南京大学

研 究 生 毕 业 论 文

（申请硕士学位）

论 文 题 目	基于卷积神经网络的实体关系抽取研究
作 者 姓 名	王　强
学科、专业名称	计算机技术
研 究 方 向	数据挖掘
指 导 教 师	李宁 副教授

2017 年 5 月 26 日

学　　号：MF1433046

论文答辩日期：2017 年 5 月 26 日

指 导 教 师：　　（签字）

南京大学申请硕士学位论文

基于卷积神经网络的实体关系抽取研究

作　　者：　**王强**

专　　业：　**计算机技术**

研究方向：　**数据挖掘**

指导教师：　**李宁　副教授**

南京大学计算机科学与技术系

2017 年 5 月

Research on Entity Relation Extraction Based on Convolutional Neural Network

Presented By

Wang Qiang

Supervised by

Prof. Li Ning

A DISSERTATION
FOR THE APPLICATION OF MASTER DEGREE
SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE
AND TECHONOLOGY OF NANJING UNIVERSITY

May 2017

声　明

本人声明所呈交的论文是我个人在导师指导下、在南京大学及导师提供的研究环境（含标明的项目资助）下作为导师领导的项目组项目整体的组成部分而完成的研究工作及取得的研究成果。

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。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了谢意。

南京大学及导师所有权保留：送交论文的复印件，允许论文被查阅和借阅；公布论文的全部或部分内容；可以采用影印、缩印或其它复制手段保存该论文。

学生签名：　　日期：

导师签名：　　日期：

Declaration

I make a declaration here that the thesis submitted is composed of the researching work by myself and its corresponding researching results finished as a constituent part of the whole project in the project team lead by my advisor. The thesis is completed with the guidance of my advisor, and under the researching circumstances offered by Nanjing University and my advisor (including the project support indicated).

The thesis does not include other people's researching results ever published or composed, except that are specially annotated and acknowledged somewhere in the article. Any contribution made to the research by my working partners is declared explicitly and acknowledged in the thesis.

Nanjing University and the advisor retain the copyright as follows: submitting the copies of the thesis, allowing the thesis to be consulted and borrowed; publicizing the whole or part of the thesis' content; keeping the thesis by photocopy, microcopy or other copy methods.

Author Signature：　　Date：

Advisor Signature：　　Date：

摘　要

随着信息技术的快速发展，互联网上产生了大量的非结构化的文本数据，其中包括：新闻，博客，政府文档，聊天日志等。如何帮助人们快速地从中获取有价值的信息成为计算机相关领域学者所关注的问题。实体关系抽取正是为了解决这样一个问题，它的主要任务是识别非结构化文本中出现的实体并确定实体之间的语义关系。目前基于监督学习的关系抽取方法已经取得了较好的效果，但这类方法比较依赖自然语言处理工具提供分类特征，而这些工具往往存在大量错误，这些错误将会在关系抽取系统中不断传播放大，最终影响关系抽取的效果。为了避免过多依靠复杂的特征工程，本文提出利用卷积神经网络来解决关系抽取问题，其将会自动从句子中学习特征，以构建句子的分布式表示作为关系分类模型的输入，从而最小化对一些 NLP 处理工具和资源的依赖。并在此基础上实现了面向互联网新闻文本的企业实体关系的抽取。

本文的工作主要包括以下几个方面：

1)针对互联网新闻当中存在的企业实体关系，本文提出了基于 Bootstrapping 技术构建关系语料库的方法，该方法克服了纯人工标注过程中费时费力的缺点；

2)针对传统词袋模型在表征句子时缺乏语义信息以及未考虑词的位置信息的缺陷，本文提出了基于词向量加权和基于卷积神经网络的方法用于构建紧凑且具有语义的句子的分布式表示，作为关系分类模型的输入；

3)在前几部分的工作基础上，结合网页正文提取、命名实体识别等关键技术实现了面向互联网新闻文本的企业关系的抽取。

关键词： 实体关系抽取；卷积神经网络；词向量；Bootstrapping

Abstract

With the rapid development of information technology, the Internet has produced a large number of unstructured text data, including news, blog, government documents, chat logs, etc. How to help people get valuable information from the web quickly becomes the problem concerned by the scholars of computer science areas. Entity Relation Extraction is generated in this context. Its main task is to identify the entities in the text and extract the semantic relations between the entities. The existing method based on supervised learning has achieved good results, but they rely heavily on POS, syntactic parsing, and other natural language processing tools to provide classification features. And these tools tend to have a lot of errors. These errors will continue to propagate in the relationship extraction system. In order to avoid relying on complex feature engineering, this paper proposes to use the convolution neural network to solve the problem. It will automatically learn features from the sentence to construct the sentence's Distributed representation as the input of the relational classification model. Thereby minimizing the reliance on some NLP processing tools and resources. And design and build a company relationship extraction system for Internet news text.

The main work of this paper is as follow：

1. In view of the existence of the company entity relation showed in the Internet News, this paper proposes a method of constructing relational corpus based on Bootstrapping technology. This method overcomes the shortcomings of time consuming and laborious in the process of manual labeling.

2. In the case of the shortcoming of the traditional word bag model lacks semantic information and the location information of the word when representing sentences, this paper propose two methods based on weighting of Word Embedding and Convolutional Neural Network to construct a compact and semantic Distributed representation of the sentences. As the input of the relational classification model.

3. On the basis of the first part of the work, combining the key technology of web page content extraction and Named Entity Recognition, this paper constructs the prototype of enterprise relationship extraction system.

Keywords: Entity Relation Extraction; Convolutional Neural Network; Word Embedding; Bootstrapping

目 录

第一章 绪论

1.1 研究背景及意义

随着信息技术的快速发展，互联网数据每天都以指数级的速度快速增长，无论是个人、企业还是政府，每天都会产生大量数据。这其中包括新闻、社交、政府网站数据。在这些数据当中蕴含着许多对人们有价值的信息，这些信息对人们的生产生活起着至关重要的作用。面对这些海量的互联网数据，传统的做法是借助于搜索引擎来获取自己所感兴趣的数据。然而搜索引擎所展示的数据通常是一系列与检索关键字相关的网页，数据的价值密度相对较低，一般需要人工进一步加工才能得到最终所需要的信息。为了应对信息爆炸所带来的挑战，亟需一种自动化的方法帮助人们快速地获取有价值的信息。

信息抽取的相关研究正是在这样的背景下产生的。信息抽取任务主要目的是从自然语言文本当中抽取出特定领域的实体、关系、事件等信息，同时抽取出实体、关系、事件的一些相关属性。更加准确的来说，信息抽取主要是将非结构化的文本信息转化为结构化或半结构化的信息，并以元组的形式存储在数据库当中。信息抽取具有很多方面的实际应用，包括自动问答(Question Answering)、知识库(knowledge base)、bio-text mining等。随着研究的进一步细化，实体关系抽取这一项子任务被越来越多的学者所关注，并取得了一系列不错的研究进展。

实体关系抽取的主要任务是识别非结构化文本中出现的实体并确定实体之间的语义关系[1]。例如，句子“丁磊是网易公司的创始人”中包含一个实体对(丁磊,网易公司)，以及这两个实体对之间的关系为创始人。因此实体关系抽取任务通常包含了两个步骤，首先是通过命名实体识别技术找出句子当中存在的实体对，然后是抽取出实体对之间存在的语义关系。概括的说实体关系抽取的任务就是给定一个句子S，识别出实体对<e_1,e_2>并确定e_1和e_2之间的语义关系 r。最终结果一般以三元组<e_1,e_2,r>的形式存储。

传统实体关系抽取的方法是通过制定模板和规则的方法，该方法需要相关领域知识，并且可移植性较差。后来人们使用统计机器学习的方法来实现实体关系抽取，首先需要人工标注大量的语料库，然后提取句子的各种特征并在此语料库上训练分类模型。然而这种做法的一个主要缺点是模型的最终分类效果将严重依赖于所提取的特征的质量，而

这些特征通常是由预先存在的自然语言处理工具产生的。实际情况是这些处理工具不可避免的会引入一些误差，这些误差将传递到关系分类的模型当中，影响最终的分类效果。

近些年来，深度学习技术(Deep Learning)[2]在图像识别[3]、语音识别[4]和自然语言处理[5]等多个人工智能领域都取得不错的效果。深度学习算法和一些浅层的学习算法相比，它会使用更多的层来对输入数据进行处理，因而可以学习到更加丰富的特征。从仿生学角度来说，深度学习的原理与人的大脑皮层一样，分层对输入数据进行处理。在每一层当中，信号被处理单元接受并做转换，如同神经元接受电信号一样。参数通过训练学习得到，最终得到数据的本质特征[6]。相比于传统的机器学习技术，深度学习技术可以自动学习到数据的特征，将人们从繁杂的特征工程中解放出来。

因此最近很多工作都尝试将深度学习技术应用到实体关系抽取的任务上。借助于词向量(Word Embedding)[1]和深层的神经网络(Deep Neural Network, DNN)[2]，可以学习得到句子级别和文档级别的分布式向量表示。该向量包含了文本的语义信息，可以作为关系抽取分类模型的输入。现有的基于深度学习技术的关系抽取方法都取得了 state-of-the-art 的效果。基于深度学习技术的关系抽取研究已经逐渐成为相关研究领域的热点。

实体关系抽取在 NLP 领域具有许多应用，如文档摘要、机器翻译、知识图谱构建、实体消歧、语言建模等。由此可见对实体关系抽取的研究是十分有意义的。

1.2 实体关系抽取研究现状

实体关系抽取作为信息抽取的重要子任务之一，主要任务是识别非结构化文本中出现的实体并确定实体之间的语义关系。给定一个句子S并标注好句子当中的实体对e_1和e_2，实体关系抽取的任务就是识别出e_1和e_2之间的语义关系。针对实体关系抽取任务，目前主流的方法包括基于半监督学习的方法和基于监督学习的方法。

在基于半监督学习的相关方法中，有的利用了实体所在的上下文信息。它是基于一种叫做分布假说的理论(Distributional hypothesis theory)[7]：出现在同一篇文章中的词往往有着相近的意思。该假说相应的可以扩展为：出现在相似文章当中的实体对往往具有类似的关系。Hasegawa[8]采用层次聚类方法来聚类实体的上下文，并简单地选择了上

[1] https:.en.wikipedia.org/wiki/Word_embedding
[2] https:.en.wikipedia.org/wiki/DNN

下文中出现最频繁的词来代表实体之间的关系。Chen[9]提出了一种基于模型顺序选择和标签识别的半监督方法，以解决这个问题。

基于有监督的学习方法是将关系抽取任务看成是多分类问题。首先是人工标注训练数据，接着是提取各种有效的特征，最后是训练分类模型。总的来说，这些方法可以分成两类：一种是基于特征向量的方法，另一种是基于核函数的方法。对于基于特征向量的方法，通常做法是将句子的一些信息例如词序列、解析树等转换成特征向量[10]。该方法在将结构化表示转换成特征向量时通常面临选择合适的特征集的问题。而基于核函数的方法则无需显示地抽取出特征，但需要定义一个比较好的核函数，目前已有的工作包括：基于卷积树的核函数[11]，子序列核函数[12]以及基于依存语法树的核函数[13]。无论是基于特征向量方法还是基于内核函数的方法，绝大多数的研究重点是如何抽取和选取有效的特征子集，以提升分类模型的性能。然而抽取这些特征不可避免的需要借助于一些自然语言处理工具，这些工具往往会存在误差，影响最终的分类结果。

基于有监督学习的关系抽取方法还面临着一个问题，就是缺少足够多标注过的训练数据集，大量的训练数据集的标注无法完全通过人力来完成。为了解决这样一个问题，Mintz 等人[14]提出了远程监督(Distant Supervision)的思想。他们将 New York Times 中的新闻文本与大规模知识图谱 Freebase[3]当中的 7000 多个关系和超过 9 亿的实体进行实体对齐。远程监督基于这样一种假设，如果一个句子当中同时包含了两个实体，那么可以认为该句子包含了该实体对在 Freebase 中对应的关系，因此可以将该句子作为一条训练语料。通过远程监督这样一种思想，便可以快速地获取大量的训练语料，有效解决了关系抽取的标注数据来源问题。但是该方法存在一个明显的问题，即标注的数据可能不正确将会导致训练数据存在噪音，影响最终模型的分类效果。为了解决这样一个问题，很多人尝试使用不同的方法对远程监督进行改进，以降低标注错误率。比如 Takamatsu 等人[15]改进了实体对齐的方法，减少了数据噪音，提高了最终抽取模型的准确率。Yao 等人[16]提出了基于无向图模型的关系抽取方法。Riedel 等人[17]则增强了远程监督的假设，将标注错误率大大降低。

远程监督思想还存在一个不合理的地方，就是它假定任何一个实体对之间只存在着一种关系。但是实际上很多实体对之间并不仅仅只存在一种关系，往往具有多种关系。

[3] https://en.wikipedia.org/wiki/Freebase

例如，“丁磊”和“网易公司”之间存在多种关系：“丁磊是网易公司的创始人”和“丁磊是网易公司的 CEO”。针对这种情况，Hoffmann 等人[18]提出利用多实例多标签方法来对实体关系抽取进行建模，有效地解决了一个实体对可能存在多种关系的问题。后来，Surdeanu 等人[19]也提出利用多实例多标签结合贝叶斯网络来进行实体关系抽取，同样取得了不错的效果。

最近随着深度学习技术变得越来越火热，很多人尝试将相关技术应用到实体关系抽取任务当中。与传统基于特征工程的方法相比，深度学习技术不需要人工筛选合适的特征子集用于构建分类模型，它会自动从标注过的训练数据中学习特征，从而避免引入一些自然语言处理工具所带来的的误差，大大提高了关系抽取模型的准确率。 Socher[20]最早尝试将递归神经网络应用于实体关系抽取问题当中，该方法首先获得句子的依存句法分析树，然后为树上的每个节点构造向量表示，并将这些节点的向量作为循环神经网络的输入，最终得到该句子的向量表示，用于分类模型的构建。由于只考虑了句子的句法信息，而没有考虑实体所在的上下文信息以及句子当中词语的语义信息，所以该方法的抽取效果提升有限。Zeng[21]最早提出使用卷积神经网络来解决实体关系抽取问题。他们将句子当中的每个词对应的词向量以及词的位置信息结合在一起作为卷积神经网络的输入层，通过卷积操作、池化操作和最终全连接层得到句子的分布式表示，用于之后关系分类模型的输入。

1.3 本文主要工作

本文主要工作围绕实体关系抽取任务的相关问题展开，主要包括三个部分，分别是句子的分布式表示、基于 Bootstrapping 技术构建关系语料库以及面向互联网新闻当中企业实体关系的抽取。需要注意的是，本文只考虑句子级别上的关系抽取。

第一部分针对传统词袋模型在表征句子时缺乏语义信息以及未考虑词的位置信息的缺陷，提出了基于词向量加权和基于卷积神经网络的方法用于构建紧凑且具有语义的句子分布式表示，作为构建关系分类模型的输入。

第二部分针对互联网新闻当中存在的企业实体关系，提出了基于 Bootstrapping 技术构建关系语料库的方法，克服了纯人工标注过程中费时费力的缺点。

第三部分是在前面工作基础上，结合网页正文提取、命名实体识别等关键技术，实现了面向互联网新闻文本的企业关系的抽取。

1.4 本文组织结构

本文后续组织结构如下：

第二章介绍了当今关系抽取领域的相关研究和基本方法，包括基于有监督学习的方法、基于半监督学习的方法和基于深度学习的方法等，其中较为详细地介绍了基于深度学习的方法，因为它们和本文的工作比较接近。并在最后列举出了一些关系抽取的评价策略。

第三章设计了基于词向量加权的方法和基于卷积神经网络的方法用于构建句子的分布式表示，作为关系分类模型的输入。并在标准数据集 SemEval-2010 Task 8 上做了一组对比实验，实验结果表明两种方法都能得到紧凑且具有语义的句子的分布式表示。

第四章主要是在前一章工作的基础上，介绍了面向互联网新闻中企业实体关系抽取的一些关键技术和实现过程。其中包括基于 Bootstrapping 技术构建关系语料库的方法、爬虫模块设计与实现、网页正文提取技术、命名实体识别技术等。

第五章是总结与展望，包括对论文系统的主要工作进行总结并针对本文工作的可完善之处进行展望和探讨。

第二章 关系抽取的相关研究

2.1 基于半监督学习的关系抽取

在许多机器学习的相关任务中，常常有很多现成的未标注数据，如何将这些未标注的数据自动转变成标注过的数据，成为许多研究者所关注的问题。比较常见的做法就是利用一些半监督的学习方法，首先人工标注少量数据，然后在这些少量的数据上学习得到一个弱分类器，在用这个弱分类器去标注剩余数据。

