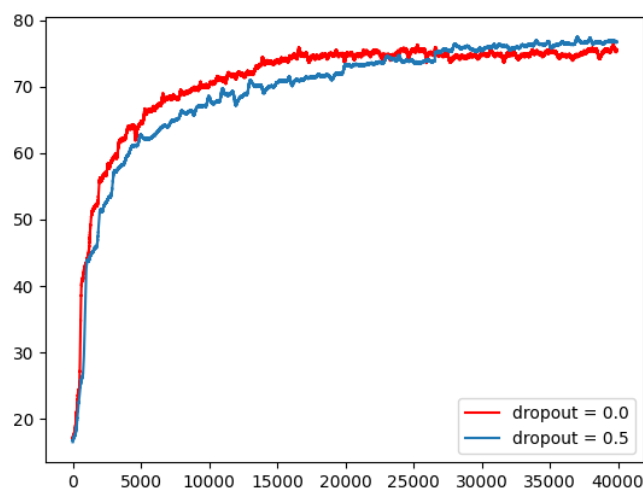


图 3-16 不同 dropout 在验证集上准确率变化 L

根据网格搜索法选出最佳参数如表3-8所示。

表 3-8 模型 FLOPs 实验参数设置

参数	参数含义	参数值
B	批大小	4
L	句子最大长度	40
H	隐藏层大小	200
train_iter	最大训练次数	40000
lr	学习率	0.1
dropout	每次不更新神经元比例	0.5

实验测得模型使用不同的预训练模型和不同采样方式在 5-way 1-shot 条件下

模型的准确率，为了对比传统方法在小样本数据集上的效果，本章对模型最后一层全连接层在验证集的支撑集中训练时进行 Finetune，得到实验结果表3-9。

表 3-9 不同采样方式实验结果

	准确率	前向传播耗时 (ms)
PCNN, Finetune	45.64	2.21
BERT <sup>[27]</sup>	79.83	1217.54
PCNN, PP	67.29	2.26
PCNN, Dense CPP	70.26	2.46
PCNN, Sparse CPP	74.12	2.13
PCNN, Sparse CPP, TDE layer	75.71	2.25

由表3-9可知，本章设计的 Sparse CPP 对比原来的分段最大池化采样效果有所提升，且使用 TDE layer 对结果也有一定提升。

## 3.2 基于双向简单神经注意力元学习器模型的关系抽取研究

### 3.2.1 引言

将处理 FewRel 数据库的模型方法应用到关系抽取系统中还存在一个问题，即是并不是每次的所有关系种类只有几个类型，如果将数据一次性放入查询集，那么会导致准确率下降。如表3-10所示, 对比起来相同条件下支撑集为 10 类时比支撑集为 5 类时效果约下降 20%。

表 3-10 在不同神经网络下 FewRel 的准确率<sup>[24]</sup>

网络模型	5-way 1-shot	10-way 1-shot
Meta Network	64.46 ± 0.54	53.96 ± 0.56
GNN	66.23 ± 0.75	46.27 ± 0.80
SNAIL	67.29 ± 0.26	53.28 ± 0.27
Prototypical Network	69.20 ± 0.20	56.44 ± 0.22

如果一次性通过在支撑集中放入所有类型的样本判断出查询集样本属于那个类型是非常困难的，折衷的解决方案是多次迭代选出最匹配的类型，如图3-17所示意，对多个支撑集判断当前查询样本的标签。

且面向一些要求比较精准的领域，最好能判断出查询集样本不属于支撑集中任意一类，故在输出查询集标签时还需要增加一个选项，即“以上都不是”。

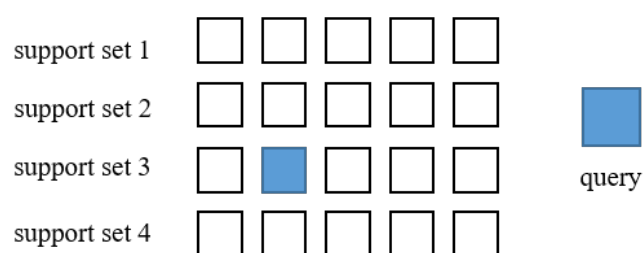


图 3-17 通过迭代所有类型样本得到查询样本类型

### 3.2.2 算法设计

本节算法与本章第二节类似，为了适配“以上都不是”的训练方式，需要在数据处理和模型结构部分作出相应调整和优化，本节对模型结构的调整主要是将单向的 SNAIL 调整为双向的结构。

Graves 等在语音识别任务上探究不同长短期记忆模型参数对语音识别错误率的影响<sup>[58]</sup>，其中 L 表示隐藏层层数，H 是隐藏层大小，UNI 表示单向。从表3-11中可以看出当 LSTM 的模型在大小相同的情况下，双向 LSTM 比单向错误率低，效果更好。

表 3-11 不同 LSTM 参数在语音识别上错误率（部分）<sup>[58]</sup>

模型参数	模型大小	训练周期次数	错误率
CTC-3L-421H-UNI	3.8M	115	19.6 %
CTC-3L-250H	3.8M	124	18.6 %

双向循环神经网络的结构如图3-18所示。

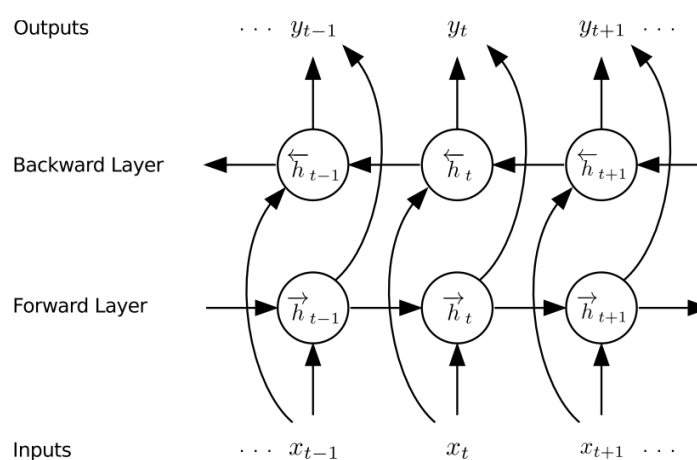


图 3-18 双向循环神经网络结构

自然语言领域热门的网络模型 Transformers 也具有双向结构, 受到以上实例的启发, 本文对 SNAIL 的模型结构进行调整, 使其具有双向结构, 能够捕获前后文, 记为双向简单神经注意力元学习器 (Bi-direction SNAIL, Bi-SNAIL)。模型结构如图3-19所示。

