面向触发词感知增强的事件抽取方法

李安南[1,2], 陆垚杰[1,2], and 林鸿宇[1]

[1] 中国科学院软件研究所, 北京
[2] 中国科学院大学, 北京
{liannan2019,yaojie2017,hongyu}@iscas.ac.cn

摘要 本文为解决在多事件情景下，论元检测过程中，使用 BERT + CRF 模型抽取给定触发词所对应的论元时，模型会错误抽取句中其它触发词所属的论元，导致精准率下降的问题。通过在输入端额外拼接当前待抽取论元所对应的触发词，提升论元抽取模型对于触发词的考虑；以及一系列针对触发词的特征，进一步强化模型对触发词的建模，进一步提升性能。最终，触发词检测模型和论元检测模型通过策略性地选择使用不同的投票阈值，取得 B 榜 F1 值 0.7999 的成绩，位列 B 榜第三。

Keywords: 事件抽取 · 多事件 · 模型融合.

1 引言

事件抽取是从无结构的自然语言文本中抽取到结构化的事件信息。事件抽取可以分为触发词检测（trigger detection）任务和论元检测（argument detection)。在本任务中，触发词是句中最能反映跟设备故障、操作/调整机器等事件发生的词语; 论元是事件的参与者，是组成事件的核心部分，它与事件触发词构成了事件的整个框架。例如在句子”XX 小区晚上 8 点之后苹果终端接入 5G 网络 失败” 中: ” 接入” 表示软硬件故障（SoftHardwareFault）事件的触发词; ” 苹果终端” 表示故障发生的位置，充当 Subject 论元角色; ”5G 网络” 表示故障相关的宾语，充当 Object 论元角色; ” 失败” 表示故障的状态，充当 State 论元角色。

通过对原始数据进行分析，我们得到如表 1所示数据特点，可以认为本数据集有着短文本、多事件而基本没有标注重叠的特点。

我们把事件抽取任务分解为流水线式地依次进行触发词检测和论元检测。这两个子任务我们都把其看作是序列标注问题。观察到数据集没有标注重叠的问题, 我们采用经典的 BERT[1] + CRF 模型进行建模解决。在多事件

表 1. 华为公司公开故障处理案例事件抽取数据集特征

文本长度 40 字以下	87.46%
一个训练实例中有 2 个及以上事件	75.00%
触发词重叠	0.01%
论元重叠	1.38%

的情景下，论元检测碰到的主要问题是在给定触发词，抽取本触发词所对应的论元时，会错误抽取句中其它触发词所属的论元，导致精准率（Precision）下降。造成这一问题的主要原因在于论元检测模型引入的触发词信息，不足以在建模时充分考虑触发词这一前提条件。因此，为了解决这一问题，我们通过在输入端额外拼接当前待抽取论元所对应的触发词，提升模型对于触发词的考虑；以及一系列针对触发词的特征，进一步强化模型对触发词的建模，更进一步提升性能。最终单模型在 B 榜测试集上的 F1 值为 0.7918。通过对触发词检测、论元检测模型使用不同的模型融合策略，在 B 榜 F1 值取得 0.7999 的成绩，最终位列第三。

2 问题定义

我们把事件抽取任务分解为流水线式依次进行触发词检测和论元检测，更进一步，触发词检测和论元检测都看作序列标注问题。即给定一个中文句子 $x = \{char_1, char_2, ..., char_n\}$，对句子中的每个字 $char_i$ 进行相应标签 y_i 的预测，获得目标标注序列 $y = \{y_1, y_2, ..., y_n\}$。本文采用如图 1所示的“BIOUL”(Begin, Inside, Other, Unit, Last) 与具体类别标签相结合的标签标注格式。

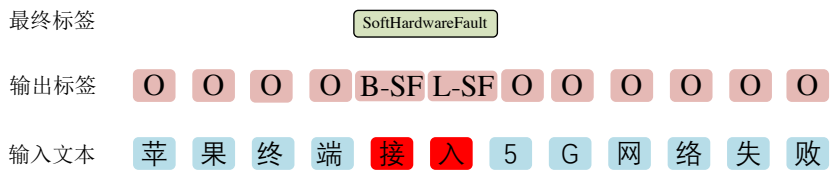

图 1. 触发词检测输出标注标签示例：触发词” 接”、” 入” 对应的输出标签分别为 B-SF、L-SF，表示” 接入” 是事件 *SoftHardwareFault* 的触发词。