本文将注意力集中在 Yarowsky[22]和 Blum＆Mitchell[23]的算法上，因为之后大部分基于半监督学习的关系抽取方法都使用了这种算法（Co-training[4]）。它们的主要思想是使用弱学习器的输出作为下一次迭代的训练数据(输入)。协同训练是一种弱监督模型，它从一小组标记数据以及使用分开但冗余的大量未标记数据（即使用不相交的特征子集来表示数据）中进行学习任务。为了确保模型的有效性，协同训练（Co-training）要求从两个不同的视角看待训练数据，它假定每个样例使用两个不同的特征集来描述，这两个特征集提供关于样例不同的补充信息。同时特征集必须满足两个条件：(1)特征集对于目标学习任务来说是有效的，(2)对于不同的特征集需要使用独立的分类器去训练。

算法 2.1：Yarowsky’s algorithm in general form

输入：一组未标记的数据集 *D* 以及一组种子数据集 *S*

重复

在数据集 S 上训练出分类器 C
用分类器 C 去标记数据集 D
N = top *n* labels that *C* was highly confident
S = *S*∪*N*
D = *D**N*

直到 达到收敛标准

Yarowsky 在论文中描述的算法框架已经在算法 2.1 给出[24]。Yarowsky 将这个算法应用到了词义消歧任务，并取得了不错的效果。他的这个算法本质上是协同训练算法的一个特例。Abney[25]提出了 Yarowsky 算法的理论分析。当探究相关协同训练的算法时，需要考虑两个主要问题：1)如何自动地获取种子数据集 2)什么样的种子集是好的，即如

[4] https://en.wikipedia.org/wiki/Co-training

何评价种子集。目前效果比较好的基于半监督的关系抽取系统有：DIPRE[26]、Snowball[27]、KnowItAll[28]以及 TextRunner[29]。

2.1.1 DIPRE

DIPRE(Dual Iterative Pattern Relation Expansion)是由 Brin 在 1998 年设计出的关系抽取系统。它主要关注的关系是互联网上(作者，书目)这样一种关系。DIPRE 从一小组集合对(作者，书目)开始，也就是初始种子集。现在假设初始种子集中只有一个条目(*Arthur Conan Doyle, The Adventures of Sherlock Holmes*).，接着该系统会去互联网上爬取所有包含种子条目的网页(作者、书目必须同时存在该网页中)。为了归纳出其中的关系模式，DIPRE 使用一个六元组来表示这一模式[顺序，作者，书名，前缀，后缀，中间]。其中顺序是 1 的话表示作者在前书名在后，0 则代表相反，前缀后缀则表示两个实体的头部和尾部所出现的字符，中间则代表两个实体之间所出现的字符。

举个例子，假设从互联网上爬取到如下文本数据：

“Read The Adventures of Sherlock Holmes by Arthur Conan Doyle online or in you email”

“know that Sir Arthur Conan Doyle wrote The Adventures of Sherlock Holmes, in 1892”

“When Sir Arthur Conan Doyle wrote the adventures of Sherlock Holmes in 1892 he was high ...”

接着抽取句子当中的六元组：

[*0, Arthur Conan Doyle, The Adventures of Sherlock Holmes, Read, online or, by*]

[*1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, know that Sir, in 1892, wrote*]

[*1, Arthur Conan Doyle, The Adventures of Sherlock Holmes, When Sir, in 1892 he, wrote*]

在抽取完所有句子的六元组之后，系统会将那些顺序值相同以及中间字符串相同的元组分到同一组当中。针对每一组，接下来系统将找出它们最长公共前缀和最长公共后缀，最终将归纳出每一组的关系模式：

[*最长公共前缀, 作者, 中间部分, 书名, 最长公共后缀*].

对于上面的例子，得到的关系模式将会是[*Sir, Arthur Conan Doyle, wrote, The Adventures of Sherlock Holmes, in 1892*]，下一步是使用通配符表达式来泛化抽取模式，于是将得到：[*Sir, .*?, wrote, .*?, in 1892*]。然后使用这样一个模式再去互联网上搜索，并假设得到一

个新的关系：(Arthur Conan Doyle, The Speckled Band)。DIPRE 会将它加入到种子集当中并重复上述过程直到满足停止条件。算法 2.2[30]描述了整个系统处理的过程。

算法 2.2：DIPRE

(1) 使用初始种子集去互联网上标记一些数据

(2) 从标记的样例数据中归纳出关系模式

(3) 将上一步归纳出的关系模式去互联网上进行匹配搜索，并得到新的关系对，接着添加到种子集当中

(4) 重复上述步骤直到满足停止条件

DIPRE 和 Yarowsky 算法相似之处在于：算法的初始化都是从一小组初始种子集开始的，DIPRE 使用的分类器就是一个模式匹配，这个关系抽取模式是从种子关系集开始迭代训练得到的。给定一个字符串，如果它和某种模式相匹配，那么它将被归为正类，并用于抽取新的关系样例。并且新的样例会被加入到种子集当中，迭代训练后会得到更多的关系模式和样例。DIPRE 可以看成是 Yarowsky 算法应用于关系抽取的范例。

通过 DIPRE 的算法也可以发现，它的主要缺点是模式匹配系统比较严格。例如两个模式只是单个标点符号不同，那么它们在 DIPRE 中就是不同的。这样一种匹配方式将会大大降低系统的召回率。

2.1.2 Snowball

Snowball 的系统结构和上面所提到的 DIPRE 有点类似，它的目标是识别文本当中的(公司，位置)这样一种关系。Snowball 也会从一个初始种子关系集开始，它所使用的分类器和 DIPRE 一样是一个匹配模式 *P*，但是这当中又有区别，Snowball 会将每个元组表示成一个向量，然后用一个相似性函数来将它们分组，而不是直接去匹配字符串。Snowball 会从原始文本中抽取出如下形式的元组[*prefix, orgnization, middle, location, suffix*]，例如(*CMU, Pittsburgh*)是种子关系集合中的一个实体关系对，对于句子：“*... go to CMU campus in Pittsburgh to meet ...*”系统将会提取出：

[*(w1, go), (w2, to), ORG, (w1, campus), (w2, in), LOC, (w1, to), (w2, meet)*]

其中 *prefix* 和 *suffix* 特征向量的长度限制在了 2 个单词。w_i代表某个词的权重，它是通过计算该词归一化的频率得到的。例如对于出现在 *suffix* 位置的单词 *meet* 权重计算如下：

$$weight(meet, suffix) = \frac{frequency\ of\ meet\ in\ suffix}{number\ of\ all\ word\ in\ suffix} \tag{2-1}$$

随着种子集的扩充，更多的元组将被加入进来，与此同时单词的权重也在不断的更新和调整。同时 Snowball 也给出了元组之间的相似度计算公式：

$$\text{Match}(tuple_i, tuple_j) = (prefix_i.prefix_j) + (suffix_i.suffix_j) + (middle_i.middle_j)$$

有了相似度计算的方法，就可以对相似性较高的元组进行聚类，对于每一类元组，Snowball 会推导出一个中心向量作为该类元组的模式 P 的一个表示。并且每一类模式 P 都会被赋予一个置信度的评分用于评价该模式的质量：

$$Confidence(P) = \frac{P_{positive}}{P_{positive} + P_{negative}} \tag{2-2}$$

其中$P_{positive}$表示在之前一轮的迭代训练过程所有满足关系模式 P 并且（organization, location）完全一样的样本数量，$P_{negative}$则表示那些满足关系模式 P 但（organization, location）不完全相同的样本数量。

为了标记新的数据样本，Snowball 会在原始数据上运行一遍命名实体识别程序来识别出文本当中所有的地理位置和公司实体。在一个句子当中，对于每一个(organization, location)关系对系统都会产生一个五元组，因此一些出现频率较高的关系对会有很多个元组和它相对应，系统将每个候选关系元组和所有模式进行匹配，最终仅保留那些相似性得分大于某个阈值的候选元组。接下来再根据所匹配模式 P 的置信度来给候选关系对评分。最终那些评分较高的会被加入到种子集合当中并进行下一轮的迭代。

与 DIPRE 相比，Snowball 的匹配系统更加的具有灵活性。Snowball 不仅仅停留在文本表面的精确匹配，而允许文本有些微小的变化，例如标点符号等。

2.1.3 KnowItAll

与 DIPRE 和 Snowball 不同，KnowItAll 是一个大规模的 Web 信息抽取系统，它只使用一小组领域无关的提取模式来标记自己的训练样例。当为特定关系实例化时，这些通用模式产生特定关系的提取规则，然后用于学习特定领域的提取规则。这些抽取规则可以被应用于互联网网页当中，例如识别搜索引擎的查询语句，并且使用从搜索引擎命中计数导出的逐点互信息（PMI）给所得到的抽取结果分配一个置信度。例如，KnowItAll

利用诸如“<NP1>如<NP2>”的通用提取模式来建议 NP2 作为类 NP1 的候选成员的实例化。接下来，系统使用频率信息来识别哪些实例化最有可能是类的成员。最后，KnowItAll 学习一组关系特定的提取模式，例如“国家的首都”，从而它可以提取更多的城市。

2.1.4 TextRunner

DIPRE，Snowball 和 KnowItAll 都是关系特定的系统，即已经指定了特定种类的关系。系统所涉及的关系集合必须事先由人工定义好，当初 TextRunner 设计之初就是为了避免这一缺点。TextRunner 以自我监督的方式从其语料库中的文本中学习关系，种类和实体。TextRunner 以三元组$t = (e_1, r, e_2)$的形式定义一个关系，其中e_1和e_2是旨在表示实体或名词短语的字符串，r 是表示e_1和e_2之间的关系的字符串。图 2-1 是 TextRunner 的系统框架图。

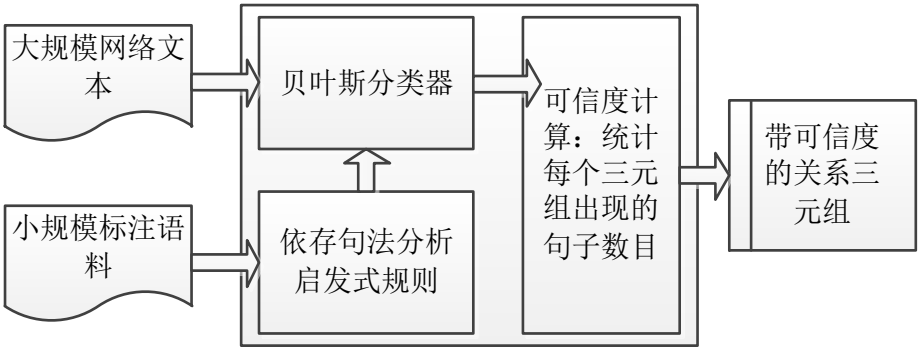

图 2-1TextRunner 系统框架图

TextRunner 包含三个主要的模块，分别是：1)自监督学习器模块；2)大规模 Web 文本抽取模块；3)可信度计算模块。

学习器首先自动将自己的训练数据标记为正样例或者负样例，然后使用该标记的数据来训练由抽取器所使用的二分类器。抽取器从每个句子当中生成一个或多个候选关系，然后运行分类器，并只保留那些标记为可信的关系。可信度计算模块基于冗余概率模型为每个保留下来的元组计算一个概率，用于表示其重要程度。

下面通过一个例子来说明 TextRunner 的运行流程：给出一个英文网页，对于网页中的每个句子，一个名词短语 chunker（步骤 1）将被调用；然后关系候选者（步骤 2）将产生可能的关系； 在以下步骤（3,4 和 5）中，运行语法解析器和依赖性解析器，并

且将使用解析树，依赖关系树和约束集合来运行关系过滤器用于标记可靠和不可信的关系（也称为正例和负例）；一旦系统标注关系后，它将每个关系映射成特征向量表示（步骤 6）； 所有功能都是领域无关的，可以在提取时进行评估，而无需使用解析器。 以下特征提取后，系统使用这组自动标记的特征向量作为二进制分类器的训练集，如 SVM，朴素贝叶斯等（步骤 7）。 分类器是针对特定语言的，但不包含关系特定或词汇特征。因此，分类器可以领域独立的方式使用。

当自监督的学习模块对分类器进行自我训练过程时，解析器只能使用一次。系统的关键思想是 Extractor 从大量文本中提取关系时并不运行依赖关系解析器。例如，我们从上 G 的英文词语语料库中提取关系的过程中，并不需要为整个语料库运行解析器，我们只需要解析一百万个单词，并使用标记的数据来训练二进制分类器。

TextRunner 最大的问题在于系统严重依赖依存句法分析工具来实现训练数据的自我标注，假如数据标注质量不高，这将直接影响接下来的抽取效果。

表 2-1 几种基于半监督的关系抽取系统对比

	DIPRE	**Snowball**	**KnowItAll**	**TextRunner**
Initial seed	*Yes*	*Yes*	*Yes*	*No*
Predefined relation	*Yes*	*Yes*	*Yes*	*No*
External NLP tools	*No*	*Yes:NER*	*Yes:NP chunker*	*Yes:dependency parser, NP chunker*
Relation types	*Binary*	*Binary*	*Unary / Binary*	*Binary*
Language dependent	*No*	*Yes*	*Yes*	*Yes*
Classifier	*Exact pattern matching*	*Matching with similarity function*	*Naïve Bayes classifier*	*Self-supervised binary classifier*
Input parameters	*2*	*9*	*>=4*	*N/A*

表格 2-1 给出几种基于半监督关系抽取系统的对比，主要从是否需要初始种子集、是否需要预先定义关系集、是否需要外部的 NLP 预处理工具、关系类型、所使用的分类器以及输入参数数量这几个维度考察了这几类系统。Snowball, KnowItAll 和 TextRunner 这几个系统初始化时需要输入一定数量的参数，相关参数的定义在系统中都

有明确解释，并提供了一些选择给用户。然而，这几个系统都没有给出参数调优的方法，用户不知道如何初始化参数以达到系统的最优效果，只能通过经验来选取。

2.2 基于监督学习的关系抽取

基于监督学习的关系抽取方法首先要有相关领域的标注数据，这里的标注一般指实体位置以及该句子所表征的实体对关系类别，之后从该训练数据集上学习对应的关系分类模型[31]。有监督关系抽取方法主要包括：基于特征向量的方法[32]，基于核函数的方法[33]，基于句法解析增强的方法[34]和基于条件随机场的方法[35]。

对于给定的句子$S = w_1, w_2, .., e_1.., w_j, .. e_2, .., w_n$,其中$e_1$和$e_2$代表实体，关系映射函数可以定义如下：

$$f_R(T(S)) = \begin{cases} +1 \; If \; e_1 \; and \; e_2 \; are \; related \; according \; to \; relation \; R \\ -1 \; Otherwise \end{cases} \tag{2-3}$$

公式(2.1)中 T(S)是从句子 S 中提取的各类特征，映射函数$f(.)$用来决定句子中的两个实体是否包含关系 R。换句话说，关系抽取任务转变成为了*实体-关系的发现*。当我们拥有一批正负关系样例的训练数据后，我们可以将$f(.)$构建成为一个分类器，例如支持向量机(SVM)，贝叶斯分类器，感知机等。这些分类器可以使用在执行文本分析之后选择的一组特征（如 POS 标注，依存句法分析等）。另一方面，输入到分类器的特征也可以是丰富的结构表示形式例如解析树。根据输入到分类器特征的性质，正如上文刚刚所提到的，基于有监督的关系抽取方法可以进一步分为基于特征向量的方法和基于核函数的方法。接下来本文会适当详细地介绍这两种方法。

2.2.1 基于特征向量的方法

当拥有带有标注的训练数据后，接下来可以从句子当中提取出语法和语义的特征。这些特征可以作为线索帮助判断句子当中的两个实体之间是否存在某种关系。其中能够抽取的语法特征包括：1)实体本身；2)两个实体所属的类；3)两个实体之间的词序列；4)两个实体之间词的个数；5)包含两个实体解析树的路径。语义特征包括两个实体之间的依存句法树路径。无论提取的是语义特征还是句法特征都以特征向量的形式呈现给分类器，用于训练或分类。Kambhatla 使用所述的特征训练对数线性模型，用于实体分类任务。另一方面，Zhao 和 Grishman 使用了 SVM 分类器去训练那些基于多项式和线性

核函数的特征，用于分类不同的实体关系。有些特征对于实体关系分类的效果很好，而有的特征则显得较弱，如何筛选出那些与分类任务较为相关的特征则显得十分重要。基于特征方法涉及特征的启发式选择，并且必须在试错法的基础上选择特征以最大限度地提高分类器的准确率。由于一般的 NLP 应用和关系提取任务需要对输入数据进行结构化表示，因此可能难以达到最优相关特征的子集。为了解决选择合适的特征集的问题，人们设计了一些专门的核函数用于关系提取，以便挖掘出输入数据的丰富表示例如浅层解析树等。这些核函数以更隐蔽的方式在更高维度的空间中穷举地探索输入表示。接下来将详细介绍相关基于核函数的方法。