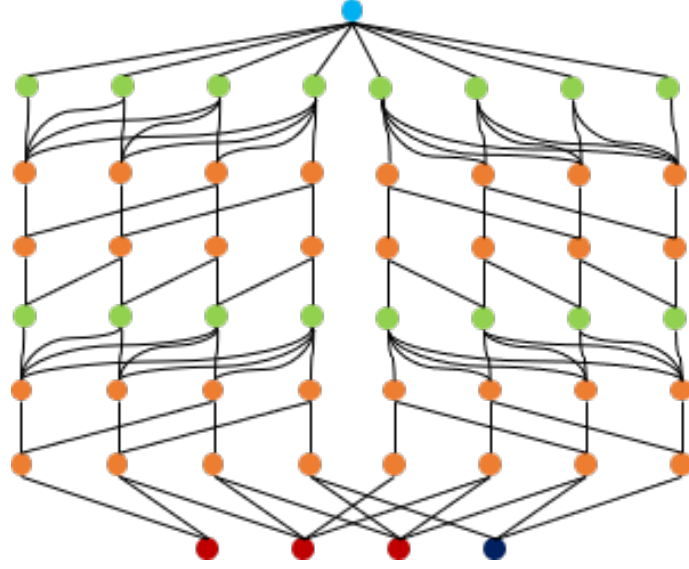


图 3-19 Bi-SNAIL 模型结构

对于模型的输出, 我们最终希望得到查询样本的标签属于支撑集中数据的标签之一或者是“以上都不是”标签, 记我们希望得到的所有的可能输出的集合为  $O$ , 则  $O$  可以表示为式 (3-24)。

$$O = \{o_1, \dots, o_N, o_{NOTA}\} \quad (3-24)$$

这样我们可以计算预测标签的概率, 从而选择最大概率作为模型标注标签, 对于输入查询样本  $x$ , 其被标注为  $r$  的标签概率可以表示为式3-25。其中  $r' \in \mathcal{R}$  表示每个 batch 的训练样本包含的实体关系类型。

$$p(y = r|x) = \frac{\exp(o_r)}{\sum_{r'} \exp(o_{r'})} \quad (3-25)$$

下面讨论如何获取输出集合  $O$ 。将查询集样本  $x$  与支撑集中每个样本如第  $r$  类第  $j$  个样本  $x_r^j$  组合成对  $(x, x_r^j)$  作为模型输入, 记模型为  $M$ 。  $o_i$  ( $i \leq N$ ) 如式 (3-26) 所示, 其中  $M(x, x_r^j)$  表示将查询样本和有标注样本联合输入, 输出为一个有两个元素的向量。  $[\cdot]_1$  表示取向量的第 2 个元素 (下标为 1)。

$$o_i = \frac{1}{K} \sum_{j=1}^K [M(x, x_r^j)]_1 \quad (3-26)$$

$o_{NOTA}$  可以参考 BERT-PAIR, 如式 (3-27) 所示。

$$o_{NOTA} = \min_{r \in r_1, \dots, r_N} \frac{1}{K} \sum_{j=1}^K [M(x, x_r^j)]_0 \quad (3-27)$$



### 3.2.3 实验设计与结果分析

#### 3.2.3.1 数据集构造

本节主要讨论如何构造模型输入对  $(x, x_r^j)$ 。设小样本关系抽取数据集包含的关系类型集合为  $\mathcal{R} = \{r_1, \dots, r_i, \dots\}$ , 其中第  $i$  类所含的样本集合为  $E_i = \{e_i^1, \dots, e_i^j, \dots\}$ 。记每个样本为  $e = \{w_1, \dots, w_l\}$ , 其中  $w$  表示句子中的 token, 即是标点符号或者单词。为了训练模型判断“以上都不是”的能力, 我们需要构造的查询集中有样本不属于任何一类支撑集样本。设训练集为  $T^{NOTA}$ , 如式 (3-28), 表示  $T^{NOTA}$  由  $n$  组支撑集和查询集组成。其中第  $i$  组的支撑集  $S_i$  包含  $N$  类样本, 每类样本  $K$  个实例。

$$T^{NOTA} = \{\{S_1, Q_1\}, \dots, \{S_n, Q_n\}\} \quad (3-28)$$

第  $i$  组支撑集可以表示为式 (3-29), 表示每类样本  $K$  个实例。

$$S_i = \{e_{r_{i,1}}^1, \dots, e_{r_{i,1}}^k, \dots, e_{r_{i,N}}^1, \dots, e_{r_{i,N}}^k\} \quad (3-29)$$

考虑到在有 25 类关系的应用中, 如果使用 5-way 的训练模型, 需要进行 5 轮的筛查才能获取真实标签, 故本章训练时查询集中以上都是不是的占比为 80%, 第  $i$  组查询集如式 (3-30), 其中  $\{e_{r_{i,1}}^1, \dots, e_{r_{i,1}}^k, e_{r_{i,1}}^{k+1}\}$  表示从  $r_i$  关系类型样本中随机抽取的  $K+1$  个样本,  $\{e_{r_{N+1,1}}, \dots, e_{r_{N+1,N}}\}$  表示不属于支撑集中类型的样本。

$$Q_i = \{e_{r_{i,1}}^{k+1}, \dots, e_{r_{i,N}}^{k+1}, e_{r_{i,N+1}}, \dots, e_{r_{i,N+3N}}\} \quad (3-30)$$

对比没有“以上都不是”的小样本关系抽取训练, 本实验在查询集中对于每类查询样本增加了一个随机其它类型的样本。对于样本  $e$ , 使用词嵌入网络层 Glove 可以得到词嵌入  $x$ , 如式 (3-31) 所示。

$$x = \text{Glove}(e) = (v_1, v_2, \dots, v_l) \quad (3-31)$$

则经过嵌入层获取的支撑集如式 (3-32) 所示, 其中  $r$  表示  $N$  类关系中的一种,  $j$  表示第  $r$  类关系中第  $j$  个样本。

$$S = \{x_1^1, \dots, x_r^j, \dots, x_N^K\} \quad (3-32)$$

则对于查询集中的样本可以同样得到句子词嵌入  $x$ , 用该句子词嵌入  $x$  与集合  $S$  中的句子词嵌入可以得到  $N \cdot K$  个输入对  $(x, x_r^j)$ 。

#### 3.2.3.2 评价指标

本文最终实现的基于神经网络的关系抽取系统有一个重要的应用场景, 即是在实体标注后, 帮助标注人员从几十种关系中推荐合适关系类型。评价指标  $\text{hit}@d$  表示如果模型输出标签概率排序前  $d$  的推荐关系中有一个属于正确的标签, 那么记为模型预测正确。设查询样本标签为  $r_q$ , 模型输出预测值概率为  $P = \{r_1, \dots, r_d\}$ , 如果  $r_q \in P$ , 则记为成功。本章主要使用  $\text{hit}@5$  与  $\text{hit}@10$  作为评

价指标。

### 3.2.3.3 模型训练

本章的关系分类实际上也是一个多分类模型，使用常用的交叉熵损失函数作为模型的损失函数，如式（3-11）所示。本章在构造训练集部分与算法3-2相同，其中对于每组数据的训练算法有所不同，算法3-3表示对每一批数据的训练过程。

#### 算法 3-3 对每一批数据使用 SNAIL 计算关系类型

**Input:** 支撑集样本类型数目  $N$ ，每类样本数量  $K$ ， $BT$  表示一批小样本分类数据集，大小为  $B$ 。

**Output:** 更新后全连接层权重  $\tilde{W}$

```

1 begin
2   for epoch in {0, ..., train_iter} do
3     (BS, BQ, BL) ← BT
4     Bx ← ∅
5     for S in BS, Q in BQ do
6       for es in S do
7         xq ← Glove(eq)
8         for eq in Q do
9           xs ← Glove(es)
10          Bx ∪ (xs, xq)
11        end
12      end
13    end
14    logits ← model(Bx)
15    ls = loss(logits, BL)
16    ls.backward()
17  end
18 end

