硕 士 学 位 论 文

基于深度学习的事件及其时序关系抽取的研究

Deep Learning Based Extraction of Events and Temporal Relations

作 者 姓 名: 黄梦佐

学科、 专业: 计算机科学与技术

学 号: 21809170

指 导 教 师: 李丽双 教授

完 成 日 期: 2021.5.1

大连理工大学

Dalian University of Technology

摘　要

作为信息抽取的重要组成部分，事件抽取旨在从大量的非结构化的自然语言文本中提取事件；事件时序关系识别则为事件抽取的下游任务，它致力于识别提取出来的事件之间的时序关系，挖掘事件之间的事理逻辑。

事件包含事件触发词与事件要素两部分，触发词标识了事件的发生与类型，要素则为事件的主要参与者，通用的事件抽取方法一般首先识别事件的触发词，然后识别特定于各个事件的要素。而通用的事件时序关系识别方法一般只考虑事件的触发词，识别文本中两个触发词之间的时序关系。

在事件抽取任务上，本文针对现有的方法对句子信息编码不充分的问题，提出了一种层次蒸馏网络（Hierarchical Distillation Network, HDN）模型来获取语义丰富的句子表示，再分别通过序列标注和成对分类的手段来预测句子中的事件。具体地说，HDN 融合了一个循环神经网络（Recurrent Neural Network, RNN）模块用于提取句子的序列信息，若干个图卷积神经网络（Graph Convolutional Network, GCN）模块用于提取句子的多阶句法信息，同时还基于双向注意力机制设计了一个蒸馏模块来减少层级编码之间的信息冗余。在实验阶段，本文分别在两个常用事件抽取数据集：ACE 2005 和 MLEE 上测试了模型的性能，实验结果表明，提出的模型较之前最好的模型在事件抽取的结果上均获得了显著提升。

在事件的时序关系抽取任务上，本文针对现有方法存在的（1）事件对的特征构造方式单一（2）整数线性规划（Integer Linear Programming, ILP）与神经网络的训练过程割裂等问题，提出了一种逻辑驱动的深度对比网络（Logic-driven Deep Comparison Network, LDCN）模型来进行时序关系的识别。LDCN 不仅使用了表征能力强的 HDN 作为句子编码器，还分别从可交换特征和不可交换特征的角度来增强事件对之间的特征表示，最后引入了可微的逻辑驱动训练框架，使得模型在训练过程中可以充分考虑时序关系之间的逻辑一致性。在实验阶段，本文分别在两个常用的事件时序识别数据集：TimeBank-Dense 和 MATRES 上测试了模型的性能，实验结果表明，提出的模型较之前最好的模型在时序关系抽取的结果上均获得了显著提升。

关键词：事件抽取；事件时序关系抽取；层次蒸馏网络；逻辑驱动深度对比网络

Deep Learning Based Extraction of Events and Temporal Relations

Abstract

As important parts of information extraction, event extraction aims to extract events from unstructured natural language texts; temporal relation extraction is dedicated to identifying the relationships between extracted events and mining the logic between them.

Event consists of trigger and argument. Trigger indicates the occurrence and the type of event, while argument is the main participant of the event. In general, event extraction methods first identify trigger and then identify the specific arguments. Accordingly, general event temporal relation extraction methods only consider the triggers and identify the temporal relation between them.

For event extraction, this thesis proposes a Hierarchical Distillation Network (HDN) model to obtain the sentence representation with rich semantics, and then predicts the events by means of sequence labeling and pairwise category respectively. Specifically, HDN integrates a recurrent neural network (RNN) module to extract sequential information of sentences, and several graph convolutional network (GCN) modules to extract multi-order syntactic information. Moreover, a distillation module is designed based on bidirectional attention mechanism to reduce information redundancy between encodings. In the experiments, this thesis tests the performance of the model on two commonly used event extraction datasets: ACE 2005 and MLEE, the experimental results show that the proposed model has achieved significant improvement in the result of event extraction than the previous best.

For event temporal relation extraction, a Logic-driven Deep Comparison Network (LDCN) model is proposed to solve the existing problems of (1) the feature construction of event pairs is too simple and (2) the integer linear programming (ILP) is separate from training process. LDCN not only employs the HDN as the encoder, but also enhances the features of events pair from the perspective of the commutative and the non-commutative features, and finally introduces the logic-driven differentiable training framework, enables the model to consider the logical consistency in training. In the experiments, this thesis tests the performance of the model on two commonly used event temporal relation extraction datasets: TimeBank-Dense and MATRES. The experimental results also show that the proposed model has achieved significant improvement in the extraction results of event temporal relation than the previous best.

Key Words：Event Extraction; Temporal Relation Extraction; Hierarchical Distillation Network; Logic-driven Deep Comparison Network

目　　录

1 绪论

本章首先介绍了事件抽取与事件时序关系识别两个任务各自的背景与意义；其次梳理了两个任务相关的研究现状；然后阐述了本文的主要研究内容；最后概括了本文的组织结构。

1.1 研究背景和意义

互联网技术自诞生以来就给人们的日常生活带来了极大的便利，一方面，每个人都可以方便地从互联网上获取各式各样的信息；另一方面，每个人也都可以成为信息的提供者，在互联网上分发各式各样的信息。但是，长此以往便造成了如今信息爆炸甚至过载的局面，日益增长的海量信息数据不仅给服务商的数据处理过程带来了难题，也使得人们难以有效地筛选符合自己预期的数据内容。在如此背景条件下，信息抽取（Information Extraction, IE）技术便应运而生了，信息抽取旨在运用自然语言处理（Natural Language Processing, NLP）的方法，从大量的非结构化的自然语言文本中提取有价值的信息结构化表示，从而方便信息在数据库中的结构化存储，提高数据的检索效率。作为信息抽取的一个重要子问题，事件及其时序关系抽取旨在以事件为最小单位解析自然语言而描述事物状态的变化规律，它在政务舆情监控[1-3]、金融风险分析[4-5]、生物分子通路模型管理[6]和生物医学数据库开发[7]等方面都具有积极促进的作用。

如自动内容抽取[8]（Automatic Content Extraction, ACE）中所描述的，事件的发生往往具有特定的时间、特定的地点，存在一个或者多个的参与者，并且通常被描述为一种状态变化的过程。事件抽取致力于从大量的非结构化的自然语言文本中提取事件，挖掘文本中的事件触发词、事件触发词与实体之间的细粒度关系。近年来，领域内为了推动事件的自动化抽取研究，许多事件抽取相关的标注数据以及共享任务被提出。其中，ACE 为最具有影响力的一个，自 2005 年起，ACE 中便囊括了事件的抽取任务并且标注了相当部分的语料供后来者研究。在之后举办的知识图谱构建[9]（Knowledge Base Population, KBP）评测中同样将事件抽取任务作为赛道任务之一。除此之外，一些特定领域的事件抽取任务评测同样吸引了许多研究者参与。比如在生物医学领域中，研究者一直对生物分子间的相互作用而导致的变化比较感兴趣，因此也尝试利用信息抽取的方法自动化提取生物医学文献中的事件表示。BioNLP-ST[10]为其中一项比较知名的赛事，它经东京大学文本挖掘中心发起，并且提供了一系列的标注数据集以及解析工具供研究者使用。为了对事件本身的结构特性有一个清晰的认知，在图 1.1 中标注了一个 ACE 2005 语料库中的事件实例，该句子中包含了一个 *Execute* 事件，该事件由触发词 *“executed”* 触

发，并且与三个命名实体构成了要素角色关系，实体 *“blasphemy”* 指明了该 *Execute* 事件的罪行，实体 *“blasphemy convict”* 指明了参与该事件的人物，实体 *“the country”* 则指明了该事件发生的地点。因此，一般而言的事件抽取任务即为，在给定命名实体标注的情况下，分别识别出句子中的事件触发词、触发词与实体之间潜在的要素角色关系。

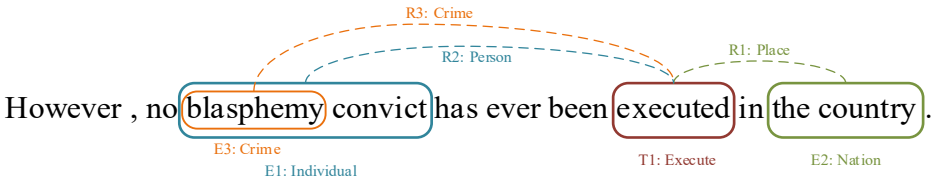

图 1.1　事件标注举例。E 为 entity 的简写；T 为 trigger 的简写；R 为 Role 的简写

Fig 1.1　The example of event annotation. ‘E’ means entity, ‘T’ means trigger and ‘R’ means role

在实体关系抽取中，研究者把实体作为信息载体的最小单位，研究实体之间存在的关系。然而这样得到的知识图谱一般只包含了静态的知识，因为，实体并不能描述事物变化的过程。而如果以事件为最小单位理解文本，研究事件与事件之间存在的关系，那么就可以在图谱中存储动态的知识。以图 1.2 为例，这里面包含了三个 *Action* 事件，并且具有显而易见的时间先后关系。 *“intended”* 事件发生在 *“rob”* 事件之前， *“rob”* 事件发生在 *“firing”* 事件之后， *“intended”* 事件发生在 *“firing”* 事件之前。这三个事件构成了一个完整的时序事件链，描述了该抢劫犯罪活动从策划到执行的完整过程。所以，研究事件之间存在的关系，可以方便在知识库中存储动态的知识，以便对事物演变、发展的规律进行进一步研究。

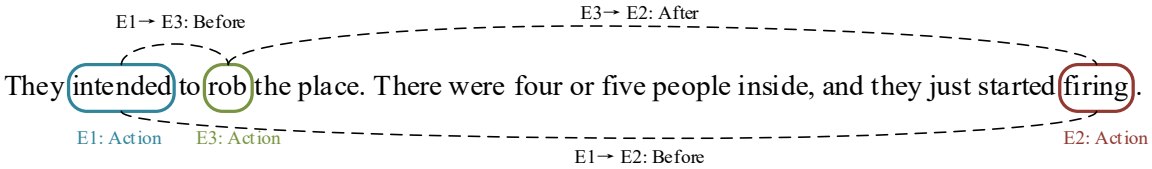

图 1.2　事件关系标注举例。E 为 event 的简写

Fig 1.2　The example of event relation annotation. ‘E’ means event

事件的关系识别任务是一个建立在事件抽取任务之上的下游任务，近年来，有越来越多的研究者投入到事件关系识别的研究当中，尤其表现在对事件时序关系的研究。在著名的词典特别兴趣组（Special Interest Group on the Lexicon，SIGLEX）发起的语义评价国际研讨会（International Workshop on Semantic Evaluation，SemEval）中，TempEval[11] 一直是重要的评测任务之一并且吸引了许多学者的参与。时至今日，信息抽取社区正逐渐将重心从传统的以实体为核心的知识库构建，转移到以事件等复杂结构知识为核心的

知识库构建，知识图谱也逐渐朝着事理图谱的方向演变。这些转变不仅有利于学术界内对一些相关下游任务，如文本摘要、阅读理解任务等的进一步研究，也对工业界内的一些实际难题，如图谱中搜索最优解面临的组合爆炸问题等都具有极大的推动作用。

1.2 研究现状和相关工作

1.2.1 事件抽取

事件抽取的研究过程可以分为三个阶段，第一个阶段为基于模式匹配（Pattern Matching）的方法；第二个阶段为基于浅层机器学习（Machine Learning）的方法；第三个阶段为基于深度学习（Deep Learning）的方法。

（1）基于模式匹配的方法

早期的事件抽取方法，由于受到计算资源和标注语料的影响，大多以基于模式匹配的方法为主。模式匹配首先需要靠人为地总结或半自动构造一些特定的事件模板，然后再在文本上进行模板的匹配，从而抽取出事件。比较具有代表性的工作为 Riloff[12] 等提出的 AutoSlog 系统。AutoSlog 从少部分人工标注的语料中总结了一些语言学模板，比如一般主谓结构、动宾结构、及物动词的被动结构等，再通过这些语言学模板和词性（Part-Of-Speech，POS）识别结果结合来提取未标注句子中的事件。在生物医学领域事件抽取中，Cohen[13] 等在 BioNLP-ST 中同样使用了基于模式匹配的方法，他们首先基于评测给定的训练集和母语使用者的语感直观对每种预定义的事件类型构建模板，然后合并现有的生物医学本体库，最后使用 OpenDMAP[14] 解析器对句子进行事件抽取。也有一些基于模板种子的半自动模板构建方法，如 Riloff[15] 、Yangarber[16] 等人的工作，但是基于模式匹配的方法都存在精确度高，召回率低且可移植性较差的特点。因此，随着标注语料规模的增大和机器学习方法的兴起，基于模式匹配的方法也逐渐被基于机器学习的方法所取代。

（2）基于浅层机器学习的方法

和基于模式匹配的方法使用槽填充（Slot Filling）方式不同，基于浅层机器学习的方法往往一般不能一次性得到完整的事件结构，它将事件抽取分为两个步骤：首先识别句子中的触发词，然后识别每个触发词对应的要素角色。事件抽取中一些常用的浅层机器学习方法包括有最大熵模型（Maximum Entropy，ME）、支持向量机（Support Vector Machine, SVM）、隐马尔可夫模型（Hidden Markov Model，HMM）等。Ahn[17] 从词法、上下文词法、依存解析等角度构建机器学习特征，使用最近邻算法（Nearest Neighbor）检测句子中的触发词，然后用最大熵模型来检测句子中的要素。Ji[18] 训练了若干个最大

熵模型分别识别句内的触发词和要素，除此之外，他还考虑了文档级别的全局特征，将这些全局证据与局部句内的事件检测结果相结合进行推理，提高了事件抽取系统的整体性能。Liao[19] 在 Ji[18] 的基础上扩展了全局信息，并应用了文档中跨事件的信息。Hong[20] 提出了一种基于 SVM 分类器的方法，并且在特征构造过程中考虑了实体类型和要素角色类型之间的潜在联系。Liao[21] 则是在事件抽取中首次引入了主题模型，他们首先计算每个文档的主题分布，然后将这些主题特征和其他基于词法的特征一起送入分类器中进行判断。以上机器学习方法都是将触发词识别阶段和要素识别阶段完全隔离，所以也称为流水线方法。除了流水线方法之外，还有部分学者尝试将触发词识别过程和要素识别过程结合起来，这些结合起来的方法统称为联合方法。Li[22] 将事件抽取看作是一个结构化预测的问题，使用结构化感知机（Structured Perceptron）和集束搜索（Beam Search）算法同时预测句子中的触发词和要素。McClosky[23] 将生物医学事件看作树形结构，并使用依存解析的方法直接在句子中预测生物医学事件。

（3）基于深度学习的方法

尽管基于浅层机器学习的方法相较于基于模式匹配的方法已经极大地减少了对专家领域知识的依赖，但它仍然需要经过复杂的特征工程阶段，并且在特征工程中受到现有解析工具性能的限制，容易在模型中引入不可逆的级联错误。近年来，随着计算机处理能力的提升，基于深度学习的方法在事件抽取任务中逐渐成为主流，深度学习模型可以不依赖手工的特征输入，借助反向传播算法让模型在优化的过程中自动地学习高维特征。Nguyen[24] 首先将卷积神经网络（Convolutional Neural Network，CNN）运用到事件的触发词抽取中，先分别从单词、位置和实体类型的角度对句子进行嵌入操作，然后用 CNN 进行编码后分类。Chen[25] 在传统 CNN 的基础上进行改进，加入了动态多池化层对句子进行分段编码后聚合，以保留更多关键的信息。Nguyen[26] 利用循环神经网络（Recurrent Neural Network, RNN）进行触发词和要素的抽取，并且通过维持一个记忆矩阵的方式建立触发词和要素之间的条件概率关系，来实现触发词和要素的联合抽取。Feng[27] 提出了一种混合的神经网络方法，分别在模型中加入了 RNN 模块以学习句子的全局信息，CNN 模块以学习上下文窗口内的局部信息。Sha[28] 为了在句子编码过程中考虑句法信息，在长短期记忆网络（Long Short Term Memory, LSTM）中加入句法树中的边形成特殊的依存桥，相较于传统的 LSTM，桥接方式使得每个单词可见的上下文范围更广从而更容易捕捉有效信息。不同于 Sha[28] 的依存桥方式，Nguyen[29] 选择在依存树上建立图模型，然后利用图卷积神经网络（Graph Convolutional Network，GCN）对句子进行编码以更有效地捕捉远距离的有用信息，最后对句子中的触发词进行抽取。Liu[30] 对普通的 GCN 进

行改进，将依存边进行嵌入并和模型一起优化，值得注意的是，不同于之前的工作中把触发词识别看作为一个句子级的分类任务，作者将触发词识别看作为一个序列标注问题以便能够一次性识别出句子中的多个事件。Yan[31] 使用了 GCN 模型的一个变种，图注意力模型（Graph Attention Network , GAT）进行触发词抽取，GAT 在卷积运算之后加入了注意力机制（Attention Mechanism）以实现对邻接结点特征的动态加权融合。Cui[32] 提出强化边信息表示的 GCN 模型用于识别触发词，设计了一个边感知的节点更新模块，通过特定的依赖类型聚合邻接结点特征来生成更有表征力的结点特征，还引入了一个节点感知的边嵌入更新模块，利用结点信息来迭代更新边的嵌入特征。Wang[33] 针对要素识别任务提出使用高层实体概念指导要素角色分类的方法，每个高层实体概念对应一个可训练的嵌入向量，然后通过注意力机制聚合高层实体概念嵌入和句子编码后的特征表示，再对聚合之后的特征表示进行分类。潘璋[34] 针对触发词抽取提出一种识别注意力模型，将触发词识别分为识别和判定类别两步；针对要素抽取提出一种卷积长短期注意力与触发词注意力融合模型，利用事件触发词的信息来增强要素的权重信息。

在生物医学领域的事件抽取中，深度学习的方法同样得到了广泛的应用。Wang[35] 在生物医学触发词抽取的任务使用了基于 CNN 的方法，同时还引入了词性（part-of-speech, POS）、位置和实体类别的特征来补充词嵌入的信息。秦美越[36] 改进了一般的 CNN 模型加入了并行多池化的机制，分段地对触发词和要素分割的文本序列池化，最终获取更为丰富的文本信息。李虹磊[37] 改进了一般的双向 BI-LSTM 模型，在其基础上使用注意力机制对文本的信息进行进一步整合，在编码句子之后，引入注意力机制后将得到句子的表示进行分类。Rahul[38] 使用基于 RNN 的方法进行触发词的抽取，首先利用 RNN 在序列层面上对给定文本的词嵌入序列进行编码，最终利用这些得到的编码进行序列标注。刘阳[39] 使用自注意力的双向门控循环网络（Gated Recurrent Unit，GRU）抽取生物医学事件的触发词，并使用多注意力的 BI-GRU 抽取生物医学要素。Li[40] 提出一种上下文标签敏感的 GRU 模型用于触发词抽取，该模型在编码过程中动态地结合了初始的词嵌入向量和门控循环单元输出的隐层向量，并且在序列标注解码过程中引入了自回归依赖。Zhang[41] 考虑到文档潜在主题对触发词抽取的好处，提出了一种联合模型来同时提取生物医学事件触发词和文档潜在主题。

1.2.2 事件时序关系识别

事件时序关系识别方法的发展过程和事件抽取类似，也经过了从基于模式匹配的方法，到基于浅层机器学习的方法，最后到基于深度学习的方法的历程。

（1）基于模式匹配的方法

在早期关于时序关系识别的研究中，同样受到了标注数据较少的制约。而这一问题的根本原因在于领域内对于时间信息的界定没有一个统一的标准，即没有统一的语料标注准则。1983 年，Allen[42] 在他的工作中定义了 13 种时序关系类型，这 13 种关系类型能够表示事件间的每一种时序关系。此后，便有基于模式匹配的方法在时序关系识别任务中被陆续提出。比较有代表性的有，Dowty[43] 利用叙述性约定（Narrative Convention）来判断事件间时序关系，其中叙述性约定指的是后面句子中的事件一定发生在之前句子的事件之后；Song[44] 从时态的可解释性出发，提出了一系列的启发式规则和约束条件用于捕捉连贯的时态序列，并将这些规则组织成算法以确定句子中情景间的时间顺序。尽管这一类算法在小规模语料上都取得了不错的效果，但是同样面临着泛化性较差的特点，并且对专家领域知识的依赖严重。随着 2003 年 TimeML[45] 标准的制定，许多大规模的时序关系语料被陆续提出，这也拉开了时序关系识别任务中机器学习方法的序幕。

