Word Cloud Texture Language Map

Xiang WEI

Design

Learning a foreign language has never been an easy task, especially for people who share a totally different mother tongue. For example, a lot of Chinese students around me, who studied English hard for ten more years, but they still can’t express themselves well. If we put Chinese and English together, we can have some clues. “你好” and “Hello” is the same word in two language, but they look extremely different. After coming to France, I found that many European students can easily master multiple languages, especially common for those who can master French, Italian, and Spanish at the same time. Except for the educational system, the great similarity between these languages provides more or less convenience. For example, “ami”, “amigo”, “amic” are the same word in French, Spanish and Italian. They look quite the same.

The English alphabet languages are the largest group of languages that exist. The several languages mentioned above all belong to it. Based on 26 English letters, there are different variants and different combining rules, or grammars, different languages are finally formed. However, Chinese is an ideogram composed of strokes. Japanese, which is slightly similar to Chinese, also has characters that look completely different from Chinese characters, such as hiragana and kana. There are also Arabic languages that even differ in writing and reading directions. When I surfed the Internet, I found such a dataset, which summarized the writing of 33 common words in 108 languages. In the performance of the same word, different languages may look completely different from other languages. By simply a quick glance, I can find that many vocabulary have great similarities, while there are also some languages that are very different from others. For example: same word may looks like "Good morning" in English, “早上好” in Chinese, "おはようございます" in Japanese, "Καλημέρα" in Greek, and so on.

Based on this discovery, I am curious as to what caused the similarities and differences in these languages. A bold conjecture is that the style of text is mostly determined by its geographic location. Adjacent languages may also influence each other, so they become more and more similar. For example, France, Italy, and Spain border each other, so French, Italian, and Spanish are relatively similar. The same applies to Chinese and Japanese. So I was wondering can we find other countries by the same analogy? Are the languages of neighboring countries related, and how might they affect each other? If we can combine some historical knowledge to explore the origin of culture, we may find out more.

So I decided to visualize these words on the map, so that readers can have a macroscopical sense of the distribution of language symbols worldwide and intuitively form an impression of the image of different texts. On top of that, people can add any knowledge to find some more interesting phenomenons.

Method

In the petite class, we used d3.geo to draw a map, which successfully demonstrated the distribution of water resources on the earth. So I decided to continue to use d3 to implement our map. And I found a geojson dataset[1] containing the geographical information of countries worldwide. With the help of d3.geo, we can generate many paths as the border of each country and combine them together to build a whole map.

Since we already have the dataset[2] that contains 33 common words corresponding to 108 different languages, the direct manifestation of a group of words that impress readers is the **word cloud**. D3 provides us with a convenient way to generate word clouds, by controlling the position and rotation of d3.text.

At this point, we have a map and a word cloud for each language. The last step is to find a set of data to link them together. So we can know to attach which the language's word cloud to which areas. Therefore, we found dataset[3]. This dataset contains attributes such as *wals_code, iso_code, glottocode, Name, latitude, longitude, genus, family, macroarea, countrycodes* etc. Attribute *Name* corresponds to the *Popularwords* in our dataset[2]. Attribute *countrycodes* also have a corresponding relationship with attribute *iso_a2* in dataset[1]. Therefore, by loading these three datasets, we obtained all the data for building our word cloud texture language map.

Format data

Use the same method we did in the petite class, we read the three datasets and store the data into the format we want. But when sorting out the data, I found that the same *countrycodes* may contain **multiple** language *Name*. In these situations, it's hard to decide which group of words should be put into this country. As a result, we don't know which word cloud should be put in that area. For simplicity, we don't consider these multilingual countries in our project. We will introduce more in detail in the following problem section (*P1* in below).

Define position

In addition to establishing the relationship between the country's geographic coordinates and its language common words group, we also need to generate a word cloud in proper size and put it in the correct position. For each irregular country's border path, a rectangular **bounding box** exists to help us define the area. We can simply replace the *width, height, x and y* position of the word cloud generation method in d3, by the value of *dx, dy, x and y* returned by the bounding box.

Clip path

Beyond having the word cloud at the right position, we also want to have it follow the geographic boundary path of the country. According to the parameter value we set for the word cloud, we will receive a rectangular boundary of each word cloud. That doesn't meet our expectations. After searching, I found that there are some ways to generate custom shape's word clouds using python or echarts. Also I found some word cloud generators with custom shapes, ie. wordArt, wordClouds, etc. But I didn't manage to understand how they achieved it.