2.2.2 基于核函数的方法

一般用于关系抽取的核函数是基于 string-kernels[5]的，string-kernels 过去一直被应用于文本分类的应用当中[36]，它是一个操作字符串的核函数，用于度量两个字符串之间的相似性。如果字符串 a 和 b 越相似，那么 string-kernels 函数$K(a,b)$的值就越大。同时每个字符串都可以映射到更高维的空间，其中每个维度对应于特定子序列的存在（加权）或不存在（表示为 0）。例如，字符串$\boldsymbol{x}=\mathbf{cat}$可以在子序列的更高维空间表示如下：

$$\emptyset\,(x=\text{cat}) = [\emptyset_{\boldsymbol{a}}(\boldsymbol{x})..\emptyset_{\boldsymbol{c}}(\boldsymbol{x})..\emptyset_{\boldsymbol{t}}(\boldsymbol{x})..\emptyset_{\boldsymbol{at}}(\boldsymbol{x})..\emptyset_{\boldsymbol{ca}}(\boldsymbol{x})..\emptyset_{\boldsymbol{ct}}(\boldsymbol{x})..\emptyset_{\boldsymbol{cat}}(\boldsymbol{x})..]$$

$$= [\,\boldsymbol{\lambda}\ ..\ \boldsymbol{\lambda}\ ..\ \boldsymbol{\lambda}\ ..\ \boldsymbol{\lambda}^{\mathbf{2}}\ ..\ \boldsymbol{\lambda}^{\mathbf{2}}\ ..\ \boldsymbol{\lambda}^{\mathbf{2}}\ ..\ \boldsymbol{\lambda}^{\mathbf{3}}\ ..\ ] \qquad (2\text{-}4)$$

其中$\boldsymbol{\lambda}\in(0,1]$ 表示衰减因子，使得较长的和不连续的子序列受到惩罚。不妨设$\boldsymbol{u}$ 表示字符串 $\boldsymbol{x}(\boldsymbol{u}=\boldsymbol{x}[\boldsymbol{i}])$ 的一个子串， $\boldsymbol{u}$的下标$\boldsymbol{i}=\boldsymbol{i}_{\mathbf{1}},\boldsymbol{i}_{\mathbf{2}},\dots,\boldsymbol{i}_{|\boldsymbol{u}|}\left(\boldsymbol{i}_{\mathbf{1}}\le\boldsymbol{i}_{\mathbf{2}}\le\cdots\le\boldsymbol{i}_{|\boldsymbol{u}|}\right)$，$\boldsymbol{l}(\boldsymbol{i})=\boldsymbol{i}_{|\boldsymbol{u}|}-\boldsymbol{i}_{\mathbf{1}}+\mathbf{1}$表示子串$\boldsymbol{u}$的长度。由于$\boldsymbol{u}$可以以多种方式存在于$\boldsymbol{x}$内部，因此字符串$\boldsymbol{x}$的坐标对应到$\boldsymbol{u}$在高维空间中的位置可以表示为

$$\emptyset_{\boldsymbol{u}}(\boldsymbol{x})=\sum_{\boldsymbol{i}:\boldsymbol{u}=\boldsymbol{x}_{\boldsymbol{i}}}\boldsymbol{\lambda}^{\boldsymbol{l}(\boldsymbol{i})} \qquad (2\text{-}5)$$

如果 U 是存在于字符串 x 和 y 中的所有可能有序子序列的集合，则 x 和 y 之间的相似度可由下式给出：

[5] https://en.wikipedia.org/wiki/String_kernel

$$\boldsymbol{K(x,y) = \emptyset(x)^T \emptyset(y)}$$

$$\boldsymbol{= \sum_{u \in U} \emptyset_u(x)^T \emptyset_u(y)} \quad (2\text{-}6)$$

直接使用公式(2-3)来计算(2-4)将会是指数级的时间复杂度，因此在实际计算过程当中一般会使用动态规划算法来进行求解。这样时间复杂度会降低至$O(|x|\,|y|^2)(|x| \geq |y|)$。 对于公式(2-4)一种更为概括的解释是：假设$x$和$y$代表两个不同的 object，这里的 object 可以是字符串、词的序列以及句子的解析树等等。在实体关系抽取任务中，如果x^+和x^-分别代表实体关系样本中的正负样例，并且 y 代表测试样例，$k(x^+,y) > k(x^-,y)$则表示y包含了这样一种关系反之则不包含。在实际应用当中，一般用某种分类器来表示相似度计算函数$k(x,y)$，例如 SVM，贝叶斯等。x^+、x^-和y可以有如下两种表现形式：1)实体周围的词序列；2)包含实体的语法分析树。根据表现形式的不同，目前主要有两种基于核函数的实现方法，分别是：Bag of features kernels 和 Tree kernels。

通过以上介绍可以发现，基于有监督学习的方法存在着以下几点不足：

1) 这些方法难以扩展到新的实体关系类型，因为需要特定领域带有标注信息的关系语料库；

2) 对高阶实体关系(Higher-order Relations)又称多元实体关系的扩展也很困难；

3) 大多数方法都没有考虑对计算复杂度的优化，随着输入数据量的增加，计算可能成为一个瓶颈；

4) 需要利用现存的 NLP 工具对输入数据进行一些预处理，例如构建句子的解析树、依存语法分析树等。实际情况是这些处理工具不可避免的会引入一些误差，这些误差将传递到关系分类的模型当中，影响最终的分类效果。

2.3 基于深度学习的关系抽取

上述工作已经证明有监督和半监督的方法能够很好地解决关系抽取任务，并产生相对较高的性能。然而，这种方法的实际效果严重依赖于所选特征的质量。随着近来深度神经网络的兴起，很多人开始尝试将深度学习的相关技术应用到实体关系抽取的任务中去。主要有基于卷积神经网络的方法和基于循环神经网络的方法。

2.3.1 基于 CNN 的方法

卷积神经网络(Convolutional Neural Networks, CNN)现在非常适用于处理自然语言处理的一些任务并且可以得到效果不错的模型。CNN 是具有多层的前馈人工神经网络的扩展架构。CNN 既可以是基于有监督的，也可以是基于无监督的，但从效果来看，有监督的方法准确率通常会比无监督的方法高一点。人工神经网络一般包括三层：输入层，隐藏层和输出层。每个隐藏层和输出层节点连接都模仿动物的视觉皮层的行为，并称作神经元。而 CNN 会在输入层上应用卷积来计算输出。因此，局部连接会被创建，其中输入的每个区域连接到输出神经元。CNN 专为最小化预处理而设计，研究人员针对不同问题提出了不同架构的 CNN 模型，以提高关系分类性能。

理解卷积的最简单的方法是将滑动窗函数应用于矩阵。滑动窗口可以命名为过滤器，特征检测器或内核。为了实现完整的卷积滤波器，将其值与原始矩阵进行乘积，然后通过在整个矩阵上滑动滤波器来为每个元素求和。CNN 中的第一层是输入层。 它可以具有单个或多个通道，取决于表示和需要，或者可以用于不同词向量的单独通道。第二层或卷积层由特征图谱组成。要从输入层移动到特征图，输入层与滤波器进行卷积，然后在池化层添加偏置。池化层仔细检查其输入，并在每个过滤器的结果处取最大值是一个常见方法。最终结果送入一个非线性层，即一个带有激活功能的神经元层。滤波器会被随机初始化，并在模型训练中被迭代更新。不同的滤波器互不相同，并且每个滤波器都对应着一个特征图谱。

最后，在经过多个卷积层和池化层之后，最后会有一个全连接层与之相连接。全连接层的神经元会与先前层中的所有激活的神经元完全连接，并且它们能否被激活可以通过矩阵乘法来评估。 通过 max pooling 生成的特征向量会被送到损失层。损失层使用损失函数作为网络训练的惩罚并产生预测和真实标签的变化。从而输出层可以提取出输入

句的关系标签。研究人员会选择超参数和正则化方案来解决其模型中的过拟合问题，以增加模型的泛化能力。

Zeng[37] 最早提出使用卷积神经网络来解决实体关系抽取问题。他们将句子当中的每个词对应的词向量以及词的位置信息结合在一起作为卷积神经网络的输入层，通过卷积操作、池化操作和最终全连接层得到句子的分布式表示，用于之后关系分类模型的输入。Zeng 的主要贡献可以总结为三个方面：1) 探索在没有复杂 NLP 预处理的情况下执行关系分类的可行性，其中卷积神经网络主要用于提取词法和句子级特征；2) 为了指定应该分配关系标签的实体对，提出了位置特征来对 CNN 中的目标名词对的相对距离进行编码。3) 使用 SemEval-2010 Task 8 数据集进行实验，实验结果表明，提出的位置特征对关系分类至关重要，以及提取的词汇和句子级别特征对于关系分类是有效的。后来，Santos[38]提出了一种比较新颖的卷积神经网络用于实体关系的抽取，它的原理是采用了新的损失函数，从而可以比较好的将不同的关系类别区分开来，提升了分类模型的准确率。

2.3.2 基于 RNN 的方法

循环神经网络(Recurrent Neural Network, RNN)也是一种深度的神经网络，其中单元之间的连接形成定向循环。这使得它适用于诸如未分段连接的手写识别或语音识别等任务。由于基于 CNN 的方法缺乏学习时间特征的能力，特别是实体对之间的长距离依赖关系处理的效果不够好，因此相关领域的研究学者尝试将 RNN 应用于关系抽取的任务中。

RNN 的思想是利用序列信息，在传统的神经网络中，通常假设所有的输入和输出是相互独立的，二者之间没有任何联系。但是对于许多任务来说事实情况并非如此。比如说预测一个句子中的下一个单词，需要考虑哪些单词来自它。RNN 称为循环，因为它们对序列的每个元素执行相同的任务，输出取决于先前的计算。考虑 RNN 的另一种方法是，它们有一个“记忆”，捕获到目前为止所计算的信息。在理论上，RNN 可以以任意长的序列使用信息，但在实践中，它们仅限于回顾前几个步骤。一个典型 RNN 结构如图 2-2 所示：

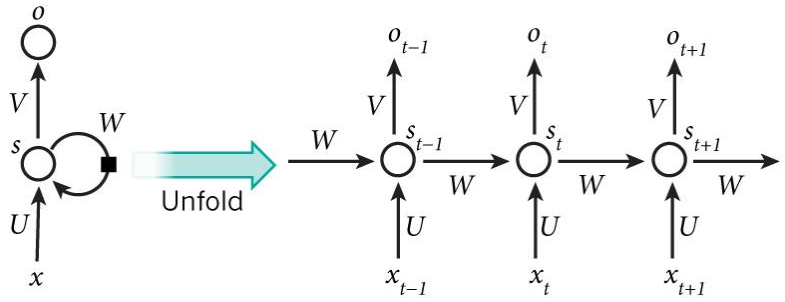

图 2-2 RNN 结构示意图

其中x_t是在时间t处的输入，举个例子，x_1可以表示句子当中第二个单词对应的 One-hot 向量表示。s_t代表时间 t 处的隐藏状态，它是网络的记忆单元。o_t代表时间 t 处的输出。

Socher[20] 最早尝试将递归神经网络应用于实体关系抽取问题当中，该方法首先获得句子的依存句法分析树，然后为树上的每个节点构造向量表示，并将这些节点的向量作为循环神经网络的输入，最终得到该句子的向量表示，用于分类模型的构建。由于只考虑了句子的句法信息，而没有考虑实体所在的上下文信息以及句子当中词语的语义信息，所以该方法的抽取效果提升有限。后来，Miwa[39]提出了一种基于端到端神经网络的实体关系抽取模型。该模型使用双向LSTM(Long-Short Term Memory，长短时记忆模型)和树形 LSTM同时对实体和句子进行建模，效果与 Socher的方法相比分类准确率有所提升。

综上所述，基于深度学习的实体关系抽取方法无论是从最终模型分类的准确性还是构建模型复杂程度来看，都是要优于传统基于特征工程的方法。但它也存在缺少训练数据集的问题，而且深度学习的相关任务所需的训练数据集更大。为此，有人尝试将基于卷积神经网络的模型与远程监督的思想相结合。该方法假设每个实体对所对应的句子集合中至少有一个句子表征了该实体对的关系，并且只用这一个句子作为该关系的训练语料。从最终效果来看确实在一些标准数据集上取得了不错的效果，但缺点也很明显，就是没有充分利用语料库中其他可能表征该关系的句子信息。

2.4 本章小结

本章主要介绍了实体关系抽取任务现有的主要研究方法，包括：基于半监督学 习的方法、基于监督学习的方法以及基于深度学习的方法。

半监督学习方法主要基于种子集的 Bootstrapping 方法。该方法首先需要根据预定义的关系类型构建相应的关系实例作为初始种子集，然后通过模式学习方法迭代生成关系描述模式集。半监督学习方法不需要手动注释语料库，只需要构建初始关系种子集，然后使用 Web 或大规模语料库信息进行高度冗余，充分挖掘相应的关系描述模式，并通过模式匹配提取新关系示例，准确高效地完成关系提取任务。然而，该方法存在几个关键问题，如初始关系种子集的生成和选择，模式组成方式，模式的质量，迭代过程的速度以及高精度，低召回率等。

基于监督学习的关系抽取方法主要有两种，一种是基于特征向量的方法，另一种是基于核函数的方法。该类方法首先需要人工标注语料库获得训练数据集，接着需要人工从训练数据中提取出各种有效特征，最后是在上述特征上学习得到各种关系分类模型。该类方法的缺点也很明显，就是最终分类模型的效果比较依赖于所选特征质量，特征质量越高，最终分类效果就越好。但实际情况是仅仅依靠人工难以选取合适的特征子集。

随着近几年深度学习技术的兴起，很多研究者尝试将相关技术应用在实体关系抽取的任务中。有基于 CNN 的关系抽取方法，也有基于 RNN 的实体关系抽取方法。该类方法一般借助于词向量，结合实体的上下文位置信息作为网络的输入向量矩阵，经过神经网络的处理得到句子的分布式表示用于分类器的输入。从而可以自动学习到相关特征，免去人工抽取特征的弊端。从实验结果来看，要优于传统基于监督学习的方法。

第三章 句子的分布式表示

3.1 引言

在实现关系抽取的相关任务时，通常做法是将其转换为句子的分类任务，即事先定义好几种关系，然后构建关系的分类模型。而在处理 NLP 相关问题时，除了诸如分词、去停用词等一系列数据清洗的必要步骤外，第一步肯定是要找一种方法将文本中的字词转为向量形式表达(Vector Representations)。自然语言处理当中最直观，最常用的词表示方法是 One-hot Representation[6]，它是一种局部表示(Local Representation)[40]方法，在这种表示方法中每个词都会被表示为一个固定长度的向量。该向量的维度一般是词表大小，并且其中某一个维度的值为 1 外，其它元素都为 0，这个为 1 的维度就表示当前词。举个例子:

“江苏”表示为 [0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 …]

“南京”表示为 [0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 …]

每个词都是茫茫 0 海中的一个 1。

这种 One-hot Representation 如果给每个词分配一个数字 ID，将会是非常的简洁的。比如上述例子中，“江苏”记为 2，“南京”记为 7。然而这种表示方法存在一个问题，即对特征的编码往往是随机的，没有提供任何关联信息，没有考虑到字词间可能存在的关系，也就是“词汇鸿沟”现象。例如，我们对“江苏”和“南京”的从属关系，地理位置关系等一无所知，我们从 3 和 8 这两个值看不出任何信息。

使用分布式表示(Distributed Representations)则可以有效地解决这个问题。分布式表示模型可以将字词转为连续值(相对于 One-Hot编码的离散值)的向量表达，并且其中意思相近的词将被映射到向量空间相近的位置。分布式表示模型在 NLP 中主要依赖的假设是 Distributional Hypothesis[41], 即在相同的语境中出现的词其语义也相近。Word2Vec[42]即是一种计算非常高效的，可以从原始语料中学习字词空间向量的预测模型，它既能表示词本身又可以考虑了词与词之间的语义距离。它主要分为 CBOW(Continues Bag of Words)和 Skip-Gram 两种模式，其中 CBOW 是从原始语句(比

[6] https:.en.wikipedia.org/wiki/One-hot

如:中国的首都是__)推测目标字词(比如:北京)；而 Skip-Gram 则正好相反，它是从目标字词推测出原始语句，其中 CBOW 对小型数据比较合适，而 Skip－Gram 在大型语料中则表现更好。在本节主要使用 Skip-Gram 模式的 Word2Vec。

对于句子和文本的分类任务来说，只有字词的语义空间向量表示还是不够的，还需要得到句子和文本的空间向量表示。最经典的文本向量表示方法要数词袋模型(Bag of Words, BOW)[7]了。通过词袋模型得到的文本向量可以直接输入到分类器，进行文本分类或情感分析等任务。下面通过一个例子来说明词袋模型的建立的过程。

下面是两个简单的例句：

(1) Jack hates to go hiking. Max also hates to go hiking.

(2) Jack and Max are good friends.

