

英译中翻译系统 项目复现记录

1. 寻找可复现源码并下载到本地

1.1 问题：寻找到合适的项目

解决方法：

1. 观察提供的数据集，寻找数据集中英文的排放规律

对于数据集的学习：

训练集：用于训练机器学习模型，使得模型获得集中的样本，学习到数据的各种特征和模式。

验证集：在训练过程中，可以使用验证集来评估模型的性能，即提供一个独立的数据集来评估模型在从未见过的数据上的表现，辅助我们构建模型。有些类似于测试集。

测试集：用来最终评估模型的效果。但是不能用于调整参数或者选择模型，否则会倒是过拟合。它只用于评估模型最终性能。

一般先使用训练集，再使用验证集，最后使用数据集。

2. 在CSDN上找到数据集排列规律最为相似的一个项目：

[机器翻译实战（英译汉）Transformer代码学习详解 transformer模型英译中-CSDN博客](#)

1.2 问题：把项目放在本地

解决方法：

1. 新建一个想要保存github项目的文件目录
2. 右键，选择 Git Bash Here
3. GitHub右上角，复制地址
4. 回到Git Bash窗口，用git clone把项目pull下来即可，如图

```
Sandiverse@Sandust MINGW64 /d/python/TranslationTask
$ git clone https://github.com/hinesboy/transformer-simple.git
Cloning into 'transformer-simple'...
remote: Enumerating objects: 58, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 58 (delta 0), reused 3 (delta 0), pack-reused 55
Receiving objects: 100% (58/58), 580.24 KiB | 1.05 MiB/s, done.
Resolving deltas: 100% (6/6), done.

Sandiverse@Sandust MINGW64 /d/python/TranslationTask
$ |
```

2. 跑项目

不管怎么说，我觉得把项目拉下来，一个个去解决报错就行。

先随便拿一个虚拟环境，在pycharm开一个项目，看看有什么地方会报错。

问题：torch报错

鉴于用到了pytorch，我对pytorch也进行了初步学习，详情见[另一个笔记](#)。

解决方法：

```
conda install torch
```

问题：项目跑起来没反应 也不报错

解决方法：逐个点开项目里的所有代码，检视什么地方标红。这当中碰到了很多不懂的地方。

一个python项目里的lib文件夹是用来做什么的？

lib文件夹一般用于python开发中，存放程序中常用的自定义模块。

在这个项目中，是**损失函数**，**优化器**等存放的位置。

相关学习记录如下：

损失函数顾名思义就是函数，在机器学习里一般用来看预测值和真实值之间的差距大小，便于调整模型的参数，损失函数最小时，模型的泛化能力一般来说是最强的（如果不考虑过拟合的话）

梯度下降：先求出所需要修改的权重关于损失函数的偏导，得出的导数就是梯度，利用梯度对权重参数进行更新，不断下降，使得损失函数达到最小值，简而言之就是沿着梯度的反方向下降，一直到权重关于损失函数的偏导为0，我们就找到了损失函数的最小值。这整个过程叫做梯度下降。

反向传播：训练过程中，根据损失函数对神经网络中权重的梯度——也就是偏导数——进行计算，利用链式法则计算每一层的梯度，一直把梯度传播到第一层。面对复杂的神经网络，可能有几百个权重，要一个个写出解析式几乎是不可能的，因此我们需要一种算法，把复杂的神经网络看作是一个图，我们可以在图上传播梯度，最终根据链式法则，把梯度求出来。简而言之就是个倒着回去用链式法则求偏导的过程。

lib库没有明显报错。直接去看model

model库环境配置问题解决

问题1：

```
from utils import clones 报错
```

解决方案：

1. 查询utils：一个实用的工具包，提供了方便的函数和类，提供数据处理等功能。
2. 如图

```
(env_first) C:\Users\Sandiverse>pip install utils
Collecting utils
  Downloading utils-1.0.2.tar.gz (13 kB)
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: utils
  Building wheel for utils (setup.py) ... done
  Created wheel for utils: filename=utils-1.0.2-py2.py3-none-any.whl size=13934 sha256=6ad9853d2fc1bf16ad99ea5fbb332092d8cfff51c95a6fd11d39335299995539
  Stored in directory: c:\users\sandiverse\appdata\local\pip\cache\wheels\15\0c\b3\674aea8c5d91c642c817d4d630bd58faa316724b136844094d
Successfully built utils
Installing collected packages: utils
Successfully installed utils-1.0.2
```

