「事件抽取」项目实践

姓名：张雪遥

学号：201928013229062

培养单位：计算技术研究所

代码说明

项目代码在github中托管：https://github.com/RMSnow/KG-Course

项目运行依赖

- Python >= 3.5.3
- Keras == 2.1.2

代码文件说明

```
- EventExtraction/
    - data/
        - preprocess/（数据预处理）
            - CEC/（原数据文件）
            - dataset.json(实验数据集)
            - preprocess.ipynb(预处理、数据分析的代码)
        - data_load.ipynb(制作模型所需要的各项输入、输出矩阵)
        - *.npy(模型的输入与输出)
    - model/
        - img/(由keras自动生成的模型架构图)
        - model/（训练好的模型参数文件）
        - predict/（模型预测输出的矩阵）
        - dataset_split.py（训练集/测试集划分）
        - DMCNN.py（DMCNN模型与CNN模型）
        - TextCNN.py（TextCNN模型）
        - train.py（训练、预测所需的各项函数）
        - *.ipynb（训练过程、模型预测、性能结果等）
    - readme.md
```

项目简介

研究任务

对事件抽取（event extraction）任务的具体实践，在该任务中，涉及到两个概念：

1. 事件触发词（Event Trigger）

2. 事件元素词（Event Argument）

在本项目中，进行的任务是**对事件触发词的检测与分类（Trigger Identification & Classification）**，事件元素词（Event Argument）可作为辅助信息，帮助对事件触发词的检测。

具体地，项目将该任务等效为一个“序列标注”任务，例如，对句子`2014年1月7日，广州番禺市桥街兴泰路的商铺发生火灾。`来说，其分词后的结果为`2014 年 1 月 7 日 广州 番禺市 桥街 兴泰路 的 商铺 发生 火灾`，其中，`火灾`一词为事件触发词，其触发的事件类型为`emergency`，`商铺`一词为该`emergency`事件的元素词，用于辅助对`火灾`一词的检测与分类。下图描述了这一流程：

```
【 样本1 】2014/ 年 / 1  / 月  / 7  / 日 /广州 /番禺市/桥街/兴泰路/ 的 / 商铺
/发生 / 火灾
【期望输出】none/none/none/none/none/none/none/none /none/none /none/none
/none/Emergency
【辅助信息】none/none/none/none/none/none/none/none /none/none
/none/Argument/none/none
```

有一些句子中存在多个事件触发词，且一部分事件触发词没有元素词，如：

```
【 样本2 】媒体 / 通报     / 7  / 日晚/ 21 / 时 /40  / 分 / 警方/接到/群众/报警
【期望输出】
none/Statement/none/none/none/none/none/none/none/none/Operation
【辅助信息】none/none     /none/none/none/none/none/none/none/none/none
```

数据集

在事件抽取（Event Extraction）领域，研究中常用的数据集是ACE、KBP等标注语料。但经过调研，发现ACE 2005数据集并不免费；TAC KBP 2017数据集虽然免费，但需要进行比赛团队注册、签署严格的使用协议等繁琐的步骤。因此，最终选用了：中文突发事件语料库（Chinese Emergency Corpus）-上海大学-语义智能实验室，该数据集在github中进行开源，其简介如下：

> **中文突发事件语料库**
>
> 中文突发事件语料库是由上海大学（语义智能实验室）所构建。根据国务院颁布的《国家突发公共事件总体应急预案》的分类体系，从互联网上收集了5类（地震、火灾、交通事故、恐怖袭击和食物中毒）突发事件的新闻报道作为生语料，然后再对生语料进行文本预处理、文本分析、事件标注以及一致性检查等处理，最后将标注结果保存到语料库中，CEC合计332篇。
>
> CEC 采用了 XML 语言作为标注格式，其中包含了六个最重要的数据结构（标记）：Event、Denoter、Time、Location、Participant 和 Object。Event用于描述事件；Denoter、Time、Location、Participant 和 Object用于描述事件的指示词和要素。此外，我们还为每一个标记定义了与之相关的属性。与ACE和TimeBank语料库相比，CEC语料库的规模虽然偏小，但是对事件和事件要素的标注却最为全面。
>
> 具体内容可参见上海大学公开发表的相关硕士博士论文，以及期刊会议论文等。

模型介绍

DMCNN

本项目主要用 Keras 框架复现了`Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks`论文中所提出的DMCNN模型，该篇文章发表在2015年的ACL会议上，也是老师课堂讲授过的一篇论文，模型的基本结构如下图：