根据这两个样本建立一个词典包含所有出现的单词，如果是未清洗过的原始文档，则以词根建立。

["Jack", "Max","hates", "to", "go", "hiking", "also", "and", "are", "good", "friends"]

根据每个词语在词典的索引和在文档中的出现频次可以对以上两个句子建立词袋向量：

(1) [1,1, 2, 2, 2, 2, 1, 0, 0, 0, 0]

(2) [1, 1,0, 0, 0, 0, 0, 1, 1, 1, 1]

每个维度的权值可以取词频或是 tf*idf，甚至是二元值，即 1 代表出现 0 代表未出现。在得到相关文本的向量表示后就可以进行其它一些比较复杂的 NLP 任务了。

然而词袋模型也存在两个比较明显的缺陷：一是它丢失了词与词之间的顺序信息，二是它没有考虑词的语义信息。因此词袋模型在 NLP 有些任务中的表现难免差强人意。

另一种向量化模型和单词输入顺序有关，可以区分"Mary loves Jack"和"Jack loves Mary"这两个句子。词语的顺序很难直接量化输入，但循环神经网络(Recurrent Neural

[7] https:.en.wikipedia.org/wiki/Bag-of-words_model

Network, RNN)能够通过时间序列的变化，实现变长词串到语义向量的映射(sequence-vector)。

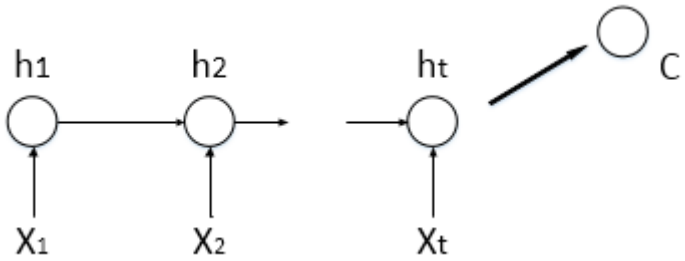

图 3-1 RNN 结构示图

如图，$x_1 x_2$..为输入文档的词串，RNN 将文档看成一个随时间变化的词序列，每输入一个新词，隐含层就进行一次更新:

$$h_{<t>} = f(h_{<t-1>}, x_t) \tag{3-1}$$

这样，隐含层充分利用了上文的历史信息，并始终保持最新状态，直到最后输出文档的语义向量 c。由于带有词序信息，RNN 训练出来的文本向量相比词袋模型更具有语义，在实验上直接体现在输入到同个分类器中，带词序信息的模型在情感分类任务的表现更好。此外，实现了变长串到向量(sequence-vector)映射的模型，可以适用更复杂的 NLP 任务[43]，比如配合语言模型可以很容易扩展成 sequence-vector-sequence 模型，以完成直接从源语言到目标语言的机器翻译任务[44]。然而 RNN 优化困难、结构复杂，且容易丢失较久之前的历史信息。

本文中尝试使用了两种方法学习句子的紧凑的分布式表示，具体过程将在 3.2 和 3.3 小节阐述。

3.2 词向量加权

词向量(Word Embedding)[8]是自然语言处理中语言模型与表征学习技术的统称。概念上而言，它是指把一个维数为所有词的数量的高维空间嵌入到一个维数低得多的连续向量空间中，每个单词或词组被映射为实数域上的向量。训练词嵌入的方法包括人工神经网络[45]、对词语同现矩阵降维[46][47][48]、概率模型[50]以及单词所在上下文的显式表示等[51]。

word2Vec[9]是由谷歌开源的一套基于神经网络的词嵌入的高效训练工具。Word2Vec 是一个双层神经网络，它的输入是文本语料，输出则是一组向量：该语料中词语的特征向量。使用 Word2Vec 训练语料能得到一些非常有趣的结果，比如意思相近的词在向量空间中的位置会接近。从一份 Google 训练超大语料得到的结果中看，诸如 Beijing、London、New York 等城市的名字会在向量空间中聚集在一起，而 Cat、Dog、Fish、等动物词汇也会聚集在一起。同时，如图 3.2 所示，Word2Vec 还能学会一些高阶的语言概念，比如计算“man”到“women”的向量(单词都是向量空间中的点，可计算两点之间的距离)，会发现它和“king”到“queen”的向量非常相似，即模型学到了男人与女人的关系；同时，“walking”到“walked”的向量和“swimming”到“swam”的向量非常相似，模型学习到了进行时与过去时的关系。

预测模型 Neural probabilistic language models 通常使用极大似然法进行训练，在给定前面的语句 h 的情况下，通过一个 softmax 函数来最大化目标词汇w_t的概率。

$$P(w_t|h) = softmax\big(score(w_t, h)\big)$$

$$= \frac{\exp\{score(w_t, h)\}}{\sum_{Word\ w'\ in\ Vocab} \exp\{score(w', h)\}} \tag{3-2}$$

其中，$score(w_t, h)$用于计算词汇w_t和上文h的兼容性(通常使用向量的点积来计算)。一般通过最大化训练集上的对数似然函数来训练这个模型，例如：

[8] https://en.wikipedia.org/wiki/Word_embedding
[9] https://code.google.com/p/word2vec/

$$J_{ML} = \log P(w_t|h)$$

$$= \text{score}(w_t, \text{h}) - \log(\sum_{Word\ w' in\ Vocab} \exp\{score(w', h)\}) \quad (3\text{-}3)$$

但这个方法存在一个比较严重的问题是计算的开销太大，因为在每一步训练过程当中需要计算并且正则化当前上下文环境 h 中所有其它单词w'的概率得分。在 Word2Vec 的 CBOW 模型中，不需要计算完整的概率模型，只需要训练一个二元分类模型，用来区分真实的目标词汇和假想的词汇(噪声)这两类。这种用少量噪声来估计的方法，类似于蒙特卡洛模拟[51]。从数学角度来看，我们的目标是最大化:

$$J_{NEG} = \log Q_\theta(D = 1|w_t, h) + k \underset{\tilde{w}\sim P_{noise}}{\mathrm{E}} \cdot [logQ_\theta(D = 0|\tilde{w}, h)] \quad (3\text{-}4)$$

其中θ是一个概率值，是通过目标单词w使用二分类逻辑回归计算得出的。在实际中，我们通过从噪声分布中绘制 k 个对比词来近似期望值。当真实的目标单词被分配到较高的概率，同时噪声单词被分配的概率很低时，目标函数也就达到最大值。

在得到每个词语的词向量表示后，对句子当中每个词进行加权取平均后即可得到句子的语义向量表示，用数学公式表示为：

$$s = \frac{\sum_{i=1}^{n} weight(wi) * embedding(wi)}{n} \quad (3\text{-}5)$$

其中wi是该文档中出现过的所有词，$embedding(wi)$表示该词的词向量，$weight$为词嵌入的加权系数函数，n为句子当中词语的总数。

词向量的权重函数 $weight(w_i)$要能够反映词w_i在所在文段中的重要性。单词的重要性主要体现在两个方面：统计信息和位置信息。统计信息可以用 tf*idf 值衡量，位置信息可以通过该词是否在标题出现过来判断：

$$weight(w_i) = (1 + \varepsilon * P(w_i)) * S(w_i) \quad (3\text{-}6)$$

其中$P(w_i)$是位置信息函数，如果w_i在标题出现值为 1，反之为 0；$S(w_i)$是统计信息函数，为w_i的 tf*idf 值。ε是个增益参数，用来放大在标题中出现过的词的权重，本文取值 0.25。

3.3 基于 CNN 的句子分类算法

在机器学习中，卷积神经网络(Convolutional Neural Network, CNN)[10]是受生物学上感受野(Receptive Field)的机制而提出的一种前馈神经网络。卷积神经网络借助于卷积、池化等操作用于局部特征的提取[52]。 最初为计算机视觉发明，随后 CNN 模型被证明对 NLP 也同样有效，并在语义解析[53]，搜索查询检索[54]，句子建模[55]和其他传统的 NLP 任务都有不错的表现。

目前的卷积神经网络一般采用交替使用卷积层和最大值池化层，然后在顶端使用多层全连接的前馈神经网络。训练过程使用反向传播算法。卷积神经网络在结构上的特点有三个：局部连接，权重共享以及次采样。这些特点使得卷积神经网络在一定程度上具有平移不变性、缩放不变性和扭曲不变性。并且和前馈神经网络相比，卷积神经网络的参数更少，因而训练速度更快。在图像识别任务上，基于卷积神经网络模型的准确率也远远超出了一般的神经网络模型。

在全连接前馈神经网络中，假设第l层有n^l个神经元节点，第$l-1$层有有m^{l-1}个神经元节点，那么连接边有$n^l * m^{l-1}$条，也就是权重矩阵有$n^l * m^{l-1}$个参数。当m和n都很大时，权重矩阵参数将会非常多，训练的速度会非常得慢。如果采用卷积的方法而不是全连接的方法，即第l层的每一个神经元都只和第$l-1$层的一个局部窗口内的神经元相连接，便可以有效的减少连接的边，也就可以减少训练参数。通常第l层的第i个神经元的输入可以通过如下公式计算得到：

$$a_i^l = f(\sum_{j=1}^{m} w_j^l . a_{i-j+m}^{l-1} + b^l) \tag{3-7}$$

$$= f(\boldsymbol{w}^l . a_{i+m-1:i}^{l-1} + b_i) \tag{3-8}$$

[10] https://en.wikipedia.org/wiki/Convolutional_neural_network

其中，$\boldsymbol{w}^l \in \mathbb{R}^m$为 m 维的滤波器，$\mathbf{a}_{i+m-1:i}^l = [a_{i+m-1}^l, \ldots, a_i^l]^T$。上述公式也可以写为:

$$\mathbf{a}^l = f(\boldsymbol{w}^l \otimes \mathbf{a}^{l-1} + b^l) \tag{3-9}$$

$\otimes$表示卷积运算，从公式 3-8 可以看出，$\boldsymbol{w}^l$对于所有的神经元来说都是相同的，这就是卷积神经网络中所谓的权重共享。因此，在卷积层里只需要设置 m + 1 个参数即可。此外第$l+1$层的神经元个数需满足$n^{l+1} = n^l - m + 1$。

为了进一步减少参数的个数，同时避免模型发生过拟合，一般经过卷积操作都会紧接着进行一个池化操作，也就是子采样(Subsampling)，构成一个子采样层。子采样层可以显著地减少特征的维数，从而可以避免过拟合情况的发生。对于卷积层得到的一个特征映射X^l，可以将X^l划分为很多区域R_k，$k = 1, \ldots, K$，这些区域可以重叠，也可以不重叠。一个子采样函数**down**(…)定义为：

$$X_k^{l+1} = f\left(Z_k^{l+1}\right), \tag{3-10}$$

$$= f(w^{l+1}.\boldsymbol{down}(R_k) + b^{l+1}) \tag{3-11}$$

其中，w^{l+1}和b^{l+1}分别是权重和偏置参数。子采样函数 **down**(.)通常是取区域内所有节点中的最大值或者平均值。

$$pool_{max}(R_k) = \max a_i (i \in R_k) \tag{3-12}$$

$$pool_{avg}(R_k) = \frac{1}{|R_k|} \sum_{i \in R_k} a_i \tag{3-13}$$

同时通过子采样这样一个操作，如果上一层神经元有一些微小形态的改变并不会对下层造成过多的影响，保持了相对的不变性。

图 3-2 是由 Kim[56]提出的用于句子分类的 CNN 模型，它一共包含四层，分别是输入层、卷积层、池化层以及最后的输出层。输入层为一个$k \times n$的词向量矩阵，其中k为词向量的维度，n为句子所包含词语的个数。输出层为一个$m \times 1$的向量，其中m为类别数，得到的是句子在不同类别上的一个概率分布。

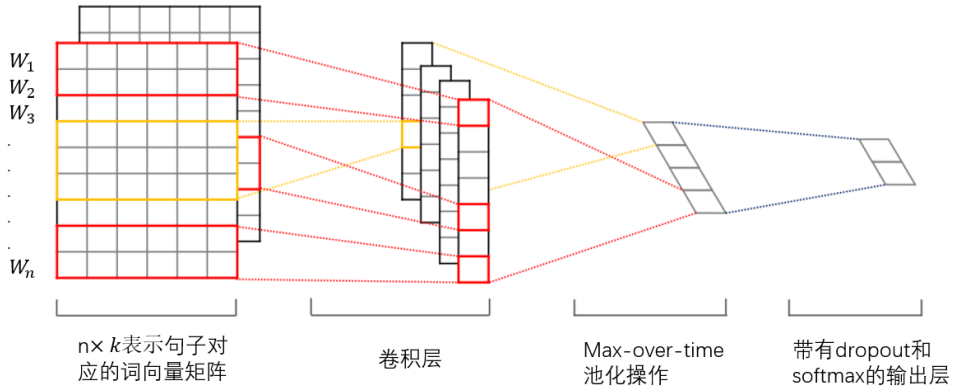

图 3-2 基于 CNN 的句子分类模型示意图

关系抽取是一个非常复杂任务，仅依靠词级别上特征还是不够的，一般还需要一些句子结构上的信息。因为有些关系是具有方向性的，比如投资关系：“2015 年 9 月 30 日，SoFi 通过官网声明软银当日以 10 亿美元领投 SoFi”。这句话中包含了两个命名实体分别是“软银”和“SoFi”，它们之间具有投资关系：软银投资了 SoFi。显而易见的是，软银是投资的主动发起方，而 SoFi 则是投资的接受方。这种单向的实体间关系使得关系抽取任务变得更加复杂，仅仅依靠上述模型无法区分出来关系的主动方与被动方。

为此，本文尝试在上述模型的基础上加入了词的位置信息，用于构建带有句子结构信息的分布式表示。具体实现过程如下：

3.3.1 位置嵌入

在关系抽取的任务中，一般距离实体词越近的词对于确定实体间关系的可能性越大。和 Zeng 的工作类似，本文使用由实体对指定的位置嵌入信息来帮助 CNN 网络记录每个词语与头尾实体的距离。它被定义为从当前单词到头实体和尾实体的相对距离的组合。举个例子，在句子“2015 年 9 月 30 日，SoFi 通过官网声明软银当日以 10 亿美元领投 SoFi”当中，经过分词、去除停用词的预处理之后可以得到如下词序列：

{ 2015 年| 9 月| 30 日| SoFi| 通过| 官网| 声明| 软银| 当日| 10 亿| 美元| 领投| SoFi }

其中头实体为“软银”，尾实体为“SoFi”，“领投”这个词与头实体“软银”的距离为 4，与尾实体“SoFi”的距离为 1。假设PE表示位置嵌入，WE 表示词向量，那么PE = $[d_1, d_2]$，其中d_1, d_2分别表示句子当中某个词与头实体和尾实体的距离。将 PE 与 WE 结合在一起即可得到词的表示：$[\mathrm{WE}, \mathrm{PE}]^T$，接着将其作为输入送入到卷积神经网络当中。

3.3.2 卷积和池化

假设$\mathrm{x}_i \in \mathbb{R}^{k+2}$表示句子当中第 i 个词语对应的 k 维词向量(本文 k 为 200)和位置嵌入的组合，一个长度为 n 的句子则可以表示为：

$$\mathrm{x}_{1:n} = \mathrm{x}_1 \oplus \mathrm{x}_2 \oplus \ldots \oplus \mathrm{x}_n \tag{3-14}$$

其中，$\oplus$表示连接操作符，一般来说$\mathrm{x}_{i:i+j}$表示将词语$\mathrm{x}_i, \mathrm{x}_{i+1}, \ldots, \mathrm{x}_{i+j}$连接之后得到的结果。之后用一个滤波器$\mathbf{w} \in \mathbb{R}^{hk}$进行卷积操作，这个滤波器将与一个窗口大小为 h 词向量矩阵产生一个特征。特征c_i可由包含词语$\mathrm{x}_{i:i+h-1}$的窗口通过如下公式产生：

$$\mathrm{c}_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b) \tag{3-15}$$

这里$b \in \mathbb{R}$代表偏置项，f是一个非线性函数例如双曲正切函数。这个滤波器通过窗口滑动与句子当中所有词语卷积得到一个特征图谱$\mathbf{c} = [c_1, c_2, \ldots, c_{n-h+1}]$，这里$\mathbf{c} \in \mathbb{R}^{n-h+1}$。然后对该特征图谱使用一个 max-overtime 的池化操作，即取$\hat{c} = \max\{\mathbf{c}\}$作为此滤波器下得到的特征。很显然，这里做池化操作的目的就是处理不同长度的句子，这样无论句子长度为多少，卷积核宽度是多少，最终到得到定长的向量表示，同时 max-pooling 也是为了捕获最重要的特征信息。紧接着使用多个滤波器(具有不同的窗口大小)来获取多个不同的特征。这些特征作为倒数第二层，并被传递到全连接的 softmax 层，其输出则是类别上的概率分布。在具体实验中，窗口大小设置为 3，4，5，每种窗口对应 100 种不同的滤波器将会对应产生 100 个特征图谱。

3.3.3 Dropout 和 L2 正则化

为了避免模型的过拟合，本文使用 Dropout[57]和 L2 正则化技术[58]来加强模型的泛化性能。Dropout是一种相当激进的技术，和 L1、L2 正则化不同，Dropout 并不依赖对代价函数的修改，而是改变了网络本身。Dropout 操作会随机地使网络中的某些节点失效，即随机删除某些神经元，同时让输入层和输出层的神经元保持不变。

