信息抽取 UIE (Universal Information Extraction)

- From: PaddleNLP 中的 **通用信息抽取 UIE(Universal Information Extraction)**
  - **CCKS 2022 通用信息抽取 -- 基于UIE的基线系统**
- 论文：Unified Structure Generation for Universal Information Extraction, ACL 2022

目录：

模型框架 UIE

论文利用 **Encoder-Decoder** 框架，把信息抽取的常用任务，如实体识别、关系抽取、事件抽取，统一到一个模型框架下。

Encoder-Decoder 用的还是Transformer（T5），关键是如何设计输入和输出格式，把各种抽取任务统一起来。

作者认为这些抽取任务都可以由2种原子操作组合而成：

1. **找点（Spotting）**：信息点类别，把信息所在的位置找出来，比如实体类型；
2. **连边（Associating）**：利用信息点关联类别把点连接起来；信息点关联类别，如关系类型、事件论元类型。

输出格式：SEL

作者定义了一种称为 **Structured Extraction Language (SEL)** 格式的任务输出样式：

```
(
  (Spot Name: Info Span
    (Asso Name: Info Span)
    (Asso Name: Info Span)
  )
)
```

其中

- **Spot Name**: 信息点类别，如实体类型；
- **Assocation Name** (asoc/asso): 信息点关联类别，如关系类型、事件论元类型；

- **Info Span**: 信息点所对应的文本片段。

比如对于 `Steve became CEO of Apple in 1997.` 这句话；

如果做NER，可以得到以下结果：

```
(
  (person: Steve)
  (organization: Apple)
  (time: 1997)
)
```

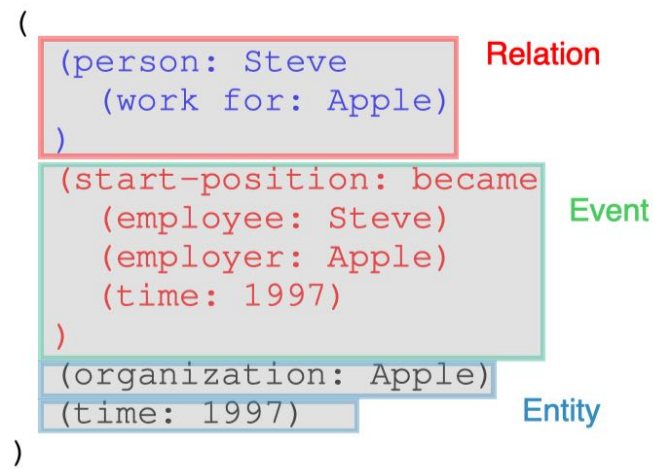
如果做事件抽取，可以得到以下结果，其中的 `start-position` 可以理解为**事件触发词**，或者**事件类别**；`employee`、`employer`、`time` 则是事件中的**角色或属性**：

```
(
  (start-position: became
    (employee: Steve)
    (employer: Apple)
    (time: 1997)
  )
)
```

如果做关系抽取，可以得到以下结果，其中的 `work for` 就是关系类别：

```
(
  (person: Steve
    (work for: Apple)
  )
  (organization: Apple)
)
```

合一块就长这样了：



一些示例（From Paper）：

Relation	CoNLL04	((location: Rome (located in: Lazio)) (location: Lazio) (location: Naples (located in: Campania)) (location: Campania))
Event	ACE05-Evt	((transport: heading (artifact: family) (destination: new hampshire) (origin: lakeland) (vehicle: plane)))
Sentiment	14/15/16-res	((aspect: staff (negative: horrible)) (opinion: horrible))

借助 SEL，就可以把各种抽取任务都统一到相同的表达框架。

UIE的具体实现中，会做以下变换：

一个**事件**抽取样例中key `"spot_asoc"` 对应的值：

```
'<extra_id_0> <extra_id_0> 大盘行情 <extra_id_5> 收
```

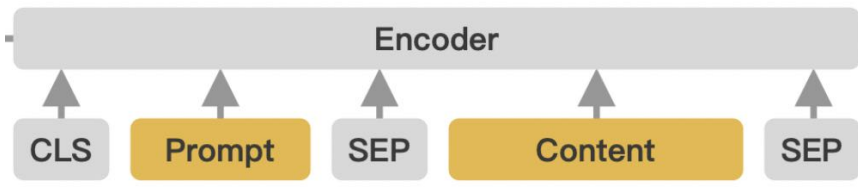
```
[{'span': '收报', 'label': '大盘行情', 'asoc': [['指数名称', '创业板指'], ['收盘价', '2321.13'], ['涨跌幅', '跌0.19%'], ['成交额', '1373.24亿']]}]
```

