基于实体对链接的单阶段实体-关系联合抽取模型

TPLinker: Single-stage Joint Extraction of Entities and Relations Through Token Pair Linking

Yucheng Wang[1], Bowen Yu[1]*, Yueyang Zhang[2]
Tingwen Liu[1], Hongsong Zhu[1], Limin Sun[1]
[1]School of Cyber Security, University of Chinese Academy of Sciences
[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]Baidu, Inc.

论文链接： https://arxiv.org/abs/2010.13415

源码链接： https://github.com/131250208/TPlinker-joint-extraction

在本文中，作者提出了基于握手标注机制的端到端序列标注模型 TPLinker，用于实体和关系的联合抽取。它是**第一个**真正意义上的单阶段联合抽取模型，可以在不受曝光误差的影响下解决单一实体重叠、实体对重叠的问题。模型在 NYT 和 WebNLG 两个关系抽取任务上均实现了当时的 **SOTA 性能**。

模型整体思想如下：

一、问题背景

关系抽取 (Relation Extraction, RE)，即从非结构化文本中抽取实体和对应关系，是自然语言处理中的基本任务。简而言之，给定一段文本，需要从中抽出（subject, predicate, object）三元组。

在抽取过程中，通常需要考虑下述三个问题：

(1) **交互缺失**：忽略了关系抽取和实体抽取的内在联系，导致实体抽取准确率高、关系抽取准确率低的问题。在传统的流水线（pipline）方法中，需要首先识别出文本中的实体，再对识别出的实体进行关系分类。这种方法忽略了实体与关系之间在文本中的信息交互关系，产生了交互缺失问题。

(2) **实体重叠**：如何从嵌套的实体中提取出多种关系是一个重要问题。在流水线方法中，先抽取实体再抽取关系的方式导致抽取出来的两个实体只能对应一种关系，无法处理实体重叠的问题。具体而言，实体间的重叠关系主要分为以下三种情况（实例见图1-1）：

- Normal：常规，即两个实体之间关系唯一且与其他实体不存在任何关系
- Single Entity Overlap (SEO)：单一实体重叠，即一个实体与多个实体具有关系
- Entity Pair Overlap (EPO)：实体对重叠，即同一对实体存在至少两种关系

(3) **曝光偏差**：来源于训练阶段与推理阶段的差异：训练时，模型接受真实的实体片段标签作为输入，实际推理时却以前一单元的输出作为当前单元的输入。当错误的输出成为下一单元的输入，会造成误差累计，损害模型的性能。

	Texts	Triplets
Normal	[The United States] President [Trump] will meet [Xi Jinping], the president of [China].	(The United States, *president*, Trump) (China, *president*, Xi Jinping)
SEO	Two of them, [Jeff Francoeur] and [Brian McCann], are from [Atlanta].	(Jeff Francoeur, *live in*, **Atlanta**) (Brian McCann, *live in*, **Atlanta**)
EPO	**[Sacramento]** is the capital city of the U.S. state of **[California].**	(**California**, *contains*, **Sacramento**) (**California**, *capital city*, **Sacramento**)

图 1-1 实体重叠的例子

为了处理问题 1，联合抽取（joint extraction）的方法应运而生，它充分利用实体与关系之间的信息交互关系，在一个模型中同时对实体和关系进行统一的抽取。目前，改进的联合抽取方法已经可以较好地处理问题 2，却无法处理曝光偏差的问题：本质上，这些基于解码机制的模型在解码阶段仍需要分多个步骤对同一三元组中的实体、关系进行

抽取。

本篇论文提供了第一个解决重叠关系并不产生曝光偏差的单阶段联合抽取模型 TPLinker，不仅成功解决了上述三个问题，而且在 NYT 和 WebNLG 两个关系抽取任务上实现了当时的 SOTA 性能。