3 模型

本文触发词检测和论元检测两个子任务都采用 BERT[3] + Bi-LSTM + CRF[3] 的基本模型架构。特别地，论元检测相比于触发词检测有些许不同：论元检测需要考虑在给定触发词的前提下，抽取到相应的论元，可以认为是带条件的序列标注问题；因此，为了让模型能在满足给定触发词的条件下进行相应论元的抽取，在论元检测阶段，我们在输入端拼接触发词的嵌入表示，加强模型对触发词这一前置条件的考虑，提升模型在给定条件下进行序列标注的性能。接下来，我们以更复杂的论元检测模型进行分析。

3.1 基础模型

论元检测模型如图 2所示，主要包含五个模块：字嵌入表示层、特征嵌入表示层、双向长短时记忆网络（Bi-directional Long Short-Term Memory, Bi-LSTM）层、多层感知机 (Multilayer Perceptron, MLP) 层和条件随机场 (Conditional Random Fields, CRF) 层，触发词检测模型无特征嵌入表示层。

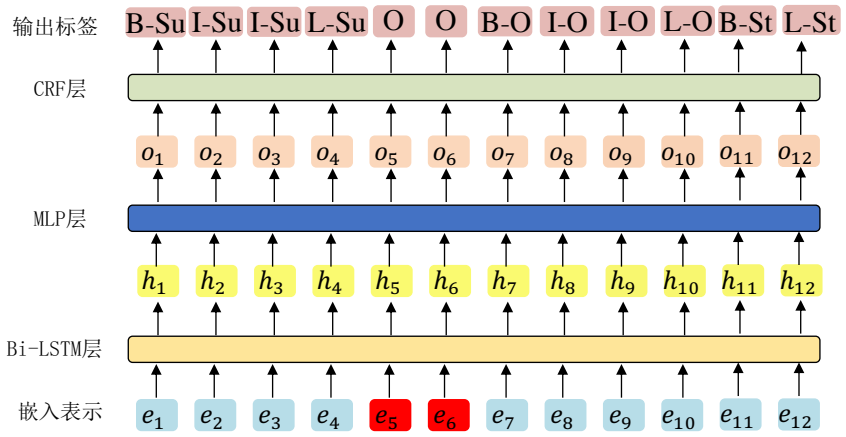

图 2. 论元检测模型结构：输出标签中”Su”、”St” 所对应的完整标签为”Subject”、”State”。

[3] 本文用 BERT 表示原生 BERT 或基于 BERT 衍生的大规模预训练语言模型。

字嵌入表示层 ：本文使用当前通用的 BERT[1] 作为字嵌入表示层，对文本进行基础特征表示。原始句子为 $S = [char_1, char_2, ..., char_n]$，经过字嵌入表示层后，变为 $E = [e_{char_1}, e_{char_2}, ..., e_{char_n}]$。BERT 作为经过大规模预训练的语言模型，除能捕获句中的语法特征外，还能赋予每个字与上下文有关的动态语义特征。本文把经过预训练的 BERT 加入训练，进行微调。

特征嵌入表示层 ：在进行论元检测的时候, 为了能加强模型对触发词这一前置条件的考虑触发词，因此把触发词经过字嵌入表示层得到如式 1所示的表示 $e_{trigger}$（trigger embedding）[4]与每个字嵌入表示 e_{char_i} 进行拼接（concate）操作。此外，为了能进一步加强模型对给定触发词的考虑，提升论元检测的性能, 我们尝试使用如 3.2所述的额外特征, 经过独热（one-hot）映射后，得到每个特征 k 的嵌入表示 $e_i^{feature_k}$（feature embedding）也与每个字嵌入表示 e_{char_i} 进行拼接操作得到 e_i。

$$e_{trigger} = Pooling([e_{char_1}^{trigger}, e_{char_2}^{trigger}, ..., e_{char_j}^{trigger}]) \tag{1}$$

$$e_i = Concate(e_{char_i}; e_{trigger}; e_i^{feature_1}; e_i^{feature_2}; ...; e_i^{feature_m}) \tag{2}$$

