

基于多标签分类和多任务学习的事件因果关系抽取方法

姜晓健¹, 徐洁馨¹, 李秋霞¹, 李怀俊¹

¹招商银行股份有限公司, 广东深圳, 518038
{jiangxiaojian, jhexinx, annielqx, lihuaijun}@cmbchina.com

摘要. 事件抽取是舆情监控和金融领域的重要任务之一, 是进行图谱推理、事件分析的必要过程。CCKS2021提出了面向金融领域篇章级事件因果关系抽取任务。针对该任务, 本文提出了一种基于多标签分类和多任务学习的事件因果关系抽取方法。该方法采用“原因事件判别-结果事件判别-因果事件要素抽取”的三段式流水线模式, 在因果事件判别中采用基于PLM的多标签分类模型, 在要素抽取中采用基于多要素联合抽取和多任务学习的阅读理解模型。实验证明, 我们的方法在最终测试集达到了0.6102的F1得分, 并取得了初赛第一, 决赛第二的优异成绩。

关键词: 多标签分类, 多要素抽取, 多任务学习.

1 引言

事件抽取是舆情监控和金融领域的重要任务之一。“金融事件”在金融领域是投资分析、资产管理的重要决策参考。事件也是知识图谱的重要组成部分, 事件抽取是进行图谱推理、事件分析的必要过程。

为了推动金融领域事件抽取的研究, 第十五届全国知识图谱与语义计算大会(CCKS 2021)设置了面向金融领域的篇章级事件因果关系抽取评测任务。该任务的目标是解决篇章级事件因果关系抽取这个核心的知识抽取问题。本次评测任务的文本语料来自于互联网上的公开新闻、报告。主办方给定一段描述因果或影响关系的文本, 参赛者需要从文本中抽取原因事件的表示和结果事件的表示, 其中事件的表示如表1所示, 包括事件类型和事件的三个要素: 影响地域、产品、行业。

表1: 因果事件表示

事件要素	事件要素描述
事件类型	原因事件有39个事件类型, 结果事件有19个事件类型
影响地域	事件影响的地域, 文本中完整的地域描述
产品	事件影响的产品。产品与行业的区别是产品的范围小, 通常只是一类具体的产品
行业	事件影响的行业, 行业与产品的区别是行业的范围大, 通常是多类产品组成一个行业

本次评测具有以下三个难点：

(1) **文本过长**：该任务属于篇章级事件抽取任务，文本平均长度为104，最大长度为4686，相较句子级事件抽取，篇章级事件抽取存在文本较长、句式复杂、语义难以准确理解等难点。

(2) **事件类型较多**：该任务包含39个原因事件类型和19个结果事件类型，因果对的组合种类繁多，且部分事件类型相似，模型既需要对相似事件进行有效地区分，又需要对因果事件进行正确的配对，难度较大。

(3) **事件要素取值为多个**：该任务每个事件要素的取值可能有零个或多个，参赛者需要识别出全部的取值，具有较高的难度。

为了解决这些难点，本文提出了一种基于多标签分类和多任务学习的篇章级事件因果关系抽取方法。具体而言，本文采用“原因事件判别-结果事件判别-因果事件要素抽取”的三段式流水线方案来解决事件因果关系抽取问题。该方案首先采用基于PLM（预训练语言模型）的多标签分类模型学习长文本语义知识，从而准确地识别文本中的多个因事件；然后将识别出的因事件作为先验知识，采用基于PLM的多标签分类模型识别每个因事件所对应的多个果事件；最后采用基于多要素联合抽取和多任务学习的阅读理解模型抽取因果事件中的地域、行业和产品三个要素。本文提出的方法具有以下三个创新点：

(1) **流水线抽取**：相比因果事件的联合抽取方法，采用流水线方法依次对因果事件进行“一对多”的抽取，一方面可以解决文本较长造成的歧义问题，另一方面可以很好地解决多因果事件配对的难题，同时，该方式允许对每一步进行模型融合，提高了因果事件判别准确率。