二、模型阐述

2.1 模型整体思想

整体而言，它将联合抽取任务转化为 **T**oken **P**air **L**inking 问题。这种创新的标注方式不仅能够正确地表示嵌套实体，同时实现了首实体、尾实体的同阶段抽取，解决了曝光偏差的问题。它从长度为 n 的句子序列中有顺序地抽取两个 tokens（称之为 token pairs），记为 p_1 和 p_2。该序列共有 n^2 个 token pairs，对于每个 token pair 的每个关系 r，TPLinker 做三个独立的分类任务：

(1) p_1 和 p_2 是否分别为同一个实体的头和尾?
(2) p_1 和 p_2 是否分别为关系 r 的两个实体的开始位置?
(3) p_1 和 p_2 是否分别为关系 r 的两个实体的结束位置?

因此，模型整体为一个标准的分类流程。通过对每个 token pair 的分类，生成对应的标注。通过对标注的解码，可以生成实体-关系三元组，如果解码后不存在三元组，则这对 token pair 之间不存在关系。

模型整体流程如下：

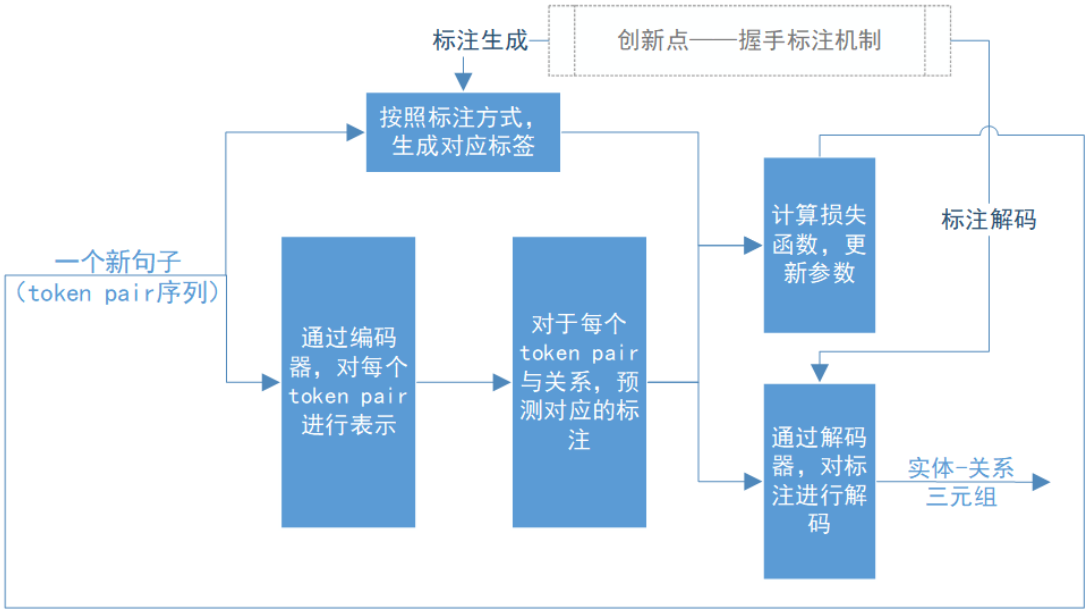

图 2-1 模型整体流程

2.2 创新点——握手标注机制

首先，对模型的创新点握手标注机制进行阐述，分别对应上图中标注标签的生成方式与对应的解码过程。

2.2.1 标注生成

给定一段文本，可以通过确定关系、确定头实体、尾实体的边界得出关系三元组。举个例子，在句子“The new mayor of [New York City] [De Blasio] is native-born”中，关系为“mayor”，头实体为“New York City”，尾实体为“De Blasio”，那么对应头、尾实体头的边界为 (4, 7)，头、尾实体尾的边界为 (6, 8)。上述边界与对应关系通过一定方式符号化，即可得到对应的三元组。

本文中，作者设计了三种链接方式，将数据转化为一个可标注的序列。

(1) Entity Head to Entity Tail（EH-to-ET）——实体头到实体尾，对应任务 1
(2) Subject Head to Object Head（SH-to-OH）——头实体头到尾实体头，对应任务 2
(3) Subject Tail to Object Tail（ST-to-OT）——头实体尾到尾实体尾，对应任务 3