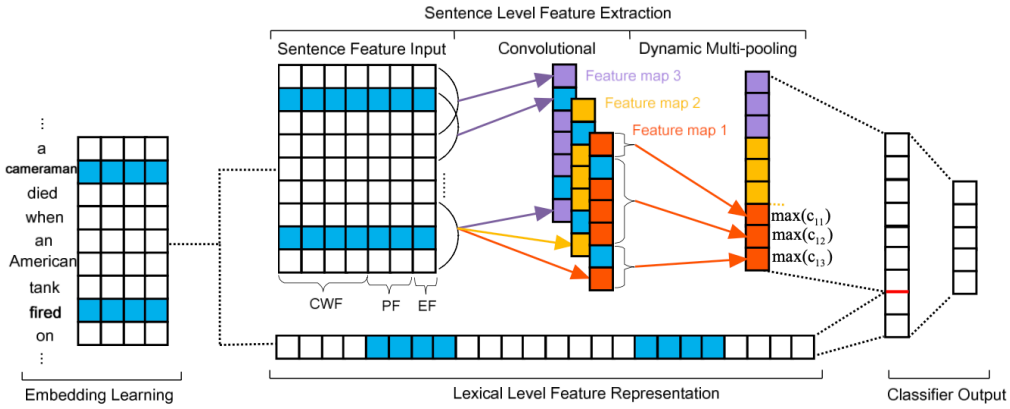

Figure 2: The architecture for the stage of argument classification in the event extraction. It illustrates the processing of one instance with the predict trigger *fired* and the candidate argument *cameraman*.

原文中的DMCNN模型，在ACE数据集上进行了**Trigger Identification & Classification**以及**Argument Idenfication & Role** 两个任务。在此项目中，由于CEC数据集所限（拥有Argument标注的语料很少），只进行**Trigger Identification & Classification**的任务，且将其视为一个序列标注任务。

Other baselines

除了DMCNN外，项目还使用两个baseline模型与之进行比较：

- CNN：即原论文中提到的`CNN`，它的输入与DMCNN相同，但没有使用`Dynamic Multi-pooling`，而采用TextCNN中通用的`max pooling`。**它用来衡量`Dynamic Multi-pooling`的重要性。**
- TextCNN：即常用作文本分类的`TextCNN`模型，它的输入只有词向量（即DMCNN中的`CWF`），而没有DMCNN中的`PF`、`Lexical Level Feature`等。**它用来衡量对于事件抽取任务而言，词级别的特征，以及位置信息等句子级别的特征的重要性。**

数据预处理

> 注：此部分的代码参见：`data/preprocess/preprocess.ipynb`

CEC数据集中的原始数据为`xml`文件，且每个`xml`文件里对应着一篇新闻（长文）的若干个句子，在预处理过程中，主要进行两个步骤：

1. xml文件格式转换为json文件格式，以方便用python进一步进行处理；
2. 将新闻长文分成若干个句子（分句是原数据集中标注好的），以采用“句子级别”的事件触发词检测。

示例

例如，原`xml`文件：

```
<Title>成都网友称震感强烈 女同事当即哭泣</Title>
<ReportTime type="absTime">2008年05月12日16:15</ReportTime>
<Content>
    <Paragraph>
        <Sentence>
            <Event eid="e1">
                <Time type="relTime" tid="t1">5月12日14时28分</Time>,
                <Location lid="l1">四川</Location>发生7.8级
                <Denoter type="emergency" did="d1">地震</Denoter>。
            </Event>
        </Sentence>
    </Paragraph>
    <Paragraph>
        <Sentence>
            <Event eid="e2">
                <Time type="absTime" tid="t2">15时50分</Time>，新民网
                <Participant sid="s1">记者</Participant>网上
                <Denoter type="action" did="d2">连线</Denoter>成都网友
                <Participant oid="o2">姚先生</Participant>
            </Event>。
        </Sentence>
```

转换为 `json` 文件：

```
{
    "event0": {
        "Denoter": {
            "#text": "地震",
            "@did": "d1",
            "@type": "emergency"
        },
        "Location": "四川",
        "Time": "5月12日14时28分"
    },
    "sentence": "5月12日14时28分 ， 四川 发生7.8级 地震 。"
},
```