而 L2 正则或叫权重衰减(weight decay)是一种最常用的正则化技术，它的思想是在代价函数中加入一个额外的正则化项。公式 3-15 是正则化之后的交叉熵：

$$C = -\frac{1}{n}\sum_{xj}\left[y_j \ln a_j^L + (1-y_j)\ln(1-a_j^L)\right] + \frac{\lambda}{2n}\sum_{w} w^2 \tag{3-16}$$

第一项是常规的交叉熵表达式,第二项是正则化项，也就是网络中所有权值的平方和。它由参数$\lambda/2n$进行调整，其中$\lambda > 0$被称为*正则化参数(regularization parameter)*，n是训练集的大小。直观来说，正则化的作用是让网络偏好学习更小的权值，而在其它的方面保持不变。选择较大的权值只有一种情况，那就是它们能显著地改进代价函数的第一部分。换句话说，正则化可以视作一种能够折中考虑小权值和最小化原来代价函数的方法。具体实验中，本文 Dropout 率设置为 0.5，λ 设置为 3，mini-batch 大小设置为 50。

3.3.4 反向传播训练

反向传播算法于 20 世纪 70 年代被提出，最早作为一种优化算法用于复杂嵌套函数的求解。但是直到 1986 年，由 David Rumelhart, Geoffrey Hinton, 和 Ronald Williams 联合发表了一篇著名论文之后[59]，才被机器学习领域的学者所认识到其存在的价值。反向传播算法已经被证明可以从人工神经网络的隐藏层中学习到高质量的特征，更重要的是，由于该算法的高效率，很多之前借助人工神经网络在常规时间内无法求解的复杂问题，现在可以被很好的解决，免去了需要相关领域专家进行人工特征提取的繁杂操作。

反向传播算法类似于计算多层前馈网络。因此和 delta rule 一样，反向传播算法需要做三件事情：

1)包含输入输出类标的训练数据集$\mathrm{X} = \{(x^i, y^i)\}$，其中$x_i$表示输入数据，$y_i$代表网络对应$x_i$的输出。一个包含 N 条训练数据的集合对表示为：$\mathrm{X} = \{(x^1, y^1), \ldots, (x^N, y^N)\}$。

2)一个前馈神经网络，其中参数集合为θ。

3)一个损失函数$\mathrm{E}(\mathrm{X}, \theta)$，其定义了在参数集合$\theta$之上一组输入-输出对的期望输出和神经网络的计算输出之间的误差。

在本文中，训练样本中的每个句子都是相互独立的。对于给定一个样本 s，输入到网络中，与训练参数θ运算将输出一个向量$\mathbf{o}$，它的第 i 维o_i表示该句子属于第 i 类的一个概率评分。为了获得条件概率$p(i|x, \theta)$,我们在关系类别之上做一个 softmax 操作:

$$p(i|x,\theta)=\frac{e^{o_i}}{\sum_{k=1}^{m}e^{o_k}} \tag{3-17}$$

对于所有的训练样本 T:(x^i,y^i)，我们可以得到参数θ的对数似然函数值：

$$J(\theta)=\sum_{i=1}^{T}\log p(y^i|x^i,\theta) \tag{3-18}$$

为了计算网络的参数θ，本文使用随机梯度下降的方法来最大化似然函数$J(\theta)$。由于参数分布在网络的不同层当中，因此我们通过反向传播算法来迭代更新θ：

$$\theta\leftarrow\theta+\lambda\frac{\partial\log p(y|x,\theta)}{\partial\theta} \tag{3-19}$$

3.4 对比实验

为了证明两种句子向量化方式的有效性以及比较两者优劣，本文做了一个文句分类的对比实验。具体实验结果参照 3.4.3 小节。

3.4.1 数据集及评价标准

本章实验数据来自 SemEval-2010 Task 8[1]会议所给出标准数据集，该数据集一共包含了 10717 条数据，定义了 9 种实体关系，其中包括：1)Cause-Effect；2)Component-Whole；3)Entity-Destination；4)Entity-Origin；5)Product-Producer；6)Member-Collection；7)Message-Topic；8)Content-Container；9)Instrument-Agency。其中 Other 用于表示训练样本中的关系不属于九种主要关系类型中的任何一种。每条数据都包含一个标记有两个实体e_1和e_2的句子，最终任务是预测两个实体间的关系。

表 3-1 语料统计

Relation	**Freq**
Cause-Effect	1331(12.4%)
Component-Whole	1253(11.7%)
Entity-Destination	1137(10.6%)
Entity-Origin	974(9.1%)
Product-Producer	948(8.8%)

Member-Collection	923(8.6%)
Message-Topic	895(8.4%)
Content-Container	732(6.8%)
Instrument-Agency	660(6.2%)
Other	1864(17.4%)
Total	10717(100%)

其中 Freq 表示每一类关系语料的绝对数量以及占整个语料库的一个比重。

本文采用分类方法中常用的三种评价方法对两个算法进行评测：分别是准确率(Precision)、召回率(recall)和 F 值。它们的计算公式如下：

$$\text{准确率 } P = \frac{\text{抽取出正确的关系对的数量}}{\text{抽取关系对的总数量}} \tag{3-20}$$

$$\text{召回率 R} = \frac{\text{抽取出正确的关系对的数量}}{\text{实际存在的关系对的数量}} \tag{3-21}$$

$$\text{F 值 F1} = \frac{2PR}{P+R} \tag{3-22}$$

3.4.2 实验设置

1)本文选取了英文维基作为训练词向量的语料库，一共包含 375 万篇文章[11]，经过简单的预处理操作后得到一个约 11G 大小的文本文件 wiki.en.text。其中训练时所用的参数见表格 3-2。

表 3-2 训练词向量参数

词向量维度	200
窗口大小	5
训练算法	hierarchical softmax
采样阈值	1e-3

[11] https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

训练结束后将得到一个所有词语的词向量矩阵，并以二进制形式存于文件 wiki_english_word2vec(Google).model 中，通过加载这个文件到内存中便可以得到指定词语的词向量以及计算词与词之间的语义距离。

2)卷积神经网络中滤波器窗口大小设置为 3，4，5，每种窗口对应 100 种不同的滤波器将会对应产生 100 个特征图谱。Dropout 率设置为 0.5，λ 设置为 3，mini-batch 大小设置为 50。实际训练中根据十折交叉法的结果来计算模型的准确率、召回率和 F 值。将总的数据集平均划分成十份，每次将其中的九份作为训练数据集，剩下的一份作为测试数据集。

表 3-2 CNN 参数设置

h	3、4、5
filter size	100
Drop rate	0.5
mini-batch	50
Learning rate	0.01
Hidden layer 1	200
Hidden layer2	100

3.4.3 实验结果及分析

表 3-3 基于词向量加权的朴素贝叶斯分类实验结果

Precision	Recall	F-Measure	Relation
0.619	0.661	0.639	1
0.701	0.724	0.712	2
0.749	0.727	0.738	3
0.723	0.680	0.701	4
0.716	0.707	0.711	5
0.667	0.652	0.659	6
0.727	0.626	0.672	7
0.653	0.768	0.706	8
0.705	0.781	0.741	9
0.695	0.702	0.698	Avg

表 3-4 基于 CNN 的分类实验结果

Precision	Recall	F-Measure	Relation
0.719	0.661	0.689	1
0.801	0.824	0.812	2
0.849	0.827	0.838	3
0.823	0.780	0.801	4
0.916	0.907	0.911	5
0.767	0.852	0.807	6
0.827	0.826	0.826	7
0.753	0.768	0.760	8
0.805	0.781	0.793	9
0.806	0.802	0.798	Avg

其中 Avg 代表的是每一列的平均值。Relation 一列当中的数字 1、2、3…9 分别对应上文提到的 9 种关系类型。根据两个表格中的 F 值可画出对比折线，如图 3-4 所示。

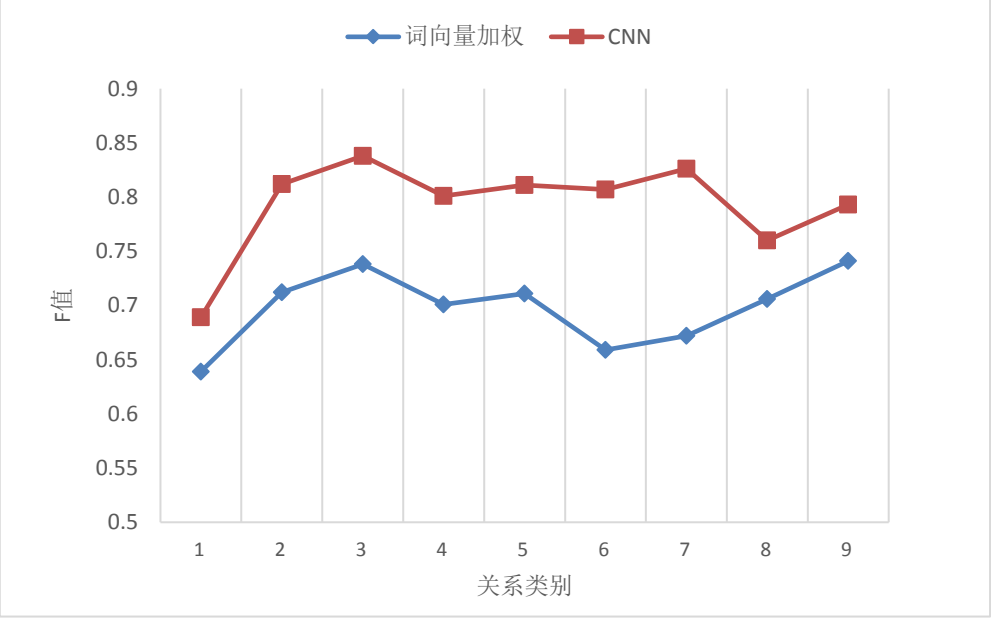

图 3-3 两种方法的 F 值对比

实验结果图像表明两种向量化方式在文句分类方面的效果都很好，说明两种模型都能学习出文本的低维语义表示。通过对比表 3-3 和 3-4 的结果可以发现，基于 CNN 的句子分类方法无论是准确率还是召回率都要比词向量加权的方法略胜一筹。

3.5 本章小结

在实现关系抽取的相关任务时，通常做法是将其转换为句子的分类任务，即事先定义好几种关系，然后构建关系的分类模型。而对于句子的分类任务来说，句子的语义向量表示是不可或缺的重要步骤。传统的词袋模型存在两个比较明显的缺陷：一是它丢失了词与词之间的顺序信息，二是它没有考虑词的语义信息。相比于词袋模型的高特征和语义缺失，我们希望能够生成紧凑和富有语义的句子向量。当前最热门、效果最好的语义向量化是 RNN，但是结构比较复杂。

本文借助于词向量尝试使用了两种方法学习句子的紧凑的分布式表示，一种是词向量加权，将复杂的句子向量化(text-vector)问题转化为相对容易的词向量化(word-vector)问题，而词向量容易通过 word2vec 训练得到。另一种方法通过卷积神经网络来构建句子的分类模型，其中通过池化操作来处理不同长度的句子，这样无论句子长度为多少，卷积核宽度是多少，最终到得到定长的句子向量表示。

经过句子分类的实验论证，两种方法训练出来的语义向量表现良好,都可以学习出句子紧凑的分布式表示。其中基于 CNN 的句子分类算法在 SemEval-2010 Task 8 标准数据集上的表现要优于词向量加权的算法。

第四章 面向互联网新闻文本的企业关系抽取

4.1 引言

互联网上每天都会更新产生大量的新闻信息，这其中包含一些企业相关的新闻报导，例如企业之间收购、合作、竞争案例等。这些存在于网页中的新闻文本包含了企业实体之间的各种关系，这种关系信息对于企业战略制定、投资方向决策等具有重要参考价值。本章将在第三章的基础上，结合网页正文提取、命名实体识别、基于 Bootstrapping 的语料库构建等技术实现对中文新闻网页当中的企业关系进行抽取。

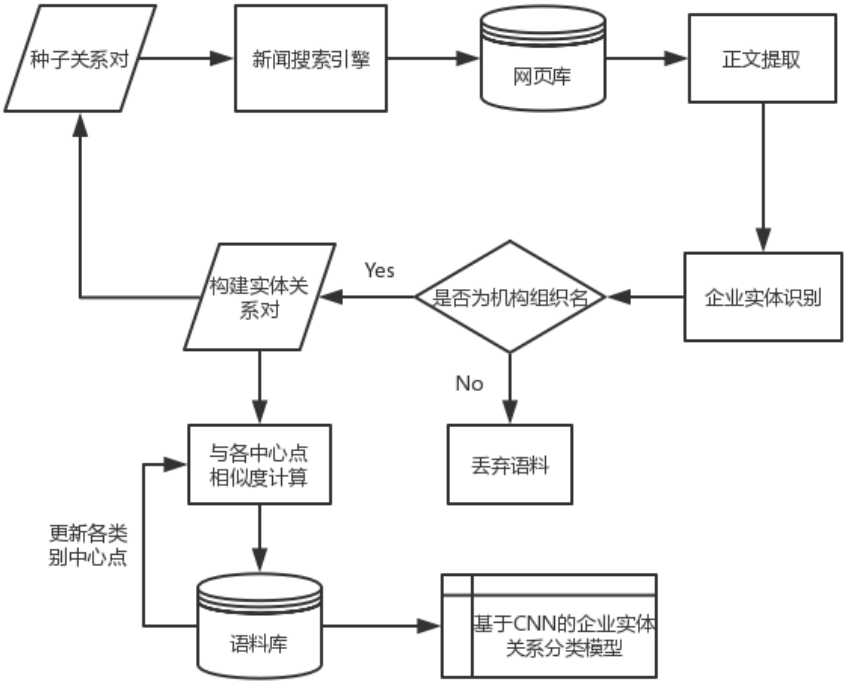

图 4-1 面向互联网新闻文本的企业关系抽取系统框架

图 4-1 是整个系统的框架设计图，主要包括两大部分，分别是语料库构建模块以及关系分类模块。其中正文提取和企业实体识别在构建语料和关系抽取过程中都有涉及，属于公共部分。所以接下来在 4.2 和 4.3 小节将着重介绍网页正文提取和命名实体识别的相关技术和实现方法。4.4 将介绍语料库的构建过程。

4.2 网页正文提取

对于新闻网页而言，真正有用的信息一般只存在于新闻标题和正文当中，网页其它部分对于我们来说是一种噪音，例如广告栏、相关链接和一些脚本语言等。因此需要借

助网页正文的提取技术将网页的正文抽取出来。针对这一问题，很多方法已经被提出来，常用方法有基于模板的方法、基于文字块密度的方法、基于 DOM 树节点统计的方法以及基于视觉的方法。

基于模板的网页正文提取算法依赖于网页 HTML 的内部结构上的一些特征。该方法假定同类型网页中有着相似的结构特征或者相似的 Dom 树(Document Object Model)结构。因而可以针对具有相似结构的网页来制定相同的模板来抽取网页的正文内容。举个例子，网易财经新闻频道当中的网页都有着相似的结构，正文都处在相同的 div 标签里。因此借助于 HTML 解析器，可以很容易地将所有该频道下网页的正文抽取出来。但这种方法的缺点也很明显，就是可移植性太差，需要针对不同结构的网页指定模板，相关工作费时费力。

基于文字块密度的方法是一种不需要建立 DOM 树的网页正文提取算法。该算法的主要思想是首先去除网页中所有的 html 标签，然后根据去除 html 标签后的文字密度判断出正文区域。最后将所有的正文区域合并在一起，从而获得网页的正文内容。该方法只适用于结构相对简单的网页，对于一些相对复杂并且不规范的网页效果并不是很好。

基于建立 DOM树的文档正文提取算法主要是先通过建立DOM 树，然后根据 DOM 树中每个节点标签的文本数量和链接数、文本长度，并将链接文本的长度作为标准来确定点是否是正文节点。最后先序遍历 DOM 树，将所有的正文节点取出并整合在一起，完成正文提取功能。这种方法主要不足在于：1)需要建立网页的 DOM 树结构、时间复杂度比较高；2)不同网页之间差异性较大、很多网页编写得很不规范，比如标签丢失、不对齐等，加大了构建 DOM 树的难度。

上述两种方法对于网页结构复杂、正文区分布零散的网页提取效果不是很好，基于视觉的网页正文提取算法弥补了它们的不足。其中比较经典的基于视觉网页的提取算法是：基于视觉的 Web 页面切割算法[60]。该算法的主要原理是从用户的视觉角度来分析网页的结构，即关心的是网页的视觉信息而不是网页的内部结构，它模拟人在阅读网页内容的过程并结合 DOM 树进行分析，可以弥补基于 DOM 树网页正文提取的一些不足之处，获得更加准确的抽取效果。图 4-2 为 VIPS 整个算法的流程图。该算法包含 3 个主要步骤：页面分块、块与块之间的权重计算、块的重构(组合)。这三个步骤组合在一起是一个完整的迭代过程。首先一个Web网页会被分割成几个大块并记录其层级结构。