仍以前面的例子为例，对应的链接方式如图2-2所示。

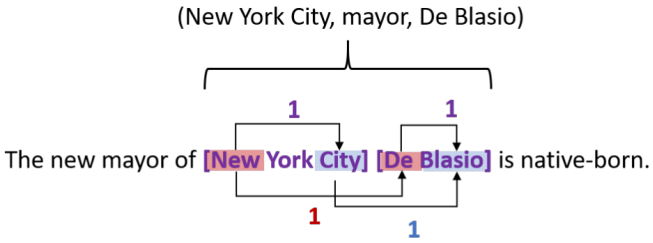

图 2-2 标注示例

其中，紫色数字表示 EH-to-EH 链接，红色数字表示 SH-to-SH 链接，蓝色数字表示 ST-to-OT 链接。上述链接可以在一个矩阵中表示，如图2-3。

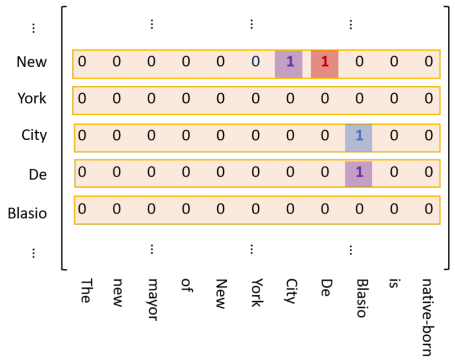

图 2-3 标注对应的矩阵

其中，横轴与纵轴分别对应原始句子中对应 token 的下标，如果 token 之间存在上述链接，则对应元素为 1，否则为 0。为了便于说明，作者将三种链接都叠加在一起展示，对应图中的三种颜色。在实际情况中，由于三种链接可能存在重叠情况，各自采用独立的矩阵分别进行表示。设句子的 token 长度为 n，由于同一对实体的头尾唯一且不重叠，可以用一个 $n \times n$ 的矩阵来表示紫色标注。同时，同一对实体可能对应多个关系，设关系数为 R，则红色、蓝色标注可以分别用 R 个 $n \times n$ 的矩阵表示，得到 $2R+1$ 个矩阵。

虽然该种标注方式能够较好地表示三元组，但所得的矩阵较为稀疏。一方面，由于实体尾部不可能出现在头部之前，EH-to-ET 矩阵中的下三角区域均为 0，造成了资源大量浪费。另一方面，由于头实体可以出现在尾实体后，SH-to-OH、ST-to-OT 对应矩阵的下三角区域也不可忽略。因此作者采用了下述映射方法，将下三角区域的非零元素映射到上三角部分：将该元素转置到上三角部分，并标记为 2。映射后，将下三角部分丢弃，所得矩阵如图2-4所示。

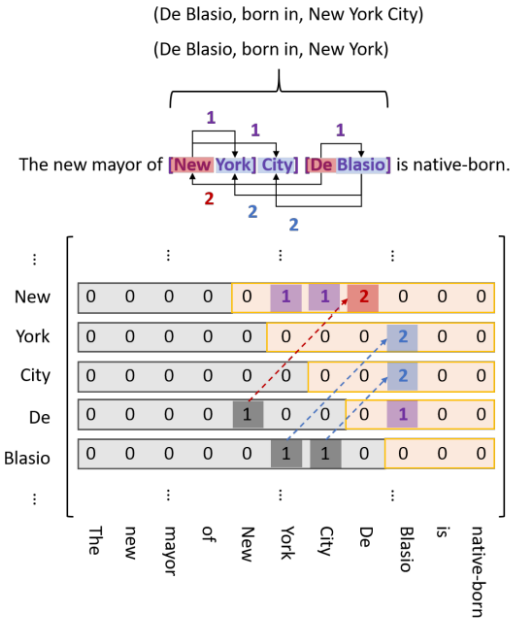

图 2-4 映射后的标注

在运算时，将所得矩阵按行拼接（图2-4中的橙色序列），得到了最终的标注。此时，对于一对 token pair，该任务被重建为 $2R+1$ 个序列标注子任务，每个子任务的序列标注长度为 $n(n+1)/2$（拼接后的上三角矩阵长度），模型的完整标注和编码示意图如图2-5所示。