最终得到的每条数据样本为：

```
{
    "sentence": "5月12日14时28分 ， 四川 发生7.8级 地震 。",
    "sentence_words": "5 月 12 日 14 时 28 分 ， 四川 发生 7.8 级 地震 。",
    "triggers": [
        {
            "event": "emergency",
            "event_trigger": "地震",
            "index_event": 1,
            "index_event_trigger": 13
        }
    ]
},
```


数据分析

> 注：经过处理后的数据文件，参见`data/preprocess/dataset.json`

经过预处理后，数据集共包含1665个句子，每个句子至少包含1个事件触发词。

事件类型

事件触发词的所属类型
有：`action`，`emergency`，`movement`，`operation`，`perception`，`stateChange`，`statement`以及`none`（无事件）。

1/1与1/N的样本统计

- 句子中只有1个事件触发词的（1/1）样本有765条，占比为46%
- 句子中至少有2个事件触发词的（1/N）样本有900条，占比为54%，其中最多的事件触发词有8个

事件元素词统计

在1665个句子中，一共包括了3406个事件触发词，其中有1875个触发词没有元素词，占比为55%；其余的1531个触发词拥有元素词（且元素词均为1个），占比为45%

最大句子长度

在1665个句子中，经分词处理后，句子中词数最多的样本包含了85个词，因此设置最大句子长度为85

训练集/测试集划分

在1665个句子中，句子中含有的事件触发词个数有：1个，2个，...，8个。在该项目中，根据句子中含有的触发词数量进行分层抽样，划分训练集/测试集的比例为4:1

性能结果

> 注：预测结果，参见`model/*.ipynb`

Lexical-Level

测试集中共有333个句子，每个句子有若干词语（最大词语数为85）。

Lexical-Level指的是：把所有句子拆分为词语后，在**词语级别**进行检测与分类。对于**Identification任务**而言，每个词语有两种类型：`是/否 为事件触发词`，即是一个二分类任务；对于**Classification任务**而言，每个任务有八种类型：`action`，`emergency`，`movement`，`operation`，`perception`，`stateChange`，`statement`以及`none`（无事件），即是一个八分类任务。

下表展示了三个模型的性能：

	Identification			**Classification**		
	Macro Precision	Macro Recall	Macro F1-Score	Macro Precision	Macro Recall	Macro F1-Score
TextCNN	0.4881	0.5000	0.4940	0.1220	0.1250	0.1235
CNN	0.9992	0.9680	0.9831	0.5137	0.4636	0.4507
DMCNN	0.9992	0.9665	0.9822	**0.6240**	**0.4929**	**0.4906**

可以看到：

- TextCNN的性能非常差，明显劣于CNN与DMCNN，说明
  - TextCNN模型虽然在文本分类任务中被广泛使用，但也许不适用于本项目中这一类似于"序列标注"的任务。
  - TextCNN模型的输入只有词向量，而忽略了事件触发词、事件元素词的位置关系、上下文关系。因此：对事件抽取任务而言，同时利用事件触发词、元素词的位置关系（即原论文中提到的`句子级别的特征`），以及上下文关系（即原论文中提到的`词级别的特征`，特指触发词、元素词左右两边的词）是十分重要的。
- CNN与DMCNN相比
  - 二者在**Identification任务**上表现性能相似，而DMCNN在**Classification任务**上性能卓越，F1值比CNN高出将近4个点。
  - CNN与DMCNN的唯一区别，即是否使用论文中提出的`Dynamic Multi-Pooling`，此结果也表明由事件触发词、事件元素词所切分而形成的`Dynamic Multi-Pooling`对于事件抽取任务的重要性。

Sentence-Level

测试集中共有333个句子，每个句子有若干词语（最大词语数为85），其中至少包含一个事件触发词。

Sentence-Level指的是：对于**整个句子**而言，进行检测与分类。对于**Identification任务**而言，对一个句子样本而言，当且仅当对其所有词语的检测均正确时，才判为检测正确；对于**Classification任务**而言，对一个句子样本而言，当且仅当对其所有词语的分类均正确时，才判为分类正确。

下表展示了三个模型的性能：

	Identification Accuracy	Classification Accuracy
TextCNN	0	0
CNN	0.8799	0.3453
DMCNN	0.8769	0.3453

结果与Lexical-Level相似：TextCNN性能很差，对333个样本，全部未能检测；而CNN与DMCNN模型的性能相似。