其中`大盘行情`为事件类型，`asoc`中每个元素的第一个元素为`事件角色`，第二个元素为`角色取值`。

```
报 <extra_id_0> 指数名称 <extra_id_5> 创业板指 <extra_id_1> <extra_id_0> 收盘价 <extra_id_5> 2321.13 <extra_id_1> <extra_id_0> 涨跌幅 <extra_id_5> 跌0.19% <extra_id_1> <extra_id_0> 成交额 <extra_id_5> 1373.24亿 <extra_id_1> <extra_id_1> <extra_id_1>'
```

输入格式：SSI

作者使用 **Structural Schema Instructor (SSI)**，把要抽取的Schema变成一串字符串，然后追加到待抽取的文本前面，一起输入Encoder，如右图。



SSI 把Schema变成一串字符串的思路也比较直观，如右图左边。基本就是`[spot] Spot-Name [asso] Asso-Name …`。

结构化模式前缀与待抽取的文本一同输入序列到结构生成模型，用于区分不同的抽取任务。 基线模型使用特殊字符`[spot]`、`[asoc]`来组织结构化模式前缀，`[spot]`对应 SEL 中的 SpotName 类别，`[asoc]`对应 AssoName。

不同任务的形式是：

- **实体抽取**：`[spot] 实体类别 [text]`，如`<spot> location <spot> organization <spot> person <spot> vehicle <spot> weapon`
- **关系抽取**：`[spot] 实体类别 [asoc] 关系类别 [text]`，如`<spot> location <spot> organization <spot> other <spot> people <asoc> kill <asoc> live in <asoc> located in <asoc> organization in <asoc> work for`
- **事件抽取**：`[spot] 事件类别 [asoc] 论元类别 [text]`，如`<spot> sue <spot> transfer money <spot> transfer ownership <spot> transport <spot> trial hearing <asoc> adjudicator <asoc> agent <asoc> artifact <asoc> attacker`
- **情感抽取**：`[spot] 评价维度 [asoc] 观点类别 [text]`，如`<spot> aspect <spot> opinion <asoc> negative <asoc> neutral <asoc> positive`

以夺冠事件为例，其对应的SSI为`[spot] 夺冠 [asoc] 夺冠事件 [asoc] 冠军 [asoc] 夺冠赛事 [text] 2月8日上午北京冬奥会自由...`。

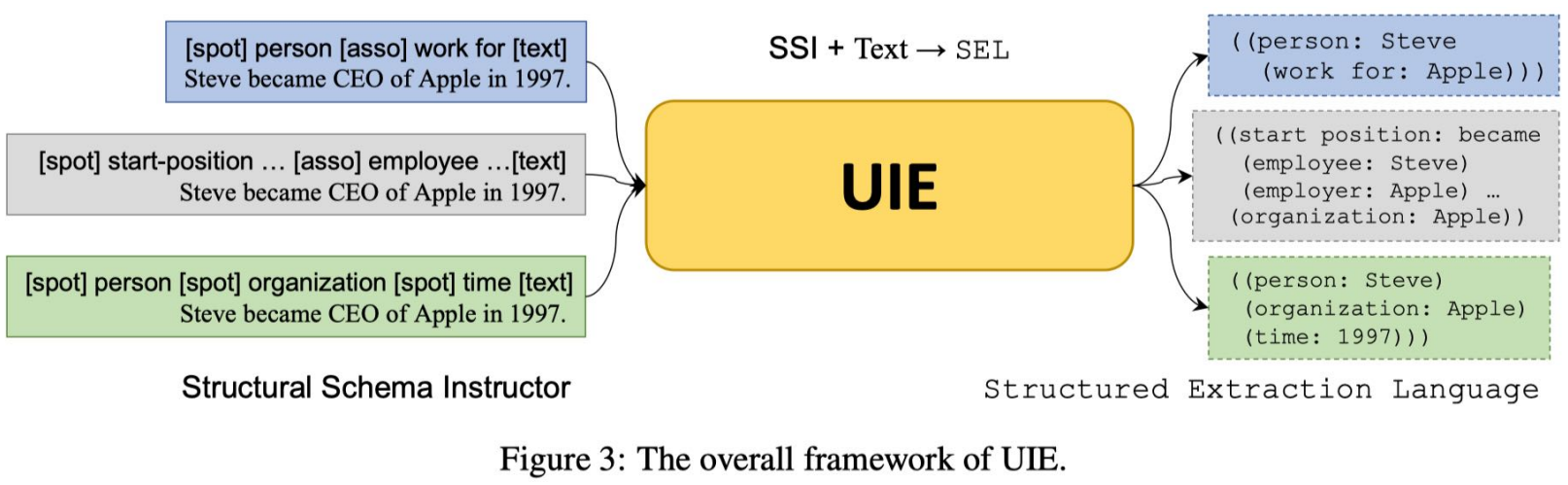


Figure 3: The overall framework of UIE.