本标注方法属于联合抽取方法，在一个模型中对实体和关系同时进行抽取，考虑了关系与实体之间的内在联系，解决了交互缺失的问题；通过确定头尾边界并设置链接方

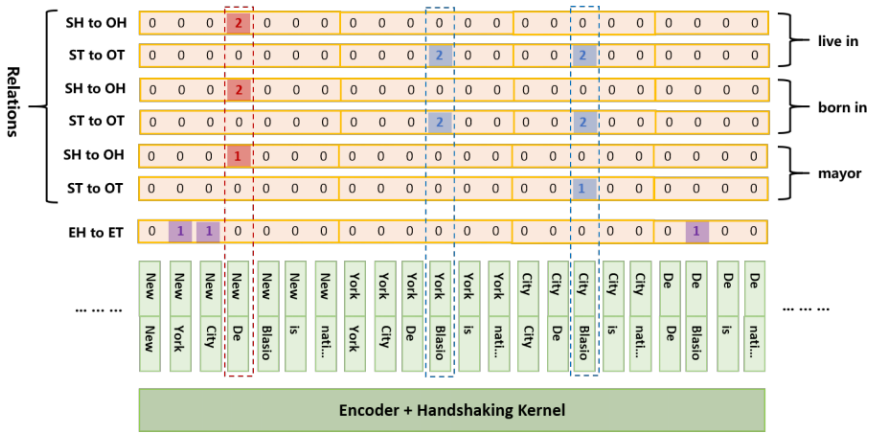

图 2-5 模型标注部分整体框架

法，较好地表示了嵌套实体，解决了实体重叠的问题；在同一解码阶段确定了首实体与尾实体，成功解决了曝光偏差的问题。

2.2.2 标注解码

对于每个关系，该模型的解码过程分为以下四步：

(1) 解码 EH-to-ET 序列，得到句子中的所有实体。将实体头作为 key，实体本身作为 value，存入字典 D。

(2) 解码 ST-to-OT 序列得到（头实体尾部，尾实体尾部）元组，并将其存入集合 E。

(3) 解码 SH-to-OH 序列，得到（头实体头部，尾实体头部）元组，从字典 D 中查找所有头部位置相同的实体。

(4) 遍历所有头实体-尾实体组合，将尾部符合集合 E 中标准的关系三元组对存入结果集 T。

结合图2-5的例子，以关系“born in”为例，对解码过程进行具体阐述：

(1) 解码 EH-to-ET 得到三个实体 {New York, New York City, De Blasio}，字典 D{New: (New York, New York City), De: (De Blasio)}。

(2) 解码 ST-to-OT 得到实体尾部集合 E：{(City, Blasio)}。

(3) 解码 SH-to-OH 得到实体头部，在字典 D 中找到所有可能的头实体集合 {New York, New York City}，尾实体集合 {De Blasio}。

(4) 遍历上述头实体和尾实体集合，检查是否符合集合 E 中的要求，得到结果集 T：{(De Blasio, born in, New York), (De Blasio, born in, New York City)}。