接着对于每个分割得到的大块对其进行处理，处理步骤即上述三个主要步骤。如此地递归处理下去直到获得足够多的小块区域并且达到某个阈值。

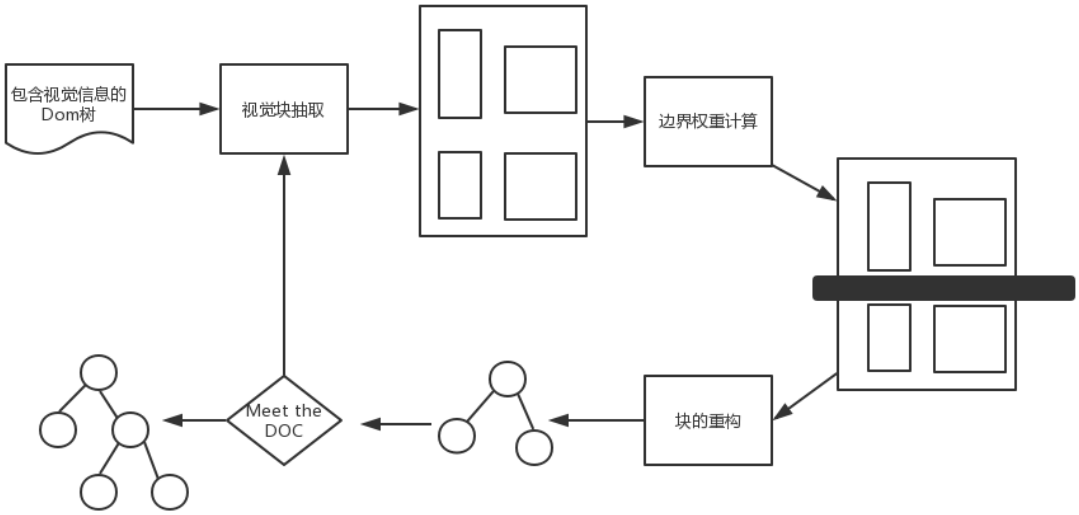

图 4-2 VIPS 算法流程图

基于视觉的网页正文提取算法充分利用网页的帧信息和视觉信息。与基于 DOM 树的网页文本提取算法相比，可以提高复杂结构和分散内容网页的提取准确性。然而，基于视觉的网页正文提取算法也有其固有的缺陷：首先，该算法需要多次迭代，最后需要语义块合并。与基于语义信息的网页正文提取算法相比，迭代更多，实现更复杂，增加时间复杂度。第二，提取网络视觉信息是耗时且费力的。由于网页的视觉信息采集和浏览器本身的层叠样式文件、脚本文件有关，访问视觉信息之前需要下载这些文件，接着浏览器内核就会调用这些文件，最后从浏览器的外部接口获取视觉页面信息此过程依赖于浏览器的内核代码并且非常耗时[61]。

本文采用了哈工大的基于行块分布函数的通用网页正文抽取[62]所提出的算法，算法流程如 4-1 所示：

算法 4-1：基于行块分布函数的通用网页正文抽取算法

Input：原始网页 HTML 代码

output：网页正文内容

(1) 首先将网页 HTML 标签去除干净，只留所有文字区域，同时保留标签去除后所有的空白位置信息，留下的正文成为 Ctext；

(2) 以 Ctext 中的行号为轴，取其周围 k 行，合起来称为一个行块 Cblock，行块 i 是以 Ctext 中行号i为轴的行块；

(3) 对于一个 Cblock，计算去掉其中的所有空白符(\n,\r,\t 等)后的字符总数做为该行块的长度；

(4) 以 Ctext 每行为轴，共有$LinesNum(Ctext)-k$个 Cblock，画出以$[1, LinesNum(Ctext)-k]$为横轴，以其各自的行块长度为纵轴的分布函数；

(5) 根据上述行块分布函数图，找出骤升点和骤降点，这两个边界点所含的区域即为网页的正文。

行块分布函数可以在$O(n)$时间求得，在行块分布函数图上可以直观的看出正文所在的区域。如图 4-3 所示的网页行块函数分布图，正文区域行号为：140-181。

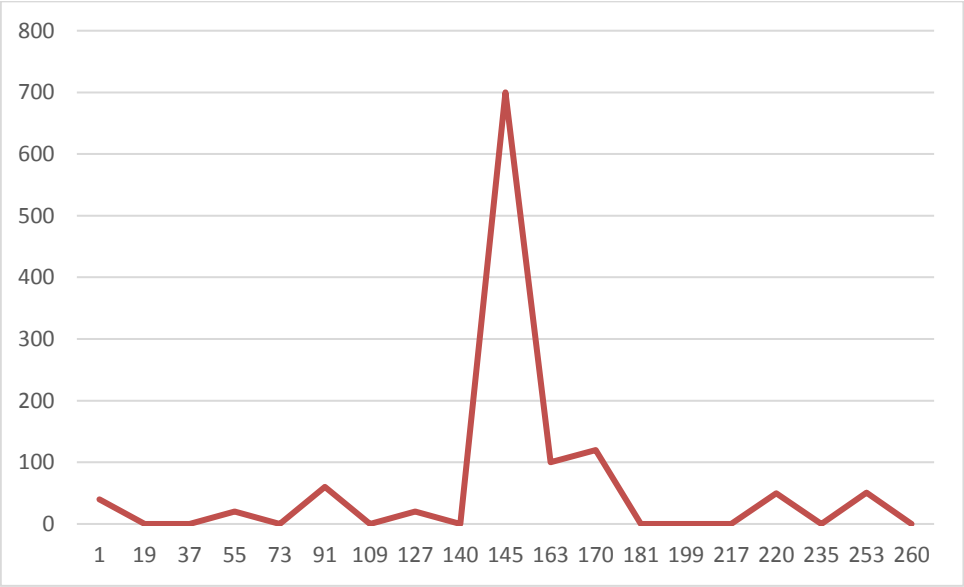

图 4-3 行块函数分布图

4.3 企业实体识别

命名实体识别(Named Entity Recognition, NER)是自然语言处理中的主要任务之一，NER 在过去的二十年中一直是比较热门的研究领域。虽然在该领域已经取得了许多不错的进展，但在很大程度上还是有比较大的提升空间。

命名实体识别任务是定位引用文本中特定实体的单词或短语的过程。 NER 任务首先出现在第六次信息理解会议（MUC-6）当中，并涉及实体名称（人和组织）、地名、时间表达式和数字表达式的识别。在 MUC-6 中，命名实体（NE）被分为三种类型，每种类型都有着特定实体类型的特定属性。实体及其标签定义如下：

1) **ENAMEX**: person, organization, location;
2) **TIMEX**: date, time;
3) **NUMEX**: money, percentage, quantity.

目前比较主流的命名实体识别方法主要还是基于监督学习的方法，该方法主要通过训练一组经过人工标注过的数据集从而学习得到识别模型。在 NER 的相关监督学习算法中，很多研究者使用隐马尔可夫模型（HMM），决策树，最大熵模型（ME）和条件随机场（CRF）进行了大量工作。通常，监督学习的方法可以基于有效特征来学习消歧规则，或尝试学习假设分布的参数，来最大化损失函数。下面就将介绍这几种模型如何应用在命名实体识别这项任务上的。

隐马尔可夫模型是最早用来解决命名实体识别的问题。该方法一般假设在一个上下文中，每一个词语只能被赋予一个表征其实体类型的标签。因此每个词语要么被赋予目标类型的标签，要么被认为是主题无关的。该模型的目标是对于一个词语序列，找出可能性最大的标签序列(NC)。

$$maxP_r(NC|W) \tag{4-1}$$

隐马尔可夫模型是一个生成式的模型，它尝试从分布参数中生成数据，单词序列 W 和标签序列 NC。

$$P_r(NC|W) = \frac{P_r(W,NC)}{P_r(W)} \tag{4-2}$$

通过维特比算法搜索整个标签序列赋值空间来最大化$P_r(W, NC)$。

和隐马尔可夫模型不同，最大熵模型是一个判别式模型。给定一组特征集合与标注过的训练数据集，模型可以直接学习出不同特征对应的权重，从而可以挑选出最优的特征子集用于构建分类模型。在最大熵模型中，目标是最大化训练数据的熵，以便尽可能地泛化训练得到的模型。在最大熵模型中，每个特征会与一个参数λ_i相关联，条件概率可以通过公式 4-1 和 4-2 得到：

$$p(f|h) = \frac{\prod_i \lambda_i^{g_i(h,f)}}{Z_\lambda(h)} \tag{4-3}$$

$$Z_\lambda(h) = \sum_f \prod_i \lambda_i^{g_i(h,f)} \tag{4-4}$$

最大化训练数据的熵可以确保对于每个特征f，对应它的值g_i可以最大限度的和经验值相等。最后，通过维特比算法可产生所需有效标签序列的条件概率并从中找到概率最高的路径。

条件随机场(CRF)最早由 Lafferty[63]等人引入作为模式识别和机器学习的统计建模工具。后来 McCallum[64]等人借助于 CRF 提出了一个特征归纳的方法用于解决命名实体识别的问题。具体实现过程如下：不妨设o $=< o_1, o_2 \ldots o_T >$表示一个输入数据序列，例如文本中的词序列。令 S 表示一组 FSM 状态，每个状态都与标签 L（如 ORG）相关联。令s $=< s_1, s_1, \ldots s_T >$是一些状态序列（T 输出节点上的值）。通过 Hammersley Clifford 定理，CRFs 定义给定输入序列的状态序列的条件概率为：

$$P(s|o) = \frac{1}{Z}\exp(\sum_{t=1}^{T} \lambda_k f_k(s_{t-1}, s_t, o, t)) \tag{4-5}$$

其中，Z 是通过所有状态序列边缘化获得的归一化因子，$f_k(s_{t-1}, s_t, o, t)$是特征函数，λ_k是每个特征函数的学习权重。通过动态规划算法，可以有效地计算两个 CRF 状态之间的状态转换。给定观察值$< o_1, o_2 \ldots o_T >$，修正后的正向值$\alpha_t(s_i)$成为到达状态s_i的“非归一化概率”。$\alpha_0(s)$被设置为在每个状态 s 中开始的概率，并且递归地计算为：

$$\alpha_{t+1}(s) = \sum_{s'} \alpha_t(s') \exp\left(\sum_k \lambda_k f_k(s', s, o, t)\right) \tag{4-6}$$

Z_o可由$\sum_s \alpha_T(s)$给出,用于从观察序列找到可能性最大的状态序列的维特比算法已经从其 HMM 形式中被修改。

由于本文的研究重点是企业实体关系的抽取，因此本文直接采用了哈工大开源的语言技术平台(Language Technology Platform，LTP)[12]来实现命名实体的识别。该平台支持人名、地名、组织机构名三类命名实体的识别。例如输入：

“2015 年 9 月 30 日，SoFi 通过官网声明软银当日以 10 亿美元领投 SoFi”

返回结果为：

“2015 年 9 月 30 日，[SoFi]Ns 通过官网声明[软银]Ns 当日以 10 亿美元领投[SoFi]Ns”

4.4 基于 Bootstrapping 构建语料的方法

包含企业间关系的语料(这里主要是指句子)存在于互联网大大小小的新闻媒体网站当中，如果单靠人工逐个审查标注的方法，将会是一项费时费力的工程。本文借助于一些中文新闻搜索引擎(例如百度新闻搜索、360 新闻搜索)，提出了一种基于 Bootstrapping 的方法用于搜集企业关系语料。算法描述如 4-2 所示。

算法 4-2：基于 Bootstrapping 构建语料的方法

Input：初始关系种子集

output：包含企业关系的语料库

---

(1) 构建关系类型体系；

(2) 初始化种子关系集，形如<name(公司名)，relation(关系)>；

(3) 将 name 和 relation 作为关键字输入爬虫模块，得到一组包含公司名 name 与关系 r 的句子集合 S;

(4) 根据集合 S 计算每一类关系的中心点C_i；

(5) 对于集合 S 当中的每个句子，逐个进行命名实体识别，找出其中出现的另一个公司名，得到实体对<name1,name2>;

(6) 将上一步得到的实体对送入爬虫模块，得到候选语料集 C

---

[12] http://www.ltp-cloud.com/

(7) 针对候选语料 C 中的每一句子S_c，计算其与每个类别中心点的距离d_i，将距离最近的中心点的类标y作为句子S_c的类别；

(8) 将(7)中的S_c加入到对应类别的种子关系集中；

(9) 判断语料库数量是否达到所设阈值，是进入(10)，否进入(3);

(10)结束

4.4.1 关系类型构建

企业与企业之间一般存在着投资(invest)、收购(acquisition)、合作(partner)、竞争(compete)的关系，企业与人一般存在的关系包括：董事长(chairman)、创始人(founder)。图 4-1 列举了一些知名企业之间的一些合作与竞争关系。从图中不难发现，有些公司之间存在一种以上关系，例如谷歌和苹果公司之间既存在竞争关系也存在合作关系。同时有些关系是单向的，有些关系是双向的。例如投资即为一种单向关系，A 投资了 B，反之并不成立。而有些关系是双向的，例如合作伙伴关系，A 是 B 的合作伙伴，反过来 B 也是 A 的合作伙伴，在离散数学中，也叫这种关系具有对称性。

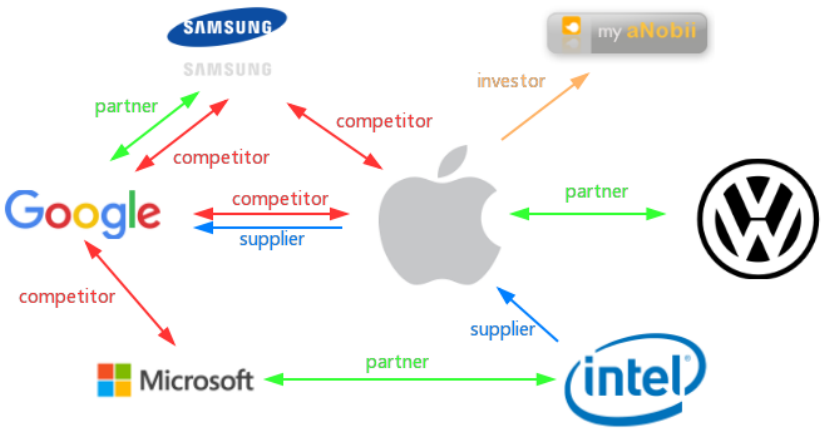

图 4-4 企业之间的关系示意图

本文搜集整理 100 多家国内外企业公司名录，用于构建企业间关系的语料库。所选企业主要分为三大类：互联网 IT 类、汽车类以及传统制造业类。其中具有代表性的企业有：微软、谷歌、阿里巴巴、吉利、大众、格力电器等。针对这些企业，本文主要围绕投资(invest)、收购(acquisition)、合作(partner)、竞争(compete)、董事长(chairman)、其他 (NA)这 6 个关系从互联网上搜集相关语料。在标注语料的过程中，本文将区分头实体与尾实体，例如“软银投资了阿里巴巴”这句话中，“软银”是头实体而“阿里巴巴”

是尾实体。而对应关系定义的方式，本文采取了对每一种关系定义一个相关关键词列表。这个关键词列表是由人工选择结合 word2Vec 扩展构建得到的。具体定义如表 4-1 所示。

表 4-1 关系类型的定义及关键词列表

关系类型	关键词列表
合作(cooperate)	合营 联合 中外合资 合资 协力 协同 协作 通力合作 合办 联手 联袂 携手 携手并肩 一并 一起 一同 分享 共享 共同
收购(acquisition)	收购 并购 竞购 竞买 承购 购进 买进 买入 议购 函购 函售 卖 卖给 抛售 售卖 转售 贷款 营收
投资(invest)	融资 投资 斥资 注资 投资额 竞得 投资者 入股
竞争(compete)	竞争者 竞争 垄断 角逐 逐鹿 竞赛 比赛
董事长	董事 董事长 常务董事 董事局 执行主席

4.4.2 初始种子集构建

为了构建初始种子集，本文选取了各个行业中市值排名靠前的企业作为初始种子企业。国内互联网 IT 类的公司包括：腾讯、阿里巴巴、百度、蚂蚁金服、京东、网易、滴滴、携程、美团点评、今日头条。汽车行业类公司包括：丰田、大众、戴姆勒、宝马集团、本田汽车、通用汽车、福特汽车、日产汽车、现代汽车、上汽集团。制造行业类公司包括：格力电器、美的集团、飞利浦、海尔、索尼、松下电器、东芝、通用电器、夏普、三星集团。

接下来将上述企业名与表 4-1 中的关键词列表一一组合成种子关系对 <entity1,relation-keyword> 。这样就可以得到一个关系对集合$S_{<e,r>}$。我们将关系对中的企业名和关系词作为检索关键词$keyword_1$与$keyword_2$作为搜索引擎爬虫的输入，爬虫将自动爬取所有包含$keyword_1$与$keyword_2$的新闻网页。

对于持久化到本地的每个网页W_i，将其送入网页正文提取器当中，得到该网页的正文C_i，以句号和分号作为句子边界对正文C_i进行划分得到句子集合S_i。最后对集合S_i进行筛选，只保留那些同时包含$keyword_1$与$keyword_2$的句子，筛选过后的句子集合为S_i'。