```

### 3.2.3.4 对比模型

本节主要使用 BERT-PAIR 作为对比模型，来说明本章模型的优缺点。BERT-PAIR 将查询集中的样本与支撑集的样本两两组成一对，将样本链接在一起后输入

BERT 中得到一个得分，得分最高的被认为特征最相似，即是一个类别来达到分类的目的。设输入为  $T$ ， $T$  如式 (3-33)。

$$T = \{S = \{(x_1^1, y_1), (x_2^1, y_1), \dots, (x_{n-k+1}^k, y_k), \dots, (x_n^k, y_k)\}, Q = \{\tilde{x}\}\} \quad (3-33)$$

类似式 (3-25) 将模型输出转为概率分布，其中输出  $o_r$  如式 (3-34) 所示。

$$o_r = \frac{1}{K} \sum_{j=1}^K [\mathbf{B}(x, x_r^j)]_1 \quad (3-34)$$

其中  $j \leq K$ ， $K$  表示每个类型有  $K$  个训练样本， $\mathbf{B}$  表示 BERT， $\mathbf{B}(\tilde{x}, x_r^j)$  输出为长度为 2 的向量，代表两种概率，其中维度 0 表示“以上都不是的概率”，维度 1 表示在支撑集中的概率。 $[\cdot]_i$  表示向量的第  $i$  个元素。

### 3.2.3.5 实验结果

本文使用网格搜索法最终确定了最优参数如表3-12所示，其中句子最大长度由于现在模型输入是合并了两个句子成对输入，所以基本增加一倍。

表 3-12 实验参数设置

参数	参数含义	参数值
B	批大小	10
L	句子最大长度	100
H	隐藏层大小	256
train_iter	最大训练次数	20000
lr	学习率	0.05
dropout	每次不更新神经元比例	0.5

本文分别对比了 5-way 1-shot 和 5-way 5-shot 的条件下，测试集中查询样本总数中出现 0%，40%，60%，80% 的“以上都不是”类型时模型的准确率，结果如表3-13所示。

表 3-13 实验结果 (%)

NOTA 占比	0%	40%	60%	80%
BERT-PAIR(5,1)	72.41	73.67	74.12	75.51
Bi-SNAIL(5,1)	74.32	72.83	70.28	69.13
SNAIL(5,1)	70.21	66.28	61.37	56.92
BERT-PAIR(5,5)	77.81	79.33	78.54	76.06
Bi-SNAIL(5,5)	80.21	78.27	75.31	72.12
SNAIL(5,5)	75.22	71.51	66.12	60.86

从表中可以看出,在 5-way 1-shot 和 5-way 5-shot 条件下,本文提出的 Bi-SNAIL 对比 SNAIL 在本文设置的 NOTA 占比条件下效果均有提升。对比 BERT-PAIR,在测试集中 NOTA 占比为 0% 时,BI-SNAIL 效果比 BERT-PAIR 的准确率高,但是当测试集中 NOTA 占比不为 0% 时,BERT-PAIR 的效果比 BI-SNAIL 高,不过差距相对比较小,差距最大的情况是在 5-way 1-shot 的条件下测试集中 NOTA 占比为 80% 时,为 6.83%,在 5-way 5-shot 时,这个差距减小到 3.94%。由本章第一节的模型复杂度实验的结论可以得知,本文的模型在响应时间上具有优势。下面说明在本文提到的实际应用中,准确率的差距是可以接受的。

考虑在向标注人员进行关系类型推荐的应用场景中,希望在输入是实体对,句子的情况下能够在推荐的前几个推荐选项中包含正确结果,因此本文还测试了模型的 hit@5 和 hit@10,结果如表3-14所示。由表3-14可知,在实际应用中,即便不使用复杂的 BERT-PAIR 模型,系统也基本能够完成关系推荐的功能,帮助标注人员快速筛选到合适的选项。

表 3-14 实验结果

NOTA 占比	成功率 (%)	平均响应时间 (ms)
BERT-PAIR (5, 1) hit@5	96.4	1023.14
Bi-SNAIL (5, 1) hit@5	94.3	2.53
Bi-SNAIL (5, 1) hit@10	96.2	2.47

### 3.3 本章小结

本章第一部分从理论上分析对比了三种词嵌入表示的浮点运算量,根据分析实验说明了获取词嵌入使用动态预训练模型会占用大量响应时间,故改而使用静态预训练模型。为了提高因为改用静态预训练模型而损失的准确率,根据一些语序交换不影响人类的阅读的启发,提出了 TDE layer 交换输入的顺序的可训练数据增强方式,验证了该方式在小样本学习中能提高模型泛化能力;再考虑到模型由于使用静态模型,降低了上下文的交互,提出上下文相关采样方式。通过使用 Glove 替换耗时的 BERT 来获取词嵌入,并且对模型其它部分优化来提升算法的响应速度的同时将准确率的损失减小到可接受程度,并且在公开数据集 FewRel 上进行实验验证了改进的效果。本章第二部分主要是在实际应用中还会出现的”“以上都不是”问题,受到自然语言处理模型中常见的双向结构启发,对 SNAIL 进行改进使其具有双向结构。实验表明,BI-SNAIL 对比单向的 SNAIL 效果有所提升;对比 BERT-PAIR,在使用 hit@5 作为评价指标时,成功率的差距仅有 2.1%。

## 第四章 小样本条件下的关系抽取系统

本章主要工作内容是设计并且实现了一个小样本条件下的关系抽取系统，利用前一章的研究成果，能够在数据量有限以及硬件资源有限条件下利用神经网络模型辅助构建精确的知识图谱。并且该系统有完善的权限管理，能够帮助研究团队更好的分工合作。

### 4.1 引言

关系抽取的目的实际上是为了构建知识图谱，知识图谱的概念最早是在 2012 年由谷歌公司工程师 Singhal 在其博客中说明的<sup>[59]</sup>，目的是增强谷歌搜索引擎的联想搜索能力。知识图谱发展到现在应用越来越广泛，比如常见的智能客服问答系统，反欺诈系统等。一般来说，知识图谱  $G$  能够表示为表达式 (4-1)。

$$G = (E, R, S) \quad (4-1)$$

其中  $E$  表示知识图谱中的实体也就是节点的集合， $R$  表示知识图谱中的关系也就是边的集合， $S \subseteq E \times R \times E$  表示图谱中的三元组集合。三元组的主要形式有两种，一种是实体与实体之间的关系：（头实体，关系，尾实体），比如（武侯祠，位于，成都）；第二种是实体自身的特征：（实体，属性，属性值），比如（张三，身高，174cm）。徐增林等给出了如图4-1所示的知识图谱的体系架构<sup>[60]</sup>，该架构说明了知识图谱的主要构建过程。