Since there is no suitable way to get a custom shaped word cloud in d3, we tried to think about it from the other aspect. Instead of generating the idea shape directly, we can cut the unnecessary area in order to get the idea shape. This works with a prerequisite : because we don't really care about the expression of each word, we don't need to accurately and completely display every word in the word cloud, so cutting words does not affect our ultimate goal. Therefore, our final method is taking the coordinate value of the bounding box corresponding to the path of each country to put the corresponding word cloud in a suitable location. Then use the path to crop the word cloud, hide the words outside the country boundary in order to realize the word cloud as the texture effect on map.

Imitate the operation in pc(s08), we used the clip path attribute. We tried to start with doing it in a single country and with the clip-path, it works pretty well (Fig.1).

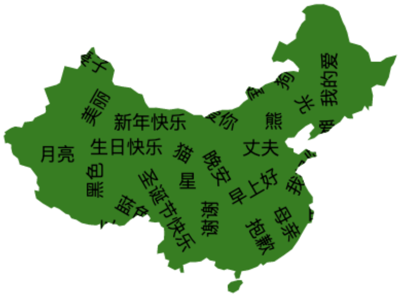


Fig.1 China Word Cloud

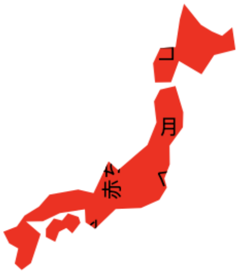


Fig.2 Japanese Word Cloud

Yet, when we wanted to apply to the whole map, we had some problems (detailed in *P3*). There are two methods I tried to achieve the combination.

The first one is to create a map background. In practical, apply the clip-path of all the countries on a background svg. Then inside that svg, I generate word cloud svgs which are also applied with the corresponding clip-path. The other way is to generate multiple svgs that have transparent backgrounds for each word cloud country. By putting them at the same original position to let them overlap with each other, we should have a complete map.

Problem

- *P1*

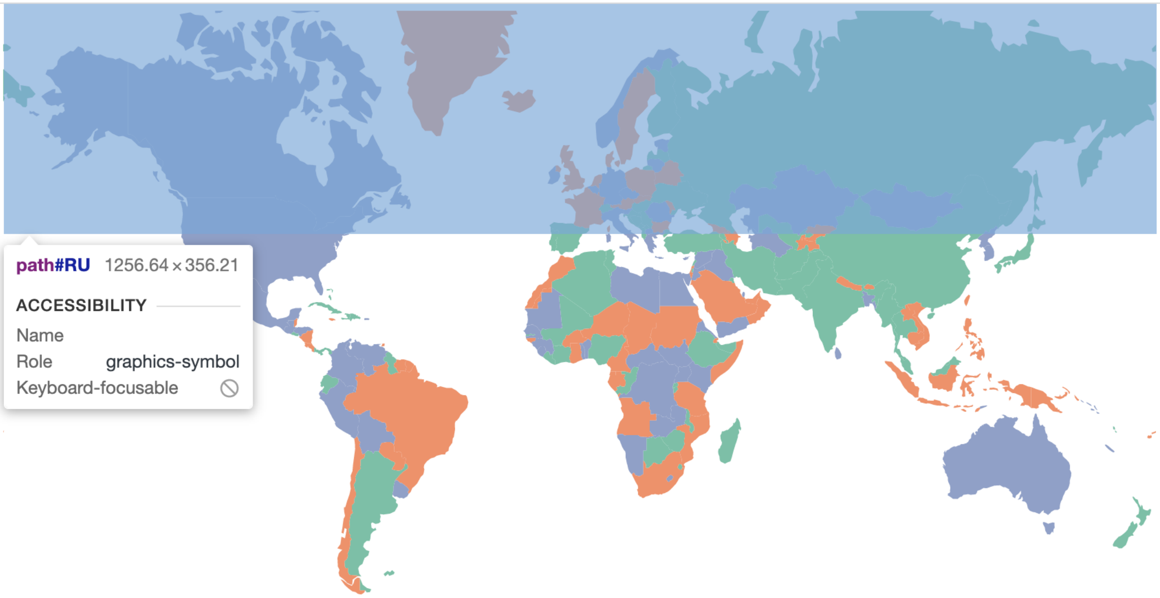
During the implementation process, many problems first appeared in the initial design. In view of the two countries I am familiar with: China and France are both monolingual countries, I ignored countries like Canada (official language is English and French), Switzerland (speak German, French, Italian, etc.) and other multilingual countries in my design. Theoretically, as long as we find a more detailed path of different language distribution areas within the country, we can apply the previous design: clip the corresponding word cloud by language area path. But due to time constraints and limitations in obtaining ideal datasets, we decide to not visualize those countries which keep more than one language. What's more, those who don't have a match in the dataset[3] won't show in our map neither.