（2）基于浅层机器学习的方法

事件时序关系识别中常用的浅层机器学习方法有多项式逻辑回归（Multinomial Logistic Regression）、ME 和 SVM 等。Mani[46] 使用了最大熵模型算法进行时序关系的识别，并选取语料库中的每个时序链接（Temporal Link, TLINK）和其类型作为特征向量。Chambers[47] 针对事件时序关系的识别提出了两阶段的方法，第一个阶段粗略地提取事件的一些时间属性，第二阶段使用一阶段得到的特征和朴素贝叶斯（Naive Bayes）算法完成时序关系识别。Bethard[48] 提出了 ClearTK 系统，该系统完成了时序信息抽取的整个流水线过程，包括时间表达式提取、事件的提取、时序关系的识别，其中用于时序关系识别的方法为最大熵模型，构建的特征包括有事件的种类、模态、极性和时态等特征。Laokulrat[49] 提出基于逻辑回归的时序关系识别模型，除了常用的文本特征之外，还将短语结构树中事件词之间的路径及路径长度、谓词参数结构及其子图中事件词之间的路径作为特征。Chambers[50] 分析了现有工作中对全局事件时序关系研究不充分的问题，提出了一种级联结构，该级联结构保证了当前的决策分类器可以对之前决策分类器的分类结果进行考虑，一定程度上解决了全局推理中需要满足的传递性约束问题。Ning[51] 在分析完时序关系与因果关系之间的潜在联系之后，提出一种基于结构化学习的联合框架对时序和因果关系联合识别，并且使用整数线性规划（Integral Linear Programming，ILP）的方法进行最大后验推断。而随着深度学习的兴起，浅层机器学习在特征工程上的弊端逐渐显露，深度学习方法成为了当前的主流方法。

（3）基于深度学习的方法

由于深度学习在自然语言处理的其他领域都陆续取得了巨大的成功，研究者们也开始考虑如何将深度学习的方法迁移到时序关系识别的任务中，尤其是从实体关系抽取的工作中借鉴经验。作为第一次的尝试，Cheng[52] 在他们的模型中引入了 BI-LSTM 的结构，并且分别从句子序列和事件结点间的依存序列两个角度进行编码，最后将两种编码聚合后进行分类。Meng[53] 提出了一种上下文感知的神经网络模型用于时序关系识别，它由四个组件组成，用于句内实体关系的 LSTM 模型，用于句间关系的 LSTM 模型，用于文档创建时间的另一个 LSTM 模型，以及用于时间表达式（Time Expression，TIMEX）对的基于规则的组件。Ning[54] 通过实验研究了多种神经网络技术对于时序关系识别基线模型的影响，如词嵌入、预训练语言模型等，最后他们还在基线模型中引入了来自于时间常识知识库（Temporal Common Sense Knowledge Base）中的先验知识用于辅助分类。Han[55] 提出一种 LSTM 和结构化支持向量机（Structured Support Vector Machine，SSVM）结合的方式，LSTM 用于学习一个关系分类器的打分函数，SSVM 则取代了传统的 ILP 过程用于全局的最大后验推断。在之后的工作中，Han[56] 也将他们这种神经网络模型与 SSVM 结合的思想用于了事件和事件时序关系的联合抽取当中。戴倩雯[57] 分别对几种兴起的关键技术进行了探索，包括 GCN 在时序识别中的运用、如何融合篇章修辞信息、以及多任务学习的训练框架的运用等。Cheng[58] 提出了一个以事件为中心的模型，允许跨多个 TLINK 管理动态的事件表示，并且以多任务学习的方式处理三个 TLINK 类别的识别，以利用完整的数据集信息。Han[59] 借鉴了预训练通用语言模型的思路，将其迁移至时序关系识别这一子领域，提出了 DEER 语言模型，该语言模型经过在大规模时序关系语料上的预训练，专注于句子中的事件时间关系，并且实验中发现在低资源的设置下比现有的通用语言模型表现更好。Ma[60] 提出一个完整的用于时序任务的 EventPlus 系统，该系统在时序关系识别阶段运用了预训练语言模型的嵌入、BI-LSTM 编码与多层感知机分类的方式来识别句子中的时序关系。

1.3 本文研究内容

在现有的事件抽取系统中，往往存在对句子编码不充分的问题。比如，目前最先进的一批事件抽取模型中都不约而同地使用了 GCN 作为 RNN 在非序列编码上的补充。然而，GCN 存在着过度平滑（Over Smoothing）的问题，多个 GCN 层的累加有的时候并不能带来性能的提升，并且在训练过程中各个 GCN 层都趋向于输出相同的特征。针对这一问题，本研究提出了一种层次蒸馏模型（Hierarchical Distillation Network, HDN）用于句子的编码。HDN 融合了一个 BI-GRU 模块用于提取句子的序列信息，若干个 GCN 模块用于提取句子的多阶句法信息，同时还基于双向注意力机制设计了一个蒸馏模块来

减少层级编码之间的信息冗余，从而获取语义丰富的句子表示。在触发词识别阶段，本研究先利用 HDN 对句子进行编码，再用序列标注（Sequence Labeling）的方式识别句子中的多个触发词；而在要素角色识别阶段中，本研究则将该任务视为成对分类（Pairwise Category）的问题，HDN 也同样被用于了句子编码，不同的是在词嵌入过程中加入了相对位置信息嵌入以标识当前的正在被分类的触发词和要素。事件抽取的具体方法实现与实验过程将在本文的第三章详细阐述。

在现有的事件时序关系识别系统中，主要存在以下两个主要的问题：（1）事件对的特征构造方式单一（2）ILP 过程与神经网络训练过程相割裂。大多先进的神经网络方法在构造事件对特征时都只简单采用了张量拼接的方式，而忽略了其他一些张量操作带来的潜在收益。针对这一问题，本研究从张量操作的可交换性和不可交换性两个角度出发，构建了多个维度的事件对特征用于最后的关系识别分类。其次，ILP 在时序关系识别任务中是非常常见的一种进行全局最大后验推断的手段，然而它却和神经网络的训练过程割裂。针对这一问题，受 Li[61] 和 Li[62] 等工作的启发，本研究在模型训练过程中引入了可微的逻辑驱动框架，使得模型在训练过程中可以充分考虑全局时序关系之间的逻辑一致性。综上所述，本研究提出一种逻辑驱动的深度对比网络（Logic-driven Deep Comparison Network, LDCN）来进行时序关系的识别以改进现有的模型。有关于具体地如何运用 LDCN 于事件时序关系识别任务，以及实验过程与分析等，都将在本文第四章得到详细阐述。

1.4 本文组织结构

第 1 章 绪论。本章主要介绍了事件抽取任务与事件时序关系识别任务的研究背景、研究意义与研究现状。除此之外还概括性地阐述了本文的主要研究内容与本文的组织架构。

第 2 章 相关背景知识与技术。本章详细地阐述了事件抽取与事件时序关系识别两个任务的定义与相关概念，还介绍了本研究使用的相关技术，最后介绍了本研究所使用的语料。

第 3 章 基于层次蒸馏网络的事件抽取。本章详细介绍了 HDN 的整个模型架构、如何将 HDN 用于事件的触发词抽取与要素角色识别、以及在两个公开的事件抽取语料库上的实验与分析。

第 4 章 基于逻辑驱动深度对比网络的事件时序关系识别。本章详细介绍了 LDCN 的整个模型架构、如何将 LDCN 用于事件时序关系的识别、以及在两个公开的事件时序关系抽取语料库上的实验与分析。

第 5 章 结论。本章进一步地总结了本研究的主要贡献，展望了未来有关事件抽取与事件时序关系识别任务的可能改进方案。

2 相关背景知识与技术

本章首先介绍了事件抽取与事件时序关系识别两个任务的相关概念；然后介绍了本研究使用的相关技术；最后介绍了本研究所使用的语料与评价标准。

2.1 事件抽取

2.1.1 事件的相关概念

事件是用于描述事物状态变化的一种结构化表示，它回答了事物状态变化发生的“5W1H”问题[1]。ACE[8]中对事件的相关概念作了如下定义：

(1) 事件提及（Event Mention）：事件提及是指描述事件发生和状态变化的一个短语，包含了一个触发词和若干个要素。

(2) 事件触发词（Event Trigger）：事件触发词是指，在一个事件中最能够描述状态变化发生的一个或多个单词，常常为动词或者名词。

(3) 事件要素（Event Argument）：事件的要素是指，在一个事件中，变化过程的参与者或属性，通常可以是实体提及（Entity Mention）、时间表达式（Time Expression）等。值得注意的是，在生物医学事件中，事件本身也可以作为另一个事件的要素参与到变化过程中。

(4) 要素角色（Argument Role）：要素角色是指，要素在事件中所扮演的角色类型。

表 2.1 MARRY 事件的事件模板举例

Tab 2.1 Schema example of MARRY event

要素角色类型	实体类型	要素角色释义	举例
Person-Arg	PER	The people who are married	[**ames**] recruited her as an informant in 1983, then married [**her**] two years later.
Time-Arg	TIME	When the marriage takes place	ames recruited her as an informant in 1983, then married her [**two years later**].
Place-Arg	GPE LOC FAC	Where the marriage takes place	We were married in [**Spain**]

[1] “5W1H”问题分别指的是“Who”、“When”、“Where”、“What”、“Why”和“How”

在封闭域（Closed Domain）下的事件抽取任务中，事件的模板往往被预先定义，即每个事件类型对应的要素类型都是确定的。表格 2.1 为 ACE 中摘取的 *MARRY* 事件的事件模板定义，可以看到，*MARRY* 事件有一个参与者槽位 *Person-Arg* 和两个属性槽位 *Time-Arg* 和 *Place-Arg*，并且每个槽位都给定了对应的实体类型或时间表达式类型。在一个事件提及中，触发词是必须被包含的，如表中的三个例句中，都包含了 *“married”* 这一触发词；但是要素却不一定被全部包含，并且不考虑模板定义之外的要素，比如表中的第一个例句中缺少了 *Place-Arg* 要素，第三个例句缺少了 *Person-Arg* 和 *Time-Arg* 两个要素。事件抽取的目标即为完成模板填充这一“填空题”，将一个自然语言形式的事件提及转化为结构化形式的事件表示。

2.1.2 事件抽取的流程

目前主流的事件抽取系统中，都遵循了 Ahn[17] 的多阶段抽取流程，它可分为以下几个主要流程：

(1) 触发词识别（Trigger Identification）：触发词识别旨在识别事件提及中的触发词。
(2) 触发词分类（Trigger Classification）：触发词分类旨在对识别出来的触发词进行分类。
(3) 要素识别（Argument Identification）：要素识别旨在判断识别出来的触发词与给定的实体或时间表达式之间是否存在要素关系。
(4) 要素分类（Argument Classification）：要素分类则是在要素识别的基础上判断要素在事件中扮演的角色类型。

根据候选类别选择的不同，触发词识别和触发词分类也可以合并为触发词抽取的过程，要素识别和要素分类也可以合并为要素角色识别的过程。触发词抽取阶段，根据抽取建模方式的不同，可以分为序列标注的方法和候选词分类的方法，而在近些年的研究中，由于序列标注一次性能够抽取多个触发词的特点，候选词分类的方法已经比较少见。要素角色识别阶段，则是早在机器学习方法中就盛行的成对分类方式一直沿用至今。

2.2 事件时序关系识别

2.2.1 事件时序关系相关概念

事件时序关系识别任务为事件抽取任务的一个下游任务，即在事件抽取的基础上，去判断一个文档中事件与事件或事件与时间表达式之间的各种关系。如上一节中所述，ACE[8] 中所定义的事件为一种结构化的表示，而在 TimeML[45] 中，对于事件的定义却

与前者定义有一定差别。TimeML 中的事件类似于 ACE 事件结构中的触发词，如下为 TimeML 中对一些相关概念的具体定义：

(1) 事件（EVENT）：事件为标识状况发生或出现的短语，一般可分为时态动词、名词性词、形容词、表语从句和介词短语等。

(2) 时间表达式（TIMEX3）：时间表达式用于标识句中的时间元素，包括有显式的时间表达、隐式的时间表达和持续的时间表达等。

(3) 链接（LINK）：链接标识元素与元素之间的关系，按功能可分为 TLINK、SLINK 和 ALINK 三种。TLINK 用于标识事件与事件之间、事件与时间表达式之间的关系，SLINK 用于标识事件之间或事件与时间信号之间的从属关系，ALINK 用于标识方面事件与其参数事件之间的关系。在时序关系识别任务中，我们只关注 TLINK 所标识的关系。

2.2.2 事件时序关系识别的流程

相较于事件抽取任务，事件时序关系识别任务的流程要简单许多。给定一个句子和句子中的事件标记与时间表达式标记，事件时序关系识别任务旨在识别出事件与事件之间或者事件与时间表达式之间存在的各种关系。这和实体关系抽取任务非常类似，在现有的方法中，也都使用成对分类的方式对其进行识别，即对事件、事件对或事件、时间对进行特征获取，再将得到的特征送入分类器中。

2.3 相关技术介绍

2.3.1 循环神经网络

在自然语言处理领域中，RNN 因为其能够处理变长输入序列的特点而被广泛地运用。RNN 在时间维度上对输入的序列信号进行建模，它具有如下的计算形式：

$$h_t = f(W_x x_t + W_h h_{t-1} + b) \tag{2.1}$$

其中，W 和 b 为模型待优化的参数矩阵和偏置，f 为非线性激活函数，x_t 为当前时间步模型的输入，h_{t-1} 为上一个时间步的输出。从计算表达式中，我们可以看出，当前模型的输出取决于两个因素，一个为上一个时刻的模型输出，另一个为当前时刻模型的输入，即 RNN 模型满足马尔可夫性质（Markov Property）。通过不断地迭代递归输入每个时间步的信息，RNN 模型则可以建立起序列的长期依赖关系，所以相较于 CNN 模型只关注局部窗口的信息，RNN 模型在处理自然语言中的文本信息时具有优势。

但是，随着序列的越来越长，反向传播中对 RNN 计算一次偏导数所需要的连乘操作也相应的增加，这便会造成梯度的弥散或者爆炸问题。为了缓解这一问题，也有若干改进 RNN 的方法被提出。比较有代表性的就是 LSTM[63] 模型和 GRU[64] 模型。

LSTM 通过维护一个长期记忆向量的方式，模拟人类学习过程中的遗忘、更新的过程。具体而言，LSTM 的计算形式如下：

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{2.2}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2.3}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{2.4}$$

$$\widetilde{c_t} = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{2.5}$$

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c_t} \tag{2.6}$$

$$h_t = o_t * \tanh(c_t) \tag{2.7}$$

其中，W 和 b 为模型待优化的参数矩阵和偏置，σ 和 tanh 为激活函数，x_t 为当前时间步模型的输入，h_{t-1} 为上一个时间步的输出，f_t 和 i_t 分别对应遗忘门和输入门，用来维护更新长期记忆向量 c_t，o_t 为输出门，用来对最后的输出向量进行更新。

GRU 则在 LSTM 的基础之上进行改进，保留了原有的门控的设计，但是却不再维护长期记忆向量。它的具体计算形式如下所述：

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{2.8}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{2.9}$$

$$\widetilde{h_t} = \tanh(W_h \cdot [r_t * h_{t-1}, x_t] + b_h) \tag{2.10}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t} \tag{2.11}$$

其中，W 和 b 为模型待优化的参数矩阵和偏置，σ 和 tanh 为激活函数，x_t 为当前时间步模型的输入，h_{t-1} 为上一个时间步的输出，z_t 和 r_t 用于重新整合输出的向量。

2.3.2 注意力机制

注意力机制最早也是基于模拟人类认知过程而提出的一种方法，人类在对外界事物做出观测时，往往倾向于选择性地注意到被观测事物的某些部分。放在神经网络模型中，注意力机制则是一种帮助模型挑选关键信息的一种手段，如在自然语言处理领域中，最早的注意力模型被用来在机器翻译过程选择最相关的词语[65] 。而后，Vaswani[66] 更是将注意力机制总结为了一种基于键值对匹配的信息整合过程，并提出完全基于注意力机

制的 Transformer 模型用于对句子进行编码。给定一个查询矩阵Q、一个键矩阵K和一个值矩阵V，注意力机制的运算可形式化为：

$$\text{Attention}(Q,K,V)=\text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{2.12}$$

其中，矩阵Q与矩阵K的乘法用于衡量两个矩阵内两两向量之间的相似性，softmax 函数用于对相似性进行归一化以得到注意力得分。多头注意力机制（Multi-head Attention）建立在注意力机制上，它的计算形式为：

$$\begin{aligned}\text{MultiHead}(Q,K,V)&=\text{Concat}(\text{head}_1,...,\text{head}_h)W^O\\ \text{where head}_i&=\text{Attention}(QW_i^Q,KW_i^K,VW_i^V)\end{aligned} \tag{2.13}$$

多头注意力首先通过多个线性变化将特征映射到多个头上，然后对每个头的特征使用注意力机制，最后利用拼接的方式将每个头的特征拼接在一起。

2.3.3 图卷积神经网络

早期的神经网络模型如 RNN、CNN 等，在处理欧几里德结构的数据输入时都能够取得不错的结果，但是却无法应对非欧几里德结构的情形。图就是一种典型的非欧几里德结构数据，如果我们要对图的数据利用神经网络进行特征提取，并不能直接将 RNN 或者 CNN 运用到图的场景下。最早，研究者为了解决图上的特征化问题，往往选择路径采样的方式将图的问题变为序列的问题，代表性的工作有 Deepwalk[67] 等，但是这类方法任务无法很好地学习图的拓扑结构。针对这个难题，研究者将传统的 CNN 在非欧氏空间上推广，而诞生了 GCN 模型。

GCN 是一种能够在图的结构上执行特征抽取的神经网络模型，并且能够和一些已有的神经网络模型叠加协同作用，以下以自然语言处理领域为例阐述 GCN 这一模型的运用。自然语言文本在一般情况下是呈现序列结构的，但是在某些特定的情形下，我们也可以把自然语言文本当成一种具有图结构的数据。如文本解析后得到的依存句法树就是一个图结构，我们可以通过在依存句法树上使用 GCN 模型来捕捉文本中的句法信息。GCN 可以诱导一个结点与其所有邻接结点发生信息交互，并且具有如下基本形式：

$$H^{(l+1)}=f(D^{-0.5}\tilde{A}D^{-0.5},H^{(l)}) \tag{2.14}$$

其中$\tilde{A}$表示图结构对应的邻接矩阵与单位阵的和，它用于聚合上一层 GCN 中的结点特征，比较常见的聚合方式有平均聚合、注意力聚合等；$D^{-0.5}(\cdot)D^{-0.5}$为对邻接矩阵的对称归一化，用于调节特征聚合时的尺度；$H^{(l)}$表示第l层图上结点上的特征；f为一个参数化的非线性函数，用于给图卷积层赋予可训练的参数。GCN 随着层数的不断累加，一个

结点可以与距离更远的结点进行信息交互，如一层 GCN 仅可以让信息在相邻的结点之间流动，而两层 GCN 则可以让信息在间隔小于等于 2 的结点之间流动，使用基于依存句法树构建 GCN 模型，则模型可分别学习到文本中的一阶句法信息和二阶句法信息。

此外，GCN 也并不是一个完全没有缺点的模型。首先的一个缺点是，在自然语言处理领域中，依存解析句法树或是一些其它靠现成的解析器解析出来的图结构，它们都会累计在解析器中产生的误差，并且这种误差对于 GCN 来说是不可逆的。然后，GCN 模型自身也存在着过度平滑的问题，所谓的过度平滑指的是，随着 GCN 层的叠加，分类的效果反而下降，并且多个 GCN 层倾向于输出类似的特征表示，即多个层之间的特征表示存在冗余。所以在本研究中，我们解决第一个问题的方法就是从消除特征表示间的冗余这一点出发的。

2.3.4 预训练的语言模型

预训练语言模型的发展也具有一段不短的历史，可以大概地描述为从基于特征（feature-based）的方法到基于微调（finetuning-based）的方法的一个发展历程。本小节沿着这一思路简要的介绍几种对自然语言处理领域影响较大的预训练语言模型。

首先，早期的预训练语言模型注重于对一个固定词表中的每一个单词预先训练一个特征向量，这些特征向量也常被称为词向量或词嵌入。如 word2vec[68] 利用连续词袋（CBOW）或跳跃元（Skip-gram）语言模型来训练词向量，然后这些词向量可以不加区分地运用到下游的任务当中。但是，这种基于特征的方法也存在着明显的缺点，一个单词仅对应一个词向量，所以无法解决在自然语言中常见的一词多义的问题。针对这一问题，研究者提出了 ELMo[69]，ELMo 通过两个不同方向的因果语言模型（Causal Language Model）来完成预训练任务，并且在下游任务中使用 ELMo 时，需要使待处理的文本经过这两个因果语言模型才能得到最终可使用的词向量。这种处理方式使得每个词向量都是上下文相关的，从而可以考虑到一词多义的情况。

ELMo 尽管解决了一词多义的问题，但它的使用方式本质上与之前的 word2vec 没有区别，即没有使用语言模型自身的已经训练好的参数。GPT[70] 的提出推动了从基于特征的方法到基于微调方法的演变。GPT 的预训练任务是训练一个单向的因果语言模型，和 ELMo 不同的是，它使用了在长距离依赖问题上更有优势的 Transformer 取代 LSTM。并且，在预训练模型的使用阶段，GPT 是直接叠加到下游任务模型的上层一起训练，如此训练方式就可以在下游任务中使用到 Transformer 中的训练好的参数。GPT 中由于要保留模型自身的生成能力，即可以胜任文本生成的任务，所以在预训练任务中保留了对因果语言模型的训练，而 BERT[71] 的提出则抛弃了因果语言模型而使用了一种遮蔽语言模型（Masked Language Model）来完成预训练任务。

