## 一、本周研究内容

**1.** 使用 semi-supervised learning (pseudo labeling) 方法，用训练好的 SVM model 对上周新增 214081 条 tweets 进行了预测，并在 KNN 和 SVM 上进行了实验。实验发现使用扩充后的数据集，模型表现都有明显提升。

**2.** Web app 对接：和 zhang juntao 同学在 web app 的需求上进行了探讨，并且在前端的实现中发现了一些问题： 地理区域划分该如何选择。

## 二、项目实施当前状态

项目目前在 text classification 部分已经完成，下一步重点在与 web app 的对接以及设定相关预警规则。

## 三、本周成果

**1. Semi-supervised learning （Pseudo labeling）**

使用 SVM 去 predict 214081 unlabeled tweets。最终数据集的构成为 221145 条数据 (7064 + 214081)，其中 label 为 0 的 211691 条，label 为 1 的 9454 条。

| | label | full_text |
|---|---|---|
| 0 | 0 | 211691 |
| 1 | 1 | 9454 |

图一： pseudo-labeled 后数据集分布

**实验：** 使用 214081 条 tweets 的 70% 154801 条 tweets 作为 training set, 30%  66344 条 tweets 作为 validation set。

```
text_train, text_test, label_train, label_test = \
train_test_split(iteration5['full_text'], iteration5['label'], test_size=0.3,random_state=42)

print(len(text_train), len(text_test), len(text_train) + len(text_test))
```

```
154801 66344 221145
```

SVM 模型在 7064 条 labeled tweets 上（supervised learning）的实验结果如下图所示， marco average F1  为 0.72。

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.96 | 0.99 | 0.97 | 1994 |
| 1.0 | 0.68 | 0.36 | 0.47 | 126 |
| accuracy | | | 0.95 | 2120 |
| macro avg | 0.82 | 0.67 | 0.72 | 2120 |
| weighted avg | 0.94 | 0.95 | 0.94 | 2120 |

图二： SVM 在 7064 条 tweets 上 classification report

SVM 模型在 221145 条 pseudo-labeled tweets 上（semi-supervised learning）的实验结果如下图所示，marco average F1 为 **0.95**。

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 1.00 | 0.99 | 1.00 | 63425 |
| 1.0 | 0.87 | 0.93 | 0.90 | 2919 |
| accuracy | | | 0.99 | 66344 |
| macro avg | 0.93 | 0.96 | 0.95 | 66344 |
| weighted avg | 0.99 | 0.99 | 0.99 | 66344 |

图三： SVM 在 221145 条 tweets 上 classification report

KNN 模型在 7064 条 labeled tweets 上（supervised learning）的实验结果如下图所示， marco average F1 为 0.66。

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.95 | 0.99 | 0.97 | 1994 |
| 1.0 | 0.71 | 0.23 | 0.35 | 126 |
| accuracy | | | 0.95 | 2120 |
| macro avg | 0.83 | 0.61 | 0.66 | 2120 |
| weighted avg | 0.94 | 0.95 | 0.94 | 2120 |

