中山大学硕士学位论文

基于深度学习和序列标注的因果知识抽取方法研究

Research on Causal Knowledge Extraction Method based on Deep Learning and Sequence Labeling

学位申请人：李肇宁

指导教师：任江涛

专业名称：工程（软件工程）

答辩委员会主席（签名）：

答辩委员会委员（签名）：

二零一八年 五月 十六日

论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究作出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日　　期：　　年　　月　　日

学位论文使用授权声明

本人完全了解中山大学有关保留、使用学位论文的规定，即：学校有权保留学位论文并向国家主管部门或其指定机构送交论文的电子版和纸质版，有权将学位论文用于非赢利目的的少量复制并允许论文进入学校图书馆、院系资料室被查阅，有权将学位论文的内容编入有关数据库进行检索，可以采用复印、缩印或其他方法保存学位论文。

学位论文作者签名：　　导师签名：

日期：　　年　　月　　日　　日期：　　年　　月　　日

论文题目：基于深度学习和序列标注的因果知识抽取方法研究

专　　业：软件工程

硕 士 生：李肇宁

指导教师：任江涛

摘 要

因果知识即了解事件发生的前因、后果等关系的知识。由于因果知识可以用于事实推理进而影响决策制定，所以其在人类认知世界的过程中扮演了十分重要的角色。对事件或实体间的因果关系进行抽取，可以了解信息之间的来龙去脉，获取信息的演化关系，有助于预测和决策。这种因果知识发现在很多领域（例如：金融、医学、生物学、环境科学等）都是非常有价值的。同时因果知识的自动抽取对于许多自然语言处理任务（例如：事件预测、文本生成、问题回答及语篇理解）也都是至关重要的一步。但由于自然语言文本的二义性及多样性，从自然语言文本中自动抽取因果知识一直是人工智能领域中一个具有挑战性的开放性问题。

针对这一问题，大多数早期的尝试都是在小规模或特定领域的数据集上，通过人工构建语言学或语法规则来抽取文本中的因果知识，这种基于规则的方法虽然可取得较高的准确率，但是其通用性较差且对领域知识的要求很高；随着计算机计算能力的不断提高以及机器学习技术的普及，现有的主流方法将规则与机器学习技术相结合，并以流水线方式——先用规则抽取候选因果关系对，然后再用机器学习算法来滤除其中的非因果关系对以抽取文本中包含的因果知识，这种方法虽不需要过多的领域知识，但是却严重依赖于文本特征的人工选择，需要在特征工程上耗费大量时间和精力。

为解决传统方法中存在的这些问题，更高效地抽取因果知识，本文将因果知识抽取归约为一个可利用深度学习模型解决的序列标注问题，其不使用任何人工特征。在此基础上，本文研究了多种基于 Bi-LSTM 网络的端到端抽取模型，以

达到直接抽取因果知识的目的，而不必同传统方法一样将因果知识抽取分为两个子任务。此外，针对因果序列标注中存在的标签类别不平衡问题，本文提出了一种以 Focal Loss（焦点损失函数）为损失函数的端到端模型：Bi-LSTM-Softmax (FL)，实验结果表明该模型可以有效提高因果之间的关联性，因而取得了显著优于其他基准模型的结果。

关键词：因果知识抽取，序列标注，Bi-LSTM 网络，焦点损失函数

Title: Research on Causal Knowledge Extraction Method based on Deep Learning and Sequence Labeling

Major: Software Engineering

Name: Zhaoning Li

Supervisor: Jiangtao Ren

Abstract

Causal knowledge is the knowledge of the relationship between the cause and effect of an incident. Causal knowledge plays a critical role in the process of human cognizing the world because it can be used for reasoning and thus influence decision making. The extraction of causality between events or entities can help people understand the sequence and the evolution of information, helping people to predict and make decisions. This kind of causal knowledge discovery is very valuable in many fields, such as finance, medicine, biology, environmental science. At the same time, automatic extraction of causal knowledge is also a crucial step for many natural language processing tasks, such as event prediction, generating future scenarios, question answering, and discourse comprehension. However, due to the ambiguity and diversity of natural language texts, causal knowledge extraction from natural language texts is a challenging open problem in artificial intelligence.

In response to this problem, most of the early attempts used manually constructed linguistic and syntactic rules to extract causal knowledge on small or domain-specific datasets. Although this rule-based method can achieve higher accuracy, its cross-domain applicability is weak, and it requires extensive domain knowledge. With the continuous improvement of computer computing capabilities and the popularity of machine learning techniques, the existing mainstream methods combine rules and machine learning techniques and treat this task in a pipeline manner. They firstly extract candidate causal pairs with rules and then use machine learning algorithms to filter non-

causal pairs among candidate pairs. This method does not require too much domain knowledge, but it relies heavily on the manual selection of text features and often requires considerable human effort and time on feature engineering.

To tackle these problems, we formulate causal knowledge extraction as a sequence labeling problem based on deep learning model, which does not use any handcrafted features. Then, we investigate different Bi-LSTM based end-to-end models to directly extract cause and effect, without extracting candidate causal pairs and identifying their relations separately. Besides, to address the tag class imbalance problem in causal sequence labeling, we propose an end-to-end model with Focal Loss as a loss function: Bi-LSTM-Softmax (FL). Experimental results show that the model can effectively enhance the association between cause and effect and thus outperforms the baseline models.

Keywords： Causal Knowledge Extraction, Sequence Labeling, Bi-LSTM Networks, Focal Loss

目 录

第 1 章 绪论

1.1 研究背景及意义

1.1.1 研究背景

因果知识就是“知道为什么”的知识，即了解一件事情发生的前因与后果等关系的知识。在自然语言文本中存在着大量可利用的因果性知识。

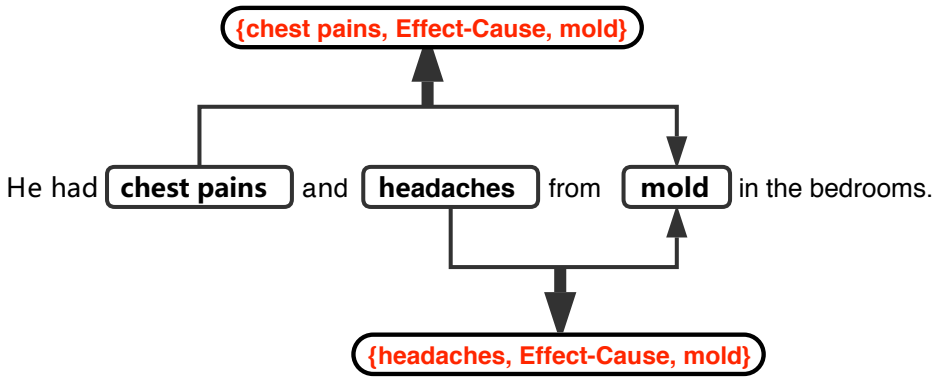

图 1-1 一个包含因果性知识的例句。在此句中，“*mold*”为原因，“*chest pains*”和“*headaches*”是“*mold*”所造成的结果

如图 1-1 所示，本文可以从图中文本抽取出：“*mold*”（霉菌）会导致“*chest pains*”（胸口痛）和“*headaches*”（头痛）这样的因果知识。但由于自然语言文本的二义性及多样性，因果知识抽取目前仍然是一个很难解决的 NLP 问题[1]。还是以图 1-1 中的句子为例：

“*He had chest pains and headaches* ***from*** *mold in the bedrooms.*”

在这个句子中，很显然，单词“*from*”为因果关系：（“*chest pains and headaches*”与“*mold*”）的触发词；但在下面这个句子中，“*from*”却没有触发任何因果关系：

“*The number of applicants* ***from*** *other countries is increasing.*”

现有的因果知识抽取方法可以分为两类：基于规则的方法[2-7]和基于规则与机器学习技术相结合的方法[8-13]。仅依赖规则进行模式匹配的方法往往通用性不强，无法兼顾精确率与召回率，在解决特定领域问题时需要大量领域知识，而且制定规则耗费大量时间和人力。基于规则与机器学习相结合的方法又往往需要大量的特征工程，严重依赖于人工选择文本特征，同样需要耗费大量时间与精力。而且后者通常以流水线方式处理因果知识抽取任务，即将其分为候选因果对抽取与因果关系分类（滤除非因果关系）两个子任务，这样处理虽使得总体任务易于处理，但是将两个子任务作为独立模型处理会忽视子任务之间的相关性，而且候选因果对抽取的结果可能会影响到因果关系分类的性能并且产生级联错误[14]。

1.1.2 研究意义

本研究的理论意义就在于针对传统因果知识抽取方法存在的问题，将因果知识抽取归约为序列标注问题，并结合深度学习技术，提出一种新型因果知识抽取方法，以达到提高模型抽取因果知识能力的同时，最大限度减少特征工程的目的；此外，针对应用序列标注后出现的标签类别不平衡问题，本文还将目标检测领域中的焦点损失函数（Focal Loss）[15]应用到模型中以更好地抽取因果知识，这对于解决自然语言处理任务中类别不平衡的问题有一定的借鉴意义。

在实际应用意义方面，随着互联网技术的不断发展，从飞速增长的文本数据中高效准确地抽取因果性知识在很多领域变得越来越重要。例如：在金融领域，从上市公司年报中抽取与财经指标相关的因果知识（如：“……毛利率下降……是因为……”），为投资者提供参考；在医疗领域，从电子病历中抽取病情描述与已确诊疾病之间的因果关系（如:“患者自述胃痛、腹泻……诊断为急性肠胃炎”），为医生提供诊断参考。同时，因果知识抽取对于许多自然语言处理任务也是不可或缺的一步，例如：信息检索[2]，问题回答[8]，事件预测[16,17]，剧本生成[18,19]以及决策处理[20]。

综上所述，本研究在理论方面和实际应用中都有着比较重要的意义。

1.2 本文主要工作

本文为解决传统因果知识抽取方法中存在的问题，受文献[21]启发，将注意力集中于因果三元组以达到直接抽取因果知识的目的。本文用因果三元组来表示自然语言文本中存在的因果性知识，在此，本文对因果三元组（Causal Triplet）给出如下定义：（其中实体也可替换为单词、短语或事件）

$$Causal\ Triplet = \{Entity\ 1, Relation\ r, Entity\ 2\} \tag{1-1}$$

其中：$\forall r \in R$，$R = \{Cause\text{-}Effect, Effect\text{-}Cause\}$。

以图 1-1 为例，其中例句所包含的因果知识可以用两个因果三元组来表达：$\{chest\ pains, Effect\text{-}Cause, mold\}$，$\{headaches, Effect\text{-}Cause, mold\}$。由此，本文可以直接对因果三元组建模以抽取因果知识，而不必将因果知识抽取分为两个子任务分别处理。基于以上考虑，本文将因果知识抽取归约为一个序列标注问题，并设计了一套因果知识标注方案来达到直接抽取因果知识的目的。在解决该问题的过程中，本文还把深度学习的一些方法和技术结合进来，最大限度减少特征工程的同时，对自然语言文本中的因果知识有效建模。

具体而言，在应用因果知识标注方案标注数据之后，本文使用在文本语义建模中性能优异[22,23]的模型——长短时记忆网络（LSTM Networks）[24]来直接抽取因果知识。此外，本文还研究了多种基于 Bi-LSTM 网络的端到端模型，以取得对因果知识抽取的最优结果。然而，在应用因果知识标注方案标注数据之后，本文发现句子中因果标签的数量远少于非因果标签的数量，在本文进行实验的数据集中，因果类标签（“B-C”，“I-C”，“B-E”，“I-E”）与非因果标签（“O”）的比例约为 1:28。由此产生模型分类难度差异问题可能会影响模型的性能。为解决序列标签中存在的类别不平衡问题，本文首次将在目标检测（Object Detection）领域表现优异的损失函数：Focal Loss 应用到自然语言处理任务中。Focal Loss 通过对原交叉熵损失函数的重构，削弱了易分类标签（well-classified tags）对总体损失的影响，并因此而聚焦于训练一些难分类的标签（hard tags）。本文将原论文中的二分类 Focal Loss 修改为多分类 Focal Loss 以应用于本文中的因果序列标注任务。