宏观来看，BERT 是已有预训练语言模型的集大成者，它综合了 ELMo 中双向和 GPT 中使用 Transformer、采取微调的优势。BERT 在预训练过程中构筑了两个任务，来使得模型在大规模语料上完成预先学习。一个是对遮蔽语言模型的训练，在训练时对句子中的部分单词进行遮蔽操作，然后要求模型在训练时预测被遮蔽的单词，这样做的好处在于预测遮蔽单词时可以同时考虑左边和右边的上下文信息。另一个预训练任务则是预测一个句子是否为另一个句子的下文，这么做的出发点为下游任务中存在一些句子对级别的任务考虑，如自然语言推断、文本相似度计算等。BERT 的运用和 GPT 一样，采取了基于微调的方式，即预先训练好的 Transformer 块与下游任务模型的参数一起继续优化。如今 BERT 俨然已成为 NLP 领域中的一个基本范式。

2.3.5 整数线性规划与逻辑驱动训练框架

传统神经网络的训练过程都依赖于网络中各个算子的可微性，所以当需要为神经网络模型额外添加离散形式的条件约束时，大部分现有工作都借助于现成的解决机（Off-the-shelf Solver）后处理神经网络模型的预测结果。如在时序关系识别任务中，按照对称一致性的常识，互逆的事件对作为模型输入应该输出互逆的时序关系，而按照传递一致性的常识，满足传递关系的三元事件输入也应该输出满足传递关系的时序关系，所以这些对称一致性常识与传递一致性常识都能作为线性规划解决机的约束条件来改善模型的预测结果。

在时序关系识别任务中，大部分现有工作是通过整数线性规划（Integer Linear Programming，ILP）的方式来后处理模型输出结果以满足上述常识性约束。通常，ILP 的一般形式如下：

$$\begin{aligned} \min \quad & c^T x \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0 \\ & x \in Z^n \end{aligned} \tag{2.15}$$

ILP 的目标函数（Objective Function）通常为线性形式，并且所有的变量都为大于等于 0 的整数。在时序关系识别任务中，目标函数的具体形式如下：

$$\text{objective}(x) = \sum_i \sum_c x_{i,c} * \log \text{prob}_{i,c} \tag{2.16}$$

其中，$\log \text{prob}_{i,c}$ 为模型最终输出的第 i 个样例在 c 类别上对应的概率分布对数值，i 下标意味着对整个数据集的样例进行遍历，c 下标意味着对所有关系类别进行遍历。时序关系识别任务中，具体要满足的约束条件有三类，独热（One-hot）约束，对称一致性约束和传递一致性约束，它们分别有以下形式：

$$\begin{aligned} &\sum_{c} x_{i,c} = 1 \\ &x_{i,c} = x_{\bar{i},\bar{c}} \\ &x_{i,p} + x_{j,q} - x_{k,r} \le 1 \end{aligned} \tag{2.17}$$

其中，在第二个约束中，$\bar{i}$ 意味着为 i 样例的逆，$\bar{c}$ 意味着关系 c 的逆；在第三个约束中，i, j, k 代表满足传递关系的三个样例，而 p, q, r 代表满足时序关系传递规则的三种时序关系。

尽管 ILP 能够一定程度上改善神经网络模型最终输出的结果，但是它与基于梯度下降算法的神经网络模型的训练过程相互割裂，使得神经网络模型在训练时对这些约束是不可见的。为了解决这一问题，Li[62] 开创性地提出了基于逻辑驱动的神经网络训练框架，它通过三角范数（T-norms）将一阶逻辑的知识转换成了可微的概率计算形式，并且作为损失函数加入到神经网络模型当中。

表 2.2　T-norms 下离散一阶逻辑到可微概率计算的映射

Tab 2.2　Mapping from discrete first-order logic to differentiable functions under T-norms

名称	一阶逻辑	Product T-norm	Gödel T-norm	Łukasiewicz T-norm
否定	$\neg A$	$1-a$	$1-a$	$1-a$
合取	$A \wedge B$	ab	$\min(a,b)$	$\max(0, a+b-1)$
析取	$A \vee B$	$a+b-ab$	$\max(a,b)$	$\min(1, a+b)$
蕴含	$A \to B$	$\min(1, \frac{b}{a})$	$\begin{cases} 1, & \text{if } b \ge a, \\ b & \text{else} \end{cases}$	$\min(1, 1-a+1)$

表 2.2 中列举出了一阶逻辑形式在 T-norms 下到概率计算公式的映射，表中的小写字母都代表模型最终输出的概率值。可以看到，通过 T-norms 的映射，一阶逻辑都具有其特定的可微计算形式，所以我们可以很自然地将这些概率计算公式作为正则项加入到神经网络模型的损失函数当中。这样做的好处在于神经网络训练时，它可以充分地考虑来源于一阶逻辑形式的常识性知识，而使得模型本身具有更好的内在逻辑一致性。

2.4 语料与评价标准

2.4.1 事件抽取语料与评价标准

本研究在事件抽取任务上分别使用了 ACE 2005[2]语料库与 MLEE[72] 语料库作为评价基准语料库。

（1） ACE 2005 语料库

ACE 定义了 8 种事件类型与 33 种事件的子类型，每一种子类型都有自己的事件结构，要素角色类型共有 36 种。表 2.3 中记录了所有 ACE 中定义的事件类型与子事件类型。

表 2.3 ACE 定义的事件类型与其子事件类型

Tab 2.3 Event type and subevent type in ACE

事件类型	事件子类型
Life	Be-Born, Marry, Divorce, Injure, Die
Movement	Transport
Contact	Meet, Phone-write
Conflict	Attack, Demonstrate
Business	Merge-org, Declare-bankruptcy, Start-org, End-position
Transaction	Transfer-money, Transfer-ownership
Personnel	Elect, Start-position, End-position, Nominate
Justice	Arrest-jail, Execute, Pardon, Release-parole, Fine, Convict, Charge-indict, Trial-hearing, Acquit, Sentence, Sue, Extradite, Appeal

表 2.4 ACE 2005 统计指标

Tab 2.4 The statistics of ACE 2005

	文档数	句子数
训练集	529	15577
开发集	30	891
测试集	40	729

ACE 2005 中一共标注了 599 个文档，为了公平的比较各个模型的性能，现有的工作大多都遵循了 Li[22] 中的数据分割方式，将 599 个文档分为：529 个文档用于训练集（Training set），30 个文档用于开发集（Development set），剩余 40 个文档用于测试集

[2] https://catalog.ldc.upenn.edu/LDC2006T06

（Test set）。不同的预处理方法所使用的分句方式也存在差异，本研究使用了 Wang[33] 的预处理程序解析原始语料，预处理过后的句子统计指标如表 2.4 中所示。

（2） MLEE 语料库

MLEE 中定义了 4 种生物医学事件类型与 19 种生物医学事件的子类型，和 ACE 2005 一样，每一种子事件类型都有自己的事件结构，要素角色类型共有 7 种。表 2.5 种记录了所有 MLEE 中定义的生物医学事件类型与子类型。

表 2.5　MLEE 定义的事件类型与其子事件类型

Tab 2.5　Event type and subevent type in MLEE

生物医学事件类型	生物医学事件子类型
Anatomical	Cell proliferation, Development, Blood Vessel Development, Death, Breakdown, Remodeling, Growth
Molecular	Synthesis, Gene Expression, Transcription, Catabolism, Phosphorylation, Dephosphorylation
General	Localization, Binding, Regulation, Positive Regulation, Negative Regulation
Planned	Planned Process

MLEE 自身已经给定了训练集、开发集和测试集的划分。本研究使用了 TEES[73] 作为预处理程序来解析原始语料。预处理过后的句子统计指标如表 2.6 中所示。

表 2.6　MLEE 统计指标

Tab 2.6　The statistics of MLEE

	文档数	句子数
训练集	131	1271
开发集	44	457
测试集	87	880

（3） 评价标准

在 ACE 2005 的评价中，现有的工作也都遵循了 Li[22] 的评价标准，即一个触发词被正确分类当且仅当预测的子事件类型与在原文中的偏移量都正确；一个要素角色被正确分类当且仅当预测的要素角色类型正确且在原文中的偏移量也正确。为了量化分类的好坏，本研究也遵循之前的工作中使用的评价指标，采用了精确率（Precision, P）、召回率（Recall, R）和 F1 值的评价指标，并且在平均多个类别指标上采用了微平均（micro average）的方法。它们的计算方法如下所示：

$$Micro\_\mathrm{P} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FP_i} \tag{2.18}$$

$$Micro\_\mathrm{R} = \frac{\sum_i TP_i}{\sum_i TP_i + \sum_i FN_i} \tag{2.19}$$

$$Micro\_\mathrm{F1} = \frac{2 * Micro\_\mathrm{P} * Micro\_\mathrm{R}}{Micro\_\mathrm{P} + Micro\_\mathrm{R}} \tag{2.20}$$

在 MLEE 的评价中，除了需要分别计算触发词和要素各自的 P、R、F1 值，还需要将整个事件结构恢复后计算事件的 P、R、F1 值。

2.4.2 事件时序关系识别语料与评价标准

本研究在事件时序关系识别任务上分别使用了 Timebank-Dense[74]（TB-Dense）语料库与 MATRES[75] 语料库作为评价基准语料库。

（1） Timebank-Dense 语料库

最初的 TimeBank[76]（TB）语料库存在 TLINK 稀疏的特点，TB-Dense 的提出就是为了解决这一稀疏性问题，它引入 *VAGUE* 标签来标识不明确的时序关系，并采取在相邻语句内强制标注的策略而使得标注 TLINK 的密度远大于之前的 TB 语料库。在 TB-Dense 语料库中一共定义了 6 种时序关系，它们分别是：*AFTER*, *BEFORE*, *SIMULTANEOUS*, *INCLUDE*, *IS_INCLUDED*, *VAGUE*，并且除了 *VAGUE* 标签，其余标签都存在自己的反标签。

在本研究中，也遵循了在现有工作中的约定，只关注事件与事件之间的时序关系。TB-Dense 中的一些统计指标如表 2.7 中所示。

表 2.7 TB-Dense 统计指标

Tab 2.7 The statistics of TB-Dense

	文档数	事件对数目
训练集	22	4032
开发集	5	629
测试集	9	1427

（2） MATRES 语料库

MATRES 语料库的构建是基于 TB-Dense 语料库上展开的，它过滤了一些非言语（non-verbal）的事件，还将事件投射到多轴上并且只保留了那些在主轴上的事件，通过

采用事件起始点来改进注释者间的一致性（Inter-Annotator Agreements, IAA），进一步提高数据质量。MATRES 包含了 4 种时序关系类型：*AFTER*，*BEFORE*，*EQUAL*，*VAGUE*。

本研究采用了 Ning[54] 的预处理程序来对原始 MATRES 语料进行解析，一些统计指标如表 2.8 所示。

表 2.8　MATRES 统计指标

Tab 2.8　The statistics of MATRES

	文档数	事件对数目
训练集	204	11444
开发集	51	1296
测试集	20	837

（3） 评价标准

和现有研究保持一致，在时序关系识别任务上，本研究也使用微平均的 P，R，F1 值作为评价指标。它们具体的计算方法可见上文的公式 (2.17)-(2.19)，值得注意的是，在两个语料中被剔除的负类有所区别。为了与以往工作进行公平的比较，在 TB-Dense 中按照约定不统计 *SIMULTANEOUS* 类型的 P，R，F1 值；在 MATRES 中按照约定不统计 *VAGUE* 类型的 P，R，F1 值。

3　基于层次蒸馏网络的事件抽取

3.1　引言

事件抽取任务作为信息抽取领域中的重要子任务，无论是在学术界还是在工业界都受到了广泛的关注。目前阶段，主流的事件抽取系统都延续了流水线的抽取思路：（1）触发词识别，识别给定句子中的触发词；（2）触发词分类，对识别出来的触发词进行分类；（3）要素识别，判断实体或时间表达式与触发词之间是否存在要素关系；（4）要素分类，判断识别出来的要素在事件中扮演的角色类型。而往往触发词的识别和分类被合并成一个触发词抽取的过程，要素的识别和分类也被合并成一个要素抽取的过程，所以在本章的后文中约定，所提到的触发词抽取就包含了触发词的识别和分类，要素抽取就包含了要素的识别和分类。随着深度学习技术的逐渐普及，尽管目前的事件抽取系统已经取得了不错的结果，但是它们往往存在对句子编码不充分的问题。针对这一个问题，本章提出了一种层次蒸馏网络（HDN）来获取语义丰富的句子表示。

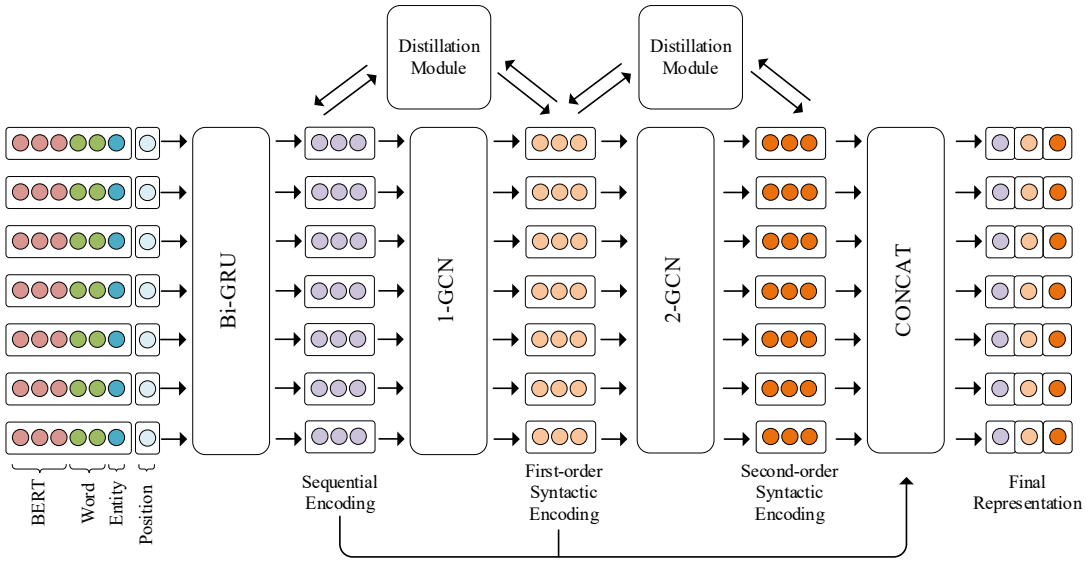

图 3.1　层次蒸馏网络架构图

Fig 3.1　The architecture of proposed HDN

3.2　层次蒸馏网络

本节先对 HDN 的网络架构进行了阐述，然后再详细叙述了如何利用提出的 HDN 模型来进行触发词抽取和要素抽取。HDN 模型的整体架构如图 3.1 中所示，它主要包括了 4 个主要的功能模块：（1）输入文本的嵌入模块，用来将一个独热（One-hot）的向量转

化成一个连续的实值向量；（2）BI-GRU 模块，用来捕捉给定文本在序列层面上的特征；（3）若干个 GCN 模块，用来捕捉给定文本在句法层面上的多阶特征；（4）蒸馏模块，也是 HDN 模型中最为核心的一个模块，它用来最大化多级编码之间的差异性，即减少多级编码之间的冗余，从而能够得到更具有表征力的句子编码。在以下的描述中，如果没有特殊说明，则约定小写字母代表一个标量，小写加粗字母代表一个列向量，大写字母则表示一个矩阵。

3.2.1 输入嵌入层

传统的词表示方法如独热表示方法，它们如果被直接地使用到构建的机器学习系统中往往会伴随着维度灾难（Dimension Disaster）的问题。而为了解决这一问题，词的连续表示方法如 word2vec[68] 、ELMo[69] 和 BERT[71] 等被相继提出，连续化的词表示方法不仅克服了独热表示方法面临的难题，还极大地丰富了单词向量的语义信息。在 HDN 中我们使用的词嵌入有以下 4 种：

(1) 上下文无关的词嵌入：在构建词嵌入的过程中，我们首先遍历整个语料库，过滤低频率的词并构建单词字典；然后用一些开源的预训练词向量对齐到构建的单词字典上来构建模型的词嵌入表。并且，我们在训练的过程中固定了上下文无关的词嵌入表。

(2) 实体类型的嵌入：和构建词嵌入的过程类似，我们也针对每一种出现的实体类型初始化了一个向量，区别在于我们没有用预训练的向量初始化它，而是让它随着模型一起优化来学习实体类型的表征。

(3) 上下文相关的词嵌入：除了以上两种独立于上下文存在的嵌入类型，我们还引入了上下文相关的词嵌入 BERT 来进一步丰富我们的语义信息。具体的，我们首先利用 BERT 的词片段分词器（Wordpiece Tokenizer）对输入的文本进行分词；然后经过预训练 BERT 的词嵌入层与 Transformer 层来得到最终的词片段表示；最后为了将词片段的表示映射回到原来的词表上，我们使用了平均的方法，即最终上下文相关的词向量为它对应的多个词片段向量的平均。

(4) 相对位置嵌入：相对位置嵌入仅用于要素抽取的过程中，它用来标识触发词与实体之间的相对位置。我们采用了以下的方式计算相对位置：

$$p_i = \begin{cases} k-i & i \le k \\ i-k-m & L \ge i \ge k+m \\ 0 & k+m \ge i \ge k \end{cases} \tag{3.1}$$

公式中 k 为触发词或候选实体在文本中的起始下标位置，m 为触发词或候选实体的长度，L 为文本本身的长度。然后和实体嵌入的操作类似，我们为每一个位置的值赋予一个初始化的向量，并让其随着模型在训练过程中一起优化。

在得到以上多个词嵌入之后，将它们进行简单地拼接。至此，我们将一个输入的单词序列 $W=[\boldsymbol{w}_1,\boldsymbol{w}_2,...,\boldsymbol{w}_L]$ 转化为了一个实值向量序列 $X=[\boldsymbol{x}_1,\boldsymbol{x}_2,...,\boldsymbol{x}_L]$。

3.2.2 带有门控单元的双向循环神经网络层

在对输入文本进行嵌入操作后，首先考虑的是利用在序列编码上具有优势的 BI-GRU 对嵌入后的实值向量序列进行编码。在小批量训练的过程中，出于对效率的考虑，将小批量的文本都固定到一个长度下，即对于那些较长的句子，修剪掉靠后的单词，而对于那些较短的句子，插入特殊的标记单词。BI-GRU 包括了两个循环的计算单元：前向计算单元和后向计算单元。它们随着序列的正方向或逆方向递归地更新，以前向计算为例，它具体的计算操作如下：

$$\boldsymbol{z}_t=\sigma(W_z\cdot[\overrightarrow{\boldsymbol{h}_{t-1}},\boldsymbol{x}_t]+b_z) \tag{3.2}$$

$$\boldsymbol{r}_t=\sigma(W_r\cdot[\overrightarrow{\boldsymbol{h}_{t-1}},\boldsymbol{x}_t]+b_r) \tag{3.3}$$

$$\widetilde{\boldsymbol{h}_t}=\tanh(W_h\cdot[\boldsymbol{r}_t*\overrightarrow{\boldsymbol{h}_{t-1}},\boldsymbol{x}_t]+b_h) \tag{3.4}$$

$$\overrightarrow{\boldsymbol{h}_t}=(1-\boldsymbol{z}_t)*\overrightarrow{\boldsymbol{h}_{t-1}}+\boldsymbol{z}_t*\widetilde{\boldsymbol{h}_t} \tag{3.5}$$

其中 $W_{(\cdot)}$ 和 $b_{(\cdot)}$ 为参数矩阵和偏置项，它们随着模型在训练过程中优化；σ 和 tanh 为非线性的激活函数，用于为 BI-GRU 层提供非线性的拟合能力；$\boldsymbol{x}_t$ 为当前时间步的输入向量，它们来自于上一个嵌入层的输出；$\overrightarrow{\boldsymbol{h}_{t-1}}$ 为上一个时间步前向计算单元的输出，它的引入使得 BI-GRU 满足了马尔可夫性，而使得能够捕捉来自序列的特征；$\boldsymbol{z}_t$ 和 $\boldsymbol{r}_t$ 为门控向量，它们的值域被 σ 函数限定在了 0 到 1 之间，而使得能够整合上一个时间步和这一个时间步的特征向量。反向计算单元与前向计算单元的计算步骤类似，只是在相反方向上执行各个步骤的更新运算。在分别得到两个方向上的句子编码 $\{\overrightarrow{\boldsymbol{h}_1},\overrightarrow{\boldsymbol{h}_2},...,\overrightarrow{\boldsymbol{h}_L}\}$ 和 $\{\overleftarrow{\boldsymbol{h}_1},\overleftarrow{\boldsymbol{h}_2},...,\overleftarrow{\boldsymbol{h}_L}\}$ 之后，将它们拼接而得到 BI-GRU 层输出的最终的句子编码 $H=\{\boldsymbol{h}_1,\boldsymbol{h}_2,...,\boldsymbol{h}_L\};\boldsymbol{h}_i=[\overrightarrow{\boldsymbol{h}_i},\overleftarrow{\boldsymbol{h}_i}]$。

