**Visualization**

I. Functions I want to achieve

1. An Echarts bar chart showing the top 10 most viewed videos in the TED videos and jumping to the corresponding video playback screen when the bar is clicked.

2. An Echarts bar chart showing the top 10 topics in the TED video in terms of number of appearances.

3. A Gephi chart with all topics as nodes and the total number of times the topic appears as the value of the node. If two topics appear in the same video, the two corresponding nodes are considered to have edges between them, and the weight of the edge is added by one for each occurrence. Exhibit the strength of the relationships between the topics and the different communities to which they belong.

4. An Echarts node-relationship graph with all topics as nodes, showing all edges with a weight of not less than 4 in the graph, and showing all nodes related to a node when the mouse is placed over it.

II. The implementation process of the important and difficult points

1. How to realize jumping to the corresponding play screen after clicking on a bar.

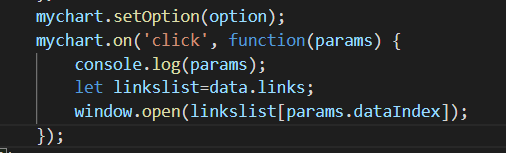
2. How to adjust the filter information so that the node information of the Gephi diagram does not interfere with each other; how to partition; How to reflect the strength of the relationship between the nodes.

3. How to obtain the weights of the edges.

4. How to import the data of nodes and edges into the Echarts node-relationship graph, Specifically, how to obtain the coordinate information of the nodes (since they do not have coordinate properties); how to control the size of the nodes so that they do not cover each other; how to set the parameters so that the edges are not so dense that they cover the nodes.

III. Solutions

1. Add the function mycharts.on() to the echarts\_views.html where the top 10 videos graph are placed, and add a property links in the views.json file where the top 10 links are placed. The code is as follows.



2. Methods to prevent interferences:

Use the Force Atlas layout in Gephi, adjust the repulsion strength to 6000, check "Adjust by size", filter by the degree of the range, and filter out nodes with a degree below 30 to prevent nodes from overlapping due to too dense; use the value of the node as a measure of the relative size of the nodes, the minimum In the preview settings, choose Show node labels and Shorten labels to prevent labels from covering each other; adjust the thickness of the edges to 0.1 and the transparency to 200.0, and choose Bend to prevent node information from being obscured due to too dense a distribution of edges or too heavy an edge color.

Partitioning.

Calculate modularity in statistics, use Modularity Class when coloring the nodes, the software will automatically give the colors and partitions, you can adjust the colors of the blocks to prevent color clashes.

Method to show relationship strength:

The color is set to mixed in the column of edges in the preview settings, the color of the displayed edge is a transition of the color of the two nodes it is connected to, showing the distance between the nodes.

3. Method of obtaining edge weight:

New sorted\_ data.py file, using Python method to obtain database data for filtering.

4. Data importation of nodes and edges:

Node attributes include: x, y, ID, name, symbolSize, itemStyle (color); edges attributes include: sourceID, targetID, weight.

Node has no coordinate attribute. In order to give node coordinate attribute, two groups of random numbers obeying normal distribution are generated by library function according to mathematical law, which are used as the horizontal and vertical coordinates of node respectively;

The node id is the sequence number arranged in descending order according to the occurrence times;

The name of the node is the name of the topic;

The symbolSize of the node is the value of the node;

The color attribute of the itemStyle of the node is set according to the colorList loop;

The sourceIF and targetID of the edge are the ID of the node;

The weight of the edge is the weight of the edge itself;

In the graph, the nodes are displayed on the canvas according to the given horizontal and vertical coordinates; the label content of the node is the name of the node; the size of the node is the symbolSize of the node;

Parameter setting:

The symbolSize of a node is not equal to its value, but is equal to the unified division of the value by 15, which can effectively prevent coincidence caused by too large nodes;

The parameter of normal distribution coordinates is expected to be 0 and the variance is 1000000. This group of parameters can effectively disperse nodes, and the number of coordinates is 348 nodes;

Set the label position of the node to the right to prevent the node labels from overlapping.

Filter the input data and eliminate the edges with weight less than 4.