本文的创新点和主要贡献主要在于以下几点：

（1）本文首次使用深度学习技术来解决英文文本中的因果知识抽取问题，最大限度地减少了特征工程(本文仅使用预训练好的词向量作为输入，见 3.2 节)，并实现了对自然语言文本中因果知识的有效建模。

（2）在本文设计的因果知识标注方案基础上，本文还研究了多种基于 Bi-LSTM 网络的端到端因果知识抽取模型，以实现因果知识抽取的最佳性能。

（3）本文首次使用 Focal Loss 来解决因果序列标注中出现的标签类别不平衡问题，并提出了一种以 Focal Loss 为损失函数的端到端因果知识抽取模型：Bi-LSTM-Softmax (FL)。后续实验结果显示，本文提出的模型取得了 state-of-the-art 的结果。

1.3 论文整体结构

本文各章的主要内容安排如下：

第一章：绪论。本章主要介绍本文所研究问题的背景及其意义，随后介绍了本文的主要工作及创新点和主要贡献点，最后对论文的整体章节安排结构进行了说明。

第二章：相关工作综述。本章主要介绍并分析了当前国内外在因果知识抽取领域的相关研究工作。

第三章：因果知识抽取方法。本章详细阐述了本文在抽取因果知识中所采用的方法以及涉及的关键技术。

第四章：实验设计及结果分析。本章介绍了本文所进行的针对因果知识抽取的实验，并给出了详细的实验设置及实验结果，最后对实验结果进行了相应的讨论和分析。

第五章：因果知识抽取系统。本章主要介绍了本文实现的一个因果知识抽取系统，该系统可为用户提供因果序列标注以及因果分析功能。

第六章：总结与展望。本章对论文的研究工作作出总结，同时分析了研究工作中存在的不足之处，并对后续工作作出展望。

第 2 章 相关工作综述

本章首先介绍因果知识抽取领域的相关研究工作，然后简要回顾使用深度学习进行序列标注这种方法近年来的发展，最后给出对这些研究工作的分析及总结。

2.1 基于规则的因果知识抽取方法

基于规则的因果知识抽取方法，大多数都是在小规模或特定领域的数据集上，通过人工预定义好的语义规则，结合语料并使用语言学、语法和语义特征进行模式匹配来抽取因果知识。

Khoo 等人[2]通过对语言学知识的大量深入研究以及对目标语料文本的深入分析，得到了一组可以代表因果关系的语言模式。他们最终实现了一个利用这些语言模式进行模式匹配，进而从华尔街日报中抽取因果性知识的自动因果知识抽取系统。

在医疗领域，Khoo 等人[3]通过可以指示因果关系的动词语义模式，在医疗数据库文本中进行模式匹配来抽取其中的因果知识，其精确率达到 68%。

Girju 等人[4]从自然语言文本中抽取出具有句法规则：$<NP1$ 因果动词 $NP2>$的候选因果关系对，然后采用一些语法语义约束将候选因果关系对划分为因果关系或非因果关系以抽取出因果知识。

Ittoo 和 Bouma[5]提出了一种基于词性、句法分析和因果关系模版的因果关系对抽取方法，他们首先从维基百科上包含因果关系的句子中抽取出因果关系模版，然后再使用这些模版去抽取其他文本中的因果关系。

干红华等人[6]提出了一种基于事件的因果关系结构分析方法，并运用因果关系的默认逻辑表达方式表达法律知识，形成规则库，实现了一个计算机辅助法律分析与解释系统[7]。

2.2 基于规则与机器学习相结合的因果知识抽取方法

基于规则与机器学习技术相结合的方法主要是以流水线（pipeline）方式处理因果知识抽取这一任务。这些方法首先根据规则或一些线索词抽取出可能具有因果关系的候选短语（或实体、事件）对，然后根据语义及语法特征或者某些统计特征采用传统机器学习算法对候选因果对进行分类，以滤除非因果关系对。

Girju[8]在一个问答系统中使用基于因果触发词的规则约束来抽取英文文本中的因果关系，然后使用 C4.5 决策树算法对这些候选因果关系进行分类，实现了 73.91%的精确率。

Sorgente 等人[9]使用预定义好的规则来抽取候选因果关系对，然后使用贝叶斯分类器和拉普拉斯平滑来滤除非因果关系对，其算法流程图如图 2-1 所示：

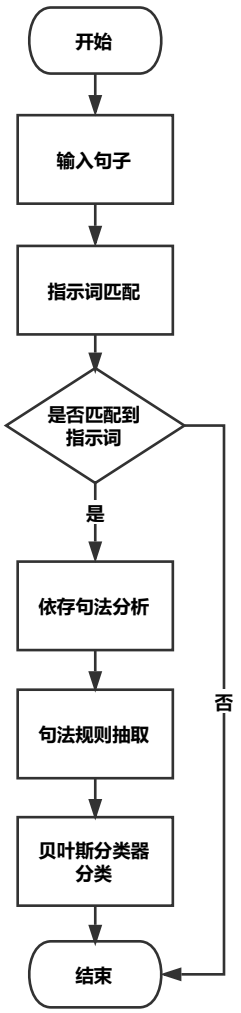

图 2-1 Sorgente 等人抽取因果知识的算法流程图

在 Sorgente 等人的工作中，他们首先检查输入语句中是否含有预定义好的因果指示词（例如：$cause$、$generate$、$result\ in$、$from$等）；若有则会使用 Stanford Paser[25]对该句进行依存句法分析，并使用预定义好的句法规则抽取出候选的因果对，在此以图 1-1 例句为例，其句法分析图如图 2-2 所示：

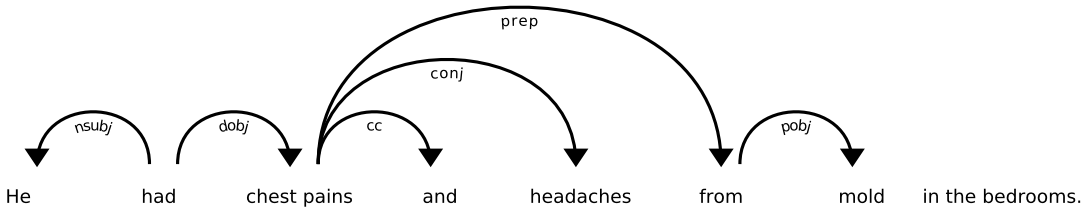

图 2-2 例句的句法分析图

在此基础上，人工定义以下 3 条规则：（“S”表示 Sentence，“C”表示 Cause，“E”表示 Effect，“$pobj$”表示介词的宾语，“$prep$”表示介词修饰，“$conj$”表示连接两个并列的词）

$$pobj(S, from, C) \rightarrow cause(S, C)$$

$$prep(S, E, from) \rightarrow effect(S, E)$$

$$effect(S, E1) \wedge conj(S, E1, E2) \rightarrow effect(S,\ E2)$$

然后，应用以上规则便可抽取出该类型句子中存在的因果对：$(mold, chest\ pains)$，$(mold, headaches)$；最后，他们使用一个基于词汇、语义和依存句法特征的贝叶斯分类器来滤除非因果关系的噪声词对。

在文献[10]中，Blanco 等人首先人工识别出可能编码因果关系的句法模式——他们发现在其使用的语料中四种最常见的因果关联词为：“$because$”、“$since$”、“as”和“$after$”，然后他们使用决策树算法对包含关联词的文本分类，以滤除其中的非因果关系文本。不过这种方法无法区分因果关系中的 Cause 和 Effect。

Zhao 等人[11]通过计算句子句法依存结构的相似性提出了一种名为“因果连接词”（Causal Connectives）的新特征。在 Zhao 等人的工作中，他们首先使用一个部分语法解析器来抽取可能表达因果关系的候选名词性短语，然后用受限隐含朴素贝叶斯学习算法结合其他特征对这些候选名词性短语对进行分类。但是他们

的方法无法区分因果知识中的 Cause 和 Effect。

Luo 等人[12]使用预定义好的因果线索词从大规模网络文本语料库（其大小约为 10TB）中抽取出共现因果词组，并统计其出现次数，然后在此基础上使用一种基于点互信息（Pointwise Mutual Information, PMI）计算的评价指标来衡量两个单词或短语乃至两段短文本之间的因果强度。Luo 等人将这种评价指标命名为“Causal Strength”（CS），CS 虽与 PMI 类似，但其对因果关系的表征能力却强于 PMI。CS 由必要性因子和充分性因子组成，对于 cause i_c和 effect j_e，必要性因子和充分性因子可由下式来定义：

$$CS_{nec}(i_c, j_e) = \frac{p(i_c|j_e)}{p^{\alpha}(i_c)} = \frac{p(i_c, i_e)}{p^{\alpha}\,(i_c)p(j_e)} \tag{2-1}$$

$$CS_{suf}(i_c, j_e) = \frac{p(j_e|i_c)}{p^{\alpha}(j_e)} = \frac{p(i_c, i_e)}{p^{\alpha}\,(j_e)p(i_c)} \tag{2-2}$$

其中$CS_{nec}(i_c, j_e)$为必要性因子，$CS_{suf}(i_c, j_e)$为充分性因子，α为一超参数，文献[12]中设置为 0.66。

在定义好必要性因子和充分性因子之后，可定义$CS(i_c, j_e)$如下：

$$CS(i_c, j_e) = CS_{nec}(i_c, j_e)^{\lambda} CS_{suf}(i_c, j_e)^{1-\lambda} \tag{2-3}$$

其中λ为一可调节的超参数。

Sasaki 等人[13]针对文献[12]中仅统计共现因果词组，忽略多词表达（如：$tired - give\ up$ 会被记为 $tired - give, tired - up$）这一问题，预定义好多词表达字典，并在抽取和统计共现因果词组时将多词表达也考虑进去，从而更好地对因果强度作出估计。

此外，付剑锋等人[26]、钟军[27]等人将针对事件的因果关系抽取转化为对事件序列的两次模式识别标注问题（即先标注事件的语义角色，再识别因果关系的边界），使用事件触发词、事件类别、事件极性等特征作为输入，采用条件随机场（Conditional random field, CRF）来抽取自然语言文本中事件之间存在的因果关系。

2.3 基于 RNN 的序列标注方法

循环神经网络（RNN），特别是长短时记忆网络（LSTM Networks）由于可以充分考虑文本的时序信息与上下文的深层语义信息，现已被广泛应用于自然语言处理任务，并在许多序列标注任务（例如：词性标注[28]、命名实体识别[28]、组块化[29]）中取得了 state-of-the-art 的效果。

另外，近年来还有很多研究人员提出了针对 LSTM 网络（或 RNN）这种结构的改进或扩展，以提高模型在序列标注任务中的表现：

Schuster 和 Paliwal[30]提出可以同时考虑前向和后向信息的双向 RNN（BRNN）。Gal 和 Ghahramani[31]提出了 variational dropout，不同于普通的 dropout[32]，variational dropout 不仅可在 LSTM 网络层的每一个时间步上应用 dropout，还可以将 dropout 应用于循环单元的内部连接上，从而更好地防止模型过拟合。在序列标注任务中，输出标签之间通常有很强的关联性，因此 Huang 等人[33], Ma 和 Hovy[28], Lample 等人[34]除使用 LSTM 网络外，还在 LSTM 层之上使用 CRF 层来联合地解码整个句子的标签，以充分使用句子级别的标签信息。Ma 和 Hovy[28]与 Lample 等人[34]还分别使用卷积神经网络（Convolutional Neural Network，CNN）[35]和 LSTM 网络将单词的字符信息编码为字符向量，这样在把字符向量与词向量结合之后，模型可在序列标注任务中获得更优异的表现。Søgaard 和 Goldberg[29]针对序列标注任务提出了一种基于 Bi-LSTM 网络的多任务学习结构，即同时针对多种任务（如：词性标注、命名实体识别等）联合训练模型。此外，Zheng 等人[21]还将偏置损失函数（Bias Loss）作为模型损失函数应用于 LSTM 序列标注模型中，以增强实体标签效果的同时削弱无效标签的影响。

2.4 分析与启发

仅依赖规则进行模式匹配以抽取因果知识的方法往往通用性不强，解决特定领域问题时可能会需要大量领域知识，同时制定规则耗费大量时间和人力。而且仅依赖规则进行模式匹配的方法对数据本身要求较高，即语言本身最好是语法严密、遵守统一规则的，在实际应用中，由于语言使用的随意性及语言本身的多样性，这显然是不现实的。

