Chatgpt 舆情热点及用户关注网络可视化

刘妮琦 计算机系 2022210899　　杜晨熙 计算机系 2022310850　　申屠泥 新闻学院 2022213203

1 技术路线

- **推文数据下载**　使用 python+selenium 框架，爬取（1）以 ChatGPT 为关键词的推文5.5w+ 条（点赞量高于 10），包含用户名、发帖时间、转评赞数目等字段;（2）OpenAI 关注用户列表前 1w+ 名的专业、地域等信息。处理后均存储于github 仓库。
- **推文文本处理**　使用 python-nltk 包进行分词、langdetect 包进行语言类别的识别。
- **推文情绪分类**　使用Twitter-Emotion-Recognition 情绪六分类模型（Anger, Disgust, Fear, Joy, Sadness, Surprise），对**英文**推文的情绪分布进行识别。
- **推文情绪分布**　使用 **PCA 降维可视化**方法将 6 维情绪向量降至 2 维平面坐标向量、雷达图显示情绪六极分布，分别建立“推文热度 ⟷ 气泡大小”以及“情绪向量 ⟷ 气泡颜色”的映射。
- **图表绘制**　主要使用 d3.js 和 echarts 进行各类图表的绘制，其中词云图的绘制使用了 d3.layout.cloud 库，地图使用了 d3-geo-projection 库。
- **交互设计**　主要通过鼠标悬停、点击的方式得到更丰富的信息。

2 架构说明

前端：d3.js(90%), echarts(10%)

后端：VSCode Liver Server

3 分工

刘妮琦：选题、twitter 数据爬取、推文情绪分类、推文散点 ⟷ 情绪分布图、OpenAI 核心用户关注网络、可视化总览框架。

杜晨熙：数据清洗、主题河流、动态词云、PPT 制作、演示视频

申屠泥：数据清洗、用户气泡图、用户领域分布图、用户地域分布图、PPT 制作、视频录制

4 遇到的困难

- **数据获取** （*）完整性：由于统计每天各时段的相关推文总量，在时间上不能有遗漏，Twitter 反爬机制严格，条件检索可选范围有限——为获取长时段的完整检索结果难度较大。尝试过 twint 等多个既有框架，最后选择手写 selenium 代码，且保证全天候的网络连接。
- **推文 vs 用户** 为确保“ChatGPT 推文分析”与“OpenaAI 关注用户社交网络”的关联，须保证所爬取的关注用户曾发表过含关键字 ChatGPT 的推文。可视化过程中，使用的是“技术路线——推文数据”中两类数据按用户取交集后的结果。其间广告类推文的过滤、用户的数量扩充较为冗杂。
- **UI 设计** D3 的交互设计具有更好的灵活性，但各功能均须从基础图元开始实现。为得到更好的视觉效果，需要大量检索、参考已有网页的代码，一致为模板进行构图美化。
- **动态词云图** 在绘制随时间推移而不断变化的动态词云图时，我们认为如果能固定同一单词前后的位置、大小和颜色，会取得更好的可视化效果，所以我们尝试建立一个从单词频数到坐标和字体大小的映射。但是由于需要兼顾字体大小和位置（不然会发生单词重叠的现象），这样的映射关系并不容易找到。于是，我们尝试使用 python 中的 **WordCloud** 库，以期望得到每个词的坐标和大小，但是 **WordCloud** 只能直接获得坐标，无法直接获得大小（只能得到一个基于频数的相对值），经过一些尝试后，均未很好地实现可视化。
- **用户气泡图** 尝试了许多方式将横轴颜色调整为白色，比如 xAxis-Group.selectAll(”line”).style(”stroke”, ”white”)；但不奏效