图四： KNN 在 7064 条 tweets 上 classification report

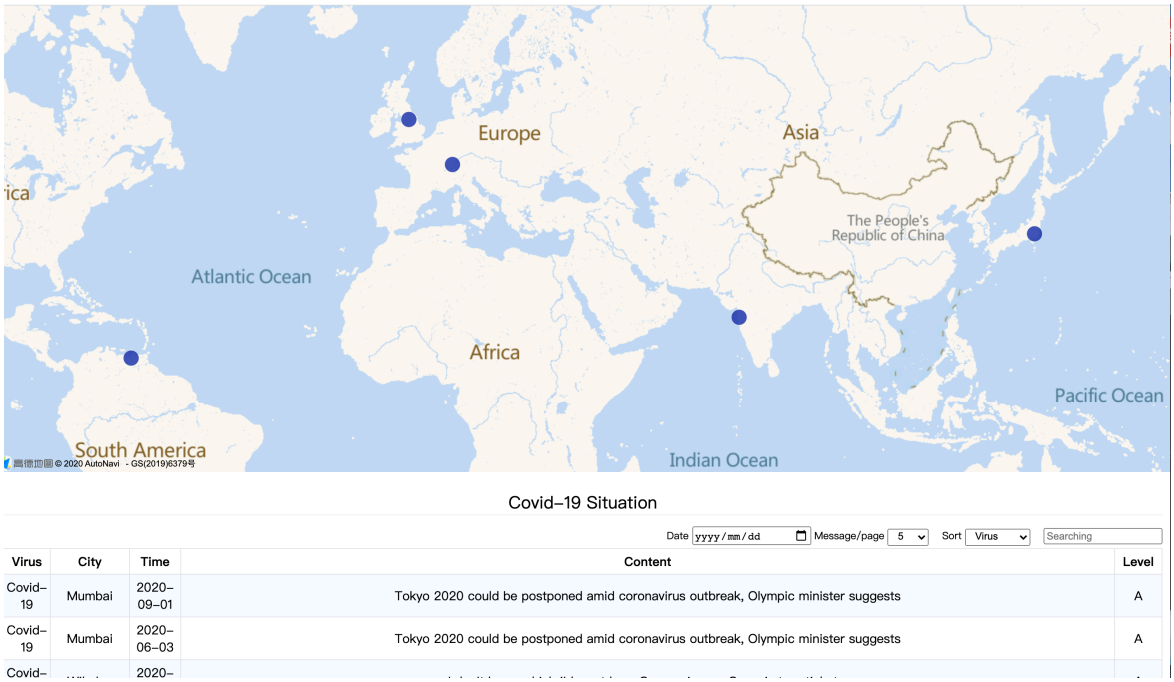
KNN 模型在 221145 条 pseudo-labeled tweets 上（semi-supervised learning）的实验结果如下图所示，marco average F1 为 **0.82**。

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0.0 | 0.98 | 0.99 | 0.99 | 63425 |
| 1.0 | 0.72 | 0.60 | 0.66 | 2919 |
| accuracy | | | 0.97 | 66344 |
| macro avg | 0.85 | 0.80 | 0.82 | 66344 |
| weighted avg | 0.97 | 0.97 | 0.97 | 66344 |

图五： KNN 在 221145 条 tweets 上 classification report

**2. Web app 对接**

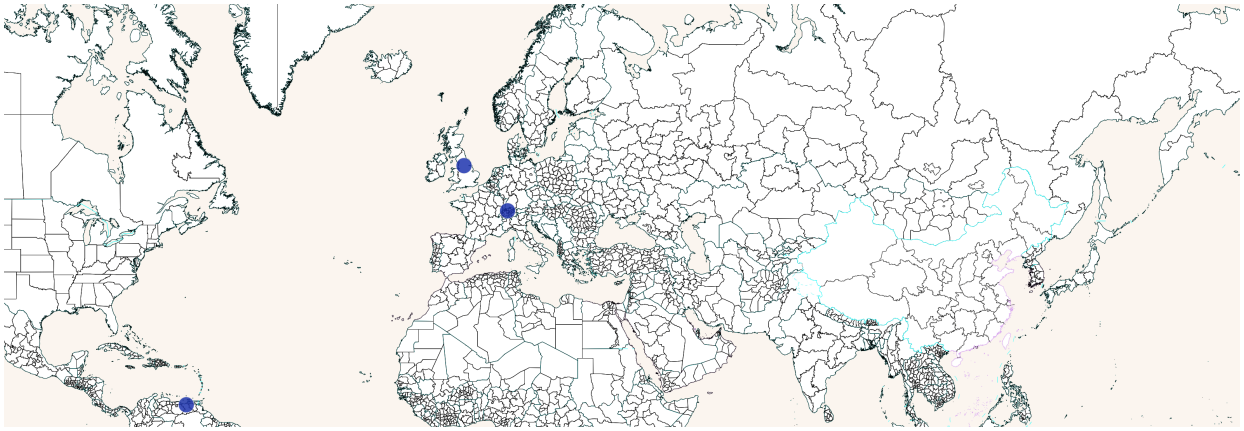
在‘预警 Web App 需求 V2’中，我把前端方面的需求归纳为三个：全球疫情预警地图，疫情预警信息 table 以及时间选择功能。Zhang juntao 同学已经将他做的符合这三个需求的 demo 截图发给了我，如下图所示。




图六: Web App 前端 Demo 页面

## 四、本周问题

Zhang juntao 同学在将疫情预警地图按照城市划分的时候，发现 UI 效果不是很理想，如下图所示。本周的问题是，能否将按照城市划分预警等级转为国家划分？



图七: Web App 按照城市划分

## 五、下周计划

落实疫情预警规则地域的设定，配合 Web App 对需要的后端数据进行处理。