

# 项目总结与回顾

## 一、项目结果评估

### 1.1 完成项目时遇到的问题？

#### ① loss 为 Nan 的情况

解决思路：1、在 github 的 issues 上提问和查看，链接 <https://github.com/abisee/pointer-generator/issues>

2、一般在 loss 部分出现问题 or attention 那里可以看看

#### ② 对数据的足够重视（任务的好坏往往与数据有关）

解决思路：1、学会运用数据的能力与思路。例如：标注数据不够，可以采用 VAE：数据 $\rightarrow$ encoder $\rightarrow$ 语义 $\rightarrow$ 反向 encoder 还原 $\rightarrow$ 还原数据，求数据与还原数据的相似度。之后拿 encoder 做模型进行 finetune.

2、公安、法律场景：数据很少、没有公开数据集（采用规则去做，收集到一定数量，再用模型）

3、数据增强的重视（回译、eda、uda）

#### ③ Model 的上限：基本上有对模型上限的认知，可用 baseline 跑跑

#### ④ NLG 的主线路

预训练模型：bert、gpt2、xlnet

生成问题 trick: 重复、OOV 等

生成问题采样：参看 gpt2 的采样代替 beam\_search

工具使用：transformers、bert4keras

#### ⑤ 竞赛的问题

一般套路：模型融合（工作中难以用到）

### 1.2 拿到数据后该如何思考？

① 统计文本长度比率、label 的比率、去除违规数据（业务）、挖掘 badcase（评估指标低的、错的）等操作

### 1.3 NLP 的基本流程

答：数据（数据认知能力） $\rightarrow$ 模型（选择 baseline 的能力） $\rightarrow$ badcase (R\_L<xx, 分析 badcase 能力) $\rightarrow$ 看论文（解决方案能力） $\rightarrow$ 模型(改进模型能力) $\rightarrow$ 上线

### 1.4 特征挖掘、融合

## 二、模型部署相关知识

### 2.1 tensorflow 模型部署

#### ①checkpoint 结构

checkpoint 只会保存参数信息，因此部署时也需要将这段源码也要带上，额外保存代码结构在 pt 文件中。

```

checkpoint
ckpt-10.data-00000-of-00002
ckpt-10.data-00001-of-00002
ckpt-10.index
ckpt-11.data-00000-of-00002
ckpt-11.data-00001-of-00002
ckpt-11.index
ckpt-12.data-00000-of-00002
ckpt-12.data-00001-of-00002

```

参数

```

v pb_model
v 1
v variables 模型
variables.data-00000-of-00001
variables.index
saved_model.pb

```

## ②Flask 方式

```

from flask import Flask, render_template, request, jsonify
import tensorflow as tf
import execute

app = Flask(__name__)

@app.route("/")
def index():
    return render_template("index.html")

@app.route("/message", methods=['POST'])
def reply():
    sentence = request.form['msg']
    translate, _ = execute.translate(sentence)
    translate = ' '.join(translate.strip().split(' ')[:-1])
    print(sentence)
    print(translate)
    return jsonify(text=translate)

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=12345)

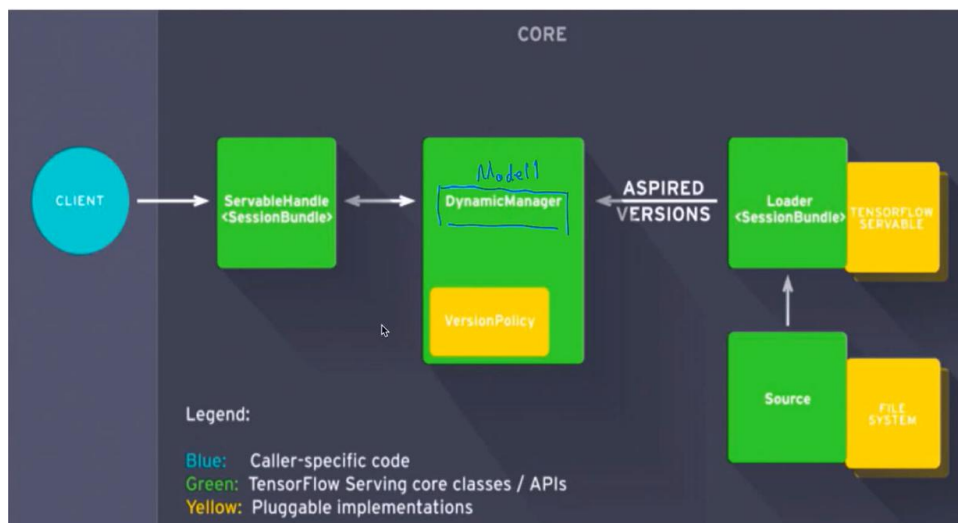
```

一般与 docker 一起用，docker 是一个容器或者一个虚拟环境，把这一段用的依赖包全部安装在当前 docker 下，然后 docker 压缩成镜像，上传到 dockerhub 上去，也可以 docker image 把镜像拉下来。

K8s 可以做集群，管理 docker,把死掉的服务拉起来，也可以用上。

缺点：太慢

### ③tf-serving



做法：①docker pull tensorflow/serving (docker 去拉官方的 tf-serving)  
 ②TESTDATA="\$HOME/.xxx/testdata" (定义 testdata 路径)  
 ③docker runn -t --rm -p 8501:8501 #启动服务  
 -v "\$TESTDATA/saved\_model\_half\_plus\_two\_cpu:/models/half\_plus\_two (模型目录)"  
 -e MODEL\_NAME=half\_plus\_two #模型名  
 Tensorflow/serving &

### 三、项目全方位总结

- ① ocr+语言模型
- ② 纠错模型: pycorrector
- ③ 魔改: github:425776024/bertsum-chinese