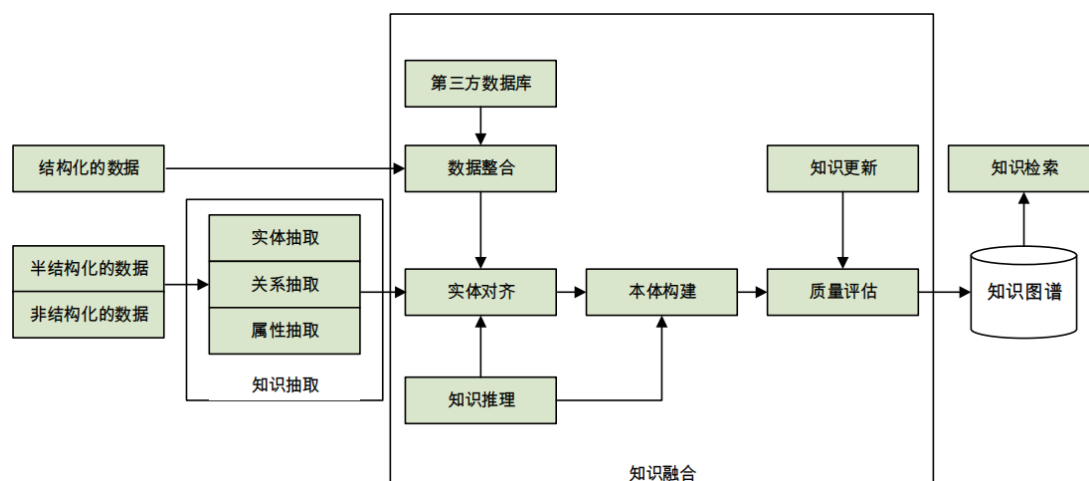


图 4-1 知识图谱体系架构<sup>[60]</sup>

在构建知识图谱之前，需要对当前领域知识体系进行本体建模。本体是领域知识、概念、实体及其关系的一种明确的、规范的概念化描述<sup>[61]</sup>。一个本体由概

念、实例、关系、公理等基本元素组成组成，此处的公理并不是数学意义上的严格公理。比如对于本体“学校”的概念可以定义为有计划、有组织、有系统的进行教育活动的重要场所，实例可以是电子科技大学，关系可以表示为本体“学校”和本体“城市”存在（“学校”，“位于”，“城市”）的关系，公理可以表示为“学校必定位于某座城市”。

本文根据本体建模理论以及图4-1的知识图谱体系架构设计实现了完善的精确知识图谱构建系统，并且实验表明，在有三个工作人员的协同工作场景下，使用本系统进行关系抽取的准确率能够达到 99% 以上。

## 4.2 系统设计

本章设计实现的系统主要的应用场景是针对某些精确数据领域的知识图谱构建系统，系统结构图如图4-2所示。本系统目标是实现一个能够团队协作的精密知识图谱构建工具，且能够利用的神经网络模型对关系进行推荐标注。系统用户应该包括管理员和不同角色的分工配合。比如在标注场景下，首先使用深度学习模型对实体关系作一个辅助推荐标注，后由角色标注员完整校验，之后数据被角色校验员审核上传到知识库。小样本关系抽取模型将被应用于关系标注推荐，具体来说就是使用所有已知关系对某句子中实体对标注，并且能够排序符合条件的标签。

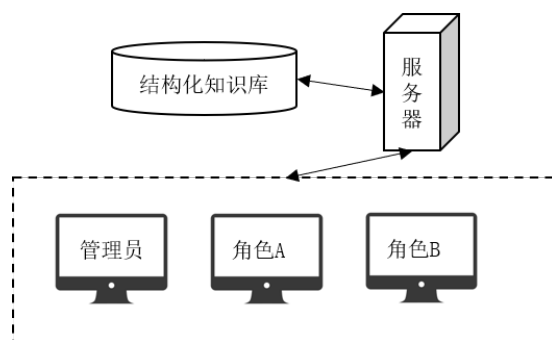


图 4-2 系统结构图

本节通过对知识图谱构建过程中的典型的工作流程进行分析，给出系统的用例图，根据系统的用例图对系统的功能模块进行设计。

### 4.2.1 流程分析

本系统是一个需要团队协作的精确知识图谱构建系统，需要为结构化知识库创建一批精确的关系数据。首先需要不同角色分工配合，故需要管理员进行角色组管理，权限管理，角色创建管理。

管理员的角色权限管理相关包括：

(1) 创建角色组，分配权限，比如标注角色组，标注角色组仅拥有查看指定项目数据，对自己的数据进行标注的权限。本系统的大部分操作都需要有权限设置，只有该角色组拥有操作权限，角色组成员才能进行相应的操作。管理员拥有所有权限，并且能够进行权限分配；

(2) 创建角色，分配角色到角色组，比如标注员张三属于标注角色组，张三仅拥有标注角色组的权限。

本体管理主要是对本体中的实体、属性、关系等相关内容进行管理, 本体模型管理应当主要包含：

(1) 本体类的创建，包括给出属性，定义以及实例；

(2) 创建本体类后还需要能够对本体类的描述进行修改，其中对本体类的修改还需要能够关联地修改知识图谱中的具体实例，如删除某个本体，那么知识图谱中相应本体类的实例应当被删除；

(3) 在创建一定本体后就可以创建本体类之间的本体关系；

(4) 系统还应当对本体模型有一个全局的预览功能。

本体建模完成后就可以对非结构化的文本数据进行实体关系抽取，实体关系抽取应当包含以下功能：

(1) 标签管理，其中关系标签和本体关系中定义应当是一一对应的，由此关系标签即是达到了约束实体之间的关系的目的是，比如人物与人物之间一般不会出现“位于”这样的关系。为了标签创建的便捷性，标签可以由本体模型自动生成；

(2) 实体关系的标注，该功能应当支持 AI 辅助标注，即是使用小样本关系抽取模型辅助标注。一个完整的知识图谱构建应当是一个不断完善的过程，本系统应当支持将数据进行分批次的任务化整合分配，然后由其它角色组的成员收到任务进行一个任务一个任务的完成，比如将一千条数据作为一个任务进行分配；