Bi-LSTM 层 ：把经过 BERT 编码得到的字嵌入表示以及特征嵌入表示进行拼接操作得到 e_i 后，接入 Bi-LSTM[2] 层得到如式 3所示的 h_i，通过 Bi-LSTM 层获取过去和将来时序的数据特征并融合上下文信息，以更好地加强 char embedding、trigger embedding 和 feature embeddings 之间的信息交互。

$$\begin{aligned} \overrightarrow{h_i} &= LSTM([e_1, e_2, ..., e_i]) \\ \overleftarrow{h_i} &= LSTM([e_n, e_{n-1}, ..., e_i]) \\ h_i &= Concate(\overrightarrow{h_i}; \overleftarrow{h_i}) \end{aligned} \tag{3}$$

MLP 层 ：对每个经过 Bi-LSTM 层的字提取到的表示 $H = [h_1, h_2, ..., h_n]$ 输入到多层感知机层中，如式 4所示。

$$O = softmax(W^{cls} H + b^{cls}) \tag{4}$$

[4] 本文中，我们选择触发词首字的 BERT char embedding 代表整个触发词，如图 3中的" 触发词特征" 所示。

其中，W^{cls} 是分类器权重矩阵，b^{cls} 是偏置项。输出矩阵 $O = [o_1, o_2, ..., o_n]$，其中 o_t 代表第 t 个字所属的不同类别的预测概率（Type Classifier Output），对于候选触发词 t 是类别 j 的概率如式 5所示

$$P(j|t, \theta) = o_t^j \tag{5}$$

CRF 层 ：虽然句中的每个字经过多层感知机后，可以得到属于每个类别的概率，但得到的只是局部最优的解码，因此我们需要在多层感知机之后加上 CRF 层，使模型可以在训练的过程中，根据标签之间的依赖关系为最终的预测标签添加约束条件，生成全局最优预测标签序列 $y = [y_1, y_2, ..., y_n]$。CRF 层定义在给定多层感知机的输出 O 下，标签序列 y 的概率为:

$$P(y \mid O) = \frac{\exp(\text{score}(O, y))}{\sum_{y'} \exp\left(\text{score}\left(O, y'\right)\right)} \tag{6}$$

根据 Lample 等 [4] 分数 $\text{score}(O, y)$ 定义为生成的转移概率矩阵与发射概率矩阵的总和，如式 7所示。在训练过程中，可以使用对数极大似然估计获取模型参数。

$$\text{score}(O, y) = \sum_{i=0}^{n} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} F_{O, y_i} \tag{7}$$

3.2 额外特征

为了能进一步加强模型对给定的触发词的考虑，提升论元检测的性能，我们额外尝试使用如图 3所示的事件类别特征、序列相对距离特征、依存相对距离特征。由于 BERT 在中文上的最小单位都是字级别的，因此以下特征都是字级别的特征，如果本身为词级别，则每个词中的所有字通过共享词级别特征，转换为字级别特征。

事件类别特征 ：当前待抽取论元所属的触发词的事件类别。

序列相对距离特征 ：当前字符距离触发词的相对距离，考虑到论元在触发词前和在触发词后是不同的，因此使用 ± 表示相对距离的方向。

依存相对距离特征 ：当前字符到触发词依存树的跳数。通过对句子进行依存分析得到的依存树，可以看作是有向图，并且是单向的（即两点直接最多

图 3. 输入文本及相应特征：事件类别特征中的"S" 表示当前待抽取论元所属的触发词对应的类别为 *SoftHardwareFault*；知识特征中的" 牌"、" 物" " 事" " 术" 所对应的完整标签为" 品牌名"、" 物体类"、" 场景事件"、" 术语类"。

只有一条有向边），把直接相连的两点间权重设为 1，其余设为 1000（无穷）。但是如果不作任何更改，直接把依存树作为有向图，会导致句中有很多的词是触发词无法到达的点，使得距离无限大；而如果把有向图改为无向图，又会使得触发词到句中任意字符的距离都基本不会超过 5，区分度不明显；因此在本实验中把直接相连的源点到终点权重设为 1，终点到源点的权重设为 10，采用非对称的权重。