基于规则与传统机器学习相结合的因果知识抽取方法，将因果知识抽取划分

为候选因果对抽取与关系分类（滤除非因果对）两个子任务。以这种流水线的方式来抽取文本中的因果知识，虽然可以将问题简化为两个易于处理的子任务，却人为切断了两个子任务之间的关联性。与此同时，前一子任务中的错误还将会传递到下一子任务，而前一子任务却得不到任何关于错误的反馈信息。

传统机器学习方法往往在特征工程上需要耗费大量时间与精力，这样模型的性能将严重依赖于人工选择特征。具体到因果知识抽取任务中，由 2.2 节可知，从本质上讲，方法[8-11]及方法[26,27]使用的都是基于词性、语义或句法的特征，这些特征一般比较简单，难以捕捉上下文的深层语义信息，很难做到对文本中的因果知识有效建模；此外，方法[12,13]虽都使用的是基于统计的特征，并用 CS 来衡量文本之间的因果强度，但是他们的方法基于一个潜在的假设，即共现因果词组所表现出的相关性可以用来衡量因果。显然，即使方法[12,13]所使用的数据量非常巨大，也不能保证其统计得出的共现特征就可以准确无误地表征因果关系——因果关系与共现关系相比要求更加严格（如：Cause 与 Effect 需满足在时间/空间上的连续性）。

不同于浅层机器学习方法，深层神经网络依靠其强大的表示学习能力，可以自动发掘特征，极大地减少了特征工程，节约了人力和时间。经过以上分析和考虑，针对传统因果知识抽取方法中存在的问题，本文决定将因果知识抽取归约为基于深度学习的序列标注问题，这样既可以最大限度地减少特征工程，又可以更有效地对因果知识建模。

综上所述，本文提出的基于深度学习和序列标注进行因果知识抽取的方法是合理的。

2.5 本章小结

本章首先介绍了因果知识抽取领域的相关研究现状，详细列举了两类因果知识抽取方法中比较典型且具有代表性的研究，然后简要回顾了基于 RNN 进行序列标注近年来的相关研究工作，最后经过 2.4 节的分析与讨论，提出了本文实现因果知识抽取的可行方案。

第 3 章 因果知识抽取方法

本章作为本文的核心内容，将首先介绍本文所要采用的因果知识标注方案，然后回顾一些广泛使用的序列标注方法，最后详细介绍本文提出的因果知识抽取模型 Bi-LSTM-Softmax (FL)，以及其中的具体细节。

3.1 因果知识标注方案

为将因果知识抽取归约为一个序列标注问题，本文设计了一套因果知识标注方案，图 3-1 即为一个应用该标注方案后的完整标注示例：

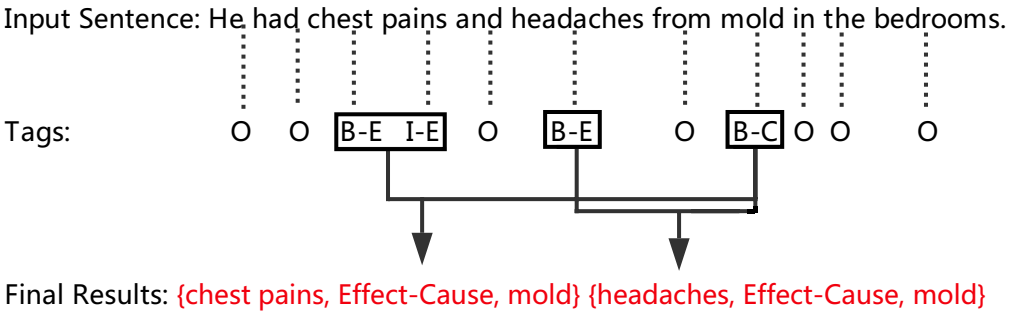

图 3-1 一个完整的标注示例

在图 3-1 中，每个单词都会被分配一个标签，用以抽取出因果三元组。标签“O”表示“$Other$”，该标签表示其所对应的单词不属于句子中的任何一个因果三元组。除了标签“O”之外，其他标签由两个部分组成：即单词在实体中的位置和关系角色。根据 Reimers 和 Gurevych[36]的建议，本文选择了“B”（$Begin$）和“I”（$Inside$）这两种记号来表示单词在实体中的位置。关系角色可由“C”（$Cause$）和“E”（$Effect$）这两种记号表示。因此，本因果知识标注方案中共有 5 类标签：“O”（$Other$）、“B-C”（$Cause\ Begin$）、“I-C”（$Cause\ Inside$）、“B-E”（$Effect\ Begin$）、“I-E”（$Effect\ Inside$）。

下面本文以图 3-1 为例，进一步说明该因果知识标注方案。

图 3-1 例句：“$He\ had\ chest\ pains\ and\ headaches\ from\ mold\ in\ the\ bedrooms.$”共包含两个因果三元组，即 $\{chest\ pains, Effect\text{-}Cause, mold\}$和

$\{headahces, Effect\text{-}Cause, mold\}$。其中“$Effect\text{-}Cause$”为预定义的关系类型（本文只关注因果关系）。单词“$chest$”、“$pains$”、“$headaches$”和“$mold$”均与最终抽取结果有关，因此，根据本文提出的因果知识标注方案，本文可以将这些单词分别标注出来。例如：单词“$pains$”是“$chest\ pains$”中的第二个单词，且由其所在因果三元组的关系类型（$Effect\text{-}Cause$）可知其关系角色是 $Effect$，所以其标签为“$I\text{-}E$”；单词“$mold$”的关系角色是Cause，故可将其标签标注为：“$B\text{-}C$”。其他与因果三元组无关的单词均可标注为“O”。

以上说明了如何将输入语句转换为其对应的可用于表达因果知识的标签序列，下面继续说明如何从标签序列转换为最终结果，也即因果三元组。

由图 3-1 中的标签序列可知，“$chest\ pains$”与“$mold$”和“$headaches$”与“$mold$”均可共享相同的关系类型“$Effect\text{-}Cause$”（即前果后因，可从关系角色相关标签得出）。因此，本文可以按照例句中的单词顺序，依次构造出两对因果三元组，也即最终结果：$\{chest\ pains, Effect\ -\ Cause, mold\}$ 和 $\{headahces, Effect\text{-}Cause, mold\}$。

需要注意的一点是：在本文中，本文仅考虑句中的单一因果关系（包括单因单果、一因多果以及多因一果三种情况），重叠关系（例如：实体 A 是实体 B 的 $Cause$，同时 A 也是实体 C 的 $Effect$）以及多因多果关系不在本文研究范围之内，本文将在这两种情况下对句中因果知识抽取的研究作为后续工作。

3.2 端到端的因果知识抽取模型

3.2.1 词嵌入模块

词嵌入（word embedding），也被称为词的分布式表示，可以从大规模无标注语料中获取词汇的语义和语法信息，近年来已经引起了研究人员的广泛关注[37]。与传统的词的独热表示（one-hot representation）相比，word embedding 是一种低维且密集的表示方式。现在，诸如 word2vec[38]和 Glove[39]这样的词向量训练工具（以及使用两种工具预训练好的词向量）已经被广泛应用于自然语言处理任务中。

如果输入语句由 n 个单词组成，即$S = \{w_1, w_2, \dots, w_n\}$，则词嵌入层会通过矩阵向量乘积的运算（相当于查表操作），将 S 中的每个单词 w_i 转换为一个实值向量 e_i：

$$e_i = W^{emb} v^i \tag{3-1}$$

其中，W^{emb} 为 embedding 矩阵，其维度为 $d \times V$，d 为词向量的维度，V为训练语料的词汇数，v^i 是维度为 V 的独热编码（one-hot vector）。输入语句转换为一实数矩阵 $embs = \{e_1, e_2, \dots, e_n\} \in \mathbb{R}^{n \times d}$后，会被输入至下一层。

为保证因果知识抽取任务的效果，根据 Reimers 和 Gurevych[36]的建议，本文选择了 Komninos 和 Manandhar[40]预训练好的词向量。此外由于字符向量[28,34]对序列标注任务结果的提升有限[36]，且使用字符向量会增加模型复杂度，增加过拟合风险，故本文未在模型中加入字符向量模块。该词向量的维度为 300，由语料大小为 20 亿词的英文维基百科语料训练而来。与传统训练词向量的方法相比，Komninos 和 Manandhar 在训练词向量时除考虑单词的上下文信息之外还考虑了文本中的依存句法信息，因而其训练的词向量不仅在句子分类任务中取得优异结果[40]，在大多数序列标注任务（词性标注、实体识别、事件触发词识别）也获得了最优表现[36]。

3.2.2 Bi-LSTM 模块

3.2.2.1 LSTM 单元

长短时记忆网络（LSTM Networks）是一种特殊的循环神经网络（RNN）模型，它克服了传统 RNN 模型由于序列过长而产生的梯度弥散和梯度爆炸问题[41, 42]。LSTM 网络模型通过特殊设计的门结构使得其可以有选择地保存上下文信息。LSTM 网络的基本构成单位是一个 memory block，其主要包括一个 memory cell（记为C）和三组具有自适应性的元素乘法门（即输入门i，遗忘门f和输出门o）。这三个门是非线性的求和单元，旨在收集 memory block 内外的信息，并且通过乘法运算控制 memory cell 的更新。图 3-2 所示为 LSTM 单元的主要结构。

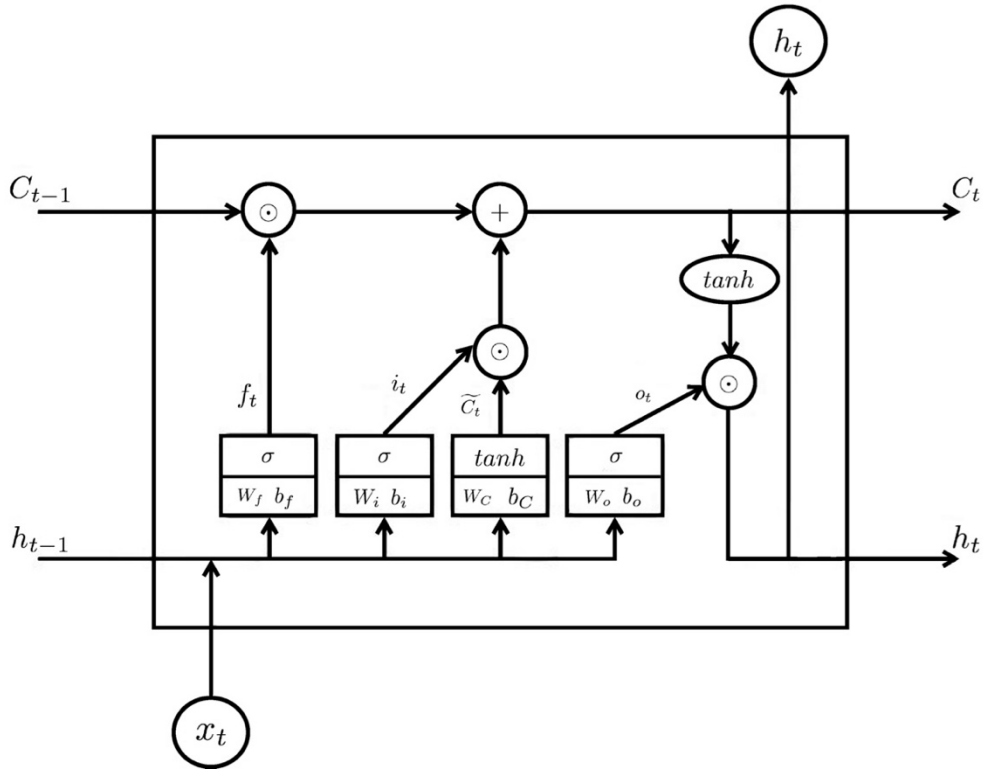

图 3-2 LSTM 单元的结构图

在时刻 t 更新一个 LSTM 单元的运算过程如下：

（1）首先，遗忘门 f_t 通过公式 (3-2) 决定有多少 $t-1$ 时刻的 memory cell 中的信息（即 C_{t-1}）可以累积到当前时刻 t 的 memory cell 中。