(3) 标注数据统计，对标注的数据查看完成进度，标注数据情况，标注差异；

(4) 标注结果审核，对于标注完成的数据进行审核导入结构化知识库，查看每条标注数据是否分配给标注人员的都是标注一致还是标注不一致，对于不一致的数据进行处理。

经过以上步骤，通过本系统可以创建一个精确的知识图谱，对于得到图谱本系统应当还需要可以提供一些基本的接口以支持一些基本的应用：

(1) 图谱预览，如导出某节点的邻接关系，查询图谱中实体关系数量等；

(2) 图谱校正，对知识图谱中的节点和边进行直接编辑，更改某节点的信息，达到在线矫正的目的。

(3) 支持一些简单的推理功能, 如查询任意两个节点的关联关系。

### 4.2.2 用例分析

通过对应用场景的典型工作流程分析，本小节给出图4-3所示的系统用例图。

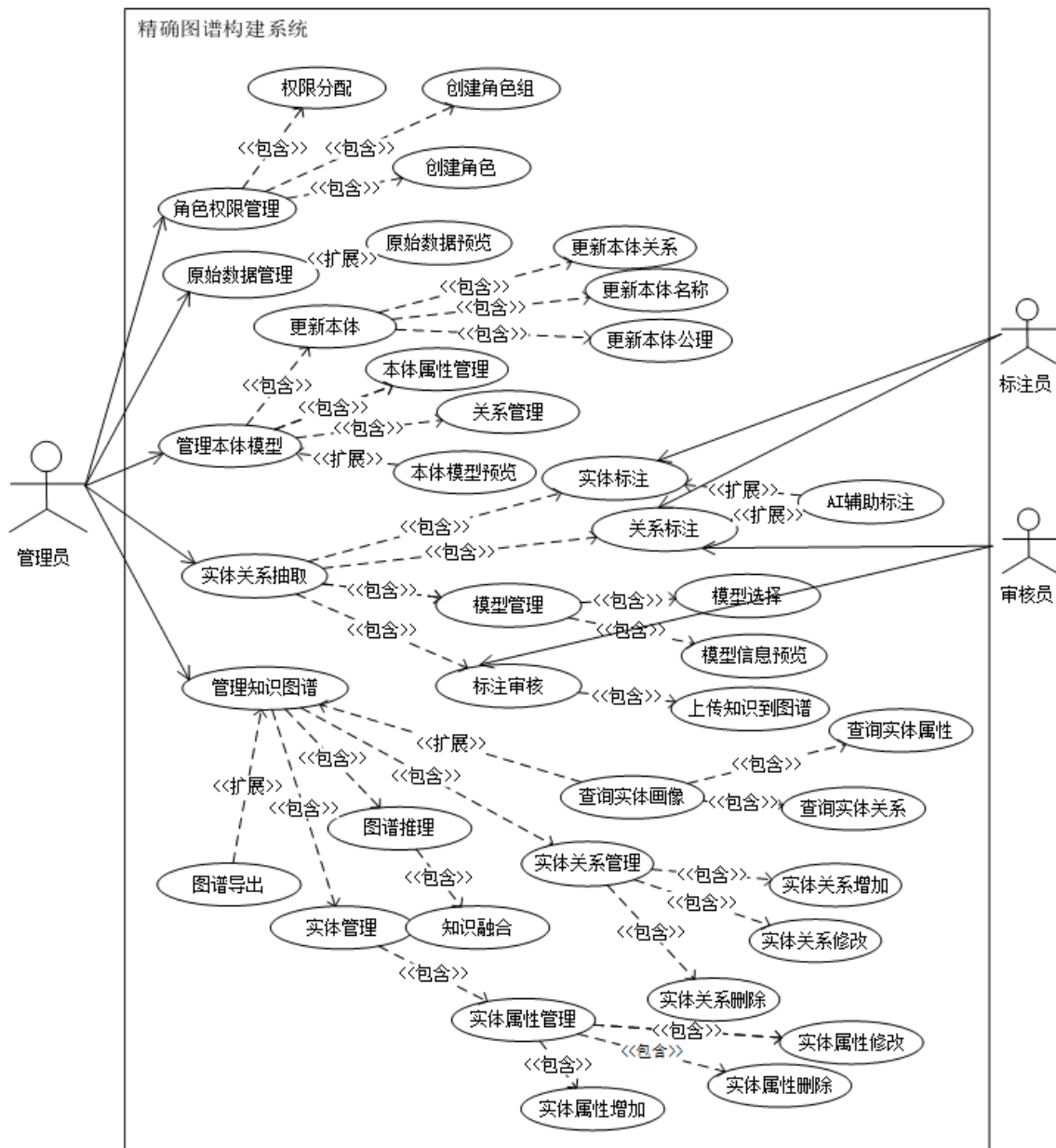


图 4-3 系统用例图图

图4-3中的标注员和审核员实际就是由管理员创建的两个角色组，其中标注员仅拥有实体标注和关系标注的权限，而审核员则是多拥有了标注审核的权限。实际应用管理员还可以创建其它不同的角色组以达到分工合作的目的。用例中的 AI 辅助标注功能的后端主要实现就是利用到第三章训练得到的模型。



### 4.2.3 系统功能设计

由以上主要工作流程可以得到系统功能模块图如图4-4所示。

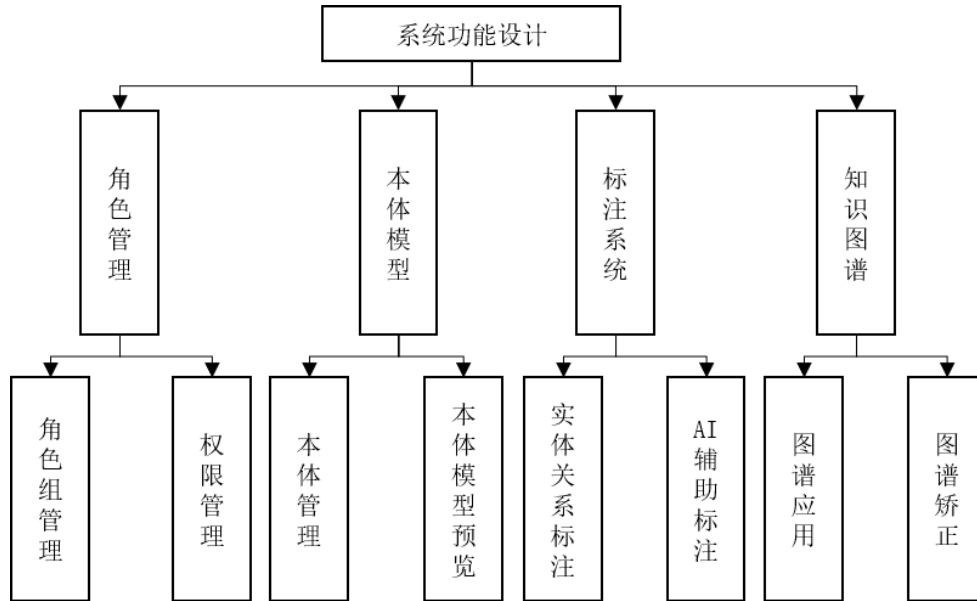


图 4-4 实体关系智能标注系统功能设计图

其中角色管理即是包含角色组权限分配，角色组创建，角色分配角色组；本体模型模块主要是负责本体模型的增删查改，以及本体模型预览；标注系统是本系统的核心，能够提供方便快速的标注功能，并且使用神经网络模型辅助标注；知识图谱模块则包含知识图谱的矫正和基本应用。

### 4.3 系统实现

为了简明扼要的说明系统的实现方法，本小节主要描述功能模块中标注系统的实现，系统实现采用模型-视图-视图模型（Model-View-ViewModel，MVVM）设计模式，MVVM 示意图如图所示。

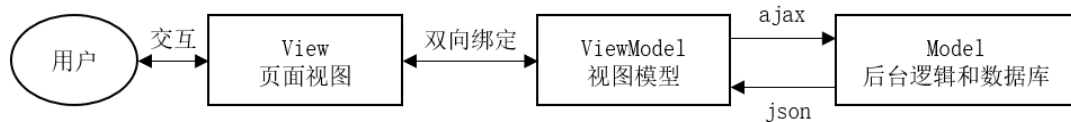


图 4-5 MVVM 示意图

本小节通过标注人员的操作过程，按照图4-5中 MVVM 的顺序说明系统的标注功能实现，其它功能模块实现步骤类似。当标注角色与页面视图交互，新选中一篇需要标注的文章时，因为“document\_id”与视图层双向绑定，此时视图模型

层监听到了“document\_id”的变化，随即向模型层发送请求；模型层随即处理相关逻辑，如加载文章内容，加载文章中的实体标注，加载实体标注的关系标注等返回给视图模型层，视图模型层随即更改相应的视图模型，视图层监听到了对应模块内容更改，再进行渲染。上述过程示意图如图4-6所示。