3.2.3 基于依存句法树的图卷积网络层

传统的 RNN 尽管在理想情况下可以捕捉任意长文本的序列信息，但是在实际训练过程中，它往往会有着梯度弥散或者梯度消失等问题，而不能从远距离的上下文中获取

指示性强的信息，LSTM 和 GRU 的提出也只是一定程度上缓解了这一问题，却不能完全解决这一问题。

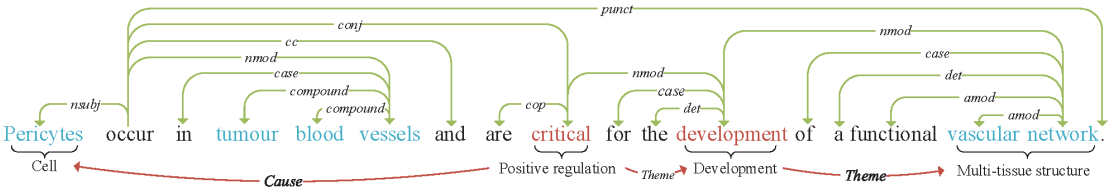

图 3.2　依存句法树举例

Fig 3.2　The example of dependency tree

图 3.2 中包含了一个来自 MLEE 语料中的文本与经过依存句法解析器解析后的结果。给定文本中存在一个 *Positive_Regulation* 事件和一个 *Development* 事件，直觉上，如果候选词 *“critical”* 能够捕捉到来自命名实体 *“Pericytes”* 的更多信息，那么模型就越能够轻易地将 *“critical”* 识别为一个事件的触发词，也可以更轻易地将 *“Pericytes”* 识别为该事件的要素。但是如果单纯地依靠 RNN 来促使两者之间的信息交互，那么需要 8 次的循环单元计算（序列角度上看 *“critical”* 与 *“Pericytes”* 相距了 8 个单词的距离），来自于前者的指示性信息不可避免地在 RNN 的运算过程中被遗忘。而如果从依存句法树的角度来计算以上两者之间的最短路径长度，那么 *“critical”* 只需要 2 跳的距离就能够与 *“Pericytes”* 发生交互（这里忽略了依存句法树中边的方向），前者也就能更轻易地获得后者所蕴含的指示性信息。

通过以上的分析，可以看到依存句法树能够给模型中结点交互带来的潜在优势，而 GCN 恰好可以发挥这一优势，它可以诱导结点间在非序列层面上发生的信息交互。本研究引入了基于依存句法树的图卷积网络层来捕捉文本中存在的多阶句法信息，具体地，本文使用了传统 GCN 的一个变种门控图网络（Gated Graph Network, GGN）来执行图上结点之间的计算，详细地计算步骤如下所示：

$$Y = D^{-0.5} A D^{-0.5} \cdot (W_Y \cdot H^{(l)}) \tag{3.6}$$

$$U = \sigma(W_U \cdot [Y, H^{(l)}]) \tag{3.7}$$

$$R = \sigma(W_R \cdot [Y, H^{(l)}]) \tag{3.8}$$

$$\tilde{H}^{(l+1)} = \mathrm{ReLU}(W_H \cdot [R * H^{(l)}, Y]) \tag{3.9}$$

$$H^{(l+1)} = (1-U) * H^{(l)} + U * \tilde{H}^{(l+1)} \tag{3.10}$$

其中 A 表示依存句法树所对应的邻接矩阵，并且本研究将有向边都转变成了无向边，还为 A 加上了单位矩阵以构成各个结点的自环连接，邻接矩阵用于对所有相邻结点的特征向量聚合；$D^{-0.5}(\cdot)D^{-0.5}$ 为对邻接矩阵的对称归一化，它用于调节特征向量聚合时的尺度，

防止聚合前后的尺度不一致问题，由此可以看出 GGN 本质上采用的是一种平均化的聚合方式；$H^{(l)}$ 为上一个图卷积层输入的结点特征矩阵，并且 $H^{(0)}$ 被初始化成 BI-GRU 的输出 H。随着 GCN 层的不断叠加，一个结点也可以和距离更远的结点发生信息交互，具体来说，一个 GCN 层可以诱导结点与其直接相邻的结点发生信息交互，因此本研究中也把第一层 GCN 的输出称为一阶句法编码；而两个叠加的 GCN 层则可以诱导结点与距离小于等于 2 的结点发生信息交互，因此本研究中也把第二层 GCN 的输出称为二阶句法编码，以此类推，通过多个 GCN 层的运算可以得到给定文本的多阶句法编码。

3.2.4 蒸馏模块

到目前为止，本文从之前的各个网络层中得到了给定句子的多个编码：来自于 BI-GRU 的序列编码，来自于 GCNs 的多阶句法编码，而为了得到最终应用于分类任务的句子表示，我们还需要探究一种合理的方式来对得到的多个编码进行整合。一个很自然的想法是直接对多个编码进行拼接，但是，这种方式存在的问题是这些编码不可避免地在特征空间中发生重叠，即存在冗余现象。比如，GCN 的过度平滑问题就是产生了冗余的表象，高层的 GCN 以低层的 GCN 输出作为输入，则高阶的句法编码就不可避免地包含了低阶句法编码中的部分特征，从而产生冗余。

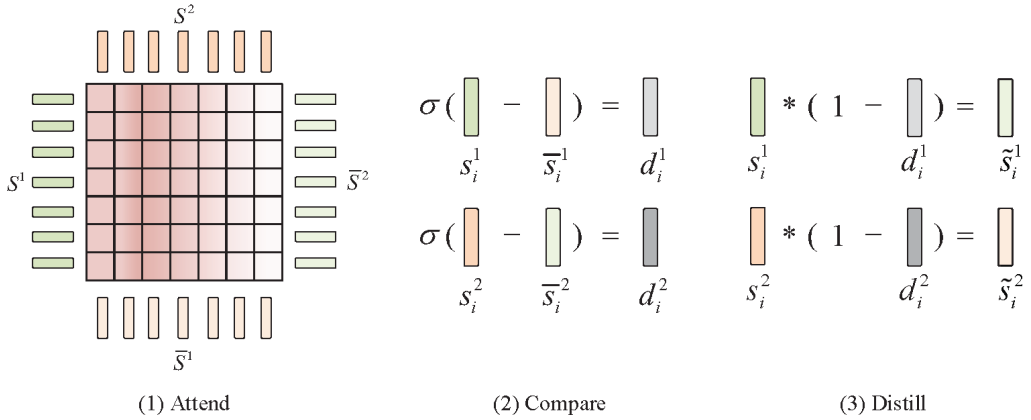

图 3.3 蒸馏模块细节图

Fig 3.3 The detail of distillation module

为了解决这一问题，本研究设计了一个新颖的蒸馏模块来对得到的多个编码进行整合，设计蒸馏模块背后的动机在于最大化编码与编码之间的异质性。图 3.3 为提出的蒸馏模块的细节图，它每次以两个待被蒸馏的编码作为输入，输出结果为两个蒸馏后的编码表示。本小节中记 $S^1 = \{\boldsymbol{s}_1^1, \boldsymbol{s}_2^1, ..., \boldsymbol{s}_L^1\}$ 与 $S^2 = \{\boldsymbol{s}_1^2, \boldsymbol{s}_2^2, ..., \boldsymbol{s}_L^2\}$ 为两个待被蒸馏的编码，蒸馏模块包含了以下三个核心的操作：

（1） 注意（Attend）

首先，本研究利用了双向的注意力机制来对齐两个待被蒸馏的编码，并且记对齐后的编码分别为 $\overline{S}^1=(\overline{\boldsymbol{s}}_{\boldsymbol{1}}^{\boldsymbol{1}},\overline{\boldsymbol{s}}_{\boldsymbol{2}}^{\boldsymbol{1}},...,\overline{\boldsymbol{s}}_{\boldsymbol{L}}^{\boldsymbol{1}})$ 和 $\overline{S}^2=(\overline{\boldsymbol{s}}_{\boldsymbol{1}}^{\boldsymbol{2}},\overline{\boldsymbol{s}}_{\boldsymbol{2}}^{\boldsymbol{2}},...,\overline{\boldsymbol{s}}_{\boldsymbol{L}}^{\boldsymbol{2}})$。它们由以下计算步骤得到：

$$e_{i,j}=\boldsymbol{s}_i^{1^T}\cdot W_E\cdot\boldsymbol{s}_j^2 \tag{3.11}$$

$$\alpha_{i,j}^1=\frac{\exp(e_{i,j})}{\sum_{k=1}^{L}\exp(e_{i,k})} \tag{3.12}$$

$$\overline{\boldsymbol{s}}_{\boldsymbol{i}}^{\boldsymbol{1}}=\sum_{j=1}^{L}\alpha_{i,j}^1\boldsymbol{s}_{\boldsymbol{j}}^{\boldsymbol{2}} \tag{3.13}$$

$$\alpha_{i,j}^2=\frac{\exp(e_{i,j})}{\sum_{k=1}^{L}\exp(e_{k,j})} \tag{3.14}$$

$$\overline{\boldsymbol{s}}_{\boldsymbol{j}}^{\boldsymbol{2}}=\sum_{i=1}^{L}\alpha_{i,j}^2\boldsymbol{s}_{\boldsymbol{i}}^{\boldsymbol{1}} \tag{3.15}$$

蒸馏模块中使用了双线性（bilinear）的注意力得分计算方式，$e_{i,j}$ 为 $\overline{S}^1$ 中第 i 个单词与 $\overline{S}^2$ 中第 j 个单词的注意力得分，它在一定程度上代表了两个单词间的匹配程度；$\alpha_{i,j}^1$ 和 $\alpha_{i,j}^2$ 则是在两个方向下的归一化注意力权重，最终得到的对齐表示 $\overline{S}^1$ 是 $\{\boldsymbol{s}_{\boldsymbol{1}}^2,\boldsymbol{s}_{\boldsymbol{2}}^2,...,\boldsymbol{s}_{\boldsymbol{L}}^2\}$ 的线性组合，$\overline{S}^2$ 是 $\{\boldsymbol{s}_{\boldsymbol{1}}^{\boldsymbol{1}},\boldsymbol{s}_{\boldsymbol{2}}^{\boldsymbol{1}},...,\boldsymbol{s}_{\boldsymbol{L}}^{\boldsymbol{1}}\}$ 的线性组合。这一步骤中，蒸馏模块所期望的是分别用 S^1 来表示 S^2，用 S^2 来表示 S^1。

（2） 比较（Compare）

我们最终的目的是需要完成对两个给定编码的蒸馏，所以在这一步骤中，通过计算得到用于蒸馏以上两个编码的比率。对齐表示 $\overline{S}^1$ 是 $\{\boldsymbol{s}_{\boldsymbol{1}}^2,\boldsymbol{s}_{\boldsymbol{2}}^2,...,\boldsymbol{s}_{\boldsymbol{L}}^2\}$ 的线性组合，$\overline{S}^2$ 是 $\{\boldsymbol{s}_{\boldsymbol{1}}^{\boldsymbol{1}},\boldsymbol{s}_{\boldsymbol{2}}^{\boldsymbol{1}},...,\boldsymbol{s}_{\boldsymbol{L}}^{\boldsymbol{1}}\}$ 的线性组合，所以 $S^1-\overline{S}^1$ 一定程度上衡量了从 S^2 变换到 S^1 的复杂程度，$S^2-\overline{S}^2$ 则一定程度上衡量了从 S^1 变换到 S^2 的复杂程度。因此可以通过它们来计算蒸馏比率如下：

$$\boldsymbol{d}_{\boldsymbol{i}}^{\boldsymbol{1}}=\sigma(\boldsymbol{s}_{\boldsymbol{i}}^{\boldsymbol{1}}-\overline{\boldsymbol{s}}_{\boldsymbol{i}}^{\boldsymbol{1}}) \tag{3.16}$$

$$\boldsymbol{d}_{\boldsymbol{i}}^{\boldsymbol{2}}=\sigma(\boldsymbol{s}_{\boldsymbol{i}}^{\boldsymbol{2}}-\overline{\boldsymbol{s}}_{\boldsymbol{i}}^{\boldsymbol{2}}) \tag{3.17}$$

$\boldsymbol{d}_{\boldsymbol{i}}^{\boldsymbol{1}}$ 是用于蒸馏 $\boldsymbol{s}_{\boldsymbol{i}}^{\boldsymbol{1}}$ 的蒸馏向量，它被 σ 函数激活后的值域限制在 0 到 1 之间，而 $\boldsymbol{d}_{\boldsymbol{i}}^{\boldsymbol{2}}$ 则用于 $\boldsymbol{s}_{\boldsymbol{i}}^2$ 的蒸馏。

（3） 蒸馏（Distill）

在得到蒸馏向量 $\boldsymbol{d}_i^1$ 与 $\boldsymbol{d}_i^2$ 之后，最终的蒸馏后的向量按以下形式计算得到：

$$\tilde{\boldsymbol{s}}_i^1 = \boldsymbol{s}_i^1 * (1 - \boldsymbol{d}_i^1) \tag{3.18}$$

$$\tilde{\boldsymbol{s}}_i^2 = \boldsymbol{s}_i^2 * (1 - \boldsymbol{d}_i^2) \tag{3.19}$$

$\tilde{\boldsymbol{s}}_i^1$ 与 $\tilde{\boldsymbol{s}}_i^2$ 为最终得到的蒸馏后的向量表示。除了用于计算蒸馏后的编码表示之外，蒸馏模块也为最终的目标函数提供了额外的惩罚项。随着优化过程的不断进行，BI-GRU 层与各个 GCN 层可以自发地越来越擅长避免在给定文本中捕获冗余的特征，换句话说，随着优化的进行，蒸馏模块所发挥的作用会越来越小，并且在理想情况下，蒸馏模块最终将成为整个模型的冗余项，即有蒸馏向量 $\boldsymbol{d}_i^1 = \boldsymbol{d}_i^2 = \boldsymbol{0}$，从而有 $\tilde{\boldsymbol{s}}_i^1 = \boldsymbol{s}_i^1$ 和 $\tilde{\boldsymbol{s}}_i^2 = \boldsymbol{s}_i^2$。出于以上考虑，为了在整体模型的复杂度与整体模型的性能之间做出权衡，我们很自然地将蒸馏向量的范数 $\| D^1 \|^2$ 与 $\| D^2 \|^2$ 加入到目标函数中作为惩罚项，其中 $D^1 = (\boldsymbol{d}_1^1, \boldsymbol{d}_2^1, ..., \boldsymbol{d}_L^1)$，$D^2 = (\boldsymbol{d}_1^2, \boldsymbol{d}_2^2, ..., \boldsymbol{d}_L^2)$，$\| \cdot \|^2$ 表示矩阵的二范数。

最后，如图 3.1 中所示，HDN 分别在序列编码、多阶句法编码的两两之间插入了蒸馏模块，以减少相邻编码之间的信息冗余。最后 HDN 再将这些蒸馏后的编码拼接，构成最终的句子表示 $R = (\boldsymbol{r}_1, \boldsymbol{r}_2, ..., \boldsymbol{r}_L)$ 服务于下游的分类任务。

3.2.5 触发词抽取

本研究中采用了序列标注的方法对句子中的触发词进行抽取，并且采用了 BILOU 的标注策略。其中 B 代表 *“begin”*，标识一个触发词的起始词；I 代表 *“inter”*，标识一个触发词的中间词；L 代表 *“last”*，标识一个触发词的结尾词；O 代表 *“other”*，标识该单词不是触发词的一部分；U 代表 *“unique”*，标识该触发词仅由这一个单词组成。序列标注的过程是一个结构化预测的过程，相较于传统的候选词分类方法，它在预测效率上有较大优势，可以一次性识别文本中的多个触发词。在每一个时间步上，序列标注分类器都对当前单词进行多分类：

$$p(y_i \mid x) = \text{softmax}(W_{te} \cdot \boldsymbol{r}_i + b_{te}) \tag{3.20}$$

其中，softmax 函数用于对最终概率的归一化。在给定训练样本 $\{(x^{(j)}, y^{(j)}); j \le N\}$ 的情况下，触发词抽取的损失函数被定义为加上正则项的负对数似然：

$$loss_{te} = -\sum_{j=1}^{N} \sum_{i=1}^{L} \log p(y_i^{(j)} \mid x^{(j)}; \theta) + \lambda \| D \|^2 \tag{3.21}$$

λ 为超参数用于调节正则项的尺度，D 为公式(3.16)-(3.17)计算得到的蒸馏向量矩阵。

3.2.6 要素抽取

本研究将要素抽取视为一个成对分类的问题，即提取触发词-实体对的编码特征后送入分类器进行分类。对每个触发词-实体对，考虑到触发词或实体可能为一个单词的序列，所以本研究采用平均单词序列向量的方法，得到各自的编码特征向量$\boldsymbol{t}$和$\boldsymbol{a}$。最后，这两个特征向量的拼接就作为该触发词-实体对的编码特征：

$$p(y \mid x) = \text{softmax}(W_{ae} \cdot [\boldsymbol{t}, \boldsymbol{a}] + b_{ae}) \tag{3.22}$$

给定训练样本$\{(x^{(j)}, y^{(j)}); j \le M\}$的情况下，要素抽取的损失函数计算如下：

$$loss_{ae} = -\sum_{j=1}^{M} \log p(y^{(j)} \mid x^{(j)}; \theta) + \lambda \| D \|^2 \tag{3.23}$$

λ为超参数用于调节正则项的尺度，D为公式(3.16)-(3.17)计算得到的蒸馏向量矩阵。

3.2.7 生物医学事件的后处理

在生物医学事件的抽取中，需要将触发词与识别的要素组合成完整的事件结构，这就依赖于后处理的步骤。在本文中，使用了 TEES[73] 中的 SVM 方法对识别的触发词和要素进行后处理，SVM 分类器所使用的主要特征有：线性跨度特征、要素组合特征和要素内容特征等。

3.3 事件抽取实验

本节分为三个部分来介绍事件抽取的实验内容以及对其进行的错误分析。首先为 HDN 在 ACE 2005 语料库上的实验，本研究列举了当前达到最先进水平（state-of-the-art，SOTA）的一批模型并进行横向比较；然后第二部分为 HDN 在 MLEE 语料库上的实验，第二部分中除了包含与 SOTA 模型的横向比较，也包含了对 HDN 本身进行的消融实验来进行模型层面的纵向分析；最后，第三部分中本研究对 ACE 2005 上的实验结果进行了一定的错误分析和讨论。

3.3.1 ACE 2005 事件抽取实验

（1） 实验设置

ACE 2005 语料库中包含了 599 个标注文档，在实验中，本文遵循了 Li[22] 的分割方式，采用 529 个文档进行训练，30 个文档作为开发集用于超参数选择，剩余 40 个文档作为测试集用于模型评估。本研究利用 AllenNLP[77] 框架构建深度学习模型，并采用 Optuna[3]作为超参数选择工具。

[3] https://optuna.org/

表 3.1 中列出了触发词模型最终的超参数设置。词嵌入部分，本研究采用了 Chen[78] 开源的利用 word2vec 训练的词向量；预训练语言模型方面，采用了 bert-base-uncased 的权重；优化器的选择方面，使用 Adam 优化器。

表 3.1　ACE 2005 触发词模型的超参数

Tab 3.1　Params of trigger extraction on ACE 2005

参数	参数含义	值
Dropout Rate	神经元丢弃率	0.48
Hidden Size	所有隐藏层的维度	726
RNN Layer Num	BI-GRU 层数	1
GCN Layer Num	GCN 层数	2
Label Weight	非 O 标签的损失权重	5
Word Embedding Size	词嵌入维度	100
Entity Embedding Size	实体嵌入维度	36
BERT	预训练 BERT	bert-base-uncased
Learning Rate	学习率	8E-5

表 3.2　ACE 2005 要素模型的超参数

Tab 3.2　Params of Argument extraction on ACE 2005

参数	参数含义	值
Dropout Rate	神经元丢弃率	0.3
Hidden Size	所有隐藏层的维度	532
RNN Layer Num	BI-GRU 层数	1
GCN Layer Num	GCN 层数	3
Word Embedding Size	词嵌入维度	100
Entity Embedding Size	实体嵌入维度	50
Position Embedding Size	相对位置嵌入维度	50
BERT	预训练 BERT	bert-base-uncased
Learning Rate	学习率	2E-4

表 3.2 中列出了要素模型最终的超参数设置。值得注意的是，在要素模型中，嵌入部分增加了相对位置的嵌入，以标识触发词-实体对的相对位置。

（2） 对比的模型

在触发词抽取实验中，本研究选取了以下具有代表性的先进的触发词抽取系统来进行横向比较：

① DMCNN[25] (2015)：DMCNN 在传统 CNN 的基础上进行改进，加入了动态多池化层对句子进行分段编码后聚合，以保留更多关键的信息，最后对每一个触发候选词进行分类。

② JRNN[26](2016)：JRNN 利用 RNN 进行触发词的抽取，并且还通过维持一个记忆矩阵的方式建立触发词和要素之间的条件概率关系，来实现要素的抽取。

③ dbRNN[28](2018)：dbRNN 在句子编码过程中考虑了句法信息，通过在 LSTM 中加入句法树中的边形成特殊的依存桥，相较于传统的 RNN，桥接方式使得每个单词可见的上下文范围更广从而更容易捕捉有效信息。

