ISOA Progress Report

Group ”200”
YiDai 2017013562
ZhaohengLi 2017050025

2020 年 3 月 15 日

1 项目介绍

我们以疫情部分的舆论方面为核心，提取舆论主题，结合时间线展示群众的关注点变化，对访问者提供舆论谣言判断服务。同时我们也注意到，信息服务平台如果缺失一方面的关键信息很容易丧失吸引力，我们还会实现政府的相关措施发布、并实现确诊数据展示。

主要实现目标如下：

- **疫情展示** 主要包括疫情确诊地图、详细数字和相关新闻或政策的时间线。
- **舆情关注展示** 对舆情进行主题提取，结合时间线展示群众的关注点变化
- **虚假新闻/谣言判断及搜索** 用户输入文本，对文本进行分析，为用户提供相关已证实谣言或者相关新闻，并给出用户输入文本为谣言的概率。

周计划制定如下：

Week	内容	成员
1/2	确定选题、了解可用资源和技术	全组
3/4	前端实现政策时间线、疫情地图等数据展示	戴翼
	后端实现疫情、政策基本数据接口	李曌珩
	完成疫情舆论数据集的构建	戴翼
5	前后端对接，实现 Demo	全组
6	对谣言判断模型进行训练和调整，提高命中率	戴翼
	后端配合实现关注点时间线的支持	全组
	谣言判断模型的部署上线	李曌珩
7/8	前端接入接口，实现谣言判断	戴翼
	准备 Demo 展示	全组
9/10	对前端视图进行优化	戴翼
	后端的细节调整	李曌珩
11/12	前端优化界面，分析可选功能，进行初步设计	戴翼
	后端实现日志监控	李曌珩
13/14	可选任务的合作完成	全组
	后端实现高可用负载均衡热更新	李曌珩
15	可选任务和基础任务的合并测试、分析、取舍	全组
	最新版本的测试	
	报告的准备	
16	Final Report	全组

2 后端部分进展

2.1 后端架构

为了降低项目的耦合性，后端使用 Web 应用程序框架 Flask 进行实现和封装。通过网络接口，可以使前端获得所需要的数据而不用关心后端的实现方法。

我们的项目所提供的信息时效性很强，为了提高服务的稳定性和实效性，我们使用了 MySQL 数据库存储收集到的数据，并且已经部署了数据定时更新的工作。

2.2 数据处理

目前收集到并整理出的数据有

- **疫情数据** 自 2020 年 1 月 22 日至今，中国所有省份、地区或直辖市及世界其他国家的所有疫情信息变化的时间序列数据（精确到市），能够追溯确诊/疑似感染/治愈/死亡人数的时间序列。疫情数据目前已经完成整理，并提供给前端供疫情可视化部分使用；
- **新闻数据** 自病发开始各个省份、地区或直辖市及世界其他国家关于疫情所发布的新闻。新闻数据目前已经按照发布时间，发布地点进行了分类整理。
- **谣言数据** 与疫情有关的谣言以及丁香园的辟谣信息。谣言数据目前已经按照“谣言”、“权威数据”和“未证实”的标签对内容进行了分类处理。

2.3 接口处理

目前后端获得的数据已经比较完善，因此计划中的接口已经全部实现。下面为列举的几个主要接口：

- **route('/getDataSummary')** 提供后端数据收集情况概览
- **route('/getTimeData')** 根据时间参数提供全国当天的疫情发展数据
- **route('/getDataPos')** 根据地区参数提供该地区的疫情历史发展数据
- **route('/getNewsData')** 根据地区和时间参数获取某个地区某个时刻的新闻列表
- **route('/getRumorData')** 根据内容类型参数提供该类型的谣言数据
- **route('/getMap')** 提供疫情分布的地图数据

2.4 遇到的问题

目前后端进展比较顺利，没有遇到较大的问题，完成了预期的计划内容。

2.5 未来计划

接下来的工作是我们的项目的核心，涉及到“提取舆论主题”、“结合时间线展示群众的关注点变化”以及“谣言判断”等相关工作，要使用神经网络模型对我们收集到的数据进行主题提取和内容分类等一系列操作。