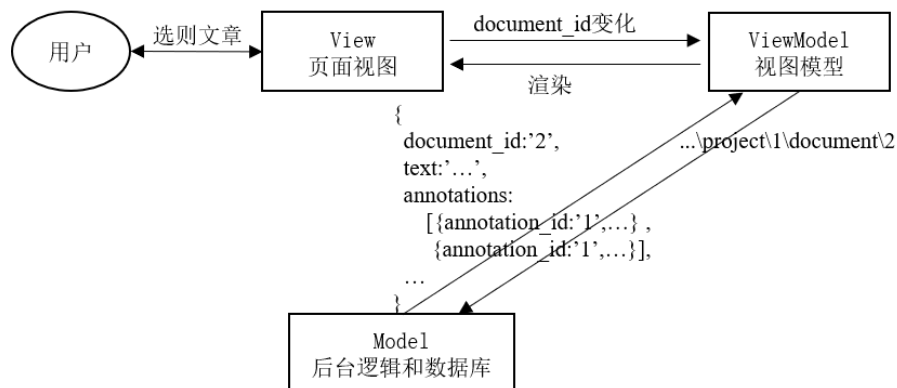


图 4-6 标注人员实际操作时各模块功能

其中模型层数据库的实体关系图如图4-7所示，为了图的简略性，图中省略了部分字段，保留了每个类的关键字段。

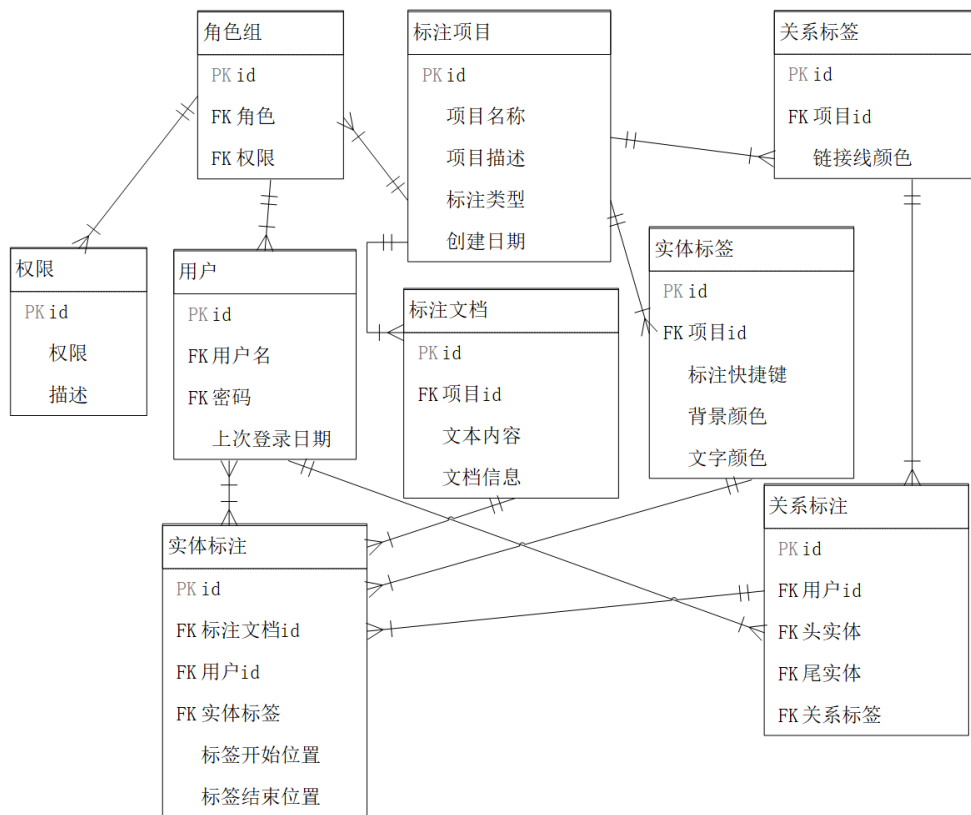


图 4-7 标注系统模型数据库设计层

### 4.3.1 系统环境和系统功能界面

本小节主要说明系统的关系标签管理界面与标注页面。系统主要分为管理员导入或者创建实体标签和关系类型标签，图4-8表示管理员创建关系标签，其中头尾实体类型从下拉框中选择，关系的选择可以约束实体之间的关系类型，标注人员在标注实体关系时可以缩小选择范围。





关联本体1:	14 Transport Hub 交通场所	 
关联本体2:	8 Place 地点	 
ID:	P931	
含义:	place served by transport hub	
公理:	<div>-</div>	
实例:	<div>Tanjung Pandan served by TJQ</div>	

图 4-8 创建关系标签页面（局部）

图4-9为标注页面，标注页面主要提供人工标注，AI 辅助标注，人工矫正标签，预览关系功能。



图 4-9 标注页面

本文实现的系统是在 `doccano` 的基础上二次开发，`doccano` 是一个开源的人工标注工具<sup>[62]</sup>，支持文本分类标注，实体标注功能。本文在该系统的基础上主要增加关系标注功能，并且将小样本关系抽取模型接入该系统，实现了人工标注关系，智能辅助标注，最后人工校验的实际应用。本文实现的系统记为 `RE-doccano`。

Han 等在 `OpenNRE` 中实现了一个在线小样本关系抽取工具需要手动输入支撑集，查询集，头实体和尾实体<sup>[63-64]</sup>，记该工具为 `openNRE-tool`。`RE-doccano` 对比 Han 等设计的系统，增加了关系标注页面，可以落地工程使用。对比 `doccano` 增加了关系标注功能，可以显示实体之间的关系连线，并且以图谱的形式预览篇章关系，功能对比如表4-1所示。

表 4-1 实验结果 (%)

功能	RE-doccano	doccano	openNRE-tool
关系标注	√	×	√
数据导入	√	√	×
推荐排序	√	×	×
对比成员标注结果	√	×	×
结构知识预览	√	×	×

### 4.3.2 人工标注实验

本节的实验分别让 4 位同学组成 2 组作为校验员使用本系统的 AI 辅助标注功能标注 1000 条数据，AI 辅助标注系统会根据每条关系的得分将推荐的关系进行排序，本系统的 AI 辅助标注功能采用 5-way 1-shot 进行训练。让 1 位同学作为管理员审核数据上传到精确知识库。使用 `FewRel` 测试集中共 24 类关系，模拟在工作中使用本系统的场景，测得数据如表4-2所示。

表 4-2 实验结果 (%)

功能	1 组-同学 A	1-组同学 B	2-组同学 A	2-组同学 B
准确率	95.4	94.2	95.7	94.8

本系统可以将不一致的标注结果统计交由管理员审核，其中 1 组和 2 组的详细统计结果如表4-3所示，管理员可以对不一致的标注进行判断，提升准确率。对比 Han 等人进行的人工标注实验准确率为 92.2，本系统将关系标注准确率提升到了 99.5%，使得本系统能够实际应用到了精密知识库构建中。

表 4-3 实验结果 (%)