④ JMEE[30] (2018)：JMEE 对普通的图卷积网络进行改进，将依存边进行嵌入并和模型一起优化，并且 JMEE 也首次采用了序列标注的方法进行触发词抽取。

⑤ MOGANED[31](2019)：MOGANED 使用了 GCN 的一个变种 GAT 进行触发词抽取，GAT 在卷积运算之后加入了注意力机制以实现对邻接结点特征的动态加权融合。

⑥ EEGCN[32] (2020)：EEGCN 采用强化边信息表示的 GCN 用于识别触发词，设计了一个边感知的节点更新模块，通过特定的依赖类型聚合邻接结点特征来生成更有表征力的结点特征，还引入了一个节点感知的边嵌入更新模块，利用结点信息来迭代更新边的嵌入特征。

在要素抽取实验中，本研究也选取了一些具有代表性的先进要素抽取模型来进行横向比较，有一些模型在上述触发词部分中已被介绍，这里仅对未被提到的系统进行叙述：

① Li’s Joint[22](2013)：Li’s Joint 把事件抽取看作是一个结构化预测的问题，使用结构化感知机和集束搜索算法同时预测句子中的触发词和要素。

② DMBERT[33] (2020)：DMBERT 把 DMCNN 中的 CNN 模块替换成了预训练的 BERT 模型来获得更好的性能。

③ HMEAE[33] (2020)：HMEAE 针对要素识别任务提出使用高层实体概念指导要素角色分类的方法，每个高层实体概念对应一个可训练的嵌入向量，然后通过注意力机制聚合高层实体概念嵌入和句子编码后的特征表示，再对聚合之后的特征表示进行分类。

（3） 实验结果

表 3.3 中为比较的各个模型以及本研究提出的 HDN 在 ACE 2005 语料库上的触发词抽取结果。可以看到，本研究提出的 HDN 模型获得了最高的 F1 值，相比于之前最好的

模型 EEGCN 在 F1 值上高出了 2.7%，并且在未使用 BERT 的情况下也高出了 1.1%，实验结果充分证明了本研究提出的 HDN 模型在触发词抽取上的有效性。除此之外，还可以观察到 HDN 相比较于以往的模型，在召回率上取得了非常大的提升，相比于之前最好的模型 EEGCN 高出了 7.7%，而较高召回率恰恰是实际环境中所需要的，高召回的神经网络模型可以充分地弥补传统意义上专家系统低召回的缺点，达到平衡互补的作用。

表 3.3　ACE 2005 触发词抽取结果 [4]

Tab 3.3　Results of trigger extraction on ACE 2005

模型	P(%)	R(%)	F1(%)
DMCNN (2015)	75.6	63.6	69.1
JRNN (2016)	66.0	73.0	69.3
dbRNN (2018)	74.1	69.8	71.9
JMEE (2018)	76.3	71.3	73.7
MOGANED (2019)	**79.5**	72.3	75.7
EEGCN (2020)	76.7	78.6	77.6
HDN*(ours)	77.0	80.4	78.7
HDN (ours)	75.0	**86.3**	**80.3**

表 3.4　ACE 2005 要素抽取结果

Tab 3.4　Results of argument extraction on ACE 2005

模型	P(%)	R(%)	F1(%)
Li's Joint (2013)	64.7	44.4	52.7
DMCNN (2015)	62.2	46.9	53.5
JRNN (2016)	54.2	56.7	55.4
dbRNN (2018)	**66.2**	52.8	58.7
DMBERT (2020)	58.8	55.8	57.2
HMEAE (2020)	62.2	56.6	59.3
HDN (ours)	56.6	**65.0**	**60.5**

表 3.4 中为比较的各个模型以及本文提出的 HDN 在 ACE 2005 语料库上的要素抽取结果。相较于之前最好的模型 HMEAE，本文的 HDN 在 F1 值上取得了 1.2%的提升，值得注意的是，HDN 模型不像 HMEAE 模型，在识别要素阶段针对要素抽取任务的特点定制了特殊的注意力模型来关注高层的实体概念信息，而是直接简单对触发词和实体

[4] "*"为不带 BERT 的结果

的向量拼接后分类，但仅仅是这一简单的策略就已经比 HMEAE 模型在 F1 值上高出了超过一个百分点。和在触发词抽取阶段中类似，在要素抽取的结果中也关注到 HDN 带来召回率的提升是足够大的，比之前最好的 JRNN 模型获得了 8.3%的召回率提升，同样这也是有利于实际运用的。

3.3.2 MLEE 事件抽取实验

（1） 实验设置

表 3.5 MLEE 触发词模型的超参数

Tab 3.5 Params of trigger extraction on MLEE

参数	参数含义	值
Dropout Rate	神经元丢弃率	0.3
Hidden Size	所有隐藏层的维度	300
RNN Layer Num	BI-GRU 层数	1
GCN Layer Num	GCN 层数	2
Word Embedding Size	词嵌入维度	200
Entity Embedding Size	实体嵌入维度	50
BERT	预训练 BERT	biobert-base-uncased
Learning Rate	学习率	1E-3

表 3.6 MLEE 要素模型的超参数

Tab 3.6 Params of argument extraction on MLEE

参数	参数含义	值
Dropout Rate	神经元丢弃率	0.3
Hidden Size	所有隐藏层的维度	300
RNN Layer Num	BI-GRU 层数	1
GCN Layer Num	GCN 层数	1
Word Embedding Size	词嵌入维度	200
Entity Embedding Size	实体嵌入维度	50
Position Embedding Size	相对位置嵌入维度	50
BERT	预训练 BERT	biobert-base-uncased
Learning Rate	学习率	1E-3

与 ACE 2005 不太相同，MLEE 语料专注于生物医学文献上的事件抽取。MLEE 中包含了 131 个文档作为训练集，44 个文档作为验证集和 87 个文档作为测试集。

表 3.5 列出了 MLEE 语料上 HDN 的各个超参数设置。在优化器上依旧选择了 Adam 优化器，词嵌入部分则是使用了基于依存句法关系训练的生物医学领域词向量[80]，BERT 也同样选用了生物医学领域特定的 biobert-base-uncased 模型。最后，在 MLEE 语料上，我们发现将 BERT 作为特征使用比微调模型效果要好，所以实验中固定了 BERT 中的可训练参数，并采用了较大的学习率来加速模型的收敛。

表 3.6 为在 MLEE 上要素模型的各个超参数设置，主要的不同点在于加入了对于相对位置的词嵌入向量。

（2） 对比的模型

在触发词抽取实验中，本研究选取了以下具有代表性的先进触发词抽取系统来进行横向比较：

① Wang's CNN[35] (2016)：使用了 CNN 的方法，同时还引入了 POS 标签、位置和实体类别的特征来补充词嵌入的信息。

② Rahul's RNN[38] (2017)：使用了 RNN 的方法进行触发词的抽取，首先利用 RNN 在序列层面上对给定文本的词嵌入序列进行编码，最终利用这些得到的编码进行序列标注。

③ Contextual-GRU[40](2019)：在编码过程中动态地结合了初始的词嵌入向量和门控循环单元输出的隐层向量，并且在序列标注解码过程中引入了自回归依赖。

④ Joint-GATE-Doc[41](2020)：该模型考虑到文档潜在主题对触发词抽取的好处，是一种联合模型，可以同时提取生物医学事件触发词和文档的潜在主题。

在生物医学的事件抽取任务中，由于需要最后对整个事件结构进行重构，所以一般不单独对要素抽取的结果进行横向比较，而是直接比较合成事件后的指标，以下为对比的一些目前最先进的生物医学事件抽取系统：

① Pyysalo's SVM[72]（2012)：Pyysalo's SVM 采用了浅层机器学习中的 SVM 方法对生物医学文本中的事件结构进行抽取。

② Zhou's Semi-supervised[81](2015)：Zhou's Semi-supervised 引入半监督学习，并结合了来自于主题的特征。

③ LSTM-ATTN[80] (2019)：LSTM-ATTN 利用 BI-LSTM 与多层级的注意力模型来改善以往事件抽取系统整体的性能。

（3） 实验结果

表 3.7 中列出了 HDN 与先进的触发词抽取模型之间的比较结果。可以看出，HDN 也取得了目前最好的触发词结果，在都不使用预训练模型 BERT 的情况下，HDN 相较于之前最好的 Joint-GATE-Doc 提升了 0.6%，在都使用 BERT 的情况下，HDN 的提升更

不那么显著。这种现象的原因存在两方面，首先 Joint-GATE-Doc 模型是一个只关注触发词抽取的模型，即它关注如何使用文档的主题信息来改善生物医学触发词抽取的结果，并且没有做要素和事件抽取。而本研究提出的 HDN 是一个通用的编码框架，HDN 不局限于触发词抽取任务，也适用于要素抽取任务；其次，BERT 通过在大规模语料上的预训练，已经具备了一定捕捉文本句法信息的能力，这可能与 HDN 在编码功能上存在部分重叠，所以导致 BERT 为 HDN 带来的收益没有在 Joint-GATE-Doc 中的收益大。

表 3.7　MLEE 触发词抽取结果

Tab 3.7　Results of trigger extraction on MLEE

模型	P(%)	R(%)	F1(%)
Wang's CNN (2016)	73.6	**83.6**	78.3
Rahul's RNN (2017)	81.1	77.2	79.1
Contextual-GRU (2019)	80.7	80.4	80.6
Joint-GATE-Doc* (2020)	80.4	81.3	80.8
Joint-GATE-Doc (2020)	82.1	82.5	82.3
HDN*(ours)	82.9	80.0	81.4
HDN (ours)	**83.5**	81.4	**82.5**

表 3.8　MLEE 事件抽取结果

Tab 3.8　Results of event extraction on MLEE

模型	P(%)	R(%)	F1(%)
Pyysalo's SVM (2012)	62.3	49.6	55.2
Zhou's Semi-supervised (2015)	55.8	**59.2**	57.4
Wang's CNN (2016)	60.6	56.2	58.3
LSTM-ATTN (2019)	**90.2**	44.5	59.6
HDN*(ours)	72.6	52.8	61.1
HDN (ours)	73.1	55.0	**62.7**

表 3.8 中列出了 HDN 与当前最先进的一批事件抽取系统之间的比较结果。可以看到，HDN 取得了当前最好的结果，相较于之前最好的 LSTM-ATTN 模型，HDN 在 F1 值上高出了 3.1%，并且在未使用 BERT 的情况下也高了 1.5%，这也充分体现了 HDN 在生物医学领域事件抽取任务上的优越性。并且，相比于 LSTM-ATTN，HDN 的结果更加均衡，即有更为接近的精确率与召回率。除了进行以上与其他先进模型的比较，本研究也在 HDN 自身架构上进行了进一步的消融实验。消融实验可以分为两个要点，其一是研究 GCN 的层数对模型性能的影响，其二是研究蒸馏模块的有无对模型性能的影响。本研究关于 HDN 的消融实验都是以 MLEE 上的触发词抽取任务展开的。

表 3.9 列举出了消融实验的结果，在实验中，控制的两个消融变量为：GCN 的层数，即表格中的“K”；是否添加蒸馏模块，即表格中的“+”，比如，“K = 1+”表示使用了一层的 GCN，并且在 GCN 与 BI-GRU 之间插入了蒸馏模块。从表中可以看出，GCN 为模型带来的提升是显著的，只有一层 GCN 的模型（K = 1）就比单纯的 BI-GRU 模型（K = 0）在 F1 值上高出了半个百分点。这也是符合预期的，GCN 可以有效地从较远的上下文中捕捉有用的信息，并以此来改善模型。其次，我们也发现 GCN 的层数也并不是越多越好的，无论是否添加蒸馏模块，GCN 都在两层时取得了最好的结果。最后，通过比较各个层是否添加蒸馏模块的结果，我们也发现添加了蒸馏模块的模型都一致地比没有添加蒸馏模块的模型结果高，这也是符合预期的，蒸馏模块一定程度上减少了各阶编码在特征空间上的冗余，有助于获得更具表征力的编码。通过以上的分析，本研究所提出的 HDN 模型是有效的，并且在当前先进的事件抽取系统中具有优越性。

表 3.9　MLEE 触发词抽取的消融实验结果

Tab 3.9　Ablation results of trigger extraction on MLEE

消融变量	P(%)	R(%)	F1(%)
K = 0	79.6	**82.8**	81.1
K = 1	80.6	82.7	81.6
K = 2	81.8	82.4	82.1
K = 3	82.6	81.2	81.9
K = 1+	82.3	81.4	81.8
K = 2+	**83.5**	81.4	**82.5**
K = 3+	83.1	81.6	82.3

3.3.3　错误分析与讨论

本小节对在 ACE 2005 上的实验结果进行了一定的错误分析，由于现有的事件抽取系统仍很大程度上受到第一阶段触发词抽取的性能影响，故本小节中着重分析了触发词抽取阶段在 ACE 2005 上的实验结果。

图 3.4 为最终 ACE 2005 上触发词抽取结果的混淆矩阵，其中颜色的深浅代表了对应位置样本数量的多少。从图中我们可以看到，在所有的错误当中，*Attack* 类别的错误最多，造成这种现象的原因有两个，一是因为本身在语料的测试集当中，*Attack* 类别的数量就较多；二就是因为模型在预测过程中，对该类别的召回较差，并且常常被识别为非触发词，即 *O* 类别。除此之外，*Meet* 类别也同样面临着召回较差的问题。此外，*End-Org* 类别的触发词在预测中也有时候会被模型预测为 *Start-Org* 类别，造成这种现象的

原因是，*End-Org* 类别和 *Start-Org* 类别本身就是两个概念上相逆的事件类型，所以它们常常会共享相同的上下文内容，因而造成模型无法很好的区分这两种类型。

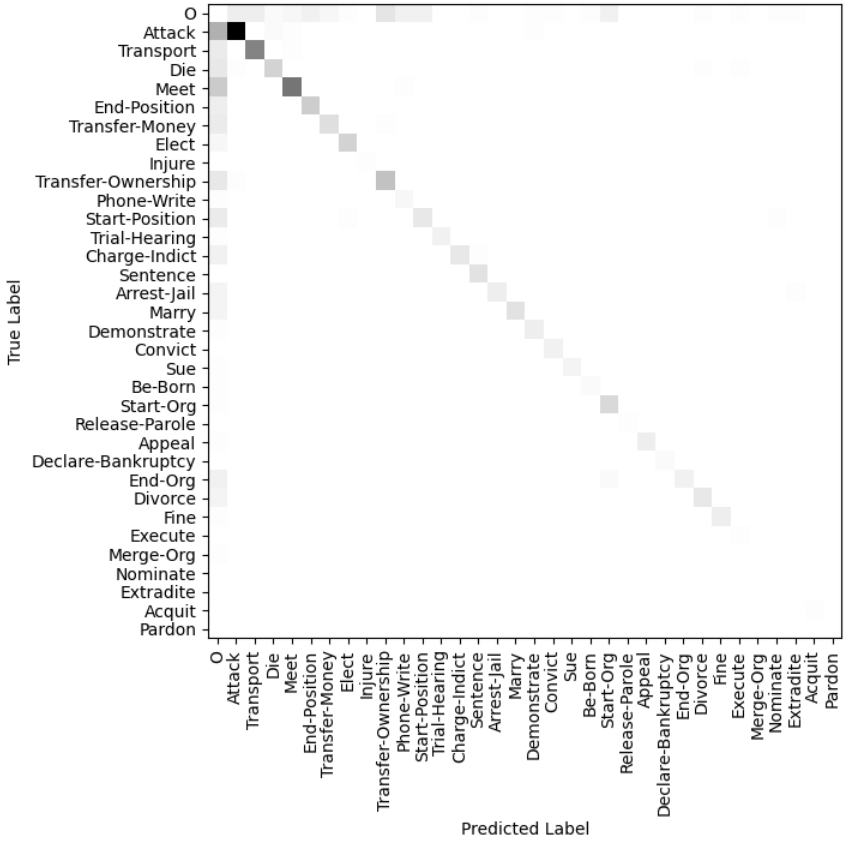

图 3.4　ACE 2005 上的触发词抽取结果混淆矩阵

Fig 3.4　The confusion matrix of trigger extraction results on ACE 2005

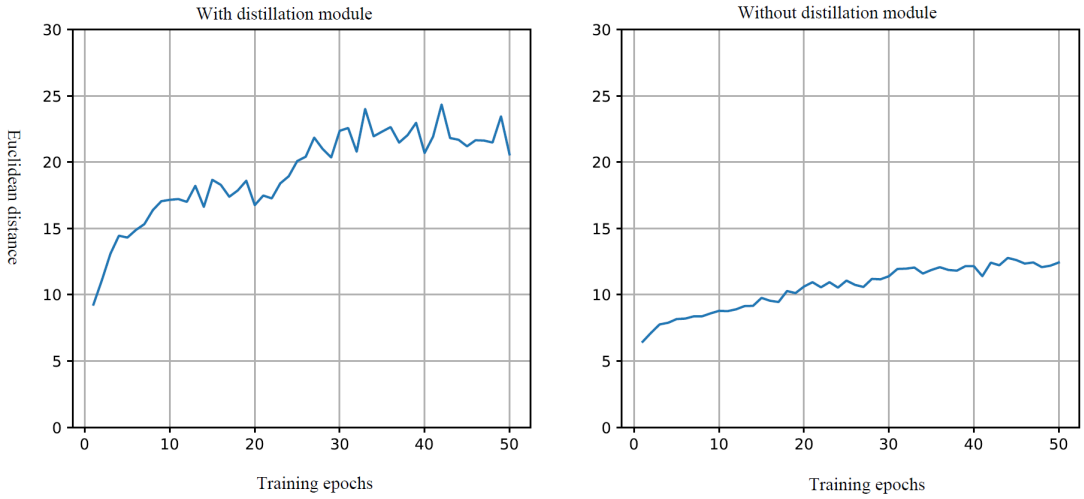

图 3.5　BI-GRU 与相邻 GCN 编码之间的欧氏距离随训练轮数变化曲线图

Fig 3.5　Euclidean distance between BI-GRU and adjacent GCN encodings as the epochs varies

除了关心实验的结果之外，我们也关心 HDN 是否如预期一样在特征空间上增强了相邻编码间的异质性。为了定量地衡量这种异质性，在每一个训练轮次结束时计算了两个给定编码之间的欧式距离，一般来说，两个编码之间的欧式距离越大，则两者在特征空间上的重叠越小，则两个编码之间的异质性越强。图 3.5 为序列编码（BI-GRU 编码）与一阶句法编码（相邻 GCN 的编码）之间的欧式距离随训练轮数变化的曲线图，左半部分为带上蒸馏模块的结果，右半部分为不带蒸馏模块的结果。可以观察到，首先，无论是否添加了蒸馏模块，序列编码与一阶句法编码之间的欧氏距离都随着训练轮次呈上升趋势，这也说明了随着优化的进行，BI-GRU 模块与 GCN 模块都能够自发地捕捉句中不同的特征；其次，添加了蒸馏模块的曲线上升的速度比没有添加蒸馏模块的曲线快，并且在峰值上几乎为后者的两倍。这也说明了蒸馏模块的确如我们期望的那样，减少了相邻编码在特征空间上的重叠，增强了它们之间的异质性。

3.4 本章小结

事件抽取任务在信息抽取中是一个很常见的任务，并且是之后的事件关系识别任务的一个基础。本章的主要贡献如下：

(1) 提出了一个新颖的 HDN 模型用于整个事件抽取任务，具体地说，HDN 融合了一个 BI-GRU 模块用于提取句子的序列信息，若干个 GCN 模块用于提取句子的多阶句法信息，以及基于双向注意力机制设计了一个蒸馏模块以减少层级编码之间的信息冗余。

(2) 本文在两个常用的、不同领域的事件抽取数据集上分别进行了实验，并与其它的先进事件抽取系统进行比较，最终验证了提出的 HDN 模型的有效性。

4 基于逻辑驱动深度对比网络的事件时序关系识别

4.1 引言

事件时序关系识别任务是一个建立在事件抽取任务之上的任务，它在构建以事件为中心的事理图谱中起着至关重要的作用。时序关系识别任务一般被描述为成对分类的问题，即首先构建事件对级别的特征，再通过分类器判断时序关系。随着深度学习技术的逐渐普及，尽管当前的时序关系识别模型都已取得了较大的改进，但它们仍存在着事件对特征构建过于简单、ILP 过程与训练过程割裂等问题。在本章中，针对以上问题提出了一种新的基于逻辑驱动的深度对比网络（LDCN）模型来完成时序关系的识别。

4.2 逻辑驱动深度对比网络

本节对本文所提出的逻辑驱动深度对比网络进行了阐述，首先着重介绍了深度对比网络（DCN）的架构，然后引入并介绍了逻辑驱动的训练框架以及具体的训练算法。DCN 模型如图 4.1 中所示，它的编码器采用了上一章所提出的在获取丰富语义编码方面具有优势的 HDN 模型，并且在构建事件对级别特征的阶段，DCN 分别从可交换特征（commutative features）与不可交换特征（non-commutative features）两个角度总结了最终用于分类器的特征。图 4.2 中描述了逻辑框架驱动下的 DCN 模型，也被称为 LDCN 模型。LDCN 采用了分阶段训练的策略，在第一阶段（First Pass）中仅对来自于监督学习本身的损失进行优化；而在第二阶段（Second Pass）中同时对监督损失与基于对称约束的损失进行优化；最后在第三阶段（Third Pass）中则是同时对监督损失与基于传递约束的损失一起进行优化。LDCN 的训练过程是一个级联的过程，即后面的训练阶段起始时都迁移了前面训练阶段结束时的 DCN 权重。在之后的小节中，将会对 DCN 的具体前向传播过程与 LDCN 的具体训练逻辑进行详细阐述。