对于后端数据部分，原来制定的计划不会有大的改变，所以接下来几周的重点会放在“舆论数据的收集和进一步的整理”以及“神经网络模型判断谣言“两部分。

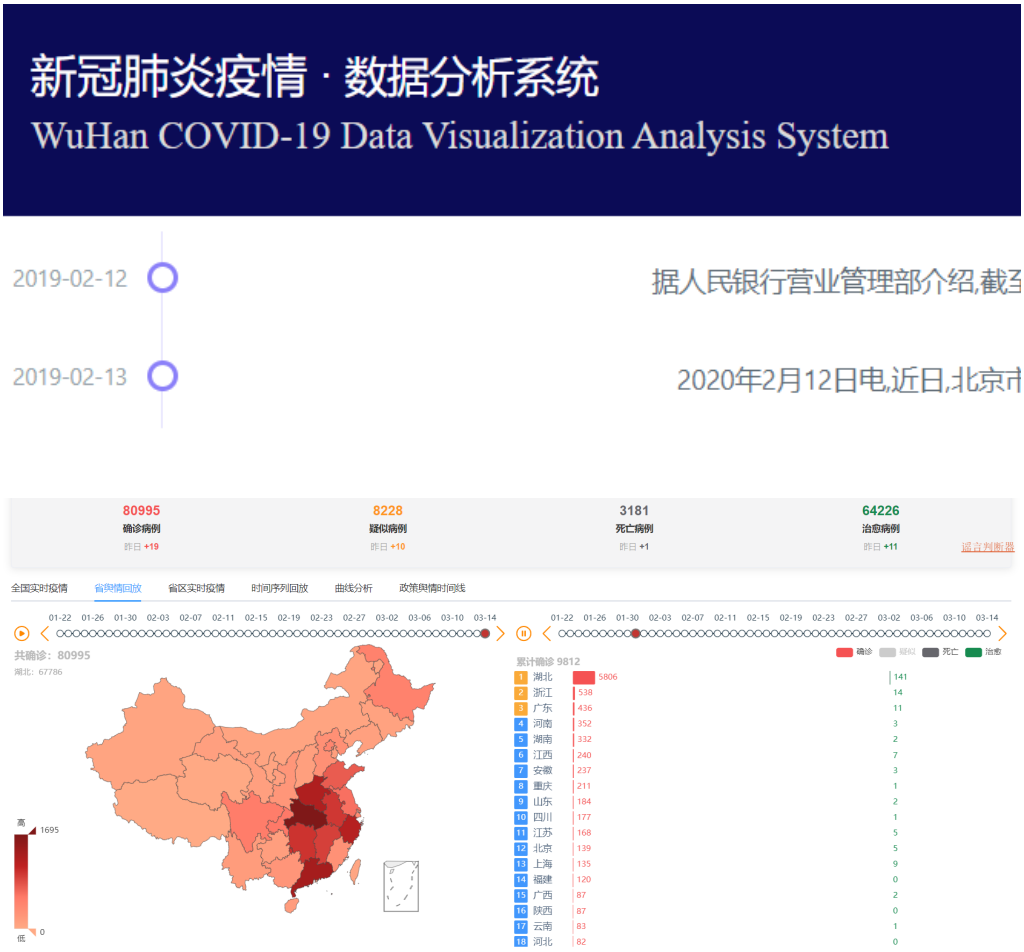
3 前端部分进展

目前疫情信息平台已经实现前后端基本端口的对接，已经完成在阿里云服务器上的自动挂载，可以通过外部 IP 访问、并查看实时疫情信息。

3.1 前端架构

前端使用 Vue.js 框架，实现了疫情实时数据 (包括舆情)、疫情新闻两个主页面，为了将页面篇幅利用率合理化，我们利用栏目切换展示疫情数据的几个角度：

- 全国实时疫情
- 全国各省疫情回放
- 省实时疫情
- 各省各项数据曲线分析
- 舆情时间线分析



依靠后端提供的地图和省区数据和 Echarts 的绘制功能，我们实现了在地图上点击即可切换地图、获取各地疫情的功能；依靠后端 MySQL 的长时间存储，结合前端的 Timeline 组件，实现了疫情、舆情的时间线回放。

4 预测模型部分进展

我们已经完成了疫情舆论数据集的构建，将丁香园提供的舆情加入传统的微博和 twitter 谣言语料库训练，并且结合 Flask 形成了外部服务接口，已经交给后端开始疫情数据提供、谣言判断两个模块的组合。

在网络结构上，我们借鉴已有的工作，用 Transformer+ 卷积层的方式对文本进行语义信息提取，其在仅 CPU 的条件下运行速度较快。

5 进展总结

5.1 遇到的问题

目前而言，前后端进展比较顺利，没有遇到较大的问题，完成了预期的计划内容。

疫情舆论数据集构建面临的情况，主要是丁香园提供数据集的不充分、且在现有条件下进行人工标注不现实，所以我们采取了折中的方法，结合了已有的舆论数据集。

此外，在缺失传统谣言判断任务中的用户 id 情况下，模型测试正确率会有降低，所以我们采取的方案是在未来结合关键词提取等 Non-Neural 的方法，提高该功能的可靠性。

5.2 未来计划

接下来的工作是我们的项目的核心，涉及到“提取舆论主题”、“结合时间线展示群众的关注点变化”以及“谣言判断”等相关工作，要使用神经网络模型对我们收集到的数据进行主题提取和内容分类等一系列操作。

对于后端数据部分，原来制定的计划不会有大的改变，所以接下来几周的重点会放在“舆论数据的收集和进一步的整理”以及“神经网络模型判断谣言“两部分。对于前端部分，需要对后端新加入的舆论数据提供进行对接，并完成神经网络模型判断谣言的前端界面。

整体而言，我们制定的 Schedule 不会发生太大改变，原本第五周仅用来对接和实现 Demo，现在我们会利用第五周对接更多接口，提高初步上线 Demo 的质量、考虑域名等细节事宜。此外，我们原定第六周将谣言预测部署上线，现在我们将提早进行部署、效果（主要是用户体验）评估，以便后期完善。