

ChatGPT: Visualization of public opinion hot spots and user attention networks

Technical Approach

• Tweet data download

Used 'python+selenium' framework, crawling:

- (1) Tweets with the keyword 'ChatGPT': 5.5w+ (> 10 likes), including user name, posting time, number of reposts, comments and likes;
- (2) The top 10,000+ 'OpenAI's followers' professional and geographical information .

• Tweet text processing

Used 'python-nl tk' package for word segmentation, 'l angdetect' package for language classification.

• Tweet sentiment classification

Used [Twitter-Emotion-Recognition Emotion](#) six-category Model (anger, disgust, fear, joy, sadness, surprise) to identify the sentiment distribution of tweets.

• Tweet sentiment distribution

Used the **PCA** dimensionality reduction visualization method, the 6-dimensional emotional vector is [reduced to a 2-dimensional plane coordinate vector](#), and the radar chart displays the six-pole distribution of emotions. Mapping is established between "[tweet popularity ↔ bubble size](#)" and "[emotional vector ↔ bubble color](#)".

• Chart drawing

Mainly used [d3.js](#) and [echarts](#) to draw various charts, including the d3.layout.cloud library for word cloud diagrams and the d3-geo-projection library for maps.

• Interaction Design

Mainly by hovering or clicking the mouse to get more information.

Architecture Description

Front end: `d3.js (90%), echarts (10%)`

Backend: `VSCode Liver Server`

Division of Labor

Me: Topic selection, twitter data crawling, tweet sentiment classification, tweet scatter ↔ Emotional distribution mapping, OpenAI core users interconnection network, and visualization overview framework.

Partener A: Data cleaning, theme river, dynamic word cloud, PPT Production, video recording

Partener B: Data cleaning, user bubble chart, user distribution chart, PPT Production, video recording

Difficulties encountered

- **Data Acquisition - Completeness:**

Due to the need for comprehensive daily tweet counts and Twitter's strict anti-scraping measures, obtaining long-term data is challenging. After trying frameworks like twint, I opted for custom Selenium code to ensure constant internet connectivity.

- **Tweets v. s. User**

To link 'ChatGPT Tweet Analysis' with 'OpenAI Social Network Following,' we confirm that followed users have posted tweets containing 'ChatGPT.' In visualization, we use intersected results from two data types in 'Technical Route—Tweet Data.' Filtering ads and expanding user data are complex tasks.

- **UI Design**

D3's interactive design offers better flexibility, but each feature must be built from basic graphic elements. To achieve superior visual effects, extensive research and referencing of existing web code are necessary, consistently using templates for aesthetic composition.

- **Dynamic word cloud**

When creating dynamic word clouds over time, we aimed to consistently fix the position, size, and color of words for better visuals. We tried mapping word frequency to coordinates and size using Python's WordCloud library. However, finding a suitable mapping proved difficult as WordCloud only provides coordinates, not exact sizes.