(2) **Circle Loss损失函数**：相比BCE Loss，Circle Loss可以在不需要调节阈值、权重等参数下较快地收敛到最好的效果。

(3) **多要素+多任务**：在因果事件要素抽取中，采用多要素联合抽取和多任务学习的阅读理解模型，使模型更准确地抽取每种要素的多个取值。

2 相关工作

近年来，事件抽取已经成为一大研究热点。事件抽取相关的研究工作大致可以分为两类：流水线抽取（Pipeline）和联合抽取（Joint）。

流水线抽取分为两步：事件类型判断和事件要素抽取。传统的流水线方法大多是基于统计模型的。Naughton等人[1]首次采用基于TF-IDF的聚类方法进行事件抽取；David Ahn等人[2]采用MegaM和Timbl机器学习方法来识别事件类型，达到了较好的识别效果。随后，Llorens等人[3]在语义角色注释中引入了条件随机场，提升了抽取准确度。随着深度学习的发展，Chen等人[4]和Nguyen等人[5]首次将神经网络方法应用到事件类型判断中，初步验证了神经网络的有效性。Feng等人[6]采用基于RNN的模型来获取文本中的序列信息，并采用卷积层来获取文本中的短语块信息，将两种信息合并后进行事件识别。Wadden等人[7]采用BERT预训练模型进行多句跨度的文本的语义表示，较大地提升了抽取效果。Zhou等人[8]采用对偶学习的方式，将事件要素抽取转化为两个对偶问题进行共同学习，提升了要素抽取的效果。

联合抽取则是对事件类型判别和事件要素抽取两个任务进行共同学习。在基于传统特征的事件抽取方法中，Li[9]等人首次通过结构化感知机模型进行两个任务的联合学习。Sha等人[10]提出一种基于RNN的模型进行事件识别和要素抽取的联合学习，并利用树结构和序列机构捕捉句子的依存信息，取得了较好的效果。Nguyen等人[11]提出了一个基于共享的隐层表示的新的模型来联合执行事件类型判别和事件要素抽取两个任务。Zhang等人[12]提出了一个Transition-based的神经网络结构，以State-transition的过程，递进地预测复杂的联合学习结构，实现了更好的信息组合。

3 方法

本章节介绍我们的方案，包括数据分析、方案整体框架和评测提分点。

3.1 数据分析

本任务数据集包含训练集共7000条样本，验证集共1000条样本（A榜）和测试集共69653条样本（B榜）。我们对训练集进行了统计：文本长度分布如图1所示（为便于展示，长度大于1000的设置1000），平均长度为104，最小长度为8，最大长度为4686；每条样本所包含的原因事件数量分布如图2所示；每个原因事件所对应的结果事件数量分布如图3所示；每个事件要素所对应的取值数量分布如图4所示。