对于S_i'中的每个句子 s，通过命名实体识别找出除$keyword_1$以外的实体 entity2。若 entity2 属于机构组织名，则将 s 进过布隆过滤器过滤判断后加入语料库中。且对于 s 来说，它的头实体为 entity1，尾实体为 entity2，关系为$keyword_2$所对应的关系。同时将<entity1, entity2>作为关系对加入种子集合$S_{<e1,e2>}$中。若 entity2 不属于机构组织名，则将该条语料丢弃。

4.4.3 句子的相似度计算及聚类

结合第二章提到的基于词向量加权的方法来获得句子的语义向量表示，我们将初始种子集中的每个句子进行向量表示，并借助 K-Means 聚类的思想对每类关系中的句子聚类求出中心点，用于之后计算待加入语料库句子类别相似度的评估。

文本向量的语义性在数学上主要体现在，语义越相似的文本之间的语义向量的距离也越近，计算方法主要包括欧式距离和夹角余弦：

$$\text{欧式距离：}\quad sim(a,b)=\sqrt{\sum_i(a_i-b_i)^2} \tag{4-7}$$

$$\text{夹角余弦：}\quad sim(a,b)=\frac{\sum_i a_i b_i}{\sqrt{\sum_i a_i{}^2\cdot\sum_i b_i{}^2}} \tag{4-8}$$

针对上小节构建的初始种子集，为了证明基于词向量加权能够很好的将同类句子聚集在一起，本文使用常见的高维数据可视化工具 t-SNE[13]对数据做了一个聚类展示。t-SNE 是一种适用于可视化的高维数据降维算法，该名称代表 t 分布随机邻域嵌入。 这个算法的思想是以低维度的方式嵌入高维点，这样就能够保留点之间的相似性信息。t-SNE 函数为高维数据创建一组低维度的点。通常，可以将低维度的点（例如平面二维）进行可视化，以查看原始高维数据中的聚类信息。

对于输入数据集X，t-SNE 首先会删除那些包含 NAN 数值的数据。经过预处理之后，t-SNE 会计算每个点x_i, x_j对之间的距离$\mathrm{d}(x_i, x_j)$，可以选择多种距离计算方式。默认

[13] https://en.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding

条件下，t-SNE 选择标准的欧几里得度量来计算距离，并且在其随后的计算中使用距离度量的平方。

然后，对于X中的每一行数据X_i，t-SNE 会计算标准误差σ_i，使得X_i的困惑度与 name-value 对困惑度相同。困惑度的定义会在接下来的高斯分布模型中定义。在原高维空间中数据点x_j和数据点x_i的相似性定义为p_{ji}，如果在以x_i为中心的高斯下以相似于其概率密度的比例选择邻居，则x_i将选择x_j作为其邻居。对于相近的数据点，p_{ji}值会相对比较大，而对于大多数分开的数据点，p_{ji}将几乎是无穷小的。条件概率p_{ji}的计算方式如下：

$$p_{j|i} = \frac{exp(-\|x_i - x_j\|^2/2{\sigma_i}^2)}{\sum_{k\neq i} exp(-\|x_i - x_k\|^2/2{\sigma_i}^2)} \tag{4-9}$$

$$p_{i|i} = 0 \tag{4-10}$$

接着通过对称条件概率来定义联合概率p_{ij}：

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \tag{4-11}$$

其中N表示输入数据X中的样本数量。令p_i表示给定数据点x_i的所有其他数据点的条件概率分布，该分布的困惑度为：

$$Perplexity(p_i) = 2^{H(p_i)} \tag{4-12}$$

其中$H(p_i)$是p_i的香农信息熵：

$$H(p_i) = -\sum_j p_{j|i} \log_2(p_{j|i}) \tag{4-13}$$

困惑度用来衡量数据点X_i有效邻居节点的个数，t-SNE 会在σ_i之上执行一个二分搜索，以便为每个点得到一个固定的困惑度。

为了将X中的数据点嵌入到低维的空间中，t-SNE 会尝试最小化高斯分布模型中的数据点与服从 Student-t 分布的数据点之间的 Kullback-Leibler 离散度。最小化过程以初

始集合 Y 开始。t-SNE 会在默认情况下创建点作为随机高斯分布点，接着会计算 Y 中每个点对之间的相似度。点y_i和y_j之间的距离分布的概率模型q_{ij}计算如下：

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_k - y_l\|^2)^{-1}} \tag{4-14}$$

$$q_{i|i} = 0 \tag{4-15}$$

联合分布P和Q之间的Kullback-Leibler离散度定义如下：

$$C = KL(P||Q) = \sum_{i \neq j} p_{i,j} log \frac{p_{ij}}{q_{ij}} \tag{4-16}$$

为了最小化Kullback-Leibler离散度，一般使用梯度下降的方法，每个Y中的点梯度值对应的离散度是：

$$\frac{dC}{d_{y_i}} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1} \tag{4-17}$$

其中正则化项为：

$$Z = \sum_k \sum_{l \neq k} (1 + \|y_k - y_l\|^2)^{-1} \tag{4-18}$$

算法4-3是用于降维可视化的t-SNE算法的详细步骤，图4-5是种子集语料库聚类的结果。算法的详细过程如下:

算法 4-3：用于降维可视化的 t-SNE 算法

Input：高维数据表示$X = x_1, \dots x_n$

output：二维数据表示

(1) Data：$X = x_1, \dots x_n$；

(2) 计算 cost function 的参数：perplexity；

(3) 优化参数：设置迭代次数 T，学习率η，动量$\alpha(t)$;

(4) 目标结果是低维数据表示$Y^T = y_1, \dots, y_n$;

(5) 开始优化

计算在给定 Perp 下的条件概率$p_{j|i}$(参见公式 4-5)；

令$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$;

用$\mathrm{N}(0, 10^{-4}I)$随机初始化Y

迭代，从$\mathrm{t} = 1$到 T，做如下操作：

1. 计算低维度下的q_{ij}(参见公式 4-6)
2. 计算梯度(参见公式 4-8)
3. 更新$Y^t = Y^{t-1} + \eta \frac{dC}{dY} + \alpha(t)(Y^{t-1} - Y^{t-2})$

结束

(6) 结束

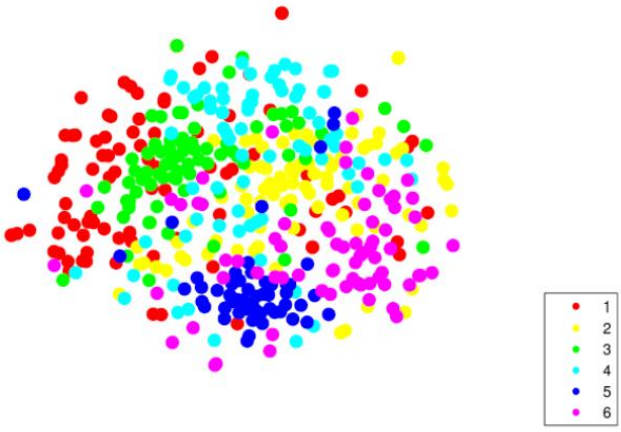

图 4-5 基于 t-SNE 种子集聚类结果

4.4.4 语料库去重

由于网络新闻中存在大量转发、转载的情况，为了保证语料库的质量，本文在构建语料库的过程中会对其进行去重处理，防止同一个句子多次出现在语料库中。去重方法是基于布隆过滤器(Bloom Filter)的方法。

布隆过滤器是一种类似于哈希表的数据结构，主要用于判断一个元素是否已经存在集合中。它支持元素的插入，但不支持删除某个元素，它的空间利用率相比较哈希表，要高出许多。但布隆过滤器存在一定的误报概率，即某个元素明明不存在集合中，但它会以为该元素存在于集合当中。但如果该元素存在于集合，布隆过滤器则不会误报它不存在。

布隆过滤器的实现核心是由一个超大的位数组和若干个哈希函数组成。哈希函数将输入数据转变为对应数组下标，假如哈希函数有 k 个，则每条数据则将被映射到数组的 k 个位置上。图 4-6 为布隆过滤器的原理示意图。

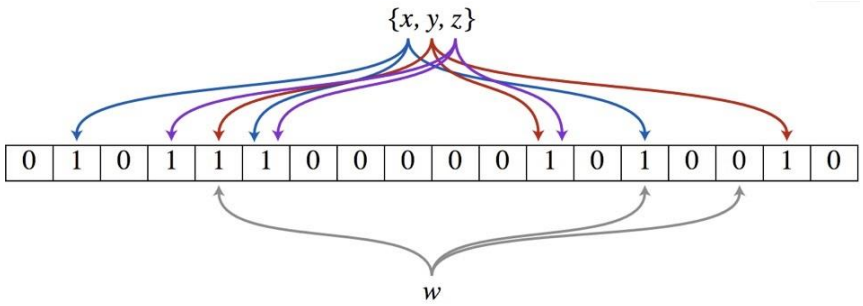

图 4-6 布隆过滤器原理示意图

算法 4-4：基于布隆过滤器的语料库去重算法

Input：存在重复数据的语料库

output：不存在重复数据的语料库

(1) Data：$X = x_1, \dots x_n$；

(2) 初始化布隆过滤器即将对应数组的每一位初始化为 0；

(3) 设计 k 个不同的哈希函数；

(4) For(i = 1 ; i <= n; i++)

将x_i送入 k 个哈希函数中，得到 k 个坐标$\{h_1 \dots h_k\}$；

检查数组对应 k 位是否为 1；

若都为 1 则表明该条数据已经存在，继续(4)进入循环；

若对应 k 位当中有一位不为 1，则说明该数据还不存在，进行插入操作；

将数组$\{h_1 \dots h_k\}$对应位置 1；

(5) 结束

4.5 实验

在本章节中，本文主要在第三章基于 CNN 的关系分类模型的基础上实施了三个实验。第一个主要是调参，针对卷积神经网络中卷积窗口大小 h 和中间隐藏层的大小来考察它们对模型分类效果的影响；第二个实验主要是对比实验，主要和传统基于特征的关系抽取方法做了个对比。第三个实验是测试系统的实际表现效果，主要方法是从互联网中随机挑选一组包含企业关系的新闻 URL，将其作为输入，将结果与人工抽取的结果做比较，得出准确率、召回率、F 值。

4.5.1 数据集及评价标准

本章节实验数据来自于 4.4 小节所得到的中文新闻当中企业关系语料。共计 1607 条，其中各个关系类型语料分布如表 4-2 所示。

表 4-2 语料统计

关系类型	频率
投资(invest)	276(17.2%)
收购(acquisition)	291(18.1%)
合作(cooperate)	290(18.0%)
竞争(compete)	256(15.9%)
董事长(chairman)	293(18.2%)
NA	201(12.5%)
Total	1607(100%)

整个系统的测试数据集为从互联网中随机选取的 100 篇包含企业关系的新闻，主要来源是网易新闻的科技频道(http://tech.163.com/)。其中所包含的企业实体以及企业实体间的关系事先已经标注好。

本章实验评价标准同第三章一样，分别是准确率(Precision)、召回率(recall)和 F 值。

4.5.2 实验设置

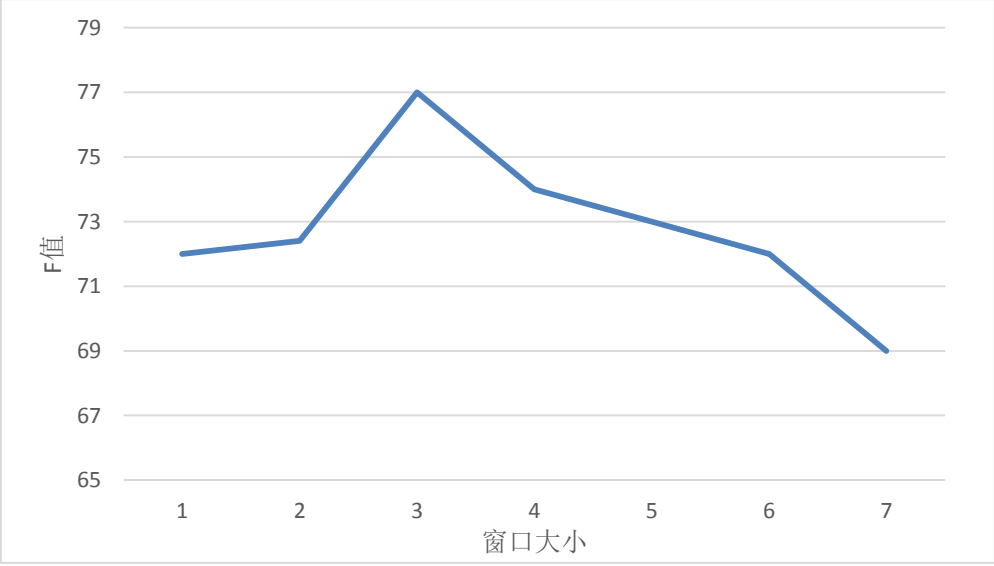

图 4-7 不同窗口大小对应 F 值的变化

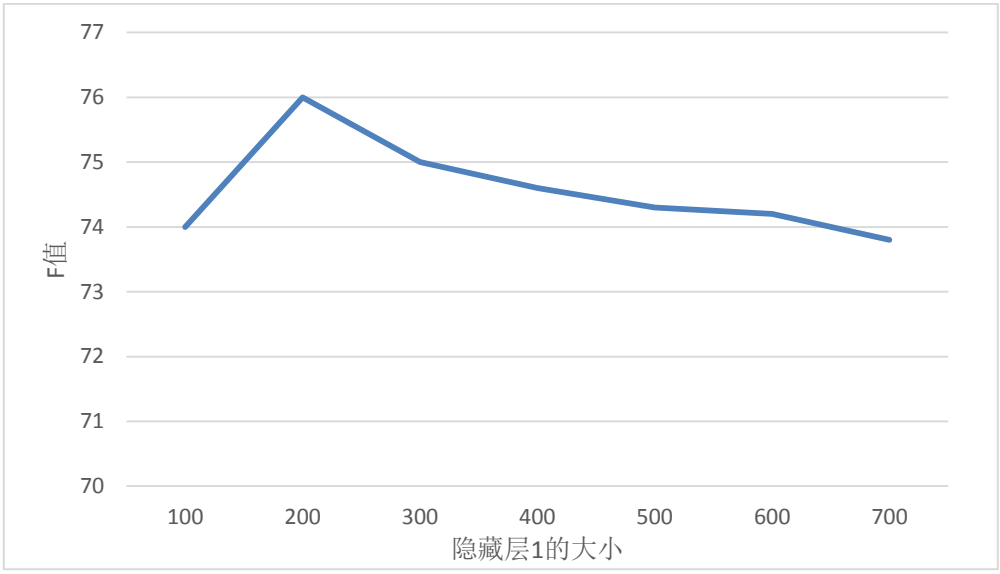

图 4-8 隐藏层 1 大小对应 F 值的变化

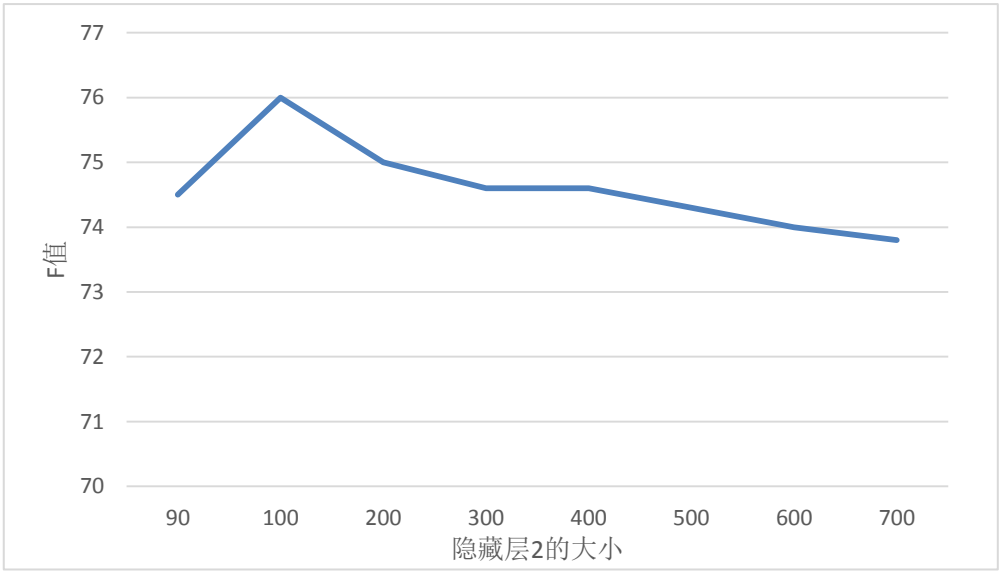

图 4-9 隐藏层 2 大小对应 F 值的变化

本小节主要通过实验来验证窗口大小和隐藏层大小对模型分类效果的影响。在图 4-7 和图 4-8 中，我们通过调整窗口大小 *h* 以及隐藏层大小 *n* 来计算不同参数下模型所对应的 F 值。最终结果是通过十折交叉验证得到的。从图中可以看出，当窗口大小为 3 时模型的 F 值最大，当在增加窗口大小时，模型的 F 值在变低。此外，由于我们使用的训练数据量较少，网络容易过拟合，特别是在使用较大的隐藏层时。从图 4-8 和图 4-9 可以看出，随着隐藏层的不断变大，对模型的影响变化很小。表 4-3 列出了本章实验的参数设置。