- *P2*

Some countries are too small or too narrow to display only one or a few characters (Fig.2). This may cause some misunderstandings. For example, the Japanese shown in Fig. 2 looks completely similar to Chinese. In fact, some typical Japanese symbols were clipped. Due to the random position of each word in the word cloud location, the worst case can be the area that happens to only cover the blank area.

Another situation related to the word cloud position is the territory that has been divided into two parts to be shown in a 2D map. For example, the bounding box of Russia is shown in Fig. 3, and the return value (x, y, dx, dy) of the bounding box is (171.6814727783203 -114.08403015136719 1256.636962890625

356.2103271484375). So if we generate a word cloud based on this bounding box coordinates, and then apply path clipping, unaligned problems will arise (Fig. 4).



path#RU 1256.64 × 356.21

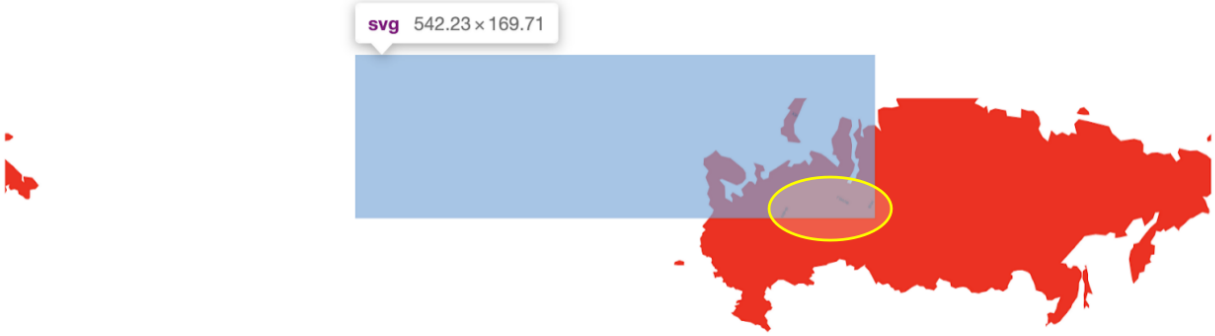
ACCESSIBILITY

Name

Role graphics-symbol

Keyboard-focusable

Fig. 3 Bounding box of Russia



svg 542.23 × 169.71

Fig. 4 Russia Word Cloud

- *P3*

When we tried to print out all the word clouds at the same time, a new problem appeared. Following the same path of making a word cloud texture to a single country, I applied all the cutting references to the same svg to show the map as a background. Then I created word clouds and applied their corresponding clip-path. Although the text is displayed in the correct position range, the clip-path applied to the word cloud svg doesn't function as expected: the word cloud isn't clipped. So the outcome contains overlapped text like shown in Fig. 5.

The blue area in Fig. 5 shows Hindi word cloud. Seems it applied all the clip-path, so the right bottom part is cut while the right top part doesn't. Same as Nepali. So it shows a mess in that area.