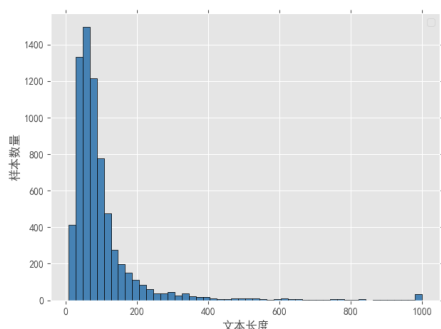


图1：文本长度分布

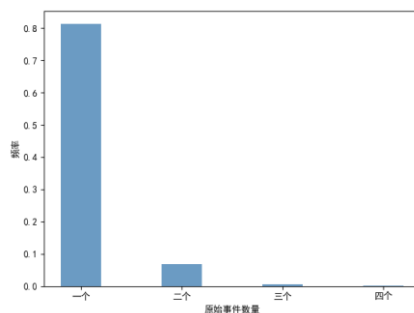


图2：原因事件数量分布

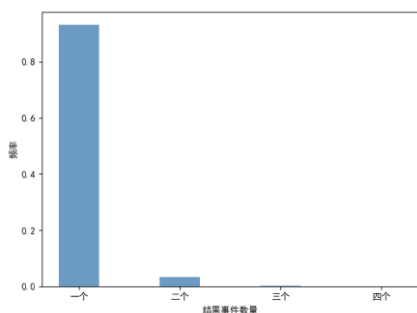


图3：结果事件数量分布

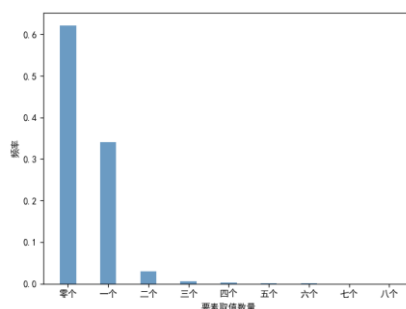


图4：要素取值数量分布

通过上图的数据分析，我们发现：一条原始文本对应多个原因事件类型，一个原因事件类型对应多个结果事件类型，同时，每个事件要素含有多种取值。在这种情况下，采用端到端的事件抽取会出现因果事件配对紊乱、事件要素重叠等问题。因此，本文提出了流水线方法来解决上述问题。

3.2 方案整体框架

本文采用“原因事件判别-结果事件判别-因果事件要素抽取”的流水线方法来完成事件因果关系抽取任务。具体分为三步：

(1) 原因事件判别：基于原始文本，采用预训练模型进行多标签原因事件分类，判断每条文本包含的所有原因事件类型；

(2) 结果事件判别：基于原始文本和(1)中抽取出的原因事件类型，采用预训练模型进行多标签结果事件分类，判断每条文本中每个原因事件对应的所有结果事件类型；

(3) 因果事件要素抽取：将(1)中的原因事件和(2)中的结果事件组成因果事件对，采用阅读理解模型抽取事件的地域、产品和行业三个要素。

3.2.1 原因事件判别

原因事件判别属于多标签分类任务，传统的多标签分类任务大多采用基于BERT-CLS + BCE Loss的模型，然而该模型存在两个缺点：(1) 单纯采用BERT-CLS向量无法准确捕获长文本的语义；(2) BCE Loss在正负标签稀疏时收敛困难，且需要频繁调整阈值和权重才可以达到较好的效果。

针对以上两个缺点，我们在语义建模阶段引入了Max-Pooling操作，对长文本进行更准确的语义表示。同时，我们使用Circle Loss[13]替换BCE Loss，从而提升了模型的学习效果。本文提出的原因事件判别模型的结构如图5所示，我们首先采用预训练语言模型(Pretrained Language Model, 以下简称PLM)获得对原始文本的向量表示，然后对PLM的最后一层输出进行最大池化(Max-Pooling)操作，将池化后的向量与CLS的输出向量进行组合，通过全连接层和Circle Loss进行多标签分类，完成多原因事件类型的判别。

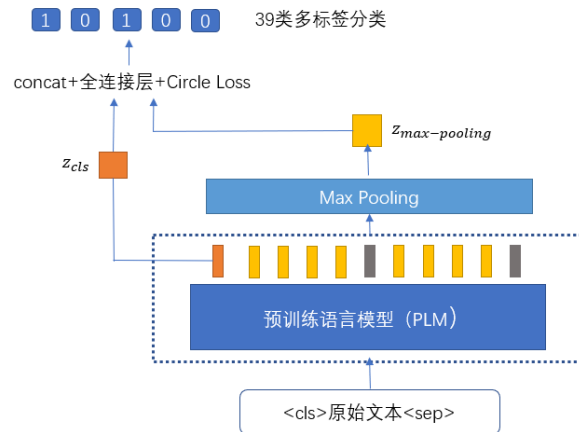


图5: 原因事件判别模型

文本的特征表示 z_{text} 为:

$$z_{max-pooling} = \text{MaxPooling}(H) \quad (1)$$

$$z_{cls} = \tanh(W h_{cls} + b) \quad (2)$$

$$z_{text} = z_{max-pooling} \oplus z_{cls} \quad (3)$$

其中 H 为PLM最后一层的输出向量; h_{cls} 为 H 中<CLS>对应的最后一层输出; $z_{max-pooling}$ 为最大池化后的向量; z_{cls} 为PLM在预训练阶段的文本表示向量; \oplus 表示向量拼接。

然后, 我们将 z_{text} 后接全连接层进行多标签分类, $z_{fcn} \in R^{39 \times 1}$ 表示在39个类别上的输出值。

$$z_{fcn} = W_{fcn} z_{text} + b_{fcn} \quad (4)$$

最后, 我们采用Circle Loss作为损失函数:

$$\text{Loss} = \log(1 + \sum_{i \in \Omega_{neg}} e^{z_{fcn}^i}) + \log(1 + \sum_{j \in \Omega_{pos}} e^{-z_{fcn}^j}) \quad (5)$$