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{3-2}$$

（2）然后，输入门 i_t 通过公式 (3-3) 决定有多少信息（即 $\tilde{C}_t$）可以流入当前时刻 t 的 memory cell。

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{3-3}$$

（3）接下来，通过（1）（2）及公式 (3-4) 、(3-5) 的计算来更新 t 时刻 memory cell 的状态。

$$\tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \tag{3-4}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{3-5}$$

（4）最后，输出门 o_t 通过公式 (3-6) 决定在当前时刻 t，memory cell 中有多

少信息可以流入隐藏状态 h_t 中。

$$h_t = o_t \odot \tanh(C_t) \tag{3-6}$$

在以上公式中 x_t 与 h_t 分别表示时刻 t 的输入向量和隐藏状态。σ 是元素运算 sigmoid 函数，$\odot$ 代表点积符号。W_i，W_f，W_o，W_C 都是网络中的权重矩阵，b_i，b_f，b_o，b_C 则代表偏置向量。

3.2.2.2 Bi-LSTM

在很多序列标注任务中，如果模型可以在一个给定时刻获取过去和未来的信息，那么对于其性能的改进和提升将是非常有益的。但是，LSTM 单元的隐藏状态 h_t 仅能考虑 t 时刻及 t 时刻之前的信息，而无法考虑未来的信息。为有效地使用上下文信息，本文可以使双向 LSTM 网络（Bi-LSTM Networks）[43]。 Bi-LSTM 网络的基本思想就是对每一个序列分别使用一个前向 LSTM 网络和一个后向 LSTM 网络，以获得两个不同的隐藏状态：$\overrightarrow{h_t}$，$\overleftarrow{h_t}$，然后通过连接两个隐藏状态来获得时刻 t 的最终输出 $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$。

3.2.3 基于 Bi-LSTM 网络和 CRF 分类器的因果知识抽取模型

3.2.2.1 CRF

条件随机场（CRF）[44]由于其能够通过考虑邻近标签的关系来获得一个全局最优的标签序列，故被广泛应用在序列标注任务中。给定一个序列 X 及其对应的标签序列 y，CRF 可根据下式给出一个实值分数：

$$score(X, y) = \sum_{t=2}^{T} \psi(y_{t-1}, y_t) + \sum_{t=1}^{T} \phi(y_t) \tag{3-7}$$

其中 $\phi(y_t)$ 是位置 t 处标签的一元势函数（unary potential），$\psi(y_{t-1}, y_t)$ 则是位置 t 和位置 $t-1$ 处的二元势函数（pairwise potential）。在序列 X 的条件下产生标签序列 y 的概率可由以下公式给出：

$$p(y|X) = \frac{1}{Z}\exp\left(score(X,y)\right) \tag{3-8}$$

给定一个新输入 X_{new}，CRF 的目标就是为 X_{new} 找到一个使其条件概率最大的标签 y^*：

$$y^* = arg\max_{y}(\sum_{i}\log(p(y|X_{new}))) \tag{3-9}$$

算法寻找最优标签的过程被称为解码。对于以上描述的线性链 CRF，其只能对输出之间的二元关系进行建模，本文可以通过动态规划来高效地训练和解码。

3.2.2.1 Bi-LSTM-CRF

在序列标注任务中，邻近标签之间通常有很强的关联性，单纯使用 Bi-LSTM 网络的效果并不理想。为解决这个问题，Huang 等人[33]提出了 Bi-LSTM-CRF 这种模型，该模型在 Bi-LSTM 网络层输出后又增添了一层 CRF，从而明确地对输出标签之间的依赖关系进行建模，并获得全局最优的标签序列。

模型 Bi-LSTM-CRF 的结构如图 3-3 所示：

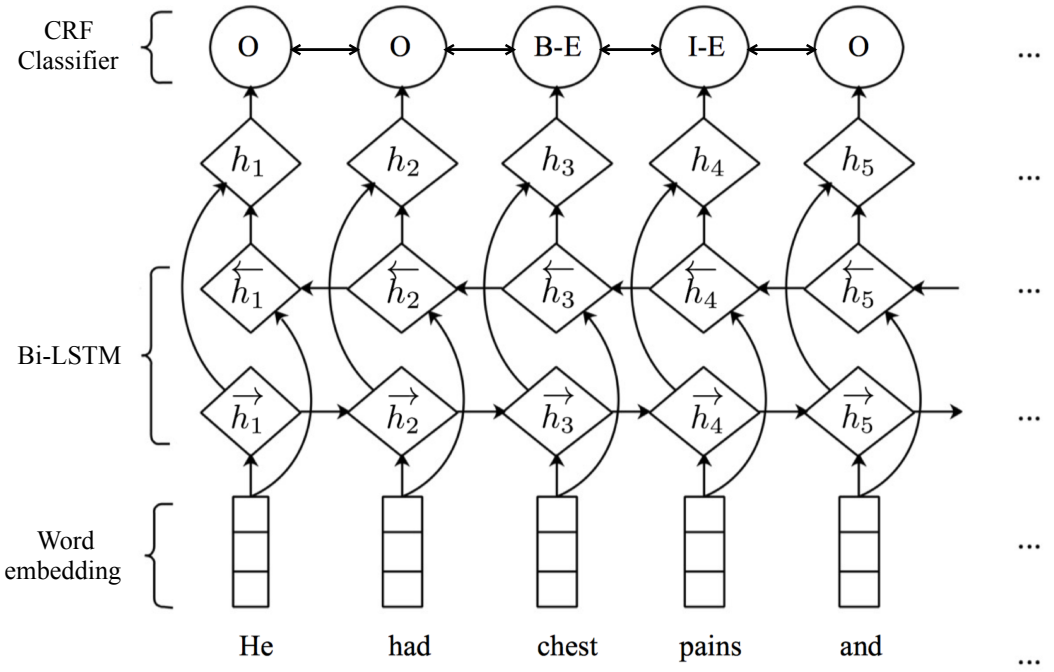

图 3-3 模型 Bi-LSTM-CRF 的结构图

对于一个给定的句子：$X=\{x_1,x_2,\ldots,x_n\}$（n 为句子长度），本文定义矩阵 P 是 X 经过词嵌入层和 Bi-LSTM 网络层后的输出结果，P_{ij} 代表句子中第 i 个单词的第 j 个标签的分数，其中 P 的维度是 $N\times k$，（N 表示单词个数，k 表示标签类别个数）。对于X的标签序列 $y=\{y_1,y_2,\ldots,y_n\}$，CRF 层可根据下式给出一个实值分数：

$$score(X,y)=\sum_{i=0}^{n}A_{y_i,y_{i+1}}+\sum_{i=1}^{n}P_{i,y_i} \tag{3-10}$$

其中，A 是转移矩阵，$A_{i,j}$ 表示由标签 i 转移到 j 的概率。y_0，y_n 则是句子起始和结束的标记，因此 A 是一个大小为 $k+2$ 的方阵。这样，在给定输入序列 X 的条件下产生标签序列 y 的概率为：

$$p(y|X)=\frac{e^{score(X,y)}}{\sum_{\tilde{y}\in Y_X}e^{score(X,\tilde{y})}} \tag{3-11}$$

本文现在可以根据下式最大化正确标签序列的对数概率：

$$\log\big(p(y|X)\big)=score(X,y)-\log\Big(\sum_{\tilde{y}\in Y_X}e^{score(X,\tilde{y})}\Big) \tag{3-12}$$

其中 Y 表示对于输入文本 X 的所有可能的标签序列，通过公式 (3-12) 本文可以得到有效合理的输出序列。预测也即解码时，由公式 (3-13) 输出整体概率最大的一组序列：

$$y^*=arg\max_{\tilde{y}\in Y_X}score(X,\tilde{y}) \tag{3-13}$$

3.2.4 基于 Bi-LSTM 网络和 Softmax 分类器的因果知识抽取模型

在基于 Bi-LSTM 网络和 Softmax 分类器的这一类模型中，对于一个输入语句：$S=\{w_1,w_2,\ldots,w_n\}$（n 为句长），模型的 embedding 层会首先将句子转换为词向量序列 $embs=\{e_1,e_2,\ldots,e_n\}$。然后，Bi-LSTM 网络层同时以正向和反向顺序读入词向量序列，并在时刻 t 输出其隐藏状态表示：h_t。最后，Softmax 层会

根据以下公式计算归一化因果标签概率：

$$s_t = W_s h_t + b_s \tag{3-14}$$

$$p_t^i = \frac{\exp(s_t^i)}{\sum_{k=1}^{K} \exp\,(s_t^k)} \tag{3-15}$$

其中，W_s 表示 Softmax 矩阵，b_s 是偏置向量，K 代表标签个数。

下面本文继续介绍三种基于 Bi-LSTM 网络和 Softamx 分类器的因果知识抽取模型，它们的区别是各自的损失函数不同。

3.2.4.1 Bi-LSTM-Softmax (CE)

对于模型 Bi-LSTM-Softmax (CE)，在其训练过程中，本文最小化分类交叉熵损失函数（categorical cross entropy，CE）。CE 的定义如下：

$$CE(Y,P) = -\sum_{j=1}^{m}\sum_{t=1}^{N_j}\sum_{k=1}^{K} y_{tk}^{(j)} \cdot \ln\,(p_{tk}^{(j)}) \tag{3-16}$$

其中 m 是批大小（batch size），N_j 是句子 S_j 的长度。当前仅当 $y_{tk}^{(j)} = 1$ 时，句子 S_j 中单词 t 的标签属于类别 k，$p_{tk}^{(j)}$ 是由公式 (3-15) 得到的模型对于标签属于类别 k 的估计概率。

3.2.4.2 Bi-LSTM-Softmax (BL)

对于模型 Bi-LSTM-Softmax (BL)，在其训练过程中，本文最小化多分类偏置损失函数（categorical biased loss，BL）。BL 是 CE 的简单扩展，Zheng 等人[21]在实体和关系的联合抽取中使用多分类偏置损失函数来提高相关实体之间的关联性，BL 的定义如下：

$$\begin{aligned} BL(Y,P) = -\sum_{j=1}^{m}\sum_{t=1}^{N_j}\sum_{k=1}^{K} & y_{tk}^{(j)} \cdot \ln\left(p_{tk}^{(j)}\right) \cdot I(O) \\ & + \alpha_{bl} \cdot y_{tk}^{(j)} \cdot \ln\left(p_{tk}^{(j)}\right) \cdot (1 - I(O)) \end{aligned} \tag{3-17}$$

其中 $I(O)$ 是一个转换函数，其定义如下：

$$I(O) = \begin{cases} 1, & tag = 'O' \\ 0, & tag \neq 'O' \end{cases} \tag{3-18}$$

$\alpha_{bl}(\alpha_{bl} \geq 1)$ 是一个可调节的偏置因子。

3.2.4.3 Bi-LSTM-Softmax (FL)

图 3-4 给出了因果知识抽取模型 Bi-LSTM-Softmax (FL) 的主要结构。在训练该模型的过程中，本文最小化多分类焦点损失函数（categorical focal loss，FL）。

Focal Loss 被用来解决单阶段目标检测场景中，在训练时出现的前景（foreground）和背景（background）类别极度失衡问题（正负例可能的比例：1:1000）[15]。本文将论文[15]中的二分类焦点损失函数扩展为多分类焦点损失函数以适应因果序列标注任务。