模型预训练

目标函数中包含了3部分。

训练 Encoder-Decoder 结构

$$\mathcal{L}_{\text{Pair}} = \sum_{(x,y)\in\mathcal{D}_{\text{pair}}} -\log p(y|x, s_{meta}; \theta_e, \theta_d) \quad (5)$$

训练 Decoder，避免Decoder解码出格式不对的结果

$$\mathcal{L}_{\text{Record}} = \sum_{y\in\mathcal{D}_{\text{record}}} -\log p(y_i|y_{<i}; \theta_d) \quad (6)$$

训练 Encoder

$$\mathcal{L}_{\text{Text}} = \sum_{x \in \mathcal{D}_{\text{text}}} -\log p(x''|x'; \theta_e, \theta_d) \qquad (7)$$

最终的Loss是上面3个之和。训练数据的选取还有一些trick，具体看论文吧。

数据处理

之前金融事件数据 `DUEE_FIN_LITE` 中包含了多个事件，但在训练UIE时，在处理代码 `examples/information_extraction/DuUIE/process_data.py:490` 中，依据不同事件类别将多事件抽取分割成多个单事件类型抽取。如果一个样例中只包含一个事件，那此样例也会作为其他事件的负样例出现。

实验分析

官方 Few-Shot 实验

作者在互联网、医疗、金融三大垂类自建测试集上进行了实验：

	金融		医疗		互联网	
	0-shot	5-shot	0-shot	5-shot	0-shot	5-shot
uie-tiny	41.11	64.53	65.40	75.72	78.32	79.68
uie-base	46.43	70.92	71.83	85.72	78.33	81.86

`0-shot` 表示无训练数据直接通过 `paddlenlp.Taskflow` 进行预测，`5-shot` 表示基于 `5` 条标注数据进行模型微调。实验表明 UIE在垂类场景可以通过少量数据（few-shot）进一步提升效果。

说明 UIE 在少样本下能获得非常不错的效果。

贝叶自己的实验

使用下面的4个样例做训练（只包含一个事件 `大盘行情`）：

```
{"text": "上证指数收报3123.11点，涨0.5%，成交额3773.65亿元。", "spot_asoc": [{"span": "收报", "label": "大盘行情", "asoc": [["指数名称", "上证指数"], ["收盘价", "3123.11"], ["涨跌幅", "涨0.5%"], ["成交额", "3773.65亿"]]}], "spot": ["大盘行情"], "asoc": ["指数名称", "收盘价", "涨跌幅", "成交额"]}
{"text": "深证成指收报11206.82点，涨0.57%，成交额4455.2亿元。", "spot_asoc": [{"span": "收报", "label": "大盘行情", "asoc": [["指数名称", "深证成指"], ["收盘价", "11206.82"], ["涨跌幅", "涨0.57%"], ["成交额", "4455.2亿"]]}], "spot": ["大盘行情"], "asoc": ["指数名称", "收盘价", "涨跌幅", "成交额"]}
{"text": "创业板指收报2321.13点，跌0.19%，成交额1373.24亿元。", "spot_asoc": [{"span": "收报", "label": "大盘行情", "asoc": [["指数名称", "创业板指"], ["收盘价", "2321.13"], ["涨跌幅", "跌0.19%"], ["成交额", "1373.24亿"]]}], "spot": ["大盘行情"], "asoc": ["指数名称", "收盘价", "涨跌幅", "成交额"]}
{"text": "上证指数以接近3400点结束本日行情。", "spot_asoc": [{"span": "", "label": "大盘行情", "asoc": [["指数名称", "上证指数"], ["收盘价", "接近3400"]]}], "spot": ["大盘行情"], "asoc": ["指数名称", "收盘价"]}
```

对下面两个样例的预测结果如下：

测试样例：

```
{"text": "创业板指今日成交额为1573.54亿元。", "spot_asoc": [{"span": "", "label": "大盘行情", "asoc": [["指数名称", "创业板指"], ["成交额", "1573.54亿"]]}], "spot": ["大盘行情"], "asoc": ["指数名称", "成交额"]}
{"text": "深证成指今日收盘价为4500。", "spot_asoc": [{"span": "", "label": "大盘行情", "asoc": [["指数名称", "深证成指"], ["收盘价", "4500"]]}], "spot": ["大盘行情"], "asoc": ["指数名称", "收盘价"]}
```

预测结果：

```
{"entity": {"offset": [], "string": []}, "relation": {"offset": [], "string": []}, "event": {"offset": [], "string": [{"trigger": "", "type": "大盘行情", "roles": [["指数名称", "创业板指"], ["收盘价", "1573.54亿"]]}]}}
{"entity": {"offset": [], "string": []}, "relation": {"offset": [], "string": []}, "event": {"offset": [], "string": [{"trigger": "", "type": "大盘行情", "roles": [["指数名称", "深证成指"], ["收盘价", "4500"], ["成交额", "深证成指"]]}]}}
```

优缺点

优点：

- zero-shot or few-shot 能力较强

缺点：

- 生成模型，速度较慢
- 追加前缀的方式不能处理事件类别数量很多的情况，比如`200+`个事件类别，那前缀的长度都不止`512`了。。