GPT2-Chinese 模型训练

1. 训练过程

1.1 小说部分

1.1.1 语料

- 斗破苍穹
- ~~试了一下古诗词，但是语料过少，导致生成结果乱码~~
- 训练集大小： 8w+ 行， 共16.1M

1.1.2 训练

- 预训练 - 使用默认循环次数

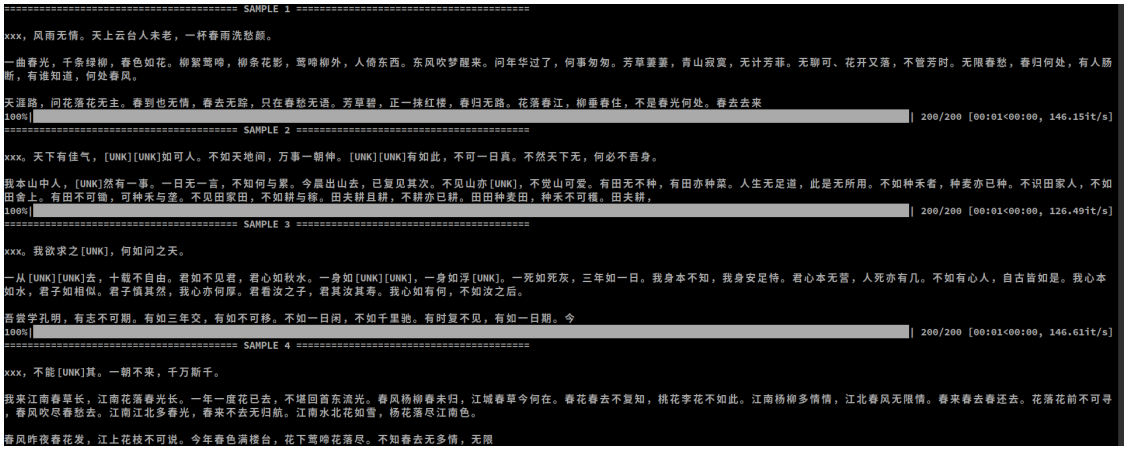
```
# 开始训练命令
python train.py --raw --raw_data_path='./data/doupo.txt'
```

  - 默认循环次数为 5
  - 修改 raw_data_path 路径为其他 默认 json 文件 "不能用" - 我也不知道为什么

```
# 模型中给出的代码：
lines = json.load(f)
# 修改后的代码：
lines = f.readlines()
```

- 生成结果：

```
# 生成文本命令
python ./generate.py --length=500 --nsamples=4 --prefix=开始 --
fast_pattern --save_samples --save_samples_path=./generate
```



2. 问题汇总

2.1 文件读取

```
# 模型中给出的代码：
lines = json.load(f)
# 修改后的代码：
lines = f.readlines()
```

- **问题描述：** 原模型中使用 train.json 文件进行多个文件的读取，但是实际只读取了 json 文件本身。
- **解决方法：** 修改默认读取文件
- **出现原因：** `待解决`

2.2 模型构造参数

- **./GPT2-Chinese/config/model_config.json**
- **./GPT2-Chinese/config/model_config_small.json**

```
# 原始值
{
  "initializer_range": 0.02,
  "layer_norm_epsilon": 1e-05,
  "n_ctx": 1024,
  "n_embd": 768,
  "n_head": 12,
  "n_layer": 12,
  "n_positions": 1024,
  "vocab_size": 21128
}

#修改值
{
  "initializer_range": 0.02,
  "layer_norm_epsilon": 1e-05,
  "n_ctx": 1024,xunlian
  "n_embd": 768,
  "n_head": 12,
  "n_layer": 12,
  "n_positions": 1024,
  "vocab_size": 21128
}
```

- **问题描述：** 使用原模型中参数，预训练过程中内存分配不足，卡死
- **解决方法：** 缩小构造参数值
- **出现原因：** 原模型中构造参数过大，更深一点的原因正在看一些 GPT2 的文章

3. 本周要做的事

- 查找一些 GPT2 相关的文章和论文进行阅读；
- 看一下模型源码；

- 重新找一些其他语料进行训练；
- 继续学习 bert 和 pytorch。

补充部分

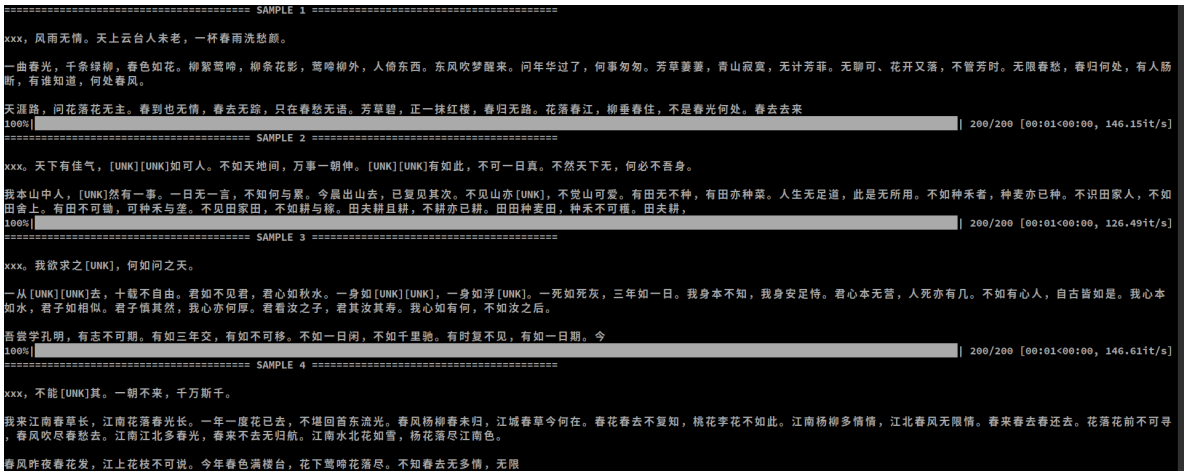
1.2 古诗词部分

1.2.1 语料

- 古诗词集
- 训练集大小： 共85w+ 行， 187M

1.2.2 训练

- 预训练 -- `使用默认循环次数-5`
  - **同上**
- 生成结果



附录：

```python
# data_processing.py -- 古诗词数据处理函数
import os


def getfilelist(rlist, path):
    # 获取 path 目录下所有文件
    for dir, floder, file in os.walk(path):
        for i in file:
            t = "%s/%s" % (dir, i)
            rlist.append(t)
    return rlist


def save2file(path):
    # 读取 && 过滤 && 保存 数据
    print('---------  Processing start  ---------')
    s = ''  # 存放文件中读取的数据
    number = 0
    for filename in getfilelist([], './data'):
        print('Current file number : %d <--> Current file : %s' %
              (number, filename))
        number += 1
```

```
        with open(filename, 'r', encoding='utf8') as f:
            next(f)  # 过滤掉第一行无用数据
            for line in f.readlines():
                line = line.replace('"', '')  # 过滤掉双引号 "
                s += line.split(',')[-1]  # 获取古诗句 过滤掉取用数据
    with open(path, 'w') as f:
        f.write('%s' % s)  # 读取数据写入文件中
    print('---------  Processing completed  ---------')


def main():
    save2file('./poetory.txt')

if __name__ == "__main__":
    main()

```