梁博云实习第一个项目【发现负面新闻】
实习目标
1）学习和掌握 json 代码的解析和生成
2）学习一种 http 服务的方式
3）学习如何使用开放 API
4）作业可以交流在群里讨论，可以上网搜索，请教他人，不要抄袭他人代码，自己独立完成。

第一步、学习和利用现有一个情感计算 API
http://www.pullword.com/baobian/
调用样例：
Linux 命令行下执行
curl -X POST 'http://baobianapi.pullword.com:9091/get.php' -d'十一月再见，十二月你好，2021 年最后一个月总会有不期而遇的温暖，和生生不息的希望' –compressed

返回的 json 结果要能解析，大于 0.5 表示正面情感，小于-0.5 表明负面情感。

或者使用其他自己熟悉的 API 也可以。推荐使用梁博公开的服务。

第二步，学习提供 http 服务的工具
C 语言选手可以通过搜狗工作流开源工具实现（参考文档见最后）。不会 C 和 C++的同学可以用 php 代码或者其他自己熟悉的语言实现。

第三步，学习和了解数据源
http://news.baidu.com/
从百度新闻首页获得当天热门新闻，用 curl 或者 wget 命令采集首页。然后提取其中的新闻标题和 URL

第四步，将今日新闻标题调用情感计算 API 得到正负面评价，提取负面结果的新闻标题和 URL，通过第二步掌握的 http 服务工具以 json 格式对外展示。
展示的内容包括标题和 URL 的列表。在自己的 linux 机器上能访问即可，有公网服务器的可以开放公网 URL 供我们检查，没有公网服务器的截图即可。

最终提交全部项目数据 1）代码　　2）可以直接访问的公网服务地址或截图
到助理邮箱 zdbb_yy@163.com。
邮箱标题格式【梁博云实习 A 班第一次作业，提交人：XXX】
最晚提交时间 5 月 8 日，截止这个时间还没提交，请自动退群。

一、搜狗工作流（http server）开源工具安装和入门

1） 预备安装一些必要的包

yum install git cmake cmake3 openssl-devel gcc-c++

2） 下载源码

git clone https://github.com/pennyliang/workflow.git

3） 编译安装

cd /workflow/

./configure

make

make install

4） 编译和运行第一个代码

在 tutorial 目录下找到样例代码 tutorial-04-http_echo_server.cc

用下面命令直接编译

g++ -std=c++11 -I /opt/sogou/include/ -o tutorial-04-http_echo_server tutorial-04-http_echo_server.cc /opt/sogou/lib64/libworkflow.a -lssl -lpthread -lcryptols

也可以直接在 tutourial 目录下运行 make 编译，上面这个编译命令方便你以后在自己的项目中编译。

运行这段代码（让这个服务开在 9090 端口上）

./tutorial-04-http_echo_server 9090

在这台机器上在运行，就可以看到正常工作了

curl "http://127.0.0.1:8080/index.html"

到这一步，基本上可以算是最低代价把这个东西算跑起来了。

二、制作一个 http web 服务

我在 2014 年用 C++自制了一个类似的 web 服务，用于跑 pullword 分词服务，这样的好处全是 C 代码，比较方便和我写的数据库联合编译，调试也方便。因为有了这个我尝试用这个来改造我的分词代码。

改动非常小。我这里只记录几个重要的改动。

只改了两个文件，一个是 main 文件，这个主要参考 tutorial-04-http_echo_server.cc，其中有一个设置比较重要 settings.handler_threads = 10; 这个控制工作线程的数量，不超过 CPU 核的 2 倍比较好，如果有大量 IO 操作，可以适当提高倍数。

另一部分就是改工作线程的回调函数，这部分改动就大了，依次如下

1） 取 request 的部分
我们知道如果 request 的内容很大，http 协议会分包，这部分这个框架已经做好了，不用考虑。protocol::HttpRequest *req = server_task->get_req(); 这个函数返回就表明所有的 request 的字符串都接收完整了。

2）发送数据

```
protocol::HttpResponse *resp = server_task->get_resp();
server_task->set_send_timeout(3*1000) ;
resp->set_http_version("HTTP/1.1");
resp->set_status_code("200");
resp->set_reason_phrase("OK");
resp->add_header_pair("Content-Type", "text/json;;charset=UTF-8");
resp->add_header_pair("Server", "Sogou WFHttpServer");
resp->append_output_body(bufs,read);
```

这里面值得说的是函数 append_output_body，第一个参数是插入发送缓存的数据，read 是插入发送缓存的这段缓存 bufs 的长度。这个函数方便之处是可以多次调用，不断追加的方式写。

程序返回后，系统会自动按照 3 秒超时的约定把数据发射出去。如果需要对超时需要使用 set_callback 函数，详见 http proxy 那个 tutorial 例子。