2.3 模型具体实现

结合图2-1，对模型其余部分的实现细节进行阐述。

2.3.1 token pair 表示

对于一个长度为 n 的句子 $[w_1, w_2, \cdots, w_n]$，采用一个 encoder(LSTM 或 BERT) 将每个 token w_i 映射为 h_i。对于上述每个分类任务，通过拼接向量，经过线性变换、激活函数后生成 token pair(w_i, w_j) 的表示

$$h_{i,j} = tanh(W_h \cdot [h_i; h_j] + b_h), j \geq i \tag{2.1}$$

其中，W_h 是权重矩阵，b_h 是偏置向量，均为训练时的参数。

2.3.2 标注预测

对于每个 token pair，预测二者之间链接的类型。

$$P(y_{i,j}) = Softmax(W_O \cdot h_{i,j} + b_O) \tag{2.2}$$

$$link(w_i, w_j) = \underset{l}{argmax}\ \ P(y_{i,j} = l) \tag{2.3}$$

其中，$P(y_{i,j} = l)$ 表示 (w_i, w_j) 之间链接为 l 的可能性。结合小节可知，对于 SH-to-OH、ST-to-OT 序列的分类任务，l 的可能取值为 0、1、2，对于 EH-to-ET 序列的分类任务，l 的可能取值为 0 和 1。

2.3.3 损失函数计算

对于每对 token pair 的每个关系 r，计算对数极大似然损失函数如下：

$$L_{link} = -\frac{1}{n} \sum_{i=1, j \geq i}^{n} \sum_{* \in \{E, H, T\}} logP(y^*_{i,j} = \hat{l}^*)$$

其中，n 为输入句子的长度，$\hat{l}$ 为真实的标注，E, H 和 T 分别表示 SH-to-OH、ST-to-OT、EH-to-ET 的标注。

2.4 模型结果

截至论文被接收，该模型在 NYT 和 WebNLG 两个关系抽取任务上都达到了当时的 SOTA 性能。

同时，通过与其他模型的对比实验，可以发现该模型在处理复杂句子具有显著优势。由于 TPLinker BERT 能够以批处理模式处理数据，而 CasRel BERT 一次只能处理一句话，TPLinker 在计算性能上也有较大优势。

Model	NYT*			NYT			WebNLG*			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging† (Zheng et al., 2017)	–	–	–	32.8	30.6	31.7	–	–	–	52.5	19.3	28.3
CopyRE‡ (Zeng et al., 2018)	61.0	56.6	58.7	–	–	–	37.7	36.4	37.1	–	–	–
MultiHead* (Bekoulis et al., 2018)	–	–	–	60.7	58.6	59.6	–	–	–	57.5	54.1	55.7
GraphRel‡ (Fu et al., 2019)	63.9	60.0	61.9	–	–	–	44.7	41.1	42.9	–	–	–
OrderCopyRE‡ (Zeng et al., 2019)	77.9	67.2	72.1	–	–	–	63.3	59.9	61.6	–	–	–
ETL-Span‡*(Yu et al., 2020)	84.9	72.3	78.1	85.5	71.7	78.0	84.0	91.5	87.6	84.3	82.0	83.1
WDec‡(Nayak and Ng, 2020)	**94.5**	76.2	84.4	–	–	–	–	–	–	–	–	–
CasRel$^{\ddagger}_{LSTM}$(Wei et al., 2020)	84.2	83.0	83.6	–	–	–	86.9	80.6	83.7	–	–	–
CasRel$^{\ddagger}_{BERT}$(Wei et al., 2020)	89.7	89.5	89.6	–	–	–	**93.4**	90.1	91.8	–	–	–
TPLinker$_{LSTM}$	83.8	83.4	83.6	86.0	82.0	84.0	90.8	90.3	90.5	**91.9**	81.6	86.4
TPLinker$_{BERT}$	91.3	**92.5**	**91.9**	**91.4**	**92.6**	**92.0**	91.8	**92.0**	**91.9**	88.9	**84.5**	**86.7**

图 2-6 TPLinker 与其他模型的性能对比

三、总结与思考

通过阅读本篇论文，收获如下：

1. 了解了一种新的实体关系联合抽取的标注方案，在解决实体重叠问题的同时克服了曝光误差的问题。
2. 学会将联合抽取任务转化为 Token Pair Linking 的分类问题，通过对链接方式的分类得出对应的标注方案。
3. 认识到该模型在关系抽取任务上均具有很高的性能，并且可以这种标注方式同样也可以处理命名实体识别的嵌套实体问题，具有较高的实用价值。