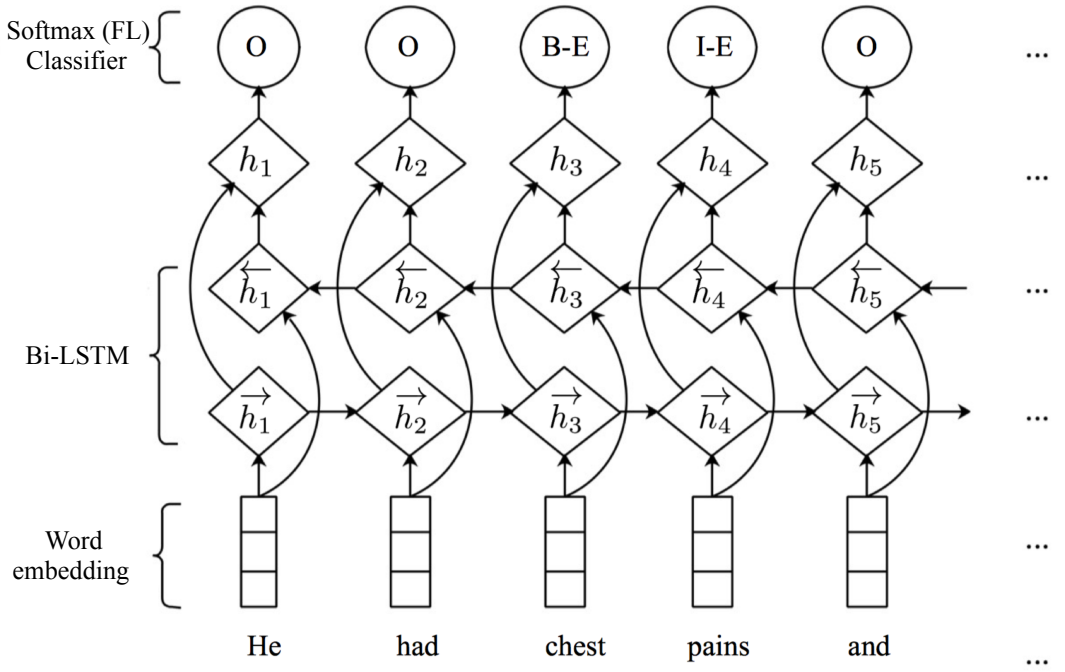

图 3-4 模型 Bi-LSTM-Softmax (FL) 的结构图。模型 Bi-LSTM-Softmax (CE) 和模型 Bi-LSTM-Softmax (BL) 的结构图与之相似（仅损失函数不同）

本文将多分类焦点损失函数定义如下：

$$FL(Y,P) = -\sum_{j=1}^{m}\sum_{t=1}^{N_j}\sum_{k=1}^{K}\alpha_{fl}\cdot\left(1-p_{tk}^{(j)}\right)^{\gamma}\cdot y_{tk}^{(j)}\cdot\ln\left(p_{tk}^{(j)}\right)\cdot(1-I(O)) \\ +\left(1-\alpha_{fl}\right)\cdot\left(1-p_{tk}^{(j)}\right)^{\gamma}\cdot y_{tk}^{(j)}\cdot\ln\left(p_{tk}^{(j)}\right)\cdot I(O) \tag{3-19}$$

其中 $\alpha_{fl}\left(\alpha_{fl}\in[0,1]\right)$ 是一个可调节的权重因子。$\left(1-p_{tk}^{(j)}\right)^{\gamma}$ 是调制因子，$\gamma(\gamma\geq 1)$ 为其中一可调节的聚焦参数。在模型 Bi-LSTM-Softmax (FL) 中，本文使用多分类焦点损失函数来解决因果序列标注中存在的类别不平衡问题及分类难度差异问题。

3.3 本章小结

本章详细描述了本文提出的因果知识抽取方法，该方法包括因果知识标注方案及端到端的因果知识抽取模型两部分。在 3.1 节中，本文结合例句详细阐述了输入语句到标签序列以及标签序列到因果三元组的转换方法，并对本文的研究范围作出了说明。在 3.2 节中，本文依次介绍了 Bi-LSTM-CRF、Bi-LSTM-Softmax (CE)、Bi-LSTM-Softmax (BL)、Bi-LSTM-Softmax (FL)这四种因果知识抽取模型，以及其中的具体细节。

第 4 章 实验设计及结果分析

为了验证本文所提出的因果知识抽取方法的有效性，本文进行了大量的实验。本章将首先介绍相关的实验设置，并报告实验结果，接下来为更好地理解本文提出的因果知识抽取模型 Bi-LSTM-Softmax (FL)，本章对模型误差及多分类焦点损失函数对结果的影响作出了分析，最后本章给出了几个利用基于 Bi-LSTM 网络的端到端模型抽取因果知识的典型案例。

4.1 实验设置

4.1.1 基准模型

本文将两种经典的流水线式因果知识抽取模型[9,12]作为实验的基准模型。Sorgente 等人[9]使用人工预定义好的规则来抽取候选因果对，然后使用贝叶斯分类器和拉普拉斯平滑来滤除 SemEval 2010 task 8[45]数据集中的非因果关系对；Luo 等人[12]提出用因果力度（Causal Strength，CS）这一将必要性因果关系与充分性因果关系结合起来的评价指标来衡量两篇短文本之间的因果关系（见 2.2 节）。为了与本文提出的模型作对比，本文向 Luo 等人[12]的方法中增添了与方法[9]相同的候选因果对抽取模块。然后，本文计算候选因果对的 CS 分数，并与预先设置好的阈值 τ 进行比较以滤除非因果关系对，即：对于 cause i_c 和 effect j_e，如果 $CS(i_c, j_e) \geq \tau$，则 (i_c, j_e) 是因果关系对，否则 (i_c, j_e) 是非因果关系对。

此外，本文还将三种典型的端到端序列标注模型：Bi-LSTM-CRF[28,33,34]，Bi-LSTM-Softmax (CE)[46,47]和 Bi-LSTM-Softmax (BL)[21]作为本文所提出模型 Bi-LSTM-Softmax (FL) 的基准模型。

4.1.2 实验数据

在本实验中，本文使用的语料库由 4600 条英文句子组成，平均句长为 20 个单词，共包含 1568 个因果三元组。训练集共包含 4000 条数据，其中 1000 条数据包含至少一个因果三元组，另外 3000 条数据则不包含任何因果三元组。 测试

集中有 600 条数据，其中 300 条数据包含至少一个因果三元组，另外 300 条数据不包含任何因果三元组。该语料库由扩展标注后的 SemEval 2010 task 8 数据集构成。本文按照文献[9]中的标注方法：若文本中存在一因多果关系或多因一果关系，除 SemEval 2010 task 8 标注者已标注出的因果对之外，本文还会将剩余的因果对都标注出来。例如：原数据集中的标注结果，

“$He\ had\ chest\ pains\ and < e1 > headaches </e1 >\ from\ < e2 > mold </e2 >\ in\ the\ bedrooms.$”，关系标签：$Cause\text{-}Effect(e_2, e_1)$，

会被扩展为“$He\ had\ < e1 > chest\ pains </e1 >\ and\ < e2 > headaches </e2 >\ from\ < e3 > mold </e3 >\ in\ the\ bedrooms.$”，关系标签：$Cause\text{-}Effect(e_3, (e_1, e_2))$。

4.1.3 评价指标

本文使用精确率（Precision，P），召回率（Recall，R）和 F1 值（F1-score, F1）作为实验的评估指标。P、R、F1 可通过以下公式计算：

$$P = \frac{\#correct\ extracted\ causal\ triplets}{\#extracted\ causal\ triplets} \tag{4-1}$$

$$R = \frac{\#correct\ extracted\ causal\ triplets}{\#total\ causal\ triplets\ in\ D} \tag{4-2}$$

$$F1 = 2\frac{P \cdot R}{P + R} \tag{4-3}$$

其中，“#”代表三元组个数，D 表示数据集中的句子集合。

本文规定：当且仅当模型预测产生的因果三元组能够精确匹配一个标注的因果三元组时，该因果三元组才被视为一个正例。在本实验中，本文通过在训练集上进行 10 折交叉验证和网格搜索来调整超参数，以取得因果知识抽取的最优结果。

4.1.4 参数设置

本文的因果知识抽取模型由词嵌入层和两层 Bi-LSTM 网络以及一个损失函数为多分类焦点损失函数的 Softmax 分类器组成。本文通过使用 Keras[48]版本

2.0.8 来实现该模型。在实验中，本文使用 Komninos 和 Manandhar 预先训练好的词向量（见 3.2 节），在模型训练过程中，为获得标注语料之外的泛化能力，及防止过拟合，本文不对词向量进行微调。本文将 LSTM 网络的隐藏层大小设置为 100。同时本文使用 dropout 率为 0.5 的 variational dropout（见 2.3 节）。本文还应用梯度归一化[42]技术并将其阈值设置为 1.0。在训练过程中，模型使用的优化方法是 Nadam[49]，学习率为 0.002。本文将 mini-batch 的大小设置为 8[50]。对于多分类偏置损失函数，本文将偏置参数 α_{bl} 设置为 2。对于多分类焦点损失函数，本文将权重因子 α_{fl} 和聚焦参数 γ 分别设置为 0.1 和 1。

4.2 实验结果

表 4-1 不同因果知识抽取模型在测试集上的结果对比

Methods	P	R	F1
Luo et al.[12]	0.6022	0.5860	0.5940
Sorgente et al.[9]	0.6894	0.5430	0.6075
Bi-LSTM-Softmax (CE)	0.8022	0.7849	0.7935
Bi-LSTM-Softmax (BL)	0.8135	**0.8091**	0.8113
Bi-LSTM-CRF	0.8260	0.8038	0.8147
Bi-LSTM-Softmax (FL)	**0.9172**	0.7446	**0.8220**

表 4-1 为使用不同模型进行因果知识抽取的结果。其中，“P”为精确率，“R”为召回率，“F1”为 F1 值；表中第一部分（前 2 行）为流水线式因果知识抽取模型的结果，第二部分（第 4 行至第 6 行）为基于 Bi-LSTM 网络的端到端因果知识抽取模型的结果。由表 4-1 可知，本文所提出的模型 Bi-LSTM-Softmax (FL) 的 F1 值显著优于其他模型，这验证了本文提出方法的有效性。此外，表 4-1 还表明直接对因果知识进行抽取的模型（即基于 Bi-LSTM 网络的端到端模型）优于传统的流水线模型。基于 Bi-LSTM 网络的端到端模型取得优越结果的原因可能在于，LSTM 网络层可以更高效地捕捉并学习到自然语言文本中因果知识的语义表

示。

此外，本文发现，模型 Bi-LSTM-CRF 的性能要优于模型 BiLSTM-Softmax (CE) 和模型 Bi-LSTM-Softmax (BL)，其 F1 值分别比二者高出了 2.67%和 0.42%。这是因为 CRF 层可以联合解码整个标签序列，不过模型 BiLSTM-CRF 的 F1 值并没有优于模型 Bi-LSTM-Softmax (FL) 从而取得最优结果（0.8147:0.8220）。

本文还发现，模型 Bi-LSTM-Softmax (BL) 通过简单地向 CE 中添加偏置权重来平衡因果标签和非因果标签对于总体损失的重要程度，仅能获得比模型 Bi-LSTM-Softmax (CE) 更高的 F1 值（提高约 2.24％），并不能在衡量模型整体性能的 F1 值上超过其他模型。

除增加一个权重因子之外，本文提出的模型 Bi-LSTM-Softmax (FL) 还在 CE 的基础上增加了一个带有可调节的聚焦参数的调制因子，以削弱易分类负例的影响（即标签“*O*”），从而聚焦于训练难分类的标签。也正因如此，本文提出的模型 Bi-LSTM-Softmax (FL) 与其他基于 Bi-LSTM 网络的模型相比，可以取得最高的 F1 值。

4.3 分析与讨论

4.3.1 误差分析

为进一步研究第 3 章所述的四种不同的基于 Bi-LSTM 网络的端到端模型的效果，本节分析了这四种模型对于 Cause、Effect 和因果对（Causal Pairs，CP）的预测结果，如表 4-2 所示。

在此本文规定：

（1）当且仅当模型预测产生的 Cause/Effect 与标注的 Cause/Effect 精确匹配时，该 Cause/Effect 实例才被视为一正例。

（2）无论因果关系的方向如何（即允许因果互换），如果因果三元组中的两个对应实体都是正确的，则该 CP 实例被视为一正例。

由表 4-2 可知，从所有三个指标来看（P、R、F1），模型在 CP 上的表现上

均低于在 Cause 和 Effect 上的表现。主要原因就是在模型预测产生的结果中，有一些预测结果相互之间不构成因果关系。也就是说，模型预测得到的一些 Cause 不能找到其真正对应的 Effect，同时也有一些预测得到的 Effect 也不能找到其正确对应的 Cause。因此，从结果来看，模型在 Cause/Effect 上的结果都比在 CP 上的结果更好。