4.2.1输入编码层

在 DCN 的编码阶段，采用了上一章节所提出的 HDN 模型作为模型编码器，HDN 的具体细节可参考上一章节，这里着重强调在编码时的不同点。首先，由于时序关系语料如 MATRES、TB-Dense 中并没有标注额外的实体信息，所以在 HDN 的嵌入层，我们并没有使用实体级别的嵌入，而是对词性（part-of-speech，POS）标签进行额外嵌入，语料中的 POS 标签皆由现成的解析器 CoreNLP[5]得到。所以，DCN 的嵌入阶段包含了对三种输入的嵌入，一个是单词级别的词嵌入，一个是词性标签的嵌入，最后一个是来自预

[5] https://stanfordnlp.github.io/CoreNLP/

训练语言模型的嵌入。在输入的嵌入层之上，DCN 同样使用了 BI-GRU 来对文本的序列信息进行抽取，使用了若干层的 GCN 来对文本的多阶句法信息进行抽取。

假定我们输入了一个单词序列 $S=[s_1,s_2,...,s_L]$，DCN 首先通过以下方式对其进行编码得到编码后的实值向量序列 $H=[h_1,h_2,...,h_L]$：

$$H=\mathrm{HDN}(S) \tag{4.1}$$

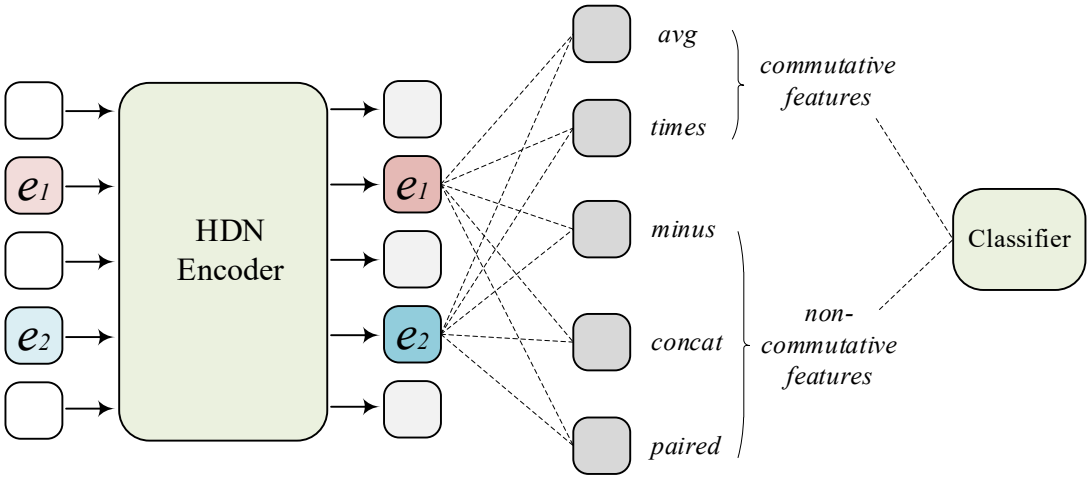

图 4.1　深度对比网络架构图

Fig 4.1　The architecture of proposed DCN

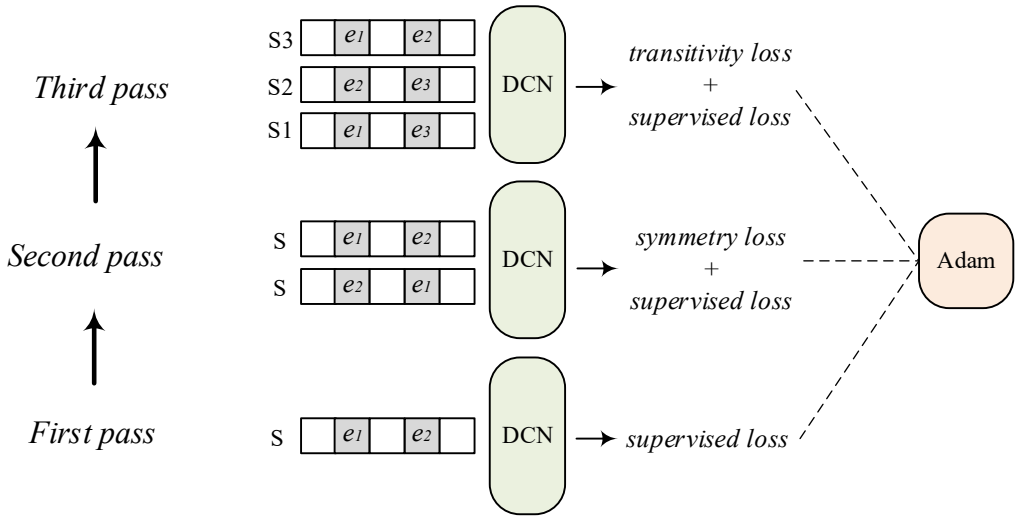

图 4.2　逻辑驱动的深度对比网络

Fig 4.2　Logic-driven deep compare network

4.2.2 特征构造层

传统的基于深度学习的时序识别模型在构造事件对特征阶段往往只是对事件对编码后的向量进行简单地拼接，而忽略了其他张量操作对模型的潜在收益。所以，在 DCN 的特征构造阶段，本研究分别使用了两大类的特征构造方式：可交换的特征构造方式与不可交换的特征构造方式。

这样的特征构造划分方式启发于我们对于语料中标签本身的观察。以 TB-Dense 中的 6 种时序关系为例，它们中有的是具有可交换性的，而有的则不具备可交换性。比如 *SIMULTANEOUS* 关系就是具有可交换性质的，具体地说，假设事件对 (e_1,e_2) 在给定文本中的关系为 *SIMULTANEOUS*，那么即使交换 e_1 与 e_2 的位置，事件对依然表现为 *SIMULTANEOUS* 关系。所以构造可交换的特征就能够有效地针对这些可交换标签的样本。再比如 *BEFORE* 关系为一种具有不可交换性质的标签，具体地说，假设事件对 (e_1,e_2) 在给定文本中的关系为 *BEFORE*，那么如果交换了 e_1 与 e_2 的位置，事件对之间的关系就会变成 *AFTER*，因为它们必须满足对称性约束。所以构造不可交换的特征就能够有效地针对这些不可交换标签的样本。

假定给定文本中事件 e_1 对应的编码后向量为 $\boldsymbol{h}_{e_1}$，事件 e_2 对应的编码后的向量为 $\boldsymbol{h}_{e_2}$，DCN 构造可交换的特征如下：

(1) 张量乘法（Times）：这里所说的张量乘法为两个向量的哈达玛积（Hadamard product），即两个张量对应位置元素的逐位相乘，它具有如下形式：

$$\boldsymbol{f}_1 = \boldsymbol{h}_{e_1} * \boldsymbol{h}_{e_2} \tag{4.2}$$

(2) 张量平均（Avg）：张量平均即为对两个事件对应向量的算术平均：

$$\boldsymbol{f}_2 = \frac{\boldsymbol{h}_{e_1} + \boldsymbol{h}_{e_2}}{2} \tag{4.3}$$

DCN 构造不可交换的特征如下：

(3) 张量减法（Minus）：张量减法为利用 e_1 的编码向量减去 e_2 的编码向量：

$$\boldsymbol{f}_3 = \boldsymbol{h}_{e_1} - \boldsymbol{h}_{e_2} \tag{4.4}$$

(4) 张量拼接（Concat）：拼接也作为一个不可交换特征在 DCN 中被使用：

$$\boldsymbol{f}_4 = [\boldsymbol{h}_{e_1}, \boldsymbol{h}_{e_2}] \tag{4.5}$$

(5) 张量对编码（Paired）：将事件对看作一个长度为 2 的序列，张量对编码操作利用 LSTM 对这一序列进行单向编码：

$$\boldsymbol{f}_5 = \mathrm{LSTM}(\{\boldsymbol{h}_{e_1}, \boldsymbol{h}_{e_2}\}) \tag{4.6}$$

最终，在送入分类器之前对以上所有特征进行拼接：

$$\boldsymbol{f} = [\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3, \boldsymbol{f}_4, \boldsymbol{f}_5] \tag{4.7}$$

4.2.3 时序关系识别

在得到最终的事件对的特征向量 $\boldsymbol{f}$ 之后，DCN 通过一个带有 softmax 激活函数的全连接层对其进行多分类：

$$p(y \mid x) = \mathrm{softmax}(W \cdot \boldsymbol{f} + b) \tag{4.8}$$

给定训练样本 $\{(x^{(j)}, y^{(j)}); j \le M\}$ 的情况下，时序关系识别的损失函数计算如下：

$$loss = -\sum_{j=1}^{M} \log p(y^{(j)} \mid x^{(j)}; \theta) \tag{4.9}$$

4.2.4 时序一阶逻辑知识

一阶逻辑（First-order Logic，FOL）也被称为谓词逻辑或一阶谓词逻辑，它在人工智能领域中常被用来对知识进行表示，所谓的“一阶”指的是谓词（predicate）作用后的个体（object）是一个命题。在时序关系识别任务中，常识知识是普遍存在的，比如事件对 (e_1, e_2) 关系为 *BEFORE* 的条件下，必然有事件对 (e_2, e_1) 关系为 *AFTER*；又比如事件对 (e_1, e_2) 关系为 *BEFORE* 且事件对 (e_2, e_3) 关系为 *BEFORE* 的情况下，必然有事件对 (e_1, e_3) 关系为 *BEFORE*，而不会出现 *AFTER* 的情况。将以上叙述表示为一阶逻辑形式，则有：

$$\mathrm{BEFORE}(e_1, e_2) \leftrightarrow \mathrm{AFTER}(e_2, e_1) \tag{4.10}$$

$$\begin{aligned} &(\mathrm{BEFORE}(e_1, e_2) \wedge \mathrm{BEFORE}(e_2, e_3) \rightarrow \mathrm{BEFORE}(e_1, e_3)) \\ &\wedge(\mathrm{BEFORE}(e_1, e_2) \wedge \mathrm{BEFORE}(e_2, e_3) \rightarrow \neg\mathrm{AFTER}(e_1, e_3)) \end{aligned} \tag{4.11}$$

表 4.1　TB-Dense 时序关系的对称一致性

Tab 4.1　Symmetry consistency in TB-Dense

$\mathrm{R}(e_1, e_2)$	$\overline{\mathrm{R}}(e_1, e_2)$
BEFORE	AFTER
AFTERE	BEFORE
SIMULTANEOUS	SIMULTANEOUS
INCLUDES	IS_INCLUDED
IS_INCLUDED	INCLUDES
VAGUE	VAGUE

以 TB-Dense 中的时序关系为例，表 4.1 中总结出了时序关系与其逆关系所满足的对称一致性，表 4.2 中总结出了时序关系中存在的传递一致性。

表 4.2　TB-Dense 时序关系的传递一致性

Tab 4.2　Transitivity consistency in TB-Dense

$P(e_1,e_2)$	$Q(e_2,e_3)$	$R(e_1,e_3)$
BEFORE	BEFORE	BEFORE
AFTER	AFTER	AFTER
SIMULTANEOUS	SIMULTANEOUS	SIMULTANEOUS
INCLUDES	INCLUDES	INCLUDES
IS_INCLUDED	IS_INCLUDED	IS_INCLUDED
VAGUE	VAGUE	VAGUE
BEFORE	VAGUE	BEFORE, INCLUDES, IS_INCLUDED, VAGUE
BEFORE	INCLUDES	BEFORE, INCLUDES, VAGUE
BEFORE	IS_INCLUDED	BEFORE, IS_INCLUDED, VAGUE
AFTER	VAGUE	AFTER, INCLUDES, IS_INCLUDED, VAGUE
AFTER	INCLUDES	AFTER, INCLUDES, VAGUE
AFTER	IS_INCLUDED	AFTER, IS_INCLUDED, VAGUE
INCLUDES	VAGUE	VAGUE, BEFORE, AFTER, INCLUDES
INCLUDES	BEFORE	BEFORE, INCLUDES, VAGUE
INCLUDES	AFTER	AFTER, INCLUDES, VAGUE
IS_INCLUDED	VAGUE	VAGUE, BEFORE, AFTER, IS_INCLUDED
IS_INCLUDED	BEFORE	BEFORE, IS_INCLUDED, VAGUE
IS_INCLUDED	AFTER	AFTER, IS_INCLUDED, VAGUE
BEFORE	SIMULTANEOUS	BEFORE
AFTER	SIMULTANEOUS	AFTER
INCLUDES	SIMULTANEOUS	INCLUDES
IS_INCLUDED	SIMULTANEOUS	IS_INCLUDED
VAGUE	BEFORE	BEFORE, INCLUDES, VAGUE, IS_INCLUDED
VAGUE	AFTER	AFTER, INCLUDES, IS_INCLUDED, VAGUE
VAGUE	INCLUDES	VAGUE, BEFORE, AFTER, INCLUDES
VAGUE	IS_INCLUDED	VAGUE, BEFORE, AFTER, IS_INCLUDED

这些离散的常识知识，或者被称为约束通常是不可微的，即无法直接融入到深度学习的模型中，所以以往的工作大多使用了 ILP 的方法进行后处理，在预测得到硬标签后通过现成的解决机（off-the-shelf solver）与约束（constraints）来修正。这种依赖于后处理的过程无法在模型训练时，为模型提供任何来源于时序关系的先验常识知识。受 Li[62] 的工作的启发，本研究通过引入三角范数（T-norms）来将这些常识知识的一阶逻辑形式映射到可微的损失函数上。

4.2.5 逻辑驱动训练过程

在逻辑驱动的训练框架中，预测结果在每一个类别标签上的概率值被作为命题布尔结果的软替代。表 4.3 中列举出了乘积三角范数（Product T-norm）下从布尔逻辑到概率可微计算的映射方式。

表 4.3 乘积三角范数的映射规则

Tab 4.3 The rules of mapping statements to differentiable functions in product t-norm

布尔逻辑	概率计算形式
$\neg A$	$1-a$
$A \wedge B$	ab
$A \vee B$	$a+b-ab$
$A \rightarrow B$	$\min(1, b/a)$

对于给定样本空间中的对称一致性，它的一阶逻辑形式通常可以写成：

$$\underset{(e_1,e_2)\in D}{\wedge}(R(e_1,e_2) \leftrightarrow \overline{R}(e_2,e_1)) \tag{4.12}$$

其中 D 为整个数据集，R 和 $\overline{R}$ 分别代表某种时序关系与其逆关系。这个命题的具体含义是，对于数据集中的所有事件对 (e_1,e_2)，如果交换 e_1 与 e_2 的位置，那么事件对则满足其逆关系。使用 Product T-norm 映射之后，我们可以得到：

$$\prod_{(e_1,e_2)\in D} \min(1, \frac{\overline{r}(e_2,e_1)}{r(e_1,e_2)}) \min(1, \frac{r(e_1,e_2)}{\overline{r}(e_2,e_1)}) \tag{4.13}$$

其中 r 与 $\overline{r}$ 分别为某种时序关系与其逆关系对应在预测结果中的概率值。最后，为了避免连乘的计算，将其转换至对数空间上，则有对称一致性损失如下：

$$loss_{sym} = \sum_{(e_1,e_2)\in D} |\log r(e_1,e_2) - \log \overline{r}(e_2,e_1)| \tag{4.14}$$

对于给定样本空间中的传递一致性，它的一阶逻辑形式通常可以写成：

$$\begin{aligned}\bigwedge_{(e_1,e_2),(e_2,e_3),(e_1,e_3)\in D}&((P(e_1,e_2)\wedge Q(e_2,e_3)\rightarrow R(e_1,e_3))\\&\wedge(P(e_1,e_2)\wedge Q(e_2,e_3)\rightarrow\neg\tilde{R}(e_1,e_3)))\end{aligned}\tag{4.15}$$

其中 P 、Q 、R 分别为满足表 4.2 中合取原则的三种时序关系，$\tilde{R}$ 为没有出现在合取结果 R 中的其他时序关系。这个命题的具体含义是，对于数据集中构成传递性关系的三个事件对，它们预测的结果需要满足表 4.2 中的合取原则。同样，使用 Product t-norm 之后，我们可以得到：

$$\prod_{(e_1,e_2),(e_2,e_3),(e_1,e_3)\in D}\min(1,\frac{r(e_1,e_3)}{p(e_1,e_2)q(e_2,e_3)})\min(1,\frac{1-\tilde{r}(e_1,e_3)}{p(e_1,e_2)q(e_2,e_3)})\tag{4.16}$$

其中 p 、q、r、$\tilde{r}$ 分别为以上提及时序关系在预测结果中对应的概率值。同样，为了避免连乘计算，我们也将其转换至对数空间上，则有传递一致性损失如下：

$$\begin{aligned}loss_{trans}=\sum_{(e_1,e_2),(e_2,e_3),(e_1,e_3)\in D}&(\mathrm{ReLU}(\log p(e_1,e_2)+\log q(e_2,e_3)-\log r(e_1,e_3))\\&+\mathrm{ReLU}(\log p(e_1,e_2)+\log q(e_2,e_3)-\log(1-\tilde{r}(e_1,e_3)))\end{aligned}\tag{4.17}$$

值得注意的是，无论是对称性损失还是传递性损失，它们都不需要使用到来源于监督标签的信息，这两个损失关注的是模型在常识性知识方面的表达错误，即识别模型其内在的一致性。

通过上述分析，我们了解了如何将来自于常识知识的一阶逻辑，通过 Product T-norm 转化为可微的损失函数的形式，而添加到我们的深度学习系统中。我们最终的逻辑驱动训练过程可以分为三步：

(1) 在整个语料库上利用给定标签进行监督学习，通过交叉熵损失来优化 DCN 模型。在这个过程中，模型训练时的输入数据流和以往一般的基于深度学习的方法是一致的，模型只使用了来自于标签的监督信息。

(2) 在第一步训练好模型的基础上，加入对称性损失进行训练，这一步的损失函数包含两部分，一部分是有监督的来源于标签的交叉熵损失，另一部分是无监督的来源于常识性知识的对称一致性损失。在这个过程中，模型训练时的输入数据流与以往不同，每一个优化步上需要对输入的事件对进行位置交换。

(3) 在第二步训练好模型的基础上，加入传递性损失进行训练，这一步的损失函数也包含了两部分，一部分是有监督的来源于标签的交叉熵损失，另一部分是无监督的来源于常识性知识的传递一致性损失。在这个过程中，模型训练时的输入数据流也发生不同，需要预先在语料库上枚举满足传递性关系的三个样例，再送入模型训练。

算法 4.1　逻辑驱动训练框架的伪代码描述

Alg 4.1　Pseudocode description of the logic-driven training framework

Algorithm: Logic-driven Training in LDCN Model

Input: Initial model – DCN, Corpus – D, Optimizer – Optim, Reversed Map – RevMap, Conjunctive Map – ConjMap

Output: Trained model – LDCN

foreach example $\{S, R(e_1, e_2)\}$ in D **do**　// First Pass

$R'(e_1, e_2) = \mathrm{DCN}(S, e_1, e_2)$

$SupervisedLoss = \mathrm{CrossEntropy}(R, R')$

$\mathrm{Optim} \leftarrow SupervisedLoss$

foreach example $\{S, R(e_1, e_2)\}$ in D **do**　//Second Pass

$R'(e_1, e_2) = \mathrm{DCN}(S, e_1, e_2)$

$\overline{R}'(e_2, e_1) = \mathrm{DCN}(S, e_2, e_1)$

$SupervisedLoss = \mathrm{CrossEntropy}(R, R') + \mathrm{CrossEntropy}(\overline{R}, \overline{R}')$

$SymLoss = 0$

foreach relation r and reversed relation $\overline{r}$ in RevMap **do**

$SymLoss \ += \ |\log r(e_1, e_2) - \log \overline{r}(e_2, e_1)|$

$\mathrm{Optim} \leftarrow SupervisedLoss + SymLoss$

// Third Pass

foreach example $\{S, P(e_1, e_2)\}$ in D **do**

foreach example $\{S, Q(e_2, e_3)\}$ in D **do**

foreach example $\{S, R(e_1, e_3)\}$ in D **do**

if (e_1, e_2), (e_2, e_3), (e_1, e_3) satisfy the transitive relations **then**

$P'(e_1, e_2) = \mathrm{DCN}(S, e_1, e_2)$, $Q'(e_2, e_3) = \mathrm{DCN}(S, e_2, e_3)$, $R'(e_1, e_3) = \mathrm{DCN}(S, e_1, e_3)$

$SupervisedLoss = \mathrm{CrossEntropy}(P, P') + \mathrm{CrossEntropy}(Q, Q') + \mathrm{CrossEntropy}(R, R')$