词语知识特征 ：此外，还额外引入知识特征 [8]，获取到更丰富的知识标注结果，该知识特征覆盖所有中文词汇的词类体系，包括各类实体词与非实体词（如概念、实体/专名、语法词等），用以辅助模型更好地理解句子语义以及句中各成分的含义。

3.3 损失函数

我们对 MLP 层的输入 h_i 使用两次 Dropout[6]，然后对这两次 Dropout 经过 MLP 层得到的结果 o_i^1 和 o_i^2 使用式 8计算两者的 KL 散度得到 R-Drop 损失 [5]。R-Drop 以一定的比例 α 与 CRF 损失相加得到最终损失，进行反向传播。

$$\mathcal{L}_i^{(KL)} = \frac{1}{2}\left[KL\left(o_i^1 \| o_i^2\right) + KL\left(o_i^2 \| o_i^1\right)\right] \tag{8}$$

3.4 模型融合

在模型融合阶段，我们首先对数据划分为 n 折，每折分别选 k 个不同种子进行训练。于是，触发词检测模型和论元检测模型能分别获得 $n \times k$ 个模型。

我们采用预测结果等权重阈值投票的模型融合策略，把触发词和论元的预测结果都转换为 span: [type, start offset, end offset] 的形式。即每个模型都为一票，span 的得票数高于阈值才作为结果，否则舍弃。

触发词和论元的阈值在设定上有些许区别。因为高阈值会提升精准率而降低召回率，而低阈值则是提升召回率降低精准率。

对于触发词 span 的预测阈值我们设置偏低，因为单模型在进行最终测试的时候，我们发现相比于精准率，召回率比较低，这意味着有相当的触发词或论元并未能找到，同时，我们注意到，一旦第一阶段的触发词检测没有抽取到，那么就会连带后面的论元都无法预测。因此，通过降低阈值主要提升召回率。

对于论元 span 的预测阈值我们设置偏高，因为在论元检测阶段，我们认为高阈值带来的精准率提升比召回率造成的损失更大。

4 实验

4.1 数据预处理

在浏览数据的时候，我们发现原始的 15000 条训练数据标注存在噪声，进行数据去重、归一化后，得到 13923 条训练实例，按 9：1 比例进行训练集 *train* 和验证集 *dev* 的划分。

4.2 参数设置

本文使用的 BERT 都为 chinese-roberta-wwm-ext[5]，触发词检测模型参数设置如表 2所示，论元检测模型参数设置如表 3所示。

4.3 验证集结果

触发词检测在验证集 *dev* 性能如表 4所示，可以看出 Bi-LSTM 网络和 R-Drop 损失都有一定的作用。其中，R-Drop 的作用可以这么理解：通过增

[5] https://huggingface.co/hfl/chinese-roberta-wwm-ext

表 2. 触发词检测模型模型设置

BERT 类型	chinese-roberta-wwm-ext($768dim12layers$)
Bi-LSTM 维度	$768 \times 300dim \times 2$
MLP 维度	$600 \times 32dim$
Dropout rate	0.3
D-Drop α	5.0
优化器	Adam
BERT learning rate	1e-5
其余 learning rate	1e-3
learning reduce factor	0.5
learning reduce patience	3
early stop patience	5
train epoch	100
batch size	16
训练数据折数 n	10
不同种子数 k	3
融合阈值	9

加一个正则项，强化模型对 Dropout 的鲁棒性，使得不同的 Dropout 下模型的输出基本一致。因为，Dropout 存在训练与预测不一致问题：在训练阶段使用 Dropout，训练的是不同 Dropout 的融合模型；而在预测阶段则是关闭 Dropout 的单模型，使得预测模型由“模型平均”变成了“权重平均”，而两者未必等价。通过 R-Drop 损失，使得不同的 Dropout 下模型的输出基本一致，降低这种不一致性，促进“模型平均”与“权重平均”的相似性，从而使得简单关闭 Dropout 的效果等价于多 Dropout 模型融合的结果，提升模型最终性能 [7]。