问题2:

```
from nltk import word_tokenize 报错
```

```
ModuleNotFoundError: No module named 'nltk'
```

解决方案:

1. 查询nltk: 一个广泛使用的自然语言处理工具库, 此处的 `word_tokenize` 用来将句子分词为单个, 应该是用于构造词典。
2. 如图

```
(env_first) C:\Users\Sandiverse>pip install -i https://pypi.tuna.tsinghua.edu.cn/simple/ nltk
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple/
Collecting nltk
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/a6/0a/0d20d2c0f16be91b9fa32a77b76c60f9baf6eba419e5ef5deca17af9c582/nltk-3.8.1-py3-none-any.whl (1.5 MB)
    1.5/1.5 MB 6.4 MB/s eta 0:00:00
Collecting click (from nltk)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/00/2e/d53fa4befbf2cfa713304affc7ca780ce4fc1fd8710527771b58311a3229/click-8.1.7-py3-none-any.whl (97 kB)
    97.9/97.9 kB ? eta 0:00:00
Requirement already satisfied: joblib in d:\anaconda3\envs\env_first\lib\site-packages (from nltk) (1.2.0)
Collecting regex<=2021.8.3 (from nltk)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/a8/01/18232f93672c1d530834e2e0568a80eaab1df12d67ae499b1762ab462b5c/regex-2023.12.25-cp311-cp311-win_amd64.whl (269 kB)
    269.5/269.5 kB ? eta 0:00:00
Collecting tqdm (from nltk)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/2a/14/e75e52d521442e2fcc9f1df3c5e456aead034203d4797867980de558ab34/tqdm-4.66.2-py3-none-any.whl (78 kB)
    78.3/78.3 kB 2.2 MB/s eta 0:00:00
Collecting colorama (from click->nltk)
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/d1/d6/3965ed04c63042e047cb6a3e6ed1a63a35087b6a609aa3a15ed8ac56c221/colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Installing collected packages: regex, colorama, tqdm, click, nltk
Successfully installed click-8.1.7 colorama-0.4.6 nltk-3.8.1 regex-2023.12.25 tqdm-4.66.2
```

问题3:

报错

```
UnicodeDecodeError: 'gbk' codec can't decode byte 0x80 in position 8: illegal
multibyte sequence
```

解决方案:

1. 查询原因:

错误的意思是: Unicode的解码 (Decode) 出现错误了, 以 `gbk` 编码的方式去解码 (该字符串变成Unicode), 但是此处通过 `gbk` 的方式, 却无法解码 (can't decode) ."illegal multibyte sequence"的意思是非法的多字节序列, 也就是说无法解码了。

2. 尝试添加encoding方式, 变为更大范围的gb18030:

```
def load_data(self, path):
    en = []
    cn = []
    with open(path, 'r', encoding='gb18030') as f:
        for line in f:
            line = line.strip().split('\t')

            en.append(["BOS"] + word_tokenize(line[0].lower()) + ["EOS"])
            cn.append(["BOS"] + word_tokenize(" ".join([w for w in line[1]])) + ["EOS"])
```

3. 尝试使用utf-8编码, 成功引发下一个报错。

问题4:

如下报错:

```

LookupError:
*****
Resource punkt not found.
Please use the NLTK Downloader to obtain the resource:

>>> import nltk
>>> nltk.download('punkt')

For more information see: https://www.nltk.org/data.html

Attempted to load tokenizers/punkt/english.pickle

Searched in:
  - 'C:\\Users\\Sandiverse\\nltk_data'
  - 'D:\\ANACONDA3\\envs\\env_first\\nltk_data'
  - 'D:\\ANACONDA3\\envs\\env_first\\share\\nltk_data'
  - 'D:\\ANACONDA3\\envs\\env_first\\lib\\nltk_data'
  - 'C:\\Users\\Sandiverse\\AppData\\Roaming\\nltk_data'
  - 'C:\\nltk_data'
  - 'D:\\nltk_data'
  - 'E:\\nltk_data'
  - ''
*****

```

解决方法：

在 `prepare_data.py` 下添加两行

```

import nltk
nltk.download('punkt')

```

成功开始下一个报错

问题5：

```

RuntimeError: CUDA error: invalid device ordinal
CUDA kernel errors might be asynchronously reported at some other API call, so the stacktrace below might be incorrect.
For debugging consider passing CUDA_LAUNCH_BLOCKING=1.
Compile with 'TORCH_USE_CUDA_DSA' to enable device-side assertions.

```

解决方法：

改parser.py里的gpu卡号，绷不住了，终于开始训练了。

问题6：

```

RuntimeError: Parent directory save does not exist.

```

解决方法：

自己改了一下args内的文件保存路径，作者貌似使用的是linux的文件路径。

之后完成了训练，没有报错。

问题7:

验证的翻译效果不太令人满意，不知道是什么情况。

解决方法:

在参数中加深transformer层数和加大循环次数。

问题8:

怎么使用BLEU分数评价结果？

解决思路:

1. 首先需要有测试集，其中包含英文原文和对应的高质量中文参考翻译。
2. 翻译测试集，使用模型将测试集中的英文原文翻译成中文
3. 对数据进行预处理，可能需要对翻译结果和参考翻译进行一些预处理，比如去除多余的空格和标点符号，统一文本格式之类的。
4. 计算BLUE分数，有很多现成的工具和库可以用来计算，我选用的项目中恰好用到了 `nltk` 库，其中的 `bleu_score` 模块就可以用来计算BLUE分数。
5. 获得每一个句子的BLUE分数平均值即可。

解决方法:

1. 我需要对数据进行预处理，因此可以通过参考项目中如何对原文和译文进行处理。
在此之前，看一下transformer到底是如何进行运作的。
在原项目代码文件中加入了注释。
看了很久模型的代码实现，觉得有点浪费时间，还是得以解决问题为导向，碰到问题了再去学习相关知识。决定直接着手利用测试集计算bleu分数。
2. 在 `run.py` 文件中添加了一部分代码，参考了 `evaluate` 模块的写法，应该只需要参照evaluate内的大部分写法，就能测试出bleu分数。
3. 在 `prepare_data.py` 的 `__init__` 内添加对于 `test.txt` 的处理。在 `parser.py` 内添加 `test.txt` 的路径。
4. 自己写了一个 `test.py`，用于计算BLUE分数

问题9

计算分数时报错为KeyError

解决方法:

由于无法找到索引中的内容，选择将其替换为UNK。

如下代码:

```
sym = data.cn_word_dict.get(out[0, j].item(), '<UNK>')
```

问题10

BLEU分数跑出来实在是太低了。现在有若干解决方法:

1. 修改参数，加长模型训练时间和加深模型
2. 修改分词，使用jieba分词器（由于最后评估的是中文句子）

尝试解决1:

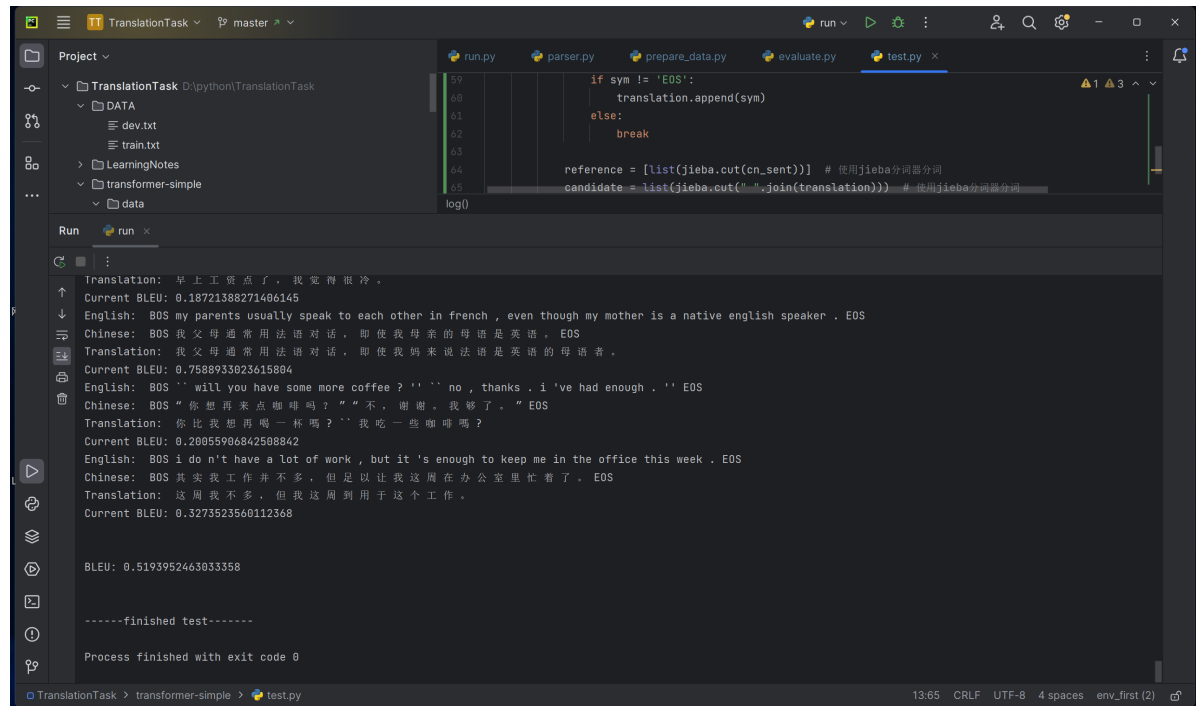
导入jieba库，尝试使用特殊的分词方法。

效果并不明显。

尝试解决2:

加大训练次数，多加了两层transformer，loss从2.0降到了大概0.11。

BLEU分数显著提高，基本上稳定在0.5左右。如图。



```
run.py  parser.py  prepare_data.py  evaluate.py  test.py x
Project v
  TranslationTask D:\python\TranslationTask
    DATA
      dev.txt
      train.txt
    LearningNotes
    transformer-simple
      data
Run run x
Translation: 早上工资点了，我觉得很冷。
Current BLEU: 0.18721388271406145
English: BOS my parents usually speak to each other in french , even though my mother is a native english speaker . EOS
Chinese: BOS 我父母通常用法语对话，即使我母亲的母语是英语，EOS
Translation: 我父母通常用法语对话，即使我母亲来说法语是英语的母语者。
Current BLEU: 0.7588933023615804
English: BOS `` will you have some more coffee ? `` `` no , thanks . i 've had enough . `` EOS
Chinese: BOS " 你想再来点咖啡吗？ " " 不，谢谢。我够了。 " EOS
Translation: 你比我想再喝一杯吗？ `` 我吃一些咖啡吗？
Current BLEU: 0.20055906842508842
English: BOS i do n't have a lot of work , but it 's enough to keep me in the office this week . EOS
Chinese: BOS 其实我工作并不多，但是足以让我这周在办公室里忙着了。EOS
Translation: 这周我不多，但我这周到用于这个工作。
Current BLEU: 0.3273523560112368

BLEU: 0.519395246303358

-----finished test-----

Process finished with exit code 0
TranslationTask > transformer-simple > test.py 13:65 CRLF UTF-8 4 spaces env_first (2)
```