$TransLoss = 0$

foreach p, q, r and $\tilde{r}$ in ConjMap **do**

$TransLoss \ += \ \mathrm{ReLU}(\log p + \log q - \log r)$

$TransLoss \ += \ \mathrm{ReLU}(\log p + \log q - \log(1 - \tilde{r}))$

$\mathrm{Optim} \leftarrow SupervisedLoss + TransLoss$

return LDCN

它们的伪代码描述如算法 4.1 中所示。完成第一阶段训练过程的模型即为不带有常识性知识先验的 DCN 模型，完成所有三阶段训练过程的模型即为我们最终的 LDCN 模型。

4.3 事件时序关系识别实验

本节分为两个部分来介绍事件时序关系识别的实验内容。首先为LDCN在TB-Dense语料库上的实验，其中包含了与最先进模型的横向对比和自身的消融对比；然后在第二部分中，介绍了 LDCN 在 MATRES 语料库上的实验过程，其中包含了与最先进模型的横向对比；最后在第三部分中，也对 LDCN 在 TB-Dense 上的结果进行了一定的错误分析与讨论。

4.3.1 TB-Dense 事件时序关系识别实验

（1） 实验设置

TB-Dense 语料库中一共包含了 36 个文档与 6088 个事件对，按照约定，其中 22 个文档与 4032 个事件对作为训练集，5 个文档与 629 个事件对作为开发集，剩余 9 个文档与1427个事件对作为测试集。本小节利用AllenNLP框架构建深度学习模型，利用Optuna作为超参数选择工具，利用 Gurobi[6]作为 ILP 的优化器。

表 4.4 TB-Dense 中各关系实例的数量

Tab 4.4 Count of temporal relation in different type on TB-Dense

	训练集	开发集	测试集
AFTER	674	172	274
BEFORE	808	156	384
SIMULTANEOUS	59	12	22
INCLUDE	206	14	56
IS_INCLUDED	273	21	53
VAGUE	2012	254	638
Overall	4032	629	1427

表 4.4 中列举了 TB-Dense 中各个时序关系对应实例的数量，从表中可以看出，除按照约定对 *SIMULTANEOUS* 类型的实例不予考虑之外，时序关系的分布仍然呈现较不均匀的状态。某几种时序关系的实例占很大比例，而如 *INCLUDE* 与 *IS_INCLUDED* 两种关系无论在训练集中还是测试集中都只有较少实例。

[6] https://www.gurobi.com/

表 4.5　TB-Dense 语料上模型的超参数

Tab 4.5　Params of model on TB-Dense

参数	参数含义	值
Hidden Size	所有隐藏层的维度	590
Dropout Rate	神经元丢弃率	0.24
RNN Layer Num	BI-GRU 层数	1
GCN Layer Num	GCN 层数	2
Word Embedding Size	词嵌入维度	100
POS Embedding Size	词性嵌入维度	50
BERT	预训练 BERT	bert-base-uncased
LR in first pass	一阶段学习率	1E-5
LR in second pass	二阶段学习率	5E-6
LR in third pass	三阶段学习率	3E-5

表 4.5 中列出了 TB-Dense 上模型最终的超参数设置。词嵌入上采用了 GLOVE 词向量；预训练语言模型方面，本研究采用了 bert-base-uncased 的权重，优化器使用了 Adam 优化器。

（2） 对比的模型

在 TB-Dense 的时序关系识别实验中，本研究选取了以下具有代表性的先进时序关系识别系统来进行横向比较：

① CAEVO[50] (2014)：该模型为两阶段方法，一个阶段粗略地提取事件的一些时间属性，第二阶段使用一阶段得到的特征和朴素贝叶斯模型完成时序关系识别。

② LSTM-Dep[52](2017)：该模型中引入了 BI-LSTM 的结构，并且分别从句子序列和事件结点间的依存序列两个角度进行编码，最后将两种编码聚合后进行分类。

③ TEA[53] (2018)：该模型是一种上下文感知的神经网络模型，它由四个组件组成，用于句内实体关系的 LSTM 模型，用于句间关系的 LSTM 的模型，用于文档创建时间的另一个 LSTM 的模型，以及用于 TIMEX 对的基于规则的组件。

④ LSTM-SSVM[56](2019)：该模型将 BI-LSTM 与传统的 SSVM 技术结合，BI-LSTM 用于学习一个关系分类器的打分函数，SSVM 则取代了传统的整数线性规划过程用于全局的最大后验推断。

⑤ SEC[58](2020)：该模型为一个以事件为中心的模型，允许跨多个 TLINKs 管理动态的事件表示，并且以多任务学习的方式处理三个 TLINK 类别的识别，以利用完整的数据集信息。

⑥ DEER[59] (2020)：DEER 借鉴了预训练通用语言模型 BERT 的思路，将其迁移至时序关系识别这一子领域，该语言模型经过在大规模时序关系语料上的预训练，专注于句子中的事件时间关系。

⑦ EventPlus[60] (2021)：EventPlus 在时序关系识别阶段运用了 BERT 嵌入、BI-LSTM 编码与多层感知机分类的方式来识别句子中的时序关系。

（3） 实验结果

表 4.6　TB-Dense 时序关系识别结果 [7]

Tab 4.6　Results of temporal relation identification on TB-Dense

	A	B	S	I	II	V	Overall
CAEVO (2014)	-	-	-	-	-	-	49.4
LSTM-Dep (2017)	44.0	46.0	0	2.5	17.0	62.4	52.9
TEA (2018)	-	-	-	-	-	-	57.0
LSTM-SSVM (2019)	63.5	64.4	0	6.9	10.2	67.8	63.2
SEC (2020)	-	-	-	-	-	-	65.0
DEER (2020)	-	-	-	-	-	-	**66.8**
EventPlus (2021)	-	-	-	-	-	-	64.5
LDCN (ours)	**64.3**	**70.0**	0	**9.1**	**29.7**	**68.6**	65.8

表 4.6 中列举出了本研究提出的 LDCN 模型与当前的最先进的一批模型在 TB-Dense 上的横向对比结果。首先可以观察到的是，除了在大规模时序关系语料上进行预训练的 DEER 模型，LDCN 取得了最高的全局 F1 值，比 SEC 模型高出了 0.8%。这是符合预期的，LDCN 在编码器上使用了获取文本丰富语义有优势的 HDN 模型取代了传统的 BI-LSTM，使得句子中的标注事件更能从上下文中获取对时序分类有利的指示性信息；而且，LDCN 所采用的逻辑训练框架使得模型在训练阶段就能考虑到来源于时序关系本身的常识性知识，从而使得模型在全局上更能获得满足一致性的结果。然后，在表中也能观察到，相较于先进的 LSTM-SSVM 基线，LDCN 在各类别上的 F1 值也提升明显，尤其在 *BEFORE* 类别上高出了 5.6 个百分点，在 *IS_INCLUDED* 类别上高出了 19.5 个百分点，这二者的收益都很大程度上来源于一致性约束的损失的添加，在训练过程中加入对称一致性的常识知识后，*BEFORE* 类别的结果不仅由其本身的标签约束，还受到 *AFTER* 类别的结果影响，同样，*IS_INCLUDED* 类别也可从 *INCLUDE* 类别的结果中受益。最后，尽管 DEER 预训练语言模型比本节提出的 LDCN 模型在 F1 值上高出了 1 个百分

[7] 表中的结果为 F1 值，且 A 代表 *AFTER*，B 代表 *BEFORE*，S 代表 *SIMULTANEOUS*，I 代表 *INCLUDE*，II 代表 *IS_INCLUDED*，V 代表 *VAGUE*

点，但是直接将 DEER 预训练语言模型的结果与其它模型的结果比较是不公平的，DEER 语言模型使用了大规模的时序关系语料来进行预训练，即它等价于使用了额外的数据资源与监督信息，反观列表中的其它模型都只是在 TB-Dense 数据集上进行数据划分、模型训练和最终预测。

表 4.7　LDCN 各阶段的时序关系识别结果

Tab 4.7　Results of temporal relation identification on each pass

	A	B	S	I	II	V	Overall
First Pass	**65.0**	70.3	0	**9.6**	22.4	66.6	64.6
Second Pass	64.1	**70.9**	0	7.9	20.0	68.4	65.4
Third Pass	64.3	70.0	0	9.1	**29.7**	**68.6**	**65.8**

表 4.7 中列举出了 LDCN 在各个阶段的时序识别结果。从表中可以看到，LDCN 在第一阶段的模型性能就已经优于了作为强基线的 EventPlus 模型。其次，*AFTER* 类别与 *INCLUDE* 类别的结果在第一阶段取得了最好性能，*BEFORE* 类别在第二阶段取得了最好性能，而 *IS_INCLUDED* 类别与 *VAGUE* 类别在第三阶段取得了最好的性能。总的来看，从第一阶段到第三阶段的训练过程中存在两个比较明显的拐点，首先，第一阶段到第二阶段的 *BEFORE* 类别取得了明显提升，这种提升得益于对称一致性损失中有关 *AFTER* 与 *BEFORE* 的对称约束；接着，第二阶段到第三阶段的 *IS_INCLUDED* 类别取得了极大的提升，这种提升则来源于对于包含 *IS_INCLUDED* 类别的三元关系的合取约束。

表 4.8　DCN 中张量操作的消融结果

Tab 4.8　Ablation results of tensor operations in DCN

	P(%)	R(%)	F1(%)
DCN	64.2	65.2	64.6
w/o Avg	63.1	64.1	63.6
w/o Concat	63.3	64.3	63.8
w/o Minus	63.8	64.8	64.2
w/o Paired	63.5	64.5	63.9
w/o Times	63.6	64.6	64.0

为了进一步剖析 DCN 中各种张量操作为模型带来的收益，本研究也对 DCN 的各个张量操作进行了消融对比，并且在消融过程中我们仅关心全局 F1 指标的变化。表 4.8 中列出了消融实验的结果，从表中可以看出，首先，各个张量操作都为最终的模型性能带

来了收益，减去 DCN 中的任意一个张量操作，性能都有一定的下降；其次，Avg 操作在其中发挥的作用是最大的，比传统的 Concat 操作带来的收益要大，这也说明以往工作中仅使用 Concat 操作构建事件对特征的方式是不全面的；最后，Minus 操作在这 5 个张量操作中所带来的收益最小，而 Paired 操作与 Times 操作带来的收益也较为显著。

表 4.9　基于逻辑的训练框架与 ILP 的对比

Tab 4.9　Comparison of logic-driven framework and ILP

	P(%)	R(%)	F1(%)
DCN	64.2	65.2	64.6
DCN+ILP	64.5	65.6	65.0
LDCN	65.3	66.3	65.8

表 4.9 中列举出了基于逻辑的训练框架与基于 ILP 的方法之间的对比结果。可以看到，ILP 的确可以在一定程度上修正时序关系识别的结果，DCN+ILP 的结果在 F1 值上比原始的 DCN 结果高出了 0.4%。其次，我们也看到基于逻辑训练框架的 LDCN 模型的结果比 DCN+ILP 的方法高出了 0.8%，这说明在训练过程中引入常识性知识的方法要优于在预测过程中引入常识性知识的方法，同时也说明在对称一致性损失的加入和传递一致性损失的加入下，模型的确可以在训练中运用、融入这些常识性知识以取得更好的性能。

4.3.2　MATRES 事件时序关系识别实验

（1） 实验设置

表 4.10　MATRES 中各关系实例的数量

Tab 4.10　Count of temporal relation in different type on MATRES

	训练集	开发集	测试集
AFTER	4051	430	271
BEFORE	5749	676	427
EQUAL	379	39	30
VAGUE	1265	151	109
Overall	11444	1296	837

MATRES 语料一共包含了 275 个文档与 13577 个事件对，其中 204 个文档与 11444 个事件对作为训练集，51 个文档与 1296 个事件对作为开发集，20 个文档与 837 个事件对作为测试集。表 4.10 中列举了 MATRES 中各个时序关系对应的时序数量，可以看到

相比于 TB-Dense，*VAGUE* 类型的占比明显减少，但是 *EQUAL* 类型的数量仍较少。在本研究中，也遵循之前研究中的约定，计算最终的指标时不统计 *VAGUE* 类型的数量。

表 4.11　MATRES 语料上模型的超参数

Tab 4.11　Params of model on MATRES

参数	参数含义	值
Hidden Size	所有隐藏层的维度	583
Dropout Rate	神经元丢弃率	0.16
RNN Layer Num	BI-GRU 层数	1
GCN Layer Num	GCN 层数	2
Word Embedding Size	词嵌入维度	100
POS Embedding Size	词性嵌入维度	50
BERT	预训练 BERT	bert-base-uncased
LR in first pass	一阶段学习率	1E-5
LR in second pass	二阶段学习率	1.5E-6
LR in third pass	三阶段学习率	2.5E-8

表 4.11 中列出了 MATRES 上模型最终的超参数设置。词嵌入上仍采用了 GLOVE 词向量；预训练语言模型方面，仍采用了 bert-base-uncased 的权重，优化器仍使用了 Adam 优化器。

（2） 对比的模型

在 MATRES 的时序关系识别实验中，本研究也选取了一些具有代表性的先进系统来进行横向比较，以下介绍了在上一小节中未提到的模型：

① AVG-Perceptron[75] (2018)：该模型构建了 n-gram 词性特征、事件对距离特征、情态动词特征、时序连词特征、WordNet 同义词特征等，并使用了平均感知机算法进行时序关系分类。

② CogCompTime[82] (2018)：该模型也使用了感知机算法作为分类方法，构建特征方面也和前一种模型类似，区别在于该模型引入了 ILP 方法对预测的结果进行全局最大后验概率推断。

③ LSTM-CSE[54] (2019)：该模型使用了 LSTM 作为模型编码器，并分别使用了成对分类的识别方法和利用锚点标记事件对的句子级分类方法来识别时序关系。同时，还引入了来源于 TEMPROB 知识库的时序关系常识知识，这里的常识知识指知识库中已经标记了时序关系的事件对。

（3） 实验结果

表 4.12　MATRES 时序关系识别结果

Tab 4.12　Results of temporal relation identification on MATRES

	A	B	E	V	Overall
AVG-Perceptron (2018)	59.0	75.0	0	18.0	69.0
CogCompTime (2018)	-	-	-	-	65.9
LSTM-SSVM (2019)	-	-	-	-	75.5
LSTM-CSE (2019)	-	-	-	-	76.7
DEER (2020)	-	-	-	-	79.3
EventPlus (2021)	-	-	-	-	75.5
LDCN (ours)	**81.3**	**83.5**	0	**22.7**	**81.1**

表 4.12 中列举出了在 MATRES 语料上 LDCN 与其他先进模型的性能比较。从表中可以看出，LDCN 获得了最高的全局 F1 值，除了在大规模语料进行预训练的 DEER 模型，它比之前最好的 LSTM-CSE 模型高出了 4.4 个百分点。尽管在 LSTM-SSVM 与 LSTM-CSE 中没有给出具体每一类时序关系上的 F1 值，但通过对比 LDCN 与 AVG-Perceptron 的结果可以得知，MATRES 语料重点关注的是 *AFTER* 与 *BEFORE* 这两类时序关系上的性能，因而，在这两类上获得了高水平的 F1 值也将获得高水平的全局 F1 值。而 LDCN 在 *AFTER* 类别上比 AVG-Perceptron 高出了 22.3 个百分点，在 *BEFORE* 类别上也高出了 8.5 个百分点，这也充分说明了本研究提出模型在识别这两类时序关系上的有效性。此外，尽管全局的 F1 值不予考虑 *VAGUE* 类别的结果，但我们也发现 LDCN 可以显著地提高在 *VAGUE* 类别上的结果。最后，我们也发现在 MATRES 语料上，LDCN 的最终结果要比大规模语料预训练的 DEER 模型高出 1.8 个百分点，造成这一现象的可能原因主要有两点，第一，MATRES 数据集由于其标注的特性而导致规模比 TB-Dense 大，而在较大规模的数据集上的模型训练可以更不那么依赖于来源于预训练的语言模型的词嵌入；第二，DEER 模型只考虑了来源于语言模型本身的词序列信息，而本研究提出的 LDCN 则在 HDN 编码时考虑了多阶句法信息，这些句法信息可能在 MATRES 上带来的收益较大而使得 LDCN 的结果更优。

表 4.13 中列举出了 LDCN 在各个阶段下模型的识别结果，同样，模型也在第三阶段取得了最好的全局 F1 值。并且，我们也可以看到，在不添加对称一致性损失与传递一致性损失的情况下，模型依然取得了比之前最好模型 LSTM-CSE 更好的结果，在全局 F1 值上高出了 3.5 个百分点，这也再次说明了 HDN 编码器相较于 LSTM 的优越性以及提出的深度对比模块在提取事件对的成对特征上的优越性。而在引入对称性损失的情况下，LDCN 在全局 F1 值上提高了 0.8 个百分点，相应的，在 *AFTER* 类别上提高了 2 个百分点。同时，也发现在 MATRES 数据集中一致性损失的引入为模型所带来提升没有

在 TB-Dense 数据集中带来的提升大，这种现象出现的原因可能是，MATRES 数据集本身的包含的三元关系较少，即在第三阶段的枚举过程中，只有较少的三元关系满足传递性需求，因而导致第三阶段模型训练样本数目缺乏。

表 4.13　LDCN 各阶段的时序关系识别结果

Tab 4.13　Results of temporal relation identification on each pass

	A	B	E	V	Overall
First Pass	79.1	83.5	0	14.0	80.2
Second Pass	81.1	**83.6**	0	22.4	81.0
Third Pass	**81.3**	83.5	0	**22.7**	**81.1**

4.3.3　错误分析与讨论

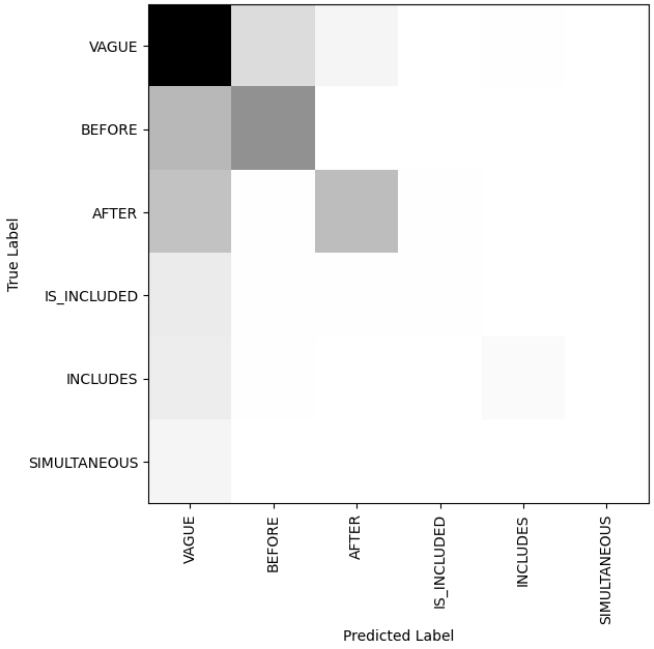

图 4.3　TB-Dense 上结果的混淆矩阵

Fig 4.3　The confusion matrix of results on TB-Dense

本小节对在 TB-Dense 上的实验结果进行了一定的错误分析，图 4.3 中为 TB-Dense 上最终结果的混淆矩阵，其中颜色越深的块表示对应该区域的样本数量越多。可以看到，*BEFORE* 类别与 *AFTER* 类别的假正例与假负例数量较多，一些原本为 *VAGUE* 类别的样本被识别为了 *BEFORE* 和 *AFTER*，也有一些原本为 *BEFORE* 和 *AFTER* 的样本被预测为了 *VAGUE*，反观 *BEFORE* 和 *AFTER* 之间的预测错误较少。这说明，在判断时序关系的是与否的问题上还有改进的空间，比如考虑先进行二分类后进行多分类的两阶段方

法。其次，我们也发现 *IS_INCLUDED* 类别的真正例数量要远小于它的假负例数量，即很多原本为 *IS_INCLUDED* 类别的样本被模型识别为了 *VAGUE* 类别。所以，在未来的工作中，如何解决如 *IS_INCLUDED* 类别的小样本问题也是一个可以着重考虑的问题。

4.4 本章小结

事件时序关系识别任务在信息抽取中是一个不可或缺的任务，它是构建事理图谱的基础，为事件与事件结点之间添加了边的信息。本章的主要贡献如下：

(1) 提出了一个深度对比网络（DCN）来更有效地提取事件对的特征，具体地，DCN 分别从特征可交换与不可交换两个角度设计了一系列的张量操作以对两个事件的特征表示进行深度地对比。

(2) 引入了逻辑驱动的训练框架以解决传统 ILP 过程与训练过程相互割裂的问题，具体地，本文中通过 Product T-norm 将常识性知识的一阶逻辑形式转化为了可微的概率计算形式，加入到损失函数中，使得模型在训练中可以考虑到常识性知识，最终得到逻辑驱动的度对比网络（LDCN）模型。

(3) 本文在两个常用的事件时序关系识别数据集上分别进行了实验，并与其它先进的事件时序关系识别系统进行比较，最终验证了提出的 LDCN 模型的有效性。

结　论

事件及其时序关系的抽取一直都作为信息抽取领域的热点话题而被广大的学者研究。事件抽取致力于从大量的非结构化的自然语言文本中提取事件，挖掘文本中的事件触发词、事件触发词与实体之间的细粒度关系；事件时序关系识别则致力于识别文本中事件之间的时序联系，挖掘事件之间的事理逻辑。