其中, pos 和 neg 表示正负标签集合。在预测阶段, 我们选择 $z_{fcn}^i > 0$ ($0 \leq i < 39$)的类别作为原因事件类型。

3.2.2 结果事件判别

结果事件判别模型的结构如图6所示。该结构与原因事件判别模型是类似的, 两者都属于基于PLM的多标签分类模型。不同的是, 在判断结果的时候, 我们可以利用上一步抽取出的原因事件作为先验知识, 加到PLM的输入中, 提升结果事件判别的准确率。

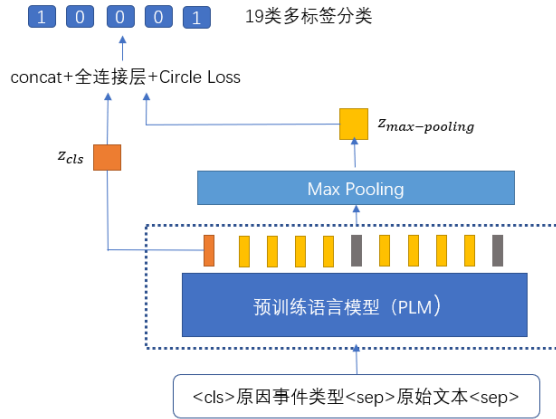


图6: 结果事件判别模型

在结果事件判别模型中, 输出为19类的多标签分类。其余公式细节与3.2.1中的原因事件判别模型一致。

3.2.3 因果事件要素抽取

基于3.2.1和3.2.2节判别出的因果事件类型, 我们可以构造因果事件对, 并采用事件要素抽取模型抽取事件的地域、产品和行业三个要素。通常而言, 事件要素抽取任务可以转化为机器阅读理解任务[14]。其中, 阅读理解任务中的问题由事件要素和相关疑问词构成, 答案表示对应要素的取值。然而, 传统的

机器阅读理解研究大多集中于单答案片段抽取[15]，即每个问题的答案只有零个或一个。本次评测的事件要素取值含有零个或多个，属于多答案片段抽取。针对多片段抽取任务，我们扩展了Hu等人[16]和Efrat等人[17]在机器阅读理解方面的工作，提出了一种基于多要素联合抽取和多任务学习的事件要素抽取模型。

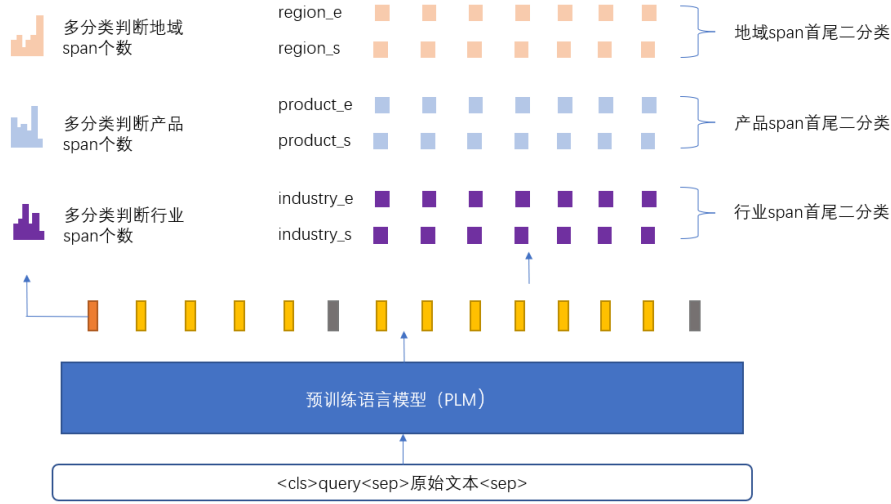


图7：因果事件要素抽取模型

如图7所示，本文提出的事件要素抽取模型包含多要素联合抽取和多任务学习两个模块。在多要素联合抽取模块，我们让地域、产品和行业三个要素的抽取共享底层的向量表示，这样可以更容易捕捉到三种要素间的关联关系。在多任务学习模块，我们将要素片段的开始、结束位置预测以及要素片段的个数预测两个任务进行联合学习，提高模型的收敛速度和最终效果。