表 4-2 基于 Bi-LSTM 网络的端到端因果知识抽取模型对于 Cause、Effect 和因果对（Causal Pairs，CP）的预测结果对比

PRF	Bi-LSTM-Softmax (CE)	Bi-LSTM-Softmax (BL)	Bi-LSTM-CRF	Bi-LSTM-Softmax (FL)
P-Cause	0.8589	0.8609	0.8720	**0.9109**
R-Cause	0.8363	0.8509	**0.8567**	0.8070
F-Cause	0.8474	0.8559	**0.8643**	0.8558
P-Effect	0.9130	0.9198	0.9304	**0.9860**
R-Effect	0.8909	**0.9030**	0.8909	0.8515
F-Effect	0.9018	0.9113	0.9102	**0.9138**
P-CP	0.8049	0.8162	0.8260	**0.9172**
R-CP	0.7876	**0.8118**	0.8038	0.7446
F-CP	0.7962	0.8140	0.8147	**0.8220**

此外，与表 4-1 中的结果相比，模型 Bi-LSTM-Softmax (CE) 和模型 Bi-LSTM-Softmax (BL) 在 CP 中的 F1 值都有所提升（F1 值分别提高 0.34％和 0.33％），而模型 Bi-LSTM-CRF 和模型 Bi-LSTM-Softmax (FL) 的 F1 值均保持不变。这说明模型 Bi-LSTM-Softmax (CE) 和模型 Bi-LSTM-Softmax (BL) 在预测因果三元组中因果关系的方向时会犯错误（即混淆关系角色标签），而模型 Bi-LSTM-CRF 和模型 Bi-LSTM-Softmax (FL) 可以很好地处理此问题。

4.3.2 多分类焦点损失函数分析

与其他基于 Bi-LSTM 网络的因果知识抽取模型不同，本文提出的模型以多分类焦点损失函数（FL）作为损失函数并取得了最高的 F1 值。该损失函数通过向原分类交叉熵损失函数（CE）中添加调制因子来解决类别不平衡问题。FL 的调制因子 $\left(1-p_{tk}^{(j)}\right)^{\gamma}$（参见第 3.4 节）可以削弱易分类标签对总体损失的影响，从而增强了那些难分类标签的重要性。

以聚焦参数 $\gamma=2$ 时为例，一个被模型以估计概率 $p_{tk}^{(j)}=0.9$ 正确分类（假设该标签的正确类别为 k）的标签的损失将会比其分类交叉熵损失小 100 倍；而一个被模型误分类的标签（同样假设该标签的正确类别为 k）至多只会被减小 4 倍（即当模型估计概率 $p_{tk}^{(j)}=0.5$ 时）。这样反而使得模型更加关心那些被误分类的标签。

表 4-3 基于 Bi-LSTM 网络的端到端因果知识抽取模型在测试集上的 RS-Cause 和 RS-Effect 结果对比

Methods	RS-Cause	RS-Effect
Bi-LSTM-Softmax (CE)	0.2231	0.3898
Bi-LSTM-Softmax (BL)	0.2285	0.4140
Bi-LSTM-CRF	0.1989	0.3844
Bi-LSTM-Softmax (FL)	**0.1559**	**0.3414**

为了进一步分析 FL 的影响，本文计算了所有基于 Bi-LSTM 网络的端到端因果知识抽取模型的 Single Cause 率（RS-Cause）和 Single Effect 率（RS-Effect），结果如表 4-3 所示。

在此，本文定义 ：

（1）Single Cause/Effect 为那些本身预测正确，但不能找到其正确对应的 Effect/Cause 的实例。

（2）RS-Cause/Effect 为 Single Cause/Effect 实例的个数占模型预测结果（即预测得到的因果三元组）总数的比重。

（3）RS-Cause/Effect 越低，则可以配对为正确因果三元组的数量就越多，反之亦然。

由表 4-3 可知，与其他基于 Bi-LSTM 网络的模型相比，模型 Bi-LSTM-Softmax (FL) 的 RS-Cause 和 RS-Effect 最低，这说明本文的模型较其他模型有效地增强了 Cause 和 Effect 之间的关联性。

4.3.3 案例研究

在表 4-4、表 4-5 和表 4-6 中，本文依次列举了三个具有代表性的例子，以显示这些基于 Bi-LSTM 网络的端到端因果知识抽取模型各自的优缺点。

表 4-4 样例 1，其例句中的 Cause 与 Effect 相对距离较远

Sentence 1	I found that the *wind(C1)* swirling around from the back, in between the front seats, caused a *draft(E1)* on the driver and passenger's necks.
True Triplets	*{wind, C-E, draft}*
Bi-LSTM-Softmax (CE)	*{wind, C-E, draft},* **{swirling, C-E, draft}**
Bi-LSTM-Softmax (BL)	**{wind swirling, C-E, draft}**
Bi-LSTM-CRF	*{wind, C-E, draft},* **{back, C-E, draft}**
Bi-LSTM-Softmax (FL)	*{wind, C-E, draft}*

对于每个样例，第一行为原始句子（Sentence）及其中包含的因果三元组（True Triplets），第 2 至 5 行展示不同模型的抽取结果。其中“C”为“Cause”的简写，“E”为“Effect”的简写。本文用蓝色斜体标示出*正确结果*，用红色粗体标示出**错误结果**。

在表 4-4 中，例句 1 内 Cause 与 Effect 的距离相对较远，显然，这会给模型

对于因果知识的学习带来困难。在本例中，本文看到只有模型 Bi-LSTM-Softmax (FL) 能够完整精确地抽取出正确的因果三元组：$\{wind, Cause\text{-}Effect, draft\}$。

图 4-5 样例 2，其例句中同时存在多个因果三元组

Sentence 2	That being said, I do love the game and play it all the time but would appreciate a little less *frustration(E1)* from *programming(C1)* and *debugging(C2)*, and sticking strictly with the frustration that comes from hitting bad shots.
True Triplets	*{frustration, E-C, programming}*, *{frustration, E-C, debugging}*
Bi-LSTM-Softmax (CE)	*{frustration, E-C, programming}*, **None**
Bi-LSTM-Softmax (BL)	*{frustration, E-C, programming}*, *{frustration, E-C, debugging}*
Bi-LSTM-CRF	*{frustration, E-C, programming}*, **None**
Bi-LSTM-Softmax (FL)	*{frustration, E-C, programming}*, *{frustration, E-C, debugging}*

在表 4-5 中，例句 2 内存在两个因果三元组：$\{frustration, Effect\text{-}Cause, programming\}$，$\{frustration, Effect\text{-}Cause, debugging\}$。在这种文本中同时存在多个因果三元组的情况下，能够完整精确地抽取出所有因果三元组对于模型也是有一定难度的。从抽取结果中本文观察到，只有模型 Bi-LSTM-Softmax (BL) 和模型 BiLSTM-Softmax (FL) 能够完全准确地抽取出例句 2 中所有正确的因果三元组。

以上两个例子都表明模型 Bi-LSTM-Softmax (FL) 较其他基于 Bi-LSTM 网络的因果知识抽取模型，可以更有效地增强 Cause 和 Effect 之间的联系，并提高因果知识抽取的性能。但是，例句 3 表明模型 Bi-LSTM-Softmax (FL) 也可能会出现致命的错误。

图 4-6 样例 3，模型 Bi-LSTM-Softmax (FL) 未能抽取出其例句中的因果三元组

Sentence 3	The constant polarization *voltage(C1)* between the two electrodes instigates the electrochemical *reaction(E1)* of the chlorine compounds on the working electrode.
True Triplets	*{voltage, C-E, reaction}*
Bi-LSTM-Softmax (CE)	*{voltage, C-E, reaction}*
Bi-LSTM-Softmax (BL)	*{voltage, C-E, reaction}*
Bi-LSTM-CRF	*{voltage, C-E, reaction}*
Bi-LSTM-Softmax (FL)	**None**

观察表 4-6 可知，模型 Bi-LSTM-Softmax (CE)、模型 BI-LSTM-Softmax (BL) 和模型 Bi-LSTM-CRF 都能够将例句 3 中的因果三元组：$\{voltage, Cause\text{-}Effect, reaction\}$ 准确无误地抽取出来；但是模型 Bi-LSTM-Softmax (FL) 却未能从例句 3 中抽取出任何一个因果三元组，并且该模型将例句 3 判定为非因果句。

显然，模型 Bi-LSTM-Softmax (FL) 并没有学习到例句 3 中的“$instigates$”一词实际上就是因果触发词。出错的原因可能是例句 3 这种类型的样本数量在本文训练的语料库中相对较少，而模型 Bi-LSTM-Softmax (FL) 与其他基于 Bi-LSTM 网络的端到端模型相比，需要相对较多一点的数据来学习这种因果知识的表达模式。

4.4 本章小结

在前一章介绍了因果知识标注方案和因果知识抽取模型以及其中涉及的诸

多细节之后，本章主要介绍了为验证本文所提出的因果知识抽取方法的有效性而进行的相关实验及结果分析。

在相关数据集上的结果显示，基于 Bi-LSTM 网络的端到端模型显著优于流水线式模型，而且本文提出的模型 Bi-LSTM-Softmax (FL) 取得了最高的 F1 值。

随后，本章对 Cause、Effect 及 Cause Pairs 等因果三元组的组成或相关部分，以及多分类焦点损失函数对结果的影响进行了详细分析，分析结果显示：模型 Bi-LSTM-Softmax (FL) 较其他模型可有效地增强 Cause 和 Effect 之间的关联性。

最后，本章对几个因果知识抽取的典型案例进行了分析。

第 5 章 因果知识抽取系统

本章介绍本文实现的一个因果知识抽取系统，包括因果知识抽取系统的体系结构、各模块功能、系统用户界面以及各功能键的作用。

5.1 因果知识抽取系统的设计

本文实现了一个英文因果知识抽取系统 Causal Knowledge Extractor，该系统提供因果序列标注以及因果分析两种功能。Causal Knowledge Extractor 主要由以下几个模块组成：文本预处理模块、序列标注模块、因果分析模块、数据存储模块。整个系统的结构如图 5-1 所示：

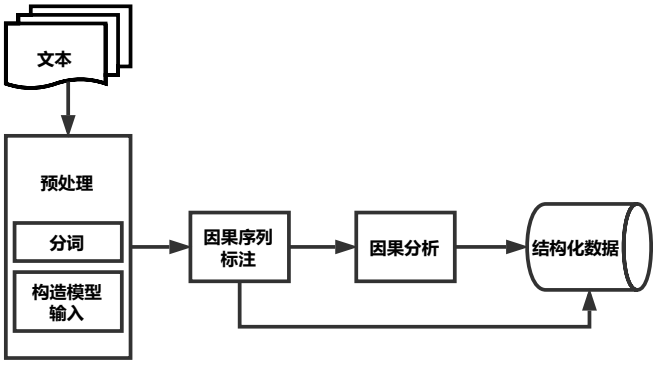

图 5-1 因果知识抽取系统结构图

5.1.1 文本预处理模块

本系统所需的文本预处理较为简单，首先对输入的英文语句进行分词，然后将单词序列替换为索引（即训练语料中单词的序号）序列，最后将索引序列（标量序列）转化为二维数组序列以便后续因果知识抽取模型的处理。数组序列的维度为 $S_{nb} \times L_{\max}$，其中：S_{nb} 为样本数目，L_{max} 为设定的最大句长。若句长 $l \leq L_{max}$，则在序列后部填充 0 以达到 L_{max}；若句长 $l > L_{max}$，则截断以使 l 匹配目标句长 L_{max}。