	1 组	2 组
A 正确, B 正确	93.2	94.1
A, B 不一致	6.1	5.5
A 错误, B 错误	0.7	0.4
管理员审核	99.2	99.5

#### 4.4 本章小结

本章首先说明了将基于神经网络的小样本关系抽取应用于实际系统中还存在需要分批次判断且需要判断标签是以上都不是的情况,再说明本文受到 Bi-LSTM 和 Transformer 的启发,将 SNAIL 修改为具有双向结构的 Bi-SNAIL。对比原型网络结合静态嵌入模型,本文的准确率有所提高。对比 BERT-PAIR 这样的上下文相关预训练模型,本文提出的模型在计算复杂度上有优势。并且说明了本文设计实现的精确知识库构建系统,实验表明在模型辅助标注的情况下,人工关系标注准确率在公开数据集 FewRel 上可以达到 99.5%。

## 第五章 全文总结与展望

自从知识图谱的概念被提出以来，知识图谱的应用产品纷纷被提出来。知识图谱的构建离不开关系抽取，小样本关系抽取能够学习“学习的能力”来利用少量标注的数据来区分无标注数据。对小样本关系抽取的研究对于将文本数据应用到实际应用中有重要意义。

### 5.1 全文总结

本文前两章主要介绍了关系抽取的背景与研究现状，并且说明了现在的研究很多依托于越来越复杂的上下文相关预训练模型，尽管这些模型在准确率方面在一些实验条件下甚至超过人类，但是复杂模型在响应时间和模型可解释性上越走越远，本文主要研究如何优化简单的网络模型 SNAIL 以及将改进的简单模型应用到实际的知识图谱构建系统。

本文首先研究小样本条件下关系抽取研究，提出的第一部分优化主要是数据增强层面以及特征采样部分，通过研究可训练的数据增强层降低了模型在训练集上的过拟合，使用上下文交互采样方式弥补了部分使用静态预训练模型的不足，在小样本关系抽取数据集上验证得到本文提出的改进对简单神经网络模型的结果有提升。本文提出的第二部分优化主要是改变 SNAIL 结构，使其具有双向结构捕捉前后数据的信息来增强模型能力。

本文还将小样本关系抽取模型应用到了精确知识图谱构建系统，介绍了知识图谱和本体构建的基本概念，并且给出了实际工作中的主要工作步骤，分析应用场景设计了系统功能模块。并且实验表明，本文提出的系统的关系推荐功能能够帮助研究人员构建精确关系抽取系统，准确率能够达到百分之九十九以上。

### 5.2 后续工作展望

本文虽然对基于神经网络的小样本关系抽取算法以及应用进行了一定的研究，但是任然有许多不足需要进一步探索：

(1) 本文主要是对 SNAIL 结构进行一定改进来提升准确率，没有充分利用 BERT 等模型丰富的语言知识，我们未来可以尝试使用复杂的预训练模型作为教师神经网络训练简单模型，使得简单模型能够获取更多的语言知识。

(2) 本文建立一个构建精密结构化知识库的应用系统，在实际的应用系统中还存在对全新领域的结构知识库构建，此时希望模型具有一定的跨领域性能，本

文在实验时发现模型能够在训练集达到百分之九十以上准确率，但是在测试集上还有所差距，我们未来可以尝试研究降低过拟合的办法，使得模型在训练集和测试集使用不同领域的数据时在测试集依然能有很好的表现。

（3）本文建立的结构化知识库也就是知识图谱在实际应用中也可以在构建到一定程度后反过来辅助模型进行判断，未来的研究可以继续使用联合知识图谱的关系抽取模型来优化整个系统。

## 致 谢

三年的时光转瞬即逝，这三年来有压力也有鼓励，有困难也有陪伴。

首先，由衷感谢我的导师傅彦教授，感谢傅老师三年前的认可，让我有机会在秋天铺满金黄色的银杏叶的校园忙碌的学习，接受各个学科博学的教授指导课程，认识实验室中一群有风趣又有学识的同学。

感谢我的责任导师陈端兵教授，感谢陈老师几乎一次不落的每周末在团队微信群中消息“今晚组会照常”，我能够每周从不同的同学汇报中吸收论文的精华，督促着我不断的学习；感谢陈教授争取的合作项目，让我能够有机会与学弟学妹一起协同工作，同时培养了我的工程实践能力。

感谢我的父母和我的阿婆，感谢阿婆每次回家总会为我准备我爱吃的火锅，粉蒸排骨，总会从她的卧室提出大包的夏威夷果；无论遇见什么样的问题，我都不会放弃，因为我知道无论什么样的困难，父母总是我最坚强的后盾。

感谢我的至爱郭敏，感谢她清晨的早安，感谢她深夜时在我身边的安静陪伴；感谢她在我烦闷时听我诉说，在我恼火时听我吐槽；感谢她在我每个生日准时的问候；感谢她清凉了我的盛夏，温暖了我的寒夜。

感谢我的研究生三年室友李凌威，我们在寝室和睦相处，一起学习，一起游戏。我们在同一片地方长大，未来也在同一个城市生活，希望我们的感情能如此这般天长地久。

最后，再次感谢三年来所有相遇的人，和你们的相遇，让我收获良多。



## 参考文献

- [1] Wang Y, Yao Q, Kwok J T, et al. Generalizing from a few examples: A survey on few-shot learning[J]. ACM Computing Surveys (CSUR), 2020, 53(3): 1-34.
- [2] Mintz M, Bills S, Snow R, et al. Distant supervision for relation extraction without labeled data [C]. Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2009: 1003-1011.
- [3] Wang Z, Lai K, Li P, et al. Tackling long-tailed relations and uncommon entities in knowledge graph completion[C]. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019: 250-260.
- [4] Han X, Gao T, Lin Y, et al. More data, more relations, more context and more openness: A review and outlook for relation extraction[A]. 2020. arXiv:2004.03186.
- [5] 谢德鹏, 常青. 关系抽取综述[J]. 计算机应用研究, 2020, 37(07): 1921-1924+1930.
- [6] Hendrickx I, Kim S N, Kozareva Z, et al. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals[J]. SEW-2009 Semantic Evaluations: Recent Achievements and Future Directions, 2009: 94.
- [7] Miller S, Fox H, Ramshaw L, et al. A novel use of statistical parsing to extract information from text[C]. 1st Meeting of the North American Chapter of the Association for Computational Linguistics, 2000: 226-233.
- [8] 崔雷. 生物医学实体关系抽取的研究[J]. 中华医学图书情报杂志, 2010, 19(05): 5-10.
- [9] 甘丽新, 万常选, 刘德喜, 等. 基于句法语义特征的中文实体关系抽取[J]. 计算机研究与发展, 2016, 53(2): 284-302.
- [10] Abe N. Query learning strategies using boosting and bagging[C]. 1998: 1-9.
- [11] Chen J, Ji D, Tan C L, et al. Relation extraction using label propagation based semi-supervised learning[C]. Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, 2006: 129-136.
- [12] Brin S. Extracting patterns and relations from the world wide web[C]. International workshop on the world wide web and databases, 1998: 172-183.
- [13] 张立邦. 基于半监督学习的中文电子病历分词和名实体挖掘[M]. 哈尔滨工业大学, 2014.

