README

Event Extraction

获取词向量，可以使用中文预训练模型，也可以参照 DMCNN 中的方法自行训练 Skip-gram

进行事件触发词抽取，即从文本中抽取标识事件发生的触发词，触发词往往为动词和名词

进行事件论元抽取，即从文本中抽取触发词所对应的事件论元，论元主要为主体、客体、时间、地点，其中主体为必备论元

可以使用 Pipeline 模型也可以使用 Joint 模型

本次实验使用数据集来自2020科大讯飞事件抽取挑战赛初赛，数据为json格式，提供了文本以及相应的触发词、主体、客体、时间、地点，其中除触发词和主体以外，其他为可选字段；数据中给出的 distant_trigger 只是远程监督标签，并不是真实的标签，真实的触发词标签是 trigger；数据集已经分为了训练集、开发集和测试集，但由于是比赛数据，因此测试集没有提供标签，请同学们在开发集上进行评测

Task description

- Identify event triggers and corresponding arguments in given text
- One sentence can have more than one event
- This implementation identifies event triggers and different types of argument, but does *not* align arguments with events

Implementation detail

- the whole pipeline is formed as a two-stage sequence-labeling
    - the first stage concerns the identification of events
    - the second stage concerns the identification and classification of arguments
        - in the second stage, the identified event triggers in the first stage is marked by `<event> trigger <event/>` in the test stage
- Chinese BERT `hfl/chinese-bert-wwm-ext` is used as the default backbone model

How to run

- Fill in TODO part in `preprocess.py` and `util.py`

Data Preparation

- put data in `data/raw` folder
- run `preprocess.py`, the results are `data/processed/argument` and `data/processed/trigger` folders, each containing `train.txt` and `dev.txt`
    - process original `train.json`, `dev.json` into data format of `char label`
    - Since an event trigger or argument may span over several tokens, the labels are in the form of `IOB2`

Train

1. run `python -u main.py --mode trigger` to run the first stage
2. run `python -u transform.py` to generate test file for second stage, after running you will see `data/processed/argument/test.txt`. **Note that if you are not using IOB2 format, you will need to rewrite some parts in `transform.py`**
3. run `python -u main.py argument` to run the second stage