我们首先将Query和原始文本拼接作为PLM模型的输入。Query的形式有两种：

- (1) 原因事件类型为xx的行业、产品、地域是什么？
 - (2) 结果事件类型为xx的行业、产品、地域是什么？
- (xx表示前一步预测出的原因或结果事件类型)。

对于一条原始文本的一个因果对，我们可以构造出两条训练样本。模型的输出包含两部分：

(1) **多要素联合抽取：**地域、产品和行业三个要素的抽取共享底层PLM的CLS向量。 softmax 损失函数对每个要素进行多分类，判断每个要素包含的片段个数：

$$p^k = \text{softmax}(W^k h_{\text{cls}} + b^k) \quad (6)$$

$$\text{loss}^k = \sum_{c=0}^G y_c^k \log(p_c^k) \quad (7)$$

其中， $k \in \{\text{industry}, \text{product}, \text{region}\}$ ， G 表示类被个数，实验中 $G = 8$ ；

(2) **多任务学习：**根据PLM最后一层的输出分别构造地域、产品和行业三个要素片段的首尾二分类器：

$$\text{loss}(\text{span}_s^k) = \text{circle_loss}(w_s^k H_{\text{span}} + b_s^k) \quad (8)$$

$$\text{loss}(\text{span}_e^k) = \text{circle_loss}(w_e^k H_{\text{span}} + b_e^k) \quad (9)$$

其中, $k \in \{\text{industry}, \text{product}, \text{region}\}$, H_{span} 表示PLM最后一层的输出, circle_loss 表示Circle Loss损失函数; $\text{loss}(\text{span}_s^k)$ 表示第 k 个span开始位置的损失函数, $\text{loss}(\text{span}_e^k)$ 表示第 k 个span结束位置的损失函数。最终的损失函数为:

$$\text{Loss} = \lambda \sum_{k=1}^3 \text{loss}^k + \mu \sum_{k=1}^3 \text{loss}(\text{span}_s^k) + \text{loss}(\text{span}_e^k) \quad (10)$$

实验中, 我们设置 $\lambda = 3$, $\mu = 1$ 。在预测的时候, 我们首先预测span的个数 n , 然后, 基于启发式的规则, 选择得分高的 n 个span作为最终的结果。每个span得分与起始和结束的预测概率成正比, 与长度成反比。以 i 起始, j 结束的span $_i^j$ 的得分计算公式如公式所示, 实验中设置 $\eta = 3$:

$$\text{score}(\text{span}_i^j) = \text{logits}_s(\text{span}_i) + \text{logits}_e(\text{span}_j) - \eta * (j - i) \quad (11)$$

其中, logits_s 和 logits_e 分别表示预测的起始概率和结束概率。

3.3 评测提分点

本次评测比赛我们采用了一些提分点, 主要包含: PLM模型继续预训练, K-Fold模型融合, 增加单原因事件判别模型, 数据反标注和规则后处理。

PLM模型继续预训练: 本文采用Roberta-wwm-ext-large[18]模型作为PLM, Gururangan等人[19]指出在特定数据集上进行领域自适应训练可以显著提高模型的理解能力。我们对Roberta-wwm-ext-large模型在给定的因果数据集上继续预训练两轮, 使其学习到该数据集的语义知识。

K-Fold模型融合: 我们在原因事件判别、结果事件判别、事件要素抽取三个模块均采用了K-Fold模型融合策略, 实验表明, 多模型融合可以显著提升最终结果。

增加单原因事件判别模型: 由于单纯使用多原因事件判别模型, 可能会存在有的样本预测全部为空的情况, 为此, 我们增加了单原因事件判别模型, 模型整体结构与图5一致, 损失函数为 softmax 。单原因事件判别模型可以至少保证有一个原因事件的输出。实验发现有一定效果。

数据反标注: 由于该任务事件要素并未给定span的起始和结束位置, 因此对于同一要素, 原文中可能会出现多个起止位置。我们尝试了三种反标注方案: (1) 标记要素的首次出现位置; (2) 标注要素最后一次出现位置; (3) 标注要素所有出现的位置, 实验结果表明采用(1)方案在初赛和决赛均取得了较好的分数。