- [14] Hasegawa T, Sekine S, Grishman R. Discovering relations among named entities from large corpora[C]. Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04), 2004: 415-422.
- [15] 王传栋, 徐娇, 张永. 实体关系抽取综述[J]. 计算机工程与应用, 2020, 56(12): 25-36.
- [16] 鄂海红, 张文静, 肖思琪, 等. 深度学习实体关系抽取研究综述[J/OL]. 软件学报, 2019, 30(06): 1793-1818. [https. DOI: 10.13328/j.cnki.jos.005817](https://doi.org/10.13328/j.cnki.jos.005817).
- [17] Socher R, Huval B, Manning C D, et al. Semantic compositionality through recursive matrix-vector spaces[C]. Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning, 2012: 1201-1211.
- [18] Sundermeyer M, Schlüter R, Ney H. Lstm neural networks for language modeling[C]. Thirteenth annual conference of the international speech communication association, 2012.
- [19] Zhang Y, Qi P, Manning C D. Graph convolution over pruned dependency trees improves relation extraction[C]. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018: 2205-2215.
- [20] Munkhdalai T, Yu H. Meta networks.[C]. Proceedings of Machine Learning Research, 2017: 2554-2563.
- [21] Lake B M, Salakhutdinov R, Tenenbaum J B. Human-level concept learning through probabilistic program induction[J]. Science, 2015, 350(6266): 1332-1338.
- [22] Satorras V G, Estrach J B. Few-shot learning with graph neural networks[C]. International Conference on Learning Representations, 2018.
- [23] Mishra N, Rohaninejad M, Chen X, et al. A simple neural attentive meta-learner[C]. International Conference on Learning Representations, 2018.
- [24] Han X, Zhu H, Yu P, et al. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation[C]. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018: 4803-4809.
- [25] Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning[C]. Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017: 4080-4090.
- [26] Soares L B, Fitzgerald N, Ling J, et al. Matching the blanks: Distributional similarity for relation learning[C]. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019: 2895-2905.

- [27] Gao T, Han X, Zhu H, et al. Fewrel 2.0: Towards more challenging few-shot relation classification [C]. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019: 6250-6255.
- [28] Rosenblatt F. The perceptron: a probabilistic model for information storage and organization in the brain.[J]. Psychological review, 1958, 65(6): 386.
- [29] 石磊, 王毅, 成颖, 等. 自然语言处理中的注意力机制研究综述[J]. 数据分析与知识发现, 2020, 4(05): 1-14.
- [30] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[M]. Advances in neural information processing systems. 2017: 5998-6008.
- [31] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]. International Conference on Machine Learning, 2010: 807-814.
- [32] Arora R, Basu A, Mianjy P, et al. Understanding deep neural networks with rectified linear units [C]. International Conference on Learning Representations, 2018.
- [33] Maas A L, Hannun A Y, Ng A Y, et al. Rectifier nonlinearities improve neural network acoustic models[C]. Proc. icml: volume 30, 2013: 3.
- [34] Nwankpa C E, Ijomah W, Gachagan A, et al. Activation functions: comparison of trends in practice and research for deep learning[C]. 2nd International Conference on Computational Sciences and Technology, 2021: 124-133.
- [35] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]. International Conference on Machine Learning, 2015: 448-456.
- [36] Santurkar S, Tsipras D, Ilyas A, et al. How does batch normalization help optimization?[C]. Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018: 2488-2498.
- [37] Li H, Xu Z, Taylor G, et al. Visualizing the loss landscape of neural nets[C]. Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018: 6391-6401.
- [38] Ba J L, Kiros J R, Hinton G E. Layer normalization[A]. 2016. arXiv:1607.06450.
- [39] 姬壮伟. 基于 pytorch 的神经网络优化算法研究[J]. 山西大同大学学报 (自然科学版), 2020, 36(06): 51-53.
- [40] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [A]. 2013. arXiv:1301.3781.

- [41] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation[C]. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014: 1532-1543.
- [42] Liu C, Sun W, Chao W, et al. Convolution neural network for relation extraction[C]. International Conference on Advanced Data Mining and Applications, 2013: 231-242.
- [43] Kumar S. A survey of deep learning methods for relation extraction[A]. 2017. arXiv:1705.03645.
- [44] Zeng D, Liu K, Lai S, et al. Relation classification via convolutional deep neural network[C]. Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014: 2335-2344.
- [45] Hu H, Liu P. 小样本关系分类研究综述 (Few-Shot Relation Classification: A Survey)[C]. Proceedings of the 19th Chinese National Conference on Computational Linguistics, 2020: 363-375.
- [46] Xu B, Wang N, Chen T, et al. Empirical evaluation of rectified activations in convolutional network[A]. 2015. arXiv:1505.00853.
- [47] 岳增营, 叶霞, 刘睿珩. 基于语言模型的预训练技术研究综述[J]. 中文信息学报, 2021, 35 (09): 15-29.
- [48] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[C]. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019: 4171-4186.
- [49] Qiu X, Sun T, Xu Y, et al. Pre-trained models for natural language processing: A survey[J]. Science China Technological Sciences, 2020: 1-26.
- [50] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016: 770-778.
- [51] Chen T, Wang N, Wang H, et al. Distant supervision for relation extraction with sentence selection and interaction representation[J]. Wireless Communications and Mobile Computing, 2021, 2021.
- [52] Zeng D, Liu K, Chen Y, et al. Distant supervision for relation extraction via piecewise convolutional neural networks[C]. Proceedings of the 2015 Conference on Empirical methods in Natural Language Processing, 2015: 1753-1762.
- [53] Goyal P, Dollár P, Girshick R, et al. Accurate, large minibatch sgd: Training imagenet in 1 hour [A]. 2017. arXiv:1706.02677.

- [54] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors[A]. 2012. arXiv:1207.0580.
- [55] adn Sam Gross A P, Chintala S, Chanan G. Document of pytorch[EB/OL]. 2019. <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>.
- [56] Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library[J]. Advances in Neural Information Processing Systems, 2019, 32: 8026-8037.
- [57] Wolf T, Debut L, Sanh V, et al. Transformers: State-of-the-art natural language processing[C/OL]. Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Online, 2020: 38-45. <https://www.aclweb.org/anthology/2020.emnlp-demos.6>.
- [58] Graves A, Mohamed A r, Hinton G. Speech recognition with deep recurrent neural networks [C]. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013: 6645-6649.
- [59] Singhal A. Introducing the knowledge graph: things, not strings[EB/OL]. <https://www.blog.google/products/search/introducing-knowledge-graph-things-not/>.
- [60] 徐增林, 盛泳潘, 贺丽荣, 等. 知识图谱技术综述[J]. 电子科技大学学报, 2016, 45(4): 589-606.
- [61] 王向前, 张宝隆, 李慧宗. 本体研究综述[J]. 情报杂志, 2016, 35(6): 163-170.
- [62] Nakayama H, Kubo T, Kamura J, et al. doccano: Text annotation tool for human[EB/OL]. 2021. <https://github.com/doccano/doccano>.
- [63] Han X, Gao T, Yao Y, et al. Online tool for relation extraction[EB/OL]. 2019. [http://opennre.thunlp.ai/#/fewshot\\_re](http://opennre.thunlp.ai/#/fewshot_re).
- [64] Han X, Gao T, Yao Y, et al. Opennre: An open and extensible toolkit for neural relation extraction [J]. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019: 169.