在事件抽取任务上，本研究遵循了触发词抽取、要素抽取的分阶段流程，并且将触发词抽取视为一个序列标注的问题，将要素抽取视为一个成对分类的问题。针对现有的事件抽取系统中存在的句子编码不充分问题，本研究提出了一种层次蒸馏网络（HDN）来进行事件的触发词分类与要素分类，HDN 融合了一个 BI-GRU 模块用于提取句子的序列信息，若干个 GCN 模块用于提取句子的多阶句法信息，同时还基于双向注意力机制设计了一个蒸馏模块来减少层级编码之间的信息冗余。为了验证提出模型的有效性，本研究分别在 ACE 2005 和 MLEE 语料上测试了提出 HDN 的性能。在 ACE 2005 的触发词结果上，HDN 相较于之前最好的模型在 F1 值上提高了 2.7 个百分点，在要素分类的结果上，相较于之前最好的模型在 F1 值上提高了 1.2 个百分点。在 MLEE 最终的事件结果上，相较于之前最好的模型在 F1 值上提高了 3.1 个百分点。这些实验结果都证明了提出模型的有效性。

在事件时序关系识别任务上，本研究也将时序关系分类视为一个成对分类的问题。针对现有系统存在的（1）事件对的特征构造方式单一（2）传统的 ILP 过程与神经网络训练过程割裂等问题，本研究提出了一种逻辑驱动的深度对比网络（LDCN）来进行时序关系的识别。LDCN 采用了第三章中所提出的 HDN 模型作为提取丰富语义的编码器；依据时序关系本身的特点设计了一个深度对比模块来增强事件对的特征表示；最后，还总结归纳了时序关系中的对称性与传递性常识性知识，并引入了逻辑驱动的训练框架来将总结出的常识性知识作为可微模块融合到深度学习系统之中。为了验证提出模型的有效性，本研究分别在 TB-Dense 和 MATRES 语料上测试了提出 LDCN 模型的性能。在 TB-Dense 语料库上，LDCN 模型相较于同一训练基准下的最好模型在全局 F1 值上提高了 0.8 个百分点。在 MATRES 语料库上，LDCN 模型相较于之前最好的模型在全局 F1 值上提高了 1.8 个百分点。这些实验结果也都证明了提出模型的有效性。

综上所述，本研究分别针对现有事件抽取系统或事件时序关系识别系统存在的各自问题，分别提出了有针对性的模型来改进事件抽取或事件时序关系识别的性能。当前，越来越多的学者加入到对事件抽取和对事件时序关系识别的研究当中，也有越来越多的方法被提出来解决某一些问题。而在未来的研究中，还有以下几个方向值得探索：

首先，提高语料库的质量与规模。语料质量与规模问题一直都是事件抽取、事件时序关系中面临的难题，而深度学习系统本身是一种对数据要求较苛刻的方法。提高语料库的质量与规模都有助于更好地使用深度学习技术。

其次，对小样本问题的研究。无论是在事件抽取还是在事件时序关系识别任务中，小样本问题都是一个不可避免的问题，它们可能只是由标注语料本身存在的标注偏差或源文本存在的领域偏差所引入的，但是在实际系统中却不可以被忽视。所以解决小样本问题也对提高系统整体的性能具有正面作用。

最后，对事件与事件时序关系的联合抽取的进一步探索。事件抽取与事件时序关系识别本身就是一个递进的过程，因此，如何有效地联合两个任务也是一个值得探索的问题，比如利用事件中的要素信息辅助时序关系的识别、利用时序关系的信息辅助事件结构的抽取等。

参 考 文 献

[1] CONLON, SUMALI J, ALAN S, et al. Terrorism information extraction from online reports [J]. Journal of Computer Information Systems 55.3 (2015): 20-28.

[2] ATKINSON, MARTIN, et al. Automated event extraction in the domain of border security [C]. International Conference on User Centric Media, pp. 321-326. Springer, Berlin, Heidelberg, 2009.

[3] TANEV, HRISTO, JAKUB P, et al. Real-time news event extraction for global crisis monitoring [C]. International Conference on Application of Natural Language to Information Systems, pp. 207-218. Springer, Berlin, Heidelberg, 2008.

[4] NUIJ W, MILEA V, HOGENBOOM F, et al. An automated framework for incorporating news into stock trading strategies [J]. IEEE Transactions on Knowledge and Data Engineering (2013), 26(4), 823-835.

[5] CAPET, PHILIPPE, THOMAS D, et al. A risk assessment system with automatic extraction of event types [C]. In International Conference on Intelligent Information Processing, pp. 220-229. Springer, Boston, MA, 2008.

[6] OHTA T, PYYSALO S, RAK R, et al. Overview of the Pathway Curation (PC) task of BioNLP shared task 2013 [C]. Proceedings of the BioNLP Shared Task 2013 Workshop. Sofia, Bulgaria: Association for Computational Linguistics, 2013: 67-75.

[7] ANANIADOU S, THOMPSON P, NAWAZ R, et al. Event-based text mining for biology and functional genomics [J]. Briefings in Functional Genomics. 2015, 14(3): 213－230.

[8] DODDINGTON, GEORGE R, ALEXIS M, et al. The automatic content extraction (ace) program-tasks, data, and evaluation [C]. In Lrec, vol. 2, no. 1, pp. 837-840. 2004.

[9] MITAMURA T, LIU Z, HOVV E, et al. Events Detection, Coreference and Sequencing: What's next? Overview of the TAC KBP 2017 Event Track [C]. In TAC, 2017.

[10] KIM J D, OHTA T, PYYSALO S, et al. Overview of BioNLP'09 shared task on event extraction[C]. Proceedings of the workshop on BioNLP shared task. Boulder, Colorado: Association for Computational Linguistics, 2009: 1-9.

[11] VERHAGEN, MARC, et al. SemEval-2010 Task 13: TempEval-2 [C]. Proceedings of the 5th international workshop on semantic evaluation. pp. 57-62. 2010.

[12] RILOFF, ELLEN. Automatically constructing a dictionary for information extraction tasks [C]. In AAAI, vol. 1, no. 1, pp. 2-1. 1993.

[13] COHEN, BRETONNEL K, KARIN V, et al. High-precision biological event extraction with a concept recognizer [C]. In Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, pp. 50-58. 2009.

[14] HUNTER, LAWRENCE, Lu Z, et al. OpenDMAP: an open source, ontology-driven concept analysis engine, with applications to capturing knowledge regarding protein transport, protein interactions and cell-type-specific gene expression [J]. BMC Bioinformatics 9, no. 1 (2008): 1-11.

[15] RILOFF, ELLEN. Automatically generating extraction patterns from untagged text [C]. In Proceedings of the national conference on artificial intelligence, pp. 1044-1049. 1996.

[16] YANGARBER, ROMAN, RALPH G, et al. Automatic acquisition of domain knowledge for information extraction [C]. In COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics. 2000.

[17] AHN, DAVID. The stages of event extraction [C]. In Proceedings of the Workshop on Annotating and Reasoning about Time and Events, pp. 1-8. 2006.

[18] JI H, RALPH G. Refining event extraction through cross-document inference [C]. In Proceedings of ACL-08: Hlt, pp. 254-262. 2008.

[19] LIAO S, RALPH G. Using document level cross-event inference to improve event extraction [C]. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 789-797. 2010.

[20] HONG Y, Zhang J, Ma B, et al. Using cross-entity inference to improve event extraction [C]. In Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, pp. 1127-1136. 2011.

[21] LIAO S, RALPH G. Acquiring topic features to improve event extraction: in pre-selected and balanced collections [C]. In Proceedings of the International Conference Recent Advances in Natural Language Processing 2011, pp. 9-16. 2011.

[22] LI Q, HENG J, LIANG H. Joint event extraction via structured prediction with global features [C]. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 73-82. 2013.

[23] MCCLOSKY D, MIHAI S, Christopher D M. Event extraction as dependency parsing [C]. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 1626-1635. 2011.

[24] NGUYEN T, RALPH G. Event detection and domain adaptation with convolutional neural networks [C]. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pp. 365-371. 2015.

[25] CHEN Y, XU L, LIU K, et al. Event extraction via dynamic multi-pooling convolutional neural networks [C]. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 167-176. 2015.

[26] NGUYEN T, CHO K, RALPH G. Joint event extraction via recurrent neural networks [C]. In Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 300-309. 2016.

[27] FENG X, HUANG L, TANG D, et al. A Language-Independent Neural Network for Event Detection [C]. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 66-71. 2016.

[28] SHA L, QIAN F, CHANG B, et al. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction [C]. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1. 2018.

[29] NGUYEN T, RALPH G. Graph convolutional networks with argument-aware pooling for event detection [C]. In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1. 2018.

[30] LIU X, LUO Z, HUANG H. Jointly multiple events extraction via attention-based graph information aggregation [C]. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 1247-1256. 2018.

[31] YAN H, JIN X, MENG X, et al. Event detection with multi-order graph convolution and aggregated attention [C]. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5770-5774. 2019.

[32] CUI S, YU B, LIU T, et al. Edge-enhanced graph convolution networks for event detection with syntactic relation [C]. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 2329-2339. 2020.

[33] WANG X, WANG Z, HAN X, et al. HMEAE: Hierarchical modular event argument extraction [C]. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 5781-5787. 2019.

[34] 潘璋. 基于注意力机制的事件抽取方法研究[D]. 大连：大连理工大学，2020.

[35] WANG J, LI H, AN Y, et al. Biomedical event trigger detection based on convolutional neural network [J]. Data Mining in Bioinformatics, 2016, 15(3): 195-213.

[36] 秦美越. 基于并行多池化 CNN 的生物医学事件抽取[D]. 大连：大连理工大学，2017.
[37] 李虹磊. 基于语义空间和神经网络的生物医学事件抽取[D]. 大连：大连理工大学, 2017.
[38] RAHUL P, SAHU S, ANAND A, et al. Biomedical event trigger identification using bidirectional recurrent neural network based models[C]. In BioNLP 2017, pp. 316-321. 2017.
[39] 刘阳. 基于深度学习的生物医学事件抽取[D]. 大连：大连理工大学，2019.
[40] LI L, HUANG M, LIU Y, et al. Contextual Label Sensitive Gated Network for Biomedical Event Trigger Extraction [J]. Journal of Biomedical Informatics, 2019: 103221.
[41] ZHANG J, LIU M, ZHANG Y. Topic-informed neural approach for biomedical event extraction [J]. Artificial Intelligence in Medicine 103 (2020): 101783.
[42] ALLEN J. Maintaining knowledge about temporal intervals [J]. Communications of the ACM, vol. 26, no. 11, pp. 832-843, 1983.
[43] DOWTY D R. The effects of aspectual class on the temporal structure of discourse: semantics or pragmatics? [J]. Linguistics and Philosophy 9, no. 1 (1986): 37-61.
[44] SONG F, COHEN R. Tense Interpretation in the Context of Narrative [C]. In AAAI, vol. 91, pp. 131-136. 1991.
[45] PUSTEJOVSKY J, CASTANO J, INGRIA R, et al. TimeML: Robust specification of event and temporal expressions in text [J]. New Directions in Question Answering 3 (2003): 28-34.
[46] MANI I, VERHAGEN M, WELLNER B, et al. Machine learning of temporal relations [C]. In Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 753-760. 2006.
[47] CHAMBERS N, WANG S, JURAFSKY D. Classifying temporal relations between events [C]. In Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions, pp. 173-176. 2007.
[48] BETHARD S. ClearTK-TimeML: A minimalist approach to TempEval 2013 [C]. In Second joint conference on lexical and computational semantics (* SEM), volume 2: proceedings of the seventh international workshop on semantic evaluation (SemEval 2013), pp. 10-14. 2013.
[49] LAOKULRAT N, MIWA M, TSURUOKA Y, et al. Uttime: Temporal relation classification using deep syntactic features [C]. In Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pp. 88-92. 2013.

[50] CHAMBERS N, CASSIDY T, MCDOWELL B, et al. Dense event ordering with a multi-pass architecture [J]. Transactions of the Association for Computational Linguistics 2 (2014): 273-284.

[51] NING Q, FENG Z, WU H, et al. Joint Reasoning for Temporal and Causal Relations. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2278-2288. 2018.

[52] CHENG F, MIYAO Y. Classifying temporal relations by bidirectional lstm over dependency paths [C]. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 1-6. 2017.

[53] MENG Y, RUMSHISKY A. Context-aware neural model for temporal information extraction [C]. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 527-536. 2018.

[54] NING Q, SURBAMANIAN S, Roth D. An Improved Neural Baseline for Temporal Relation Extraction [C]. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6204-6210. 2019.

[55] HAN R, HSU I, YANG M, et al. Deep Structured Neural Network for Event Temporal Relation Extraction [C]. In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), pp. 666-106. 2019.

[56] HAN R, Ning Q, PENG N. Joint Event and Temporal Relation Extraction with Shared Representations and Structured Prediction [C]. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 434-444. 2019.

[57] 戴倩雯. 事件时序关系识别关键技术研究[D]. 苏州：苏州大学, 2020.

[58] CHENG F, ASAHARA M, KOBAYASHI I, et al. Dynamically Updating Event Representations for Temporal Relation Classification with Multi-category Learning [C]. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings, pp. 1352-1357. 2020.

[59] HAN R, REN X, PENG N. DEER: A Data Efficient Language Model for Event Temporal Reasoning [EB/OL]. arXiv preprint arXiv:2012.15283 (2020).

[60] MA M, SUN J, YANG M, et al. EventPlus: A Temporal Event Understanding Pipeline [EB/OL]. arXiv preprint arXiv:2101.04922 (2021).

[61] LI T, SRIKUMAR V. Augmenting Neural Networks with First-order Logic [C]. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 292-302. 2019.

[62] LI T, GUPTA V, MEHTA M, et al. A logic-driven framework for consistency of neural models [C]. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 3924–3935. 2019.

[63] HOCHREITER S, SCHMIDHUBER J. Long short-term memory [J]. Neural Computation 9, no. 8 (1997): 1735-1780.

[64] CHO K, MERRIENBOER B, GULCEHRE C, et al. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation [C]. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724-1734. 2014.

[65] BAHDANAU D, CHO K, BENGIO Y. Neural Machine Translation by Jointly Learning to Align and Translate[C]. Proceedings of the Conference on International Conference on Learning Representations, 2015.

[66] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is All you Need [J]. Advances in Neural Information Processing Systems 30 (2017): 5998-6008.

[67] PEROZZI B, Al-RFOU R, SKIENA S, et al. Deepwalk: Online learning of social representations [C]. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 701-710. 2014.

[68] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality [C]. Proceedings of Advances in Neural Information Processing Systems 26 (NIPS 2013).

[69] PETERS M, NEUMANN M, IYYER M, et al. Deep Contextualized Word Representations [C]. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227-2237. 2018.

[70] RADFORD A, NARASIMHAN K, SALIMANS T, et al. Improving language understanding by generative pre-training [EB/OL]. OpenAI BLOG (2018).

[71] DEVLIN J, CHANG M, LEE K, et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [C]. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 4171-4186. 2019.

[72] PYYSALO S, OHTA T, MIWA M, et al. Event extraction across multiple levels of biological organization[J]. Bioinformatics. 2012, 28(18): i575–i581.

[73] BJORNE J, SALAKOSKI T. TEES 2.1: Automated annotation scheme learning in the BioNLP 2013 Shared Task [C]. In Proceedings of the BioNLP shared task 2013 workshop (pp. 16-25).

[74] CASSIDY T, MCDOWELL B, CHAMBERS N, et al. An Annotation Framework for Dense Event Ordering [C]. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 501-506. 2014.

[75] NING Q, WU H, ROTH D. A Multi-Axis Annotation Scheme for Event Temporal Relations [C]. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1318-1328. 2018.

[76] PUSTEJOVSKY J, HANKS P, SAURI R, et al. The timebank corpus [J]. In Corpus linguistics, vol. 2003, p. 40. 2003.

[77] GARDNER M, GRUS J, NEUMANN M, et al. AllenNLP: A Deep Semantic Natural Language Processing Platform [C]. In Proceedings of Workshop for NLP Open Source Software (NLP-OSS), pp. 1-6. 2018.

[78] CHEN Y, YANG H, LIU K, et al. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms [C]. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 1267-1276. 2018.

[79] KINGMA D, BA J. Adam: A method for stochastic optimization [C]. In 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

[80] HE X, LI L, SONG X, et al. Multi-level attention based BLSTM neural network for biomedical event extraction [J]. IEICE TRANSACTIONS on Information and Systems 102, no. 9 (2019): 1842-1850.

[81] ZHOU D, ZHONG D. A semi-supervised learning framework for biomedical event extraction based on hidden topics [C]. Artificial intelligence in medicine 64, no. 1 (2015): 51-58.

[82] NING Q, ZHOU B, Feng Z, et al. CogCompTime: A tool for understanding time in natural language [C]. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 72-77. 2018.

攻读硕士学位期间发表学术论文情况

1　Contextual label sensitive gated network for biomedical event trigger extraction. Lishuang Li, **Mengzuo Huang**, Yang Liu, Shuang Qian, Xinyu He. Journal of biomedical informatics, 2019: 103221. 主办单位：Elsevier。SCI 检索期刊，SCI 检索号：000525695600008

2　Hierarchical Distillation Network for Biomedical Event Extraction. Lishuang Li, **Mengzuo Huang**, Beibei Zhang. In 2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 287-292), Regular paper. 主办单位：IEEE。EI 检索、CCF 推荐 B 类会议，EI 检索号：20210609885520（本硕士学位论文第三章）

致　　谢

转眼三年的硕士求学生涯即将结束，我的滨城大连之旅也即将告一段落，不免思绪万千，有窃喜，有伤感，亦有不舍。我从来都不是一个擅长表达情感的人，但行文至此，我想真诚地对所有这三年来帮助过我的人说一声谢谢。

我感谢李丽双老师三年来对我的指导与帮助。李老师对大家有花不完的细心与耐心，起初的我，对于要研究的内容都是一头雾水，正是有李老师的细心解答与耐心的教导才使得我在入门时少走了很多弯路。李老师对我也非常包容，在选择研究方向上给了我很大的自主空间，对各个研究方向的调研也使得我的视野得到了开阔。李老师对大家也非常负责，每周都组织大家开会，既增进了大家的对于新文献的把握，也锻炼了大家的语言表达能力。论文写作上，李老师对我严格要求，也正是这份严格，使我在之后的写作过程中避免了很多逻辑上和表达上的错误。所以在此，我要对李老师由衷地说一声谢谢。

我感谢我的父母，在每一个迷茫的人生岔路口，他们都选择了支持与信任。高中毕业，选择随遇而安的我没有被责备，只有道不尽的鼓励；大学毕业，选择得过且过的我也没有被责备，只有关心与包容；乃至选择考研，等待着匪夷所思的、破罐子破摔的我的依然不是责备，而是打气与相信，大幸莫过于此。

我还要感谢实验室的大伙三年来对我的帮助。感谢野哥和做成，是他们带初来乍到的我熟悉东北的风俗习惯，感受东北的风土人情。感谢老贝、哈希和阳哥，他们在学习上和论文上都给予了我很大的帮助。感谢晓哥、辉哥和泰森，他们总能在大家被学习的枯燥所充斥时为实验室带来欢笑。感谢世一师兄、陆师姐和何师姐，在我刚进入实验室时提供了很多帮助。感谢钱爽师姐和郭元凯师兄，在我找工作的时候给予了很多建议。感谢连师妹、蒋助、泽昊和姜媛，给我研三的紧张生活增添了乐趣。当然，也还要感谢实验室其他的师兄师姐师弟师妹，是他们在日常的实验室生活中包容了我的种种缺点，并且营造了良好的学习氛围。

三年说长不长，但却遇上了形形色色的人，遇上了许许多多的事，这一切都是我人生中最宝贵的财富，最后，感谢经历。

本文的所有研究工作，来自于国家自然科学基金项目 No. 61672126, No. 62076048、大连市科技创新基金 2020JJ26GX035 的资助。

大连理工大学学位论文版权使用授权书

本人完全了解学校有关学位论文知识产权的规定，在校攻读学位期间论文工作的知识产权属于大连理工大学，允许论文被查阅和借阅。学校有权保留论文并向国家有关部门或机构送交论文的复印件和电子版，可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印、或扫描等复制手段保存和汇编本学位论文。

学位论文题目：基于深度学习的事件及其时序关系抽取的研究

作者签名：黄梦佐　　日期：2021 年 6 月 13 日

导师签名：李丽双　　日期：2021 年 6 月 13 日

大连理工大学学位论文独创性声明

作者郑重声明：所呈交的学位论文，是本人在导师的指导下进行研究工作所取得的成果。尽我所知，除文中已经注明引用内容和致谢的地方外，本论文不包含其他个人或集体已经发表的研究成果，也不包含其他已申请学位或其他用途使用过的成果。与我一同工作的同志对本研究所做的贡献均已在论文中做了明确的说明并表示了谢意。

若有不实之处，本人愿意承担相关法律责任。

学位论文题目：基于深度学习的事件及其时序关系抽取的研究

作 者 签 名：黄梦佐　　日期：2021 年 6 月 13 日