规则后处理: 截掉过长的要素取值、去除重复预测结果等。

4 实验

本章节介绍我们的实验, 包括实验参数设置和实验结果分析。

4.1 实验参数设置

对于多标签分类模型, 我们采用继续预训练后的Roberta-wwm-ext-large模型作为初始预训练模型, 对数据集采用10折交叉验证切分, Batch大小为4, 预训练模型的最大编码长度为400。我们使用Adam优化算法, 权重衰减系数设置为

0.01，学习率设置为 $2e-5$ 。在原因事件预测时，我们对10个单原因事件判别模型和10个多原因事件判别模型的预测结果进行投票，选择出现次数大于8的事件作为最终的原因事件。在结果事件预测时，我们对10个多结果事件判别模型的预测结果进行投票，选择出现次数大于7的因果事件对作为最终事件抽取的结果。

对于事件要素抽取模型，我们采用原始的Roberta-wwm-ext-large模型作为初始预训练模型，对数据集采用10折交叉验证切分，Batch大小为8，问题的最大长度为30，答案的最大个数为8，文本最大长度为512，滑动窗口大小为128。我们使用Adam优化算法，学习率设置为 $2e-5$ 。在事件要素预测时，我们对10个模型的预测结果进行投票，选择出现次数大于5的要素作为最终结果。

本文的所有实验均是在NVIDIA-Tesla-V100的GPU上进行。

4.2 实验结果及分析

针对因果事件判别和事件要素抽取，我们分别设计了多个实验来验证本文提出的方案的有效性。由于本次任务我们无法得知A榜和B榜数据集的真实标签，因此，我们将原始的7000条数据集拆分成6500条训练集，500条测试集，并以此进行实验验证。在进行因果事件判别实验时，我们采用预测事件对与真实事件对的Macro-F1值作为评价指标，在进行事件要素抽取实验时，我们固定最好的因果事件判别模型，并将最终预测结果与真实结果的Macro-F1值作为评价指标。

4.2.1 因果事件判别验证

表2: 不同PLM模型对因果事件判别的影响

PLM模型	Macro-F1
Bert-base	0.726
Bert-large	0.751
Roberta-wwm-ext	0.738
Roberta-wwm-ext-large	0.759

表2展示了使用不同的PLM模型对因果事件判别的影响。由表可知，采用Roberta-wwm-ext-large模型可以取得较好的实验效果。

表3: 损失函数和模型结构对因果事件判别的影响

	BCE Loss	Circle Loss
BiLSTM	0.736	0.755
Max-Pooling	0.744	0.759

表3展示了使用不同损失函数和模型结构对因果事件判别的影响。由表可知，相比BCE Loss，Circle Loss可以显著提升模型的收敛效果。同时，Max-Pooling效果要略好于BiLSTM，我们猜测可能由于Transformer本身已经较好地捕捉到了文本的序列化信息，Max-pooling可以将捕捉到的显著信息更好地提取出来。

表4: 不同融合策略对因果事件判别的影响

模型融合策略	Macro-F1
I. 10个多原因+10个多结果	0.744
II. 10个单原因+10个多原因+10个多结果	0.759
III. 10个多原因+10个单结果+10个多结果	0.746
IV. 10个单原因+10个多原因+10个单结果+10个多结果	0.760

表4展示了不同模型融合策略对因果事件判别的影响。由表可知，增加单原因和单结果的融合可以保证每条样本至少输出一个因果对，不会出现空的情况，因此，确实会提升事件判别的准确率。对比方案II.和方案IV.，我们发现增加单结果事件判别模型对最终效果的提升有限，但却会较大增加模型预测的时间，因此我们最终选择了方案II.作为最终融合策略。

4.2.2 事件要素抽取验证

表5: 不同事件要素抽取模型对最终结果的影响

	单要素分别抽取	多要素联合抽取
单任务学习	0.5892	0.6031
多任务学习	0.6276	0.6365

表5展示了不同事件抽取模型对最终结果的影响。其中，单要素分别抽取表示一次只抽取一个要素，单任务学习表示去掉要素取值个数的判别，只预测起始和结束位置。由表可知，相比单要素分别抽取，多要素联合抽取可以更好地捕捉地域、产品和行业三种要素之间的关联关系，提升抽取的准确率。同时，多任务学习可以帮助模型更快地收敛到最优值，对于要素取值有多个的情况，效果提升明显。