论元检测在验证集 *dev* 性能如表 5所示，可以看出，在 5 项特征嵌入表示中，移除触发词特征嵌入表示，F1 分数下降的最多，这说明触发词特征在 5 项特征嵌入表示中最重要。此外，注意到，在移除触发词特征后，相比于召回率，精准率大幅下降，这也验证了本文所遇到的问题：在多事件的情景下，抽取本触发词所对应的论元时，会错误抽取句中其它触发词所属的论元。通过拼接触发词嵌入表示后，论元检测模型已经能充分考虑在给定触发词条件下，抽取该触发词所对应的论元。

表 3. 论元检测模型模型设置

BERT 类型	chinese-roberta-wwm-ext($768dim12layers$)
触发词 Pooling 类型	触发词首字 pooling
触发词特征	$768dim$
事件类别特征	$10 \times 50dim$(随机初始化)
带方向的相对距离特征	$986 \times 50dim$(随机初始化)
相对依存距离特征	$79 \times 50dim$(随机初始化)
知识特征	$62 \times 50dim$(随机初始化)
Bi-LSTM 维度	$1686 \times 300dim \times 2$
MLP 维度	$600 \times 70dim$
Dropout rate	0.3
D-Drop α	1.0
优化器	Adam
BERT learning rate	1e-5
其余 learning rate	1e-3
learning reduce factor	0.5
learning reduce patience	3
early stop patience	5
train epoch	100
batch size	16
训练数据折数 n	10
不同种子数 k	3
融合阈值	21

表 4. 触发词检测实验结果

模型	**精准率（P）**	**召回率（R）**	**调和平均数（F1）**
触发词检测模型	95.51	95.78	95.64
(-)Bi-LSTM	95.25	95.25	95.25
(-)R-Drop	95.05	95.05	95.05

表 5. 论元检测实验结果

模型	精准率（P）	召回率（R）	调和平均数（F1）
论元检测模型	93.59	91.57	92.57
(-) 触发词特征	49.28	84.96	62.38
(-) 事件类别特征	91.91	90.68	91.29
(-) 序列相对距离特征	90.01	90.57	90.29
(-) 依存相对距离特征	92.61	92.11	92.36
(-) 知识特征	93.49	91.82	92.64

4.4 B 榜测试集结果

B 榜测试集结果如表 6所示，通过简单对触发词检测模型和论元检测模型设定使用不同的投票阈值，可以取得更进一步的提升。

表 6. 触发词检测实验结果

模型	精准率（P）	召回率（R）	调和平均数（F1）
触发词、论元单模型	82.39	76.21	79.18
触发词单模型、论元融合模型	**83.77**	75.96	79.67
触发词、论元融合模型	83.13	**77.07**	**79.99**

5 结论

本文为解决在多事件情景下，论元检测过程中，使用 BERT + CRF 模型抽取给定触发词所对应的论元时，模型会错误抽取句中其它触发词所属的论元，导致精准率下降的问题。通过在输入端额外拼接当前待抽取论元所对应的触发词，提升论元抽取模型对于触发词的考虑；以及一系列针对触发词的特征，进一步强化模型对触发词的建模，进一步提升性能。最终，触发词检测模型和论元检测模型通过策略性地选择使用不同的投票阈值，带来更进一步提升。

参考文献

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. 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. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://www.aclweb.org/anthology/N19-1423

2. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (1997)
3. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. CoRR **abs/1508.01991** (2015), http://arxiv.org/abs/1508.01991
4. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 260–270. Association for Computational Linguistics, San Diego, California (2016)
5. Liang, X., Wu, L., Li, J., Wang, Y., Meng, Q., Qin, T., Chen, W., Zhang, M., Liu, T.: R-drop: Regularized dropout for neural networks. CoRR **abs/2106.14448** (2021), https://arxiv.org/abs/2106.14448
6. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(56), 1929–1958 (2014)
7. Su, J.: 又是 dropout 两次！这次它做到了有监督任务的 sota (2021)
8. Zhao, M., Qin, H., Zhang, G., Lyu, Y., Zhu, Y.: Termtree and knowledge annotation framework for chinese language understanding (2020)