5.1.2 序列标注模块

序列标注模块是本系统中非常重要的一个模块，该模块主要负责标注文本中存在的因果知识，3.2 节详细介绍了基于 Bi-LSTM 网络的因果序列标注模型。

5.1.3 因果分析模块

该模块主要负责的将序列标注结果转换为因果三元组，3.1 节详细介绍了转换方法。

5.1.4 数据存储模块

无论是序列标注，还是因果分析，最终输出的结果都是结构化的数据，数据存储模块负责将这些结果以结构化文档的形式保存在本地文件中。

5.2 Causal Knowledge Extractor 系统的介绍

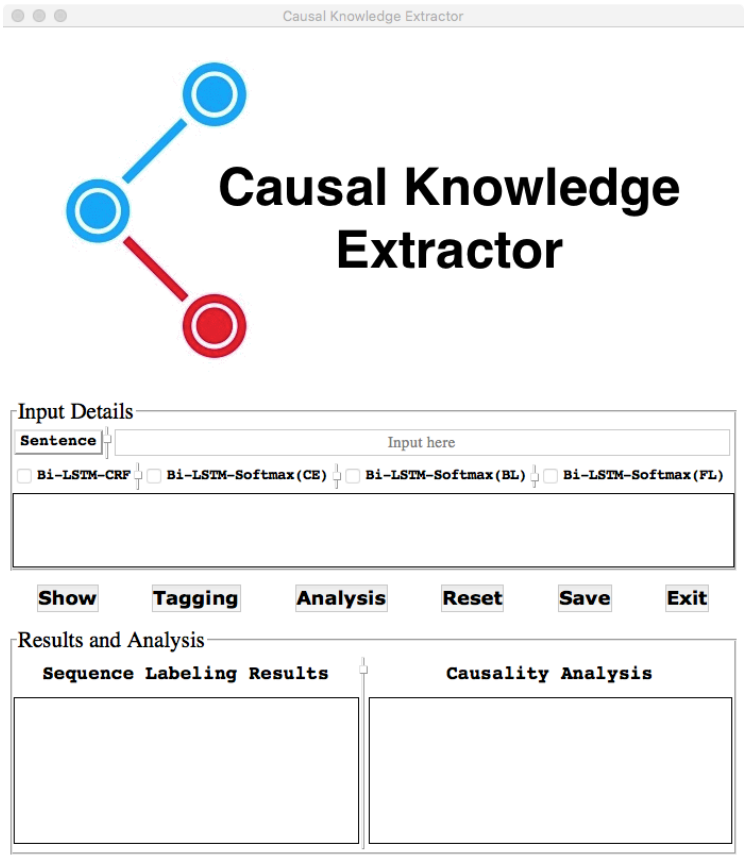

图 5-2 因果知识抽取系统 Causal Knowledge Extractor

Causal Knowledge Extractor 系统在 macOS High Sierra 操作系统下开发，采用 Python 语言作为开发语言，图 5-2 给出了系统的用户界面。

在图 5-2 中，系统界面中间一栏为工具栏（包含“Show”、“Tagging”、“Analysis”、“Reset”、“Save”、“Exit”等 6 个按钮），工具栏上方为输入文本及模型选择区域（“Input Details”），工具栏下方为显示序列标注结果和因果分析部分（“Result and Analysis”）。

5.2.1 输入

用户可以在输入框内输入有待进行因果知识抽取的句子（目前仅支持英文），并在输入框下方选择模型。在用户完成文本输入及选择模型，单击工具栏中的“Show”（显示）键后，系统就会在模型选项下方显示出输入文本及模型，以便用户修改。图 5-3 显示了用户完成输入及选择，点击“Show”键后的系统界面。

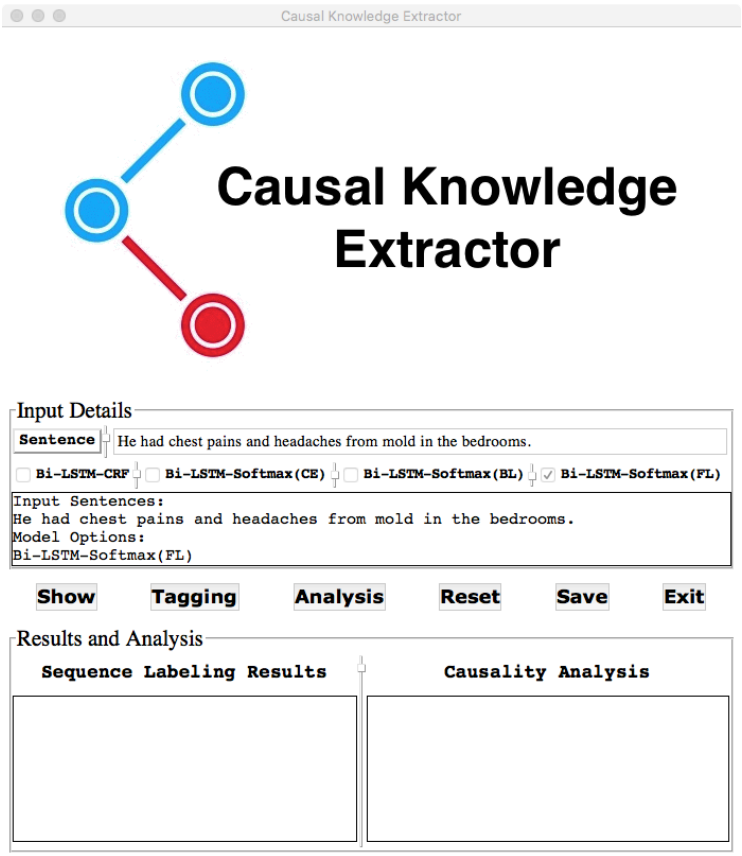

图 5-3 用户完成文本输入及模型选择后的系统界面

5.2.2 序列标注

在用户完成文本输入及模型选择后，用户单击“Tagging”（标注）按键，系统就会根据用户选择的模型调用相关因果知识抽取模型，完成对输入文本的因果序列标注，并在“Sequence Labeling Results”（序列标注结果）对应窗口显示出序列标注结果。图 5-4 和图 5-5 给出了序列标注结果显示界面。

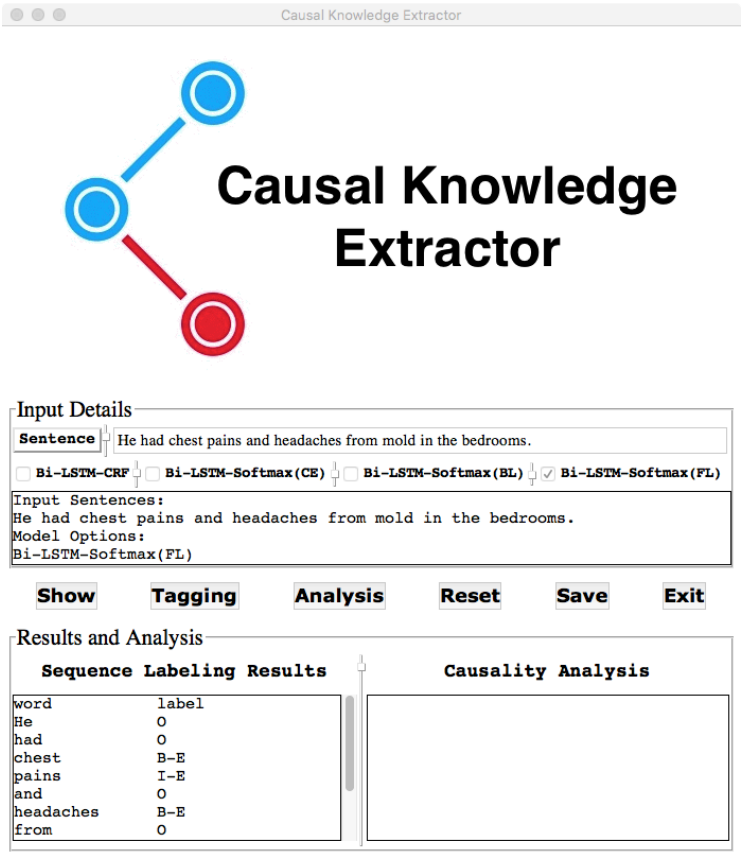

图 5-4 序列标注后的系统界面

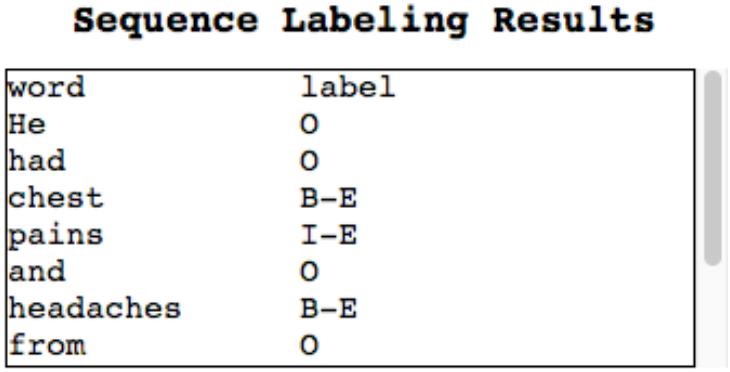

图 5-5 序列标注结果

图 5-5 显示的序列标注结果中：左侧一列为用户输入语句中的单词，右侧一

列为单词对应的标签结果，用户可通过右侧滚动条来查看完整的序列标注结果。

5.2.3 因果分析

在用户完成序列标注操作之后，可继续单击“Analysis”（分析）按键，系统就会根据因果序列标注的结果，在“Causality Analysis”（因果分析）对应的窗口中显示出因果分析结果。图 5-6 和图 5-7 给出了因果分析结果的显示。

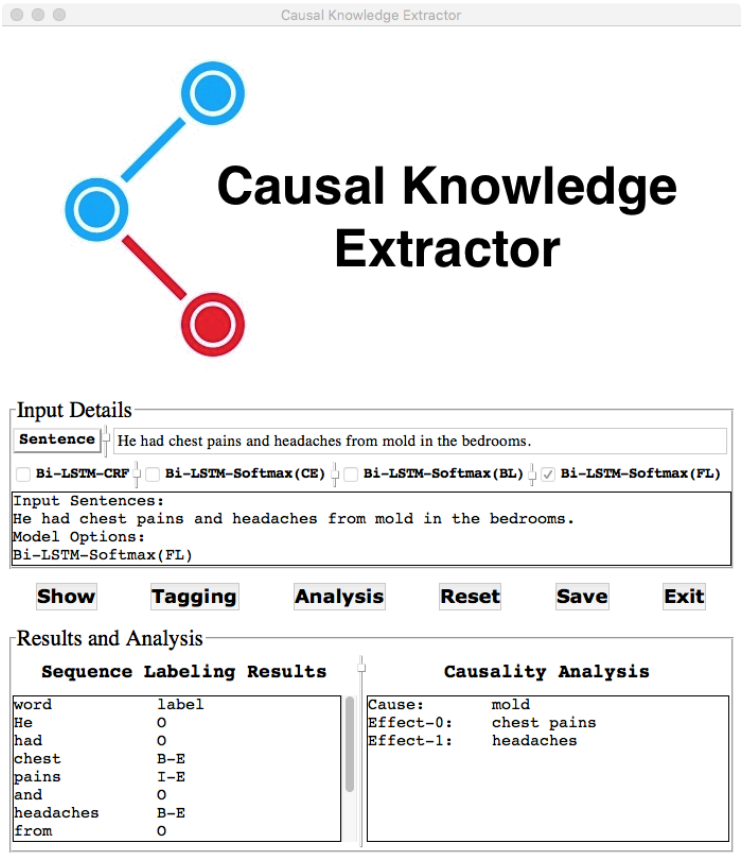

图 5-6 因果分析后的系统界面

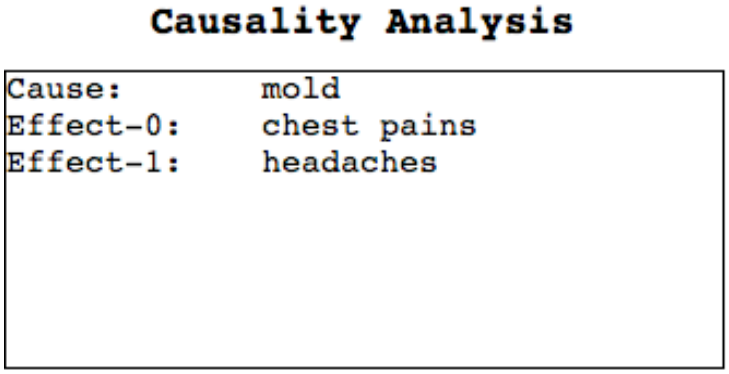

图 5-7 因果分析结果

在图 5-7 显示的因果分析结果中：“Cause”表示原因，“Effect”表示由“Cause”导致的结果，“Cause/Effect-n”表示按用户输入语句中单词顺序排列后的第 n 个 Cause/Effect。

5.2.4 保存结果

用户可单击“Reset”（重置）按键清空输入及结果，或单击“Save”（保存）按键将结果保存入本地文件中，也可单击“Exit“（退出）按键退出本系统。

5.3 本章小结

本章详细描述了本文实现的一个英文因果知识抽取系统 Causal Knowledge Extractor。本文从 Causal Knowledge Extractor 的体系结构开始介绍该系统的模块组成及功能，然后详细介绍并展示了系统用户界面和功能键作用。