5 结论

事件因果关系抽取在金融领域具有较为广泛的应用，可以帮助用户发现事件之间的关联关系，提升舆情预警的及时性和准确性。本文基于CCKS 2021主办方提供的事件因果关系抽取数据集，提出了基于多标签分类和多任务学习事件因果关系抽取方法，并取得了初赛第一，决赛第二的好成绩。

致谢

我们对本次赛题的主办方CCKS组委会、蚂蚁集团和中科院自动化所表示感谢，对本论文的审稿人员表示感谢。

参考文献

1. Naughton, M., Kushmerick, N., & Carthy, J. (2006, July). Event extraction from heterogeneous news sources. In proceedings of the AAAI workshop event extraction and synthesis (pp. 1-6).
2. Ahn, D. (2006, July). The stages of event extraction. In Proceedings of the Workshop on Annotating and Reasoning about Time and Events (pp. 1-8).
3. Llorens, H., Saquete, E., & Navarro, B. (2010, August). TimeML events recognition and classification: learning CRF models with semantic roles. In Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010) (pp. 725-733).
4. Chen, Y., Xu, L., Liu, K., Zeng, D., & Zhao, J. (2015, July). Event extraction via dynamic multi-pooling convolutional neural networks. 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).
5. Nguyen, T. H., & Grishman, R. (2015, July). Event detection and domain adaptation with convolutional neural networks. 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).
6. Feng, X., Qin, B., & Liu, T. (2018). A language-independent neural network for event detection. *Science China Information Sciences*, 61(9), 1-12.
7. Wadden, D., Wennberg, U., Luan, Y., & Hajishirzi, H. (2019). Entity, relation, and event extraction with contextualized span representations. arXiv preprint arXiv:1909.03546.
8. Zhou, Y., Chen, Y., Zhao, J., Wu, Y., Xu, J., & Li, J. (2021, May). What the role is vs. What plays the role: Semi-supervised Event Argument Extraction via Dual Question Answering. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, No. 16, pp. 14638-14646).
9. Li, Q., Ji, H., & Huang, L. (2013, August). Joint event extraction via structured prediction with global features. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 73-82).
10. Sha, L., Qian, F., Chang, B., & Sui, Z. (2018, April). Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In Thirty-Second AAAI Conference on Artificial Intelligence.
11. Nguyen, T. M., & Nguyen, T. H. (2019, July). One for all: Neural joint modeling of entities and events. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 33, No. 01, pp. 6851-6858).
12. Zhang, J., Qin, Y., Zhang, Y., Liu, M., & Ji, D. (2019). Extracting Entities and Events as a Single Task Using a Transition-Based Neural Model. In IJCAI (pp. 5422-5428).
13. Sun, Y., Cheng, C., Zhang, Y., Zhang, C., Zheng, L., Wang, Z., & Wei, Y. (2020). Circle loss: A unified perspective of pair similarity optimization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 6398-6407).
14. Du, X., & Cardie, C. (2020). Event extraction by answering (almost) natural questions. arXiv preprint arXiv:2004.13625.

15. Liu, S., Zhang, X., Zhang, S., Wang, H., & Zhang, W. (2019). Neural machine reading comprehension: Methods and trends. *Applied Sciences*, 9(18), 3698.
16. Hu, M., Peng, Y., Huang, Z., & Li, D. (2019). A multi-type multi-span network for reading comprehension that requires discrete reasoning. *arXiv preprint arXiv:1908.05514*.
17. Efrat, A., Segal, E., & Shoham, M. (2019). Tagbased multi-span extraction in reading comprehension. *arXiv preprint arXiv:1909.13375*.
18. Cui, Y., Che, W., Liu, T., Qin, B., Yang, Z., Wang, S., & Hu, G. (2019). Pre-training with whole word masking for chinese bert. *arXiv preprint arXiv:1906.08101*.
19. Gururangan, S., Marasović, A., Swayamdipta, S., Lo, K., Beltagy, I., Downey, D., & Smith, N. A. (2020). Don't stop pretraining: adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.