表 4-3 CNN 参数设置

Windows size	3
filter size	100
Drop rate	0.5
mini-batch	50
Learning rate	0.01
Hidden layer1	200
Hidden layer1	100

4.5.3 实验结果及分析

为了和传统基于人工特征的方法做对比，本文做了一组对比实验，其中所选特征主要包括实体对、句子的依存句法树结构信息，实体对之间的词序列以及词性标注的信息。实验结果见表格 4-4。

表 4-4 对比实验结果

分类器	特征集合	准确率	召回率	F 值
SVM	词序列	0.652	0.613	0.632
SVM	词序列、词性标注	0.681	0.621	0.649
SVM	词序列、词性标注、依存句法树结构	0.694	0.701	0.697
SVM	词序列、词性标注、依存句法树结构、实体对位置信息	0.712	0.694	0.703
CNN	词向量、位置嵌入	**0.786**	**0.752**	**0.768**

由表 4-4 可见，基于特征的方法更多的是依赖于所选特征集合的质量，所提特征越多，模型的表现越好，但是提升空间有限。这种提升可以通过从训练到测试数据的语义泛化的需要来解释。传统特征的质量依赖于之前的经验和相关 NLP 工具，因此很难人工构建最优的特征子集。从最终结果来看，本文所提方法效果表现最好。

4.6 本章小结

互联网上每天都会更新产生大量的新闻信息，这其中包含一些企业相关的新闻报导，例如企业之间收购、合作、竞争案例等。这些存在于网页中的新闻文本包含了企业实体之间的各种关系，这种关系信息对于企业战略制定、投资方向决策等具有重要参考价值。本章将在第三章的基础上，结合网络爬虫、网页正文提取、命名实体识别等关键技术实现对中文新闻文本当中的企业、人物关系进行抽取。针对预料构建过程当中人工标注的效率低下的问题，提出了基于 Bootstrapping 技术构建关系语料库的方法，大大提高了获取语料库的效率。

第五章 总结与展望

5.1 工作总结

本文主要工作围绕实体关系抽取任务的相关问题展开，主要包括三个部分，分别是句子的分布式表示、基于 Bootstrapping 技术构建关系语料库以及面向互联网新闻当中企业实体关系的抽取。需要注意的是，本文只考虑句子级别上的关系抽取。

第一部分针对传统词袋模型在表征句子时缺乏语义信息以及未考虑词的位置信息的缺陷，提出了基于词向量加权和基于卷积神经网络的方法用于构建紧凑且具有语义的句子分布式表示，作为构建关系分类模型的输入。

第二部分针对互联网新闻当中存在的企业实体关系，提出了基于 Bootstrapping 技术构建关系语料库的方法，克服了纯人工标注过程中费时费力的缺点。

第三部分是在前面工作基础上，结合网页正文提取、命名实体识别等关键技术，实现了面向互联网新闻文本的企业关系的抽取。

5.2 未来展望

本文有关实体关系抽取的工作还存在着一些不足，比如目前只是考虑句子级的实体关系抽取，而没有考虑段落级和篇章级的实体关系抽取；另一方面本文主要面向的是互联网新闻领域的企业实体关系抽取，需要构建相关的语料库，无法直接迁移到别的领域。因此未来的工作主要专注于两个方面：

1) 文档级关系抽取：在完成句子级的关系抽取之后，通过引入实体的上下文以及等价关系，进行一些实体关系的推理，从而实现文档级的关系抽取任务。最终可以尝试构建一套实体关系图。

2) 领域自适应的关系抽取：可以尝试使用迁移学习的方法，在已有领域经过标注了的训练语料库的基础上，实现其他领域的实体关系抽取任务[65]。

参考文献

[1] Hendrickx I, Kim S, Kozareva Z. SemEval-2010 task 8: multi-way classification of semantic relations between pairs of nominals[C]. In *Proceedings of ACL*. Singapore ,2009.

[2] Deng L, Yu D. Deep Learning: Methods and Applications[J]. *Foundations & Trends in Signal Processing*, 2014, 7(3):197-387.

[3] Farabet C, Couprie C, Najman L. Learning Hierarchical Features for Scene Labeling[J]. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2013, 35(8):1915-1929.

[4] Dahl G E, Yu D, Deng L. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition[J]. *IEEE Transactions on Audio Speech & Language Processing*, 2012, 20(1):30-42.

[5] Collobert R, Weston J, Bottou L, et al. Natural Language Processing (Almost) from Scratch[J]. *Journal of Machine Learning Research*, 2011, 12(1):2493-2537.

[6] Bengio Y. Learning Deep Architectures for AI[J]. *Foundations & Trends® in Machine Learning*, 2009, 2(1):1-127.

[7] Harris Z S. Distributional structure[M]. *Springer Netherlands:Papers in structural and transformational linguistics*, 1970: 775-794.

[8] Hasegawa T, Sekine S, Grishman R. Discovering relations among named entities from large corpora[C]. In *Proceedings of ACL*,Barcelona,2004.

[9] Chen J, Ji D, Tan C L, et al. Unsupervised feature selection for relation extraction[C].In *Proceedings of IJCNLP*. Jeju Island,Korea,2005.

[10] Suchanek F M, Ifrim G, Weikum G. Combining linguistic and statistical analysis to extract relations from web documents[C]. In *Proceedings of SIGKDD*. Philadelphia, USA ,2006.

[11] Qian L, Zhou G, Kong F, et al. Exploiting constituent dependencies for tree kernel-based semantic relation extraction[C]. *In Proceedings of COLING*. Manchester, UK ,2008.

[12] Bunescu R C, Mooney R J. Subsequence Kernels for Relation Extraction[J]. *Advances in Neural Information Processing Systems*, 2005:171-178.

[13] Bunescu R C, Mooney R J. A shortest path dependency kernel for relation extraction[C]. In *Proceedings of ACL*. Sydney, Australia ,2005.

[14] Mintz M, Bills S, Snow R, et al. Distant supervision for relation extraction without labeled data[C] In *Proceedings of ACL*. Singapore,2009.

[15] Takamatsu S, Sato I, Nakagawa H. Reducing wrong labels in distant supervision for relation extraction[C] In *Proceedings of ACL*. Jeju Island, South Korea,2012.

[16] Yao L, Riedel S, McCallum A. Collective cross-document relation extraction without labelled data[C]. In *Proceedings of ACL*.Uppsala,Sweden,2010.

[17] Riedel S, Yao L, McCallum A. Modeling relations and their mentions without labeled text[J]. *Machine learning and knowledge discovery in databases*.2010: 148-163.

[18] Hoffmann R, Zhang C, Ling X, et al. Knowledge-based weak supervision for information extraction of overlapping relations[C]. In *Proceedings of ACL*.Oregon,USA,2011.

[19] Surdeanu M, Tibshirani J, Nallapati R, et al. Multi-instance multi-label learning for relation extraction[C]. In *Proceedings of ACL*. Jeju Island, South Korea,2012.

[20] Socher R, Huval B, Manning C D, et al. Semantic compositionality through recursive matrix-vector spaces[C]. In *Proceedings of ACL*. Jeju Island, South Korea,2012.

[21] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional

deep neural network[C]. In *Proceedings of COLING*, Dublin, Ireland,2014.

[22] Yarowsky D. Unsupervised word sense disambiguation rivaling supervised methods[J]. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, 1970:189--196.

[23] Blum A, Mitchell T M. Combining Labeled and Unlabeled Sata with Co-Training[C]. *Eleventh Conference on Computational Learning Theory*, Wisconsin, Usa, 1998.

[24] McDonald R. Extracting relations from unstructured text[J]. *Rapport technique, Department of Computer and Information Science*,University of Pennsylvania, 2005.

[25] Abney S. Understanding the Yarowsky Algorithm[J]. *Computational Linguistics*, 2006, 30(3):365-395.

[26] Brin S. Extracting Patterns and Relations from the World Wide Web[J]. *Lecture Notes in Computer Science*, 1998, 1590:172-183.

[27] Agichtein E, Gravano L. Snowball : extracting relations from large plain-text collections[C]. In *Proceedings of ACM Conference on Digital Libraries*, San Antonio, USA,2000.

[28] Etzioni O, Cafarella M, Downey D, et al. Unsupervised named-entity extraction from the Web: An experimental study [J]. *Artificial Intelligence*, 2005, 165(1):91-134.

[29] Banko M, Cafarella M J, Soderland S, et al. Open information extraction from the web[C]. In *Proceedings of* IJCAI. Hyderabad, India,2008.

[30] Brin, Sergey. Extracting Patterns and Relations from the World Wide Web[M]. *The World Wide Web and Databases. Springer Berlin Heidelberg*, 1998:172-183.

[31] Huang X, You H, Yu Y. A Review of Relation Extraction[J]. *New Technology of Library & Information Service*, 2013.

[32] Kambhatla, Nanda. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations[C]. In *Proceedings of ACL*,Barcelona,2004.

[33] Zhao S, Grishman R. Extracting relations with integrated information using kernel methods[C]. In *Proceedings of ACL*. Sydney, Australia,2005.

[34] Miller S, Fox H, Ramshaw L, et al. A novel use of statistical parsing to extract information from text[C]. In *Proceedings of ACL*. Hong Kong, 2000.

[35] Culotta A, Mccallum A, Betz J. Integrating probabilistic extraction models and data mining to discover relations and patterns in text[C]. In *Proceedings of ACL*. Sydney, Australia, 2006.

[36] Lodhi H, Saunders C, Shawe-Taylor J, et al. Text classification using string kernels[J]. *Journal of Machine Learning Research*, 2002, 2(3):419-444.

[37] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. Relation classification via convolutional deep neural network[C]. In *Proceedings of COLING*, Dublin, Ireland,2014.

[38] Santos C N, Xiang B, Zhou B. Classifying relations by ranking with convolutional neural networks[J]. *Computer Science*, 2015, 30(3):365-395.

[39] Miwa M, Bansal M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures[C]. In *Proceedings of ACL*. Jeju Island, South Korea,2012.

[40] Alperin J L. Local representation theory[M]. *England:Cambridge University Press*, 1986.

[41] Sahlgren M. The distributional hypothesis[J]. *Italian Journal of Linguistics*, 2008, 20(1): 33-54.

[42] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. *Computer Science*, 2013,30(3):365-395.

[43] Le Q V, Mikolov T. Distributed Representations of Sentences and Documents[J]. *Computer Science*, 2014, 4:1188-1196.

[44] Sutskever I, Vinyals O, Le Q V. Sequence to Sequence Learning with Neural Networks[J]. *Advances in Neural Information Processing Systems*, 2014, 4:3104-3112.

[45] Mikolov T, Sutskever I, Chen K, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. *Advances in neural information processing systems*, 2013, 26:3111-3119.

[46] Lebret R, Collobert R. Word Emdeddings through Hellinger PCA[J]. *Computer Science*, 2013, 4:3104-3112.

[47] Levy O, Goldberg Y. Neural word embedding as implicit matrix factorization[J]. *Advances in Neural Information Processing Systems*, 2014, 3:2177-2185.

[48] Li Y, Xu L, Tian F, et al. Word embedding revisited: a new representation learning and explicit matrix factorization perspective[C]. *In Proceedings of AAAI*, Austin,USA,2015.

[49] Globerson A, Chechik G, Pereira F, et al. Euclidean Embedding of Co-occurrence Data[J]. *Journal of Machine Learning Research*, 2004, 8(4):2265-2295.

[50] Levy O, Goldberg Y. Linguistic Regularities in Sparse and Explicit Word Representations[C]. In *Proceedings of COLING*. Dublin, Ireland,2014.

[51] Nicholas Metropolis, S. Ulam. The Monte Carlo Method[J]. *Journal of the American Statistical Association*, 1949, 60(247):252.

[52] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. *Proceedings of the IEEE*, 1998, 86(11):2278-2324.

[53] Yih W T, He X, Meek C. Semantic Parsing for Single-Relation Question Answering[C]. In *Proceedings of ACL*. Jeju Island, South Korea,2014.

[54] Shen Y, He X, Gao J, et al. Learning semantic representations using convolutional neural networks for web search[C].Companion Publication of the, International Conference on World Wide Web Companion. 2014:373-374.

[55] Kalchbrenner, E. Grefenstette, P. Blunsom. 2014. A Convolutional Neural Network for Modelling Sentences. In *Proceedings of ACL*.Maryland,USA,2014.

[56] Kim Y. Convolutional Neural Networks for Sentence Classification[J]. *Eprint Arxiv*, 2014,8(4):2265-2295.

[57] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. *Journal of Machine Learning Research*, 2014, 15(1):1929-1958.

[58] Hinton G E, Srivastava N, Krizhevsky A, et al. Improving neural networks by preventing co-adaptation of feature detectors[J]. *Computer Science*, 2012, 3(4):212-223.

[59] Rumelhart D E, Hinton G E, Williams R J. Learning representations by back-propagating errors[J]. *Parallel Distributed Processing Explorations in the Microstructure of Cognition*, 1986, 323(6088):533-536.

[60] Cai D, Yu S, Wen J R, et al. VIPS: a Vision-based Page Segmentation Algorithm[J]. *Microsoft Research*, 2003.

[61] 赵明明, 陶华, 伏虎,等. 网页正文提取方法研究[J].*中国科技论文在线*,2011,39(10):24-27

[62] 陈鑫. 基于行块分布函数的通用网页正文抽取[J].http://code. google. com/p/cx-extractor, 2010

[63] Lafferty J D, Mccallum A, Pereira F C N. Conditional Random Fields: Probabilistic Models For Segmenting And Labeling Sequence Data[J]. In *Proceedings of ICML*. San Francisco, USA, 2002, 3(2):282--289.

[64] Mccallum A, Li W. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons[C]. In *Proceedings of ACL*.Maryland,USA,2014.

[65] 王莉峰. 领域自适应的中文实体关系抽取研究[D]. 哈尔滨工业大学, 2011.

致谢

时光飞逝，转眼间即将告别我的研究生生活。三年前乘坐地铁二号线来到南京大学仙林校区复试的场景依然历历在目，现已不知不觉在这里生活学习了三年。这三年短暂而又美好，让我学习到了很多东西、遇到了很多优秀的人。

首先要感谢我的导师王崇骏教授和李宁副教授，在这三年里给了我许许多多的指导和帮助，在我科研迷茫时鼓励指点我，在我懒散浮躁时不断鞭策我，他们就像一盏明灯指引着我这三年以及今后的人生道路。

其次要感谢我的父母，求学二十载，不计辛苦不求回报地默默在背后为我付出，帮助我完成学业。

最后要感谢实验室的小伙伴们，我们一起出差、一起做项目、一起找工作、一起憋论文，是你们让我这三年的研究生生活变得丰富多彩，他们是高杨、汤兆亮、陈嘉伟、冯翰洋以及韩建军。同时还要感谢我的师妹夏丽，感谢她帮我做实验、标注数据集，没有她的帮助我的论文不可能这么快完成。祝愿你们在今后的人生道路上能够实现梦想、充满希望。

附录

研究生期间专利

[1] 张雷,刘焕锐,资帅,**王强**,吴和生,谢俊元,一种中药方剂核心药物的发现方法:201510183745.0；

[2] 吴骏,**王强**,李振兴,李宁,一种基于卷积神经网络的企业实体关系抽取的方法:201710371463.2；

研究生期间参与项目

[1] 科技部重点研发计划：跨时空异构数据的结构化描述和语义协同，2016YFB1001102；

[2] 某企业资助项目：互联网数据富集工具软件系统；

[3] 重庆市交委资助：重庆市交委企业互联网征信项目；

附件二

《学位论文出版授权书》

本人完全同意《中国优秀博硕士学位论文全文数据库出版章程》(以下简称“章程”)，愿意将本人的学位论文提交“中国学术期刊(光盘版)电子杂志社”在《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》中全文发表。《中国博士学位论文全文数据库》、《中国优秀硕士学位论文全文数据库》可以以电子、网络及其他数字媒体形式公开出版，并同意编入《中国知识资源总库》，在《中国博硕士学位论文评价数据库》中使用和在互联网上传播，同意按“章程”规定享受相关权益。

作者签名：_____

2017 年 5 月 26 日

论文题名	基于卷积神经网络的实体关系抽取研究				
研究生学号	MF1433046	所在院系	计算机科学与技术	学位年度	2017
论文级别	☐硕士 ☐博士		☐硕士专业学位 ☐博士专业学位		(请在方框内画钩)
作者 Email	wqchina007@126.com				
导师姓名	李宁 副教授				

论文涉密情况：

☐ 不保密

☐ 保密，保密期(_____年_____月_____日 至 _____年_____月_____日)

注：请将该授权书填写后装订在学位论文最后一页(南大封面)。