第 6 章 总结与展望

6.1 研究工作总结

因果知识是一类非常重要的知识，因果知识的抽取在很多领域都是非常有价值的。自动抽取因果知识的方法主要分为基于规则的方法与基于规则和机器相结合的方法两类，然而，两种方法都有其固有的缺点和问题。

为解决传统方法中存在的问题，更高效地抽取因果知识，本文将因果知识抽取归约为一个序列标注问题，并提出了相应的因果知识标注方案。在此基础上，本文研究了多种基于 Bi-LSTM 网络的端到端模型来抽取因果知识，以达到直接抽取文本中因果知识的目的。此外，本文还针对因果序列标注中存在的因果标签与非因果标签类别不平衡问题及分类难度差异问题，提出了一种基于 Bi-LSTM 网络和多分类焦点损失函数的模型：Bi-LSTM-Softmax (FL)。对比实验表明，本文提出的模型能够有效地增强因果之间的关联性，并因此而优于基准模型。

6.2 后续工作展望

本文提出的因果知识抽取方法虽然取得了很好的实验结果，但本文仍存在一些不足之处：

（1）在本文进行的因果知识抽取中，本文仅以句子为单位，考虑了句中的单一显式因果关系（即句子中有明显的因果触发词）抽取，没有考虑到段落或篇章中的因果知识抽取，也没有考虑到文本中的重叠因果关系和多因多果关系以及隐式因果关系（即文本中没有明显的因果触发词）等关系的抽取及识别。在下一步工作中，本文将参考目前对句间隐式关系抽取的最新研究成果[51-54]，抽取文本中存在的跨句（限两个句子之间）隐式因果关系。

（2）本文所提出的基于 Bi-LSTM 网络的端到端因果知识抽取模型需要高质量的已标注好的数据来训练，而构建这种因果知识库是非常耗时耗力的。在未来的工作中，本文会根据一些最新的研究成果[55]，将本文的模型与远程监督[56]和强

化学习[57]这两种技术相结合，这样本文就不必专门构建因果知识库来训练模型，而且这样训练得到的模型也会比只针对某一语料库训练的模型更通用。

（3）现有针对中文财经领域因果知识抽取的研究比较少，且不够深入，未来本文可以将本文的因果知识抽取方法应用于中文财经文本上，以挖掘其中的因果知识（如：与财经指标有关的因果知识）。

参考文献

[1] Asghar N. Automatic Extraction of Causal Relations from Natural Language Texts: A Comprehensive Survey[J]. arXiv preprint arXiv:1605.07895, 2016.

[2] Khoo C S G, Kornfilt J, Oddy R N, et al. Automatic Extraction of Cause-Effect Information from Newspaper Text Without Knowledge-based Inferencing[J]. Literary & Linguistic Computing, 1998, 13(4):177-186.

[3] Khoo C S G, Chan S, Niu Y. Extracting Causal Knowledge from a Medical Database Using Graphical Patterns[C]// Meeting of the Association for Computational Linguistics. 2000. 336--343.

[4] Girju R, Dan I M. Text Mining for Causal Relations[C]// Fifteenth International Florida Artificial Intelligence Research Society Conference. AAAI Press, 2002. 360-364.

[5] Ittoo A, Bouma G. Extracting explicit and implicit causal relations from sparse, domain-specific texts[C]// International Conference on Natural Language Processing and Information Systems. Springer-Verlag, 2011. 52-63.

[6] 干红华, 潘云鹤. 一种基于事件的因果关系的结构分析方法[J]. 模式识别与人工智能, 2003, 16(1):000056-62.

[7] 干红华. 基于事件的因果关系可计算化分析研究[D]. 浙江大学, 2003.

[8] Girju R. Automatic Detection of Causal Relations for Question Answering[C]// The Workshop on in the 41 St Meeting of the Association for Computational Linguistics. 2003.

[9] Sorgente A, Vettigli G, Mele F. Automatic Extraction of Cause-Effect Relations in Natural Language Text[J]. DART@ AI* IA, 2013, 2013:37-48.

[10] Blanco E, Castell N, Dan M. Causal Relation Extraction[J]. Universitat Politècnica De Catalunya, 2008.

[11] Zhao S, Liu T, Zhao S, et al. Event causality extraction based on connectives analysis[J]. Neurocomputing, 2016, 173(P3):1943-1950.

[12] Luo Z, Sha Y, Zhu K Q, et al. Commonsense causal reasoning between short texts[C]// Fifteenth International Conference on Principles of Knowledge Representation and Reasoning. AAAI Press, 2016. 421-430.

[13] Sasaki S, Takase S, Inoue N, et al. Handling Multiword Expressions in Causality Estimation[C]//IWCS 2017—12th International Conference on Computational Semantics—Short papers. 2017.

[14] Li Q, Ji H. Incremental Joint Extraction of Entity Mentions and Relations[C]// Meeting of the Association for Computational Linguistics. 2014. 402-412.

[15] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[J]. arXiv preprint arXiv:1708.02002, 2017.

[16] Silverstein C, Brin S, Motwani R, et al. Scalable Techniques for Mining Causal Structures[J]. Data Mining & Knowledge Discovery, 2000, 4(2-3):163-192.

[17] Radinsky K, Davidovich S, Markovitch S. Learning causality for news events prediction[C]// International Conference on World Wide Web. ACM, 2012. 909-918.

[18] Riaz M, Girju R. Another Look at Causality: Discovering Scenario-Specific Contingency Relationships with No Supervision[C]// IEEE Fourth International Conference on Semantic Computing. IEEE Computer Society, 2010. 361-368.

[19] Hashimoto C, Torisawa K, Kloetzer J, et al. Toward Future Scenario Generation: Extracting Event Causality Exploiting Semantic Relation, Context, and Association Features[C]// The, Meeting of the Association for Computational Linguistics. 1977.

[20] Ackerman E J M. Extracting a Causal Network of News Topics[C]// Otm Confederated International Conferences "on the Move To Meaningful Internet Systems. Springer Berlin Heidelberg, 2012. 33-42.

[21] Zheng S, Wang F, Bao H, et al. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017. 1: 1227-1236.

[22] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural

networks[C]//Advances in neural information processing systems. 2014. 3104-3112.

[23] Cheng J, Dong L, Lapata M. Long Short-Term Memory-Networks for Machine Reading[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016. 551-561.

[24] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997, 9(8):1735-1780.

[25] De Marneffe M C, MacCartney B, Manning C D. Generating typed dependency parses from phrase structure parses[C]//Proceedings of LREC. 2006, 6(2006): 449-454.

[26] 付剑锋, 刘宗田, 刘炜,等. 基于层叠条件随机场的事件因果关系抽取[J]. 模式识别与人工智能, 2011, 24(4):567-573.

[27] 钟军, 禹龙, 田生伟,等. 基于双层模型的维吾尔语突发事件因果关系抽取[J]. 自动化学报, 2014, 40(4):771-779.

[28] Ma X, Hovy E. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016, 1:1064-1074.

[29] Søgaard A, Goldberg Y. Deep multi-task learning with low level tasks supervised at lower layers[C]// Meeting of the Association for Computational Linguistics. 2016. 231-235.

[30] Schuster M, Paliwal K K. Bidirectional recurrent neural networks[M]. IEEE Press, 1997.

[31] Gal Y, Ghahramani Z. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks[J]. Statistics, 2015. 285-290.

[32] 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):págs. 212-223.

[33] Huang Z, Xu W, Yu K. Bidirectional LSTM-CRF Models for Sequence Tagging[J]. Computer Science, 2015.

[34] Lample G, Ballesteros M, Subramanian S, et al. Neural Architectures for Named

Entity Recognition[C]//Proceedings of NAACL-HLT. 2016. 260-270.

[35] Lecun Y, Boser B, Denker J S, et al. Backpropagation Applied to Handwritten Zip Code Recognition[J]. Neural Computation, 1989, 1(4):541-551.

[36] Reimers N, Gurevych I. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks[J]. arXiv preprint arXiv:1707.06799, 2017.

[37] Lai S, Liu K, He S, et al. How to Generate a Good Word Embedding[J]. IEEE Intelligent Systems, 2016, 31(6):5-14.

[38] Mikolov T, Sutskever I, Chen K, et al. Distributed representations of words and phrases and their compositionality[C]//Advances in neural information processing systems. 2013. 3111-3119.

[39] Pennington J, Socher R, Manning C. Glove: Global Vectors for Word Representation[C]// Conference on Empirical Methods in Natural Language Processing. 2014. 1532-1543.

[40] Komninos A, Manandhar S. Dependency Based Embeddings for Sentence Classification Tasks[C]// Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016. 1490-1500.

[41] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult[J]. IEEE Transactions on Neural Networks, 2002, 5(2):157-166.

[42] Pascanu R, Mikolov T, Bengio Y. On the difficulty of training recurrent neural networks[C]//International Conference on Machine Learning. 2013. 1310-1318.

[43] Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures[J]. Neural Networks, 2005, 18(5-6): 602-610.

[44] Lafferty J D, Mccallum A, Pereira F C N. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data[C]// Eighteenth International Conference on Machine Learning. Morgan Kaufmann Publishers Inc. 2001. 282-289.

[45] Hendrickx I, Su N K, Kozareva Z, et al. SemEval-2010 task 8: multi-way classification of semantic relations between pairs of nominals[C]// The Workshop

on Semantic Evaluations: Recent Achievements and Future Directions. Association for Computational Linguistics, 2009. 94-99.

[46] Wang P, Qian Y, Soong F K, et al. Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Recurrent Neural Network[J]. Computer Science, 2015.

[47] Vaswani A, Bisk Y, Sagae K, et al. Supertagging with lstms[C]//Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016: 232-237.

[48] F. Chollet, et al., Keras[EB/OL].[2015]. https://github.com/fchollet/keras, 2015.

[49] Dozat T. Incorporating nesterov momentum into adam[C]// ICLR Workshop. 2016. (1):2013-2016.

[50] Reimers N, Gurevych I. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging[C]//Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017. 338-348.

[51] Qin L, Zhang Z, Zhao H. A stacking gated neural architecture for implicit discourse relation classification[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016: 2263-2270.

[52] Qin L, Zhang Z, Zhao H, et al. Adversarial Connective-exploiting Networks for Implicit Discourse Relation Classification[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017, 1: 1006-1017.

[53] Lei W, Xiang Y, Wang Y, et al. Linguistic Properties Matter for Implicit Discourse Relation Recognition: Combining Semantic Interaction, Topic Continuity and Attribution[J]. 2018.

[54] Wang Y, Li S, Yang J, et al. Tag-Enhanced Tree-Structured Neural Networks for Implicit Discourse Relation Classification[J]. arXiv preprint arXiv:1803.01165, 2018.

[55] Feng J, Huang M, Zhao L, et al. Reinforcement Learning for Relation Classification from Noisy Data[C]// Association for the Advancement of Artificial Intelligence, 2018.

[56] Mintz, Mike, Steven, et al. Distant supervision for relation extraction without labeled data[C]// Joint Conference of the, Meeting of the ACL and the, International Joint Conference on Natural Language Processing of the Afnlp: Volume. Association for Computational Linguistics, 2009. 1003-1011.

[57] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. Cambridge: MIT press, 1998.

攻读硕士学位期间的研究成果

[1] Zhaoning Li（李肇宁）, Jiangtao Ren. Causality Extraction based on Bi-directional LSTM Networks with Focal Loss. Knowledge-Based Systems.（Under Review）（与硕士学位论文第三章相关）

[2] 基于深度学习的因果知识抽取系统 [简称：因果知识抽取系统] V1.0，登记号：2018SR275268，证书号：软著登字第 2604363 号.（与硕士学位论文第五章相关）

致谢

两年紧张而又充实的研究生生活即将画上句号，我也即将跨出校门，告别承载自己六年青春的母校。在此，我要向所有曾经给予我帮助和关怀的人致以最诚挚的谢意。

感谢我的导师任江涛副教授。任老师是我在自然语言处理领域的启蒙老师，在我研究生学习期间，任老师对我遇到的困难和疑惑总是能够给予悉心指导，本文从选题到完成，每一步都倾注了老师大量的心血和精力。

感谢我的父母，他们在背后的默默支持是我前进的动力。

感谢我的朋友、同学，我将带着你们的鼓励与信任，保持一颗好奇心，努力前行。