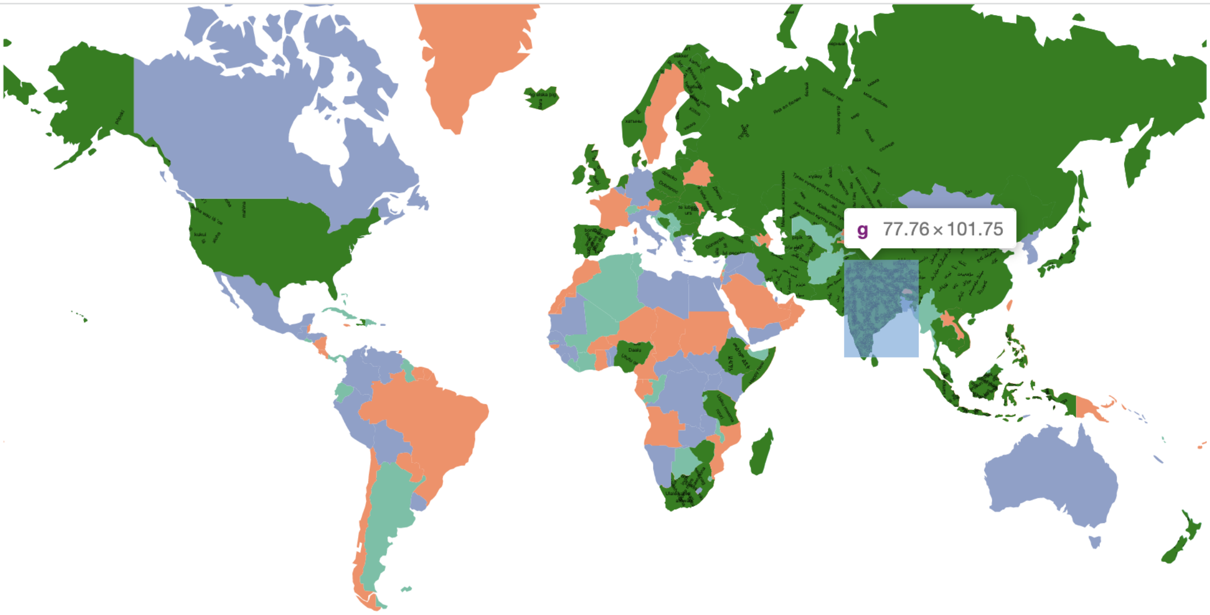
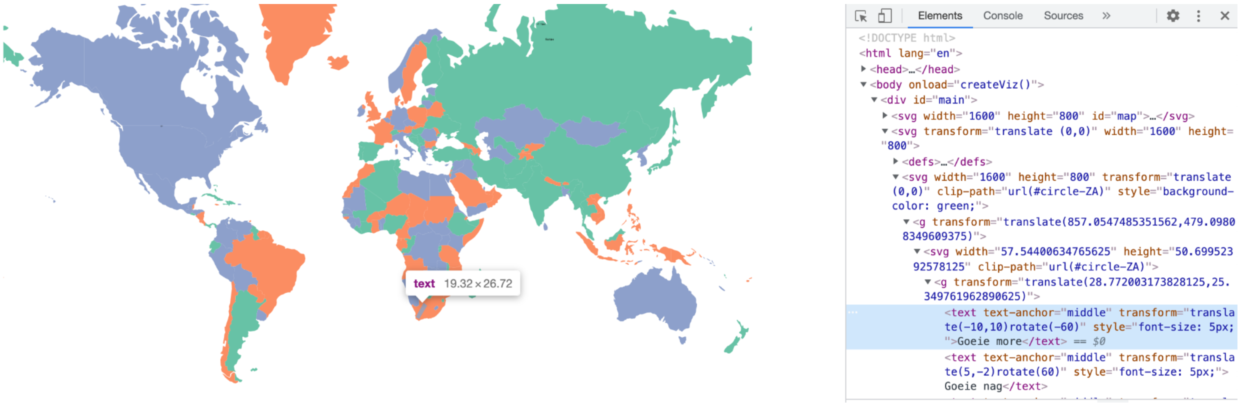


Fig. 5 Clip-path on background and each word cloud

I did check the dom tree and found that each word cloud is applied with its corresponding clip-path, not the entire one. So I didn't find out the reason why the clip-path doesn't function well on a word cloud.

Since a single textured country works well, I tried another solution to stack the outcome of each single textured country. In that case, we need to force the background color to transparent and the position should be at the same spot (eg. all at (0,0)). But after practice, it didn't work as we expected neither. As we can see in Fig. 6, we can find the inserted text in the dom tree and the corresponding position (blue box) on the map page, but we cannot see the text displays on the page.



```
<!DOCTYPE html>
<html lang="en">
▸<head>…</head>
▾<body onload="createViz()">
  ▾<div id="main">
    ▸<svg width="1600" height="800" id="map">…</svg>
    ▾<svg transform="translate (0,0)" width="1600" height=
    "800">
      ▸<defs>…</defs>
      ▾<svg width="1600" height="800" transform="translate
      (0,0)" clip-path="url(#circle-ZA)" style="background-
      color: green;">
        ▾<g transform="translate(857.0547485351562,479.0980
        8349609375)">
          ▾<svg width="57.54400634765625" height="50.699523
          92578125" clip-path="url(#circle-ZA)">
            ▾<g transform="translate(28.772003173828125,25.
            349761962890625)">
                <text text-anchor="middle" transform="transla
                te(-10,10)rotate(-60)" style="font-size: 5px;
                ">Goeie more</text> == $0
                <text text-anchor="middle" transform="transla
                te(5,-2)rotate(60)" style="font-size: 5px;">
                Goeie nag</text>
```

Fig. 6 Each word cloud country overlap on each other

The above bug looks so weird and I couldn't find any helpful solutions on the internet. One guess is that using an absolute position may cause problems. But when I try to test whether the forced position is valid by simply adding an svg at the specified position and using the red background color, it works well and returns me a screenwide red rectangle. On top of that, I tried to apply clip-path on each svg, but I still have the whole screen red. By checking the dom tree manually, I found that there are some clip-paths that don't work properly. Some of them are because of lack of data while some of them are because of some unknown reasons. So I decided to use the *mouseover* function based on the existing map to guarantee the existence of data. And surprisingly, it works pretty well and drives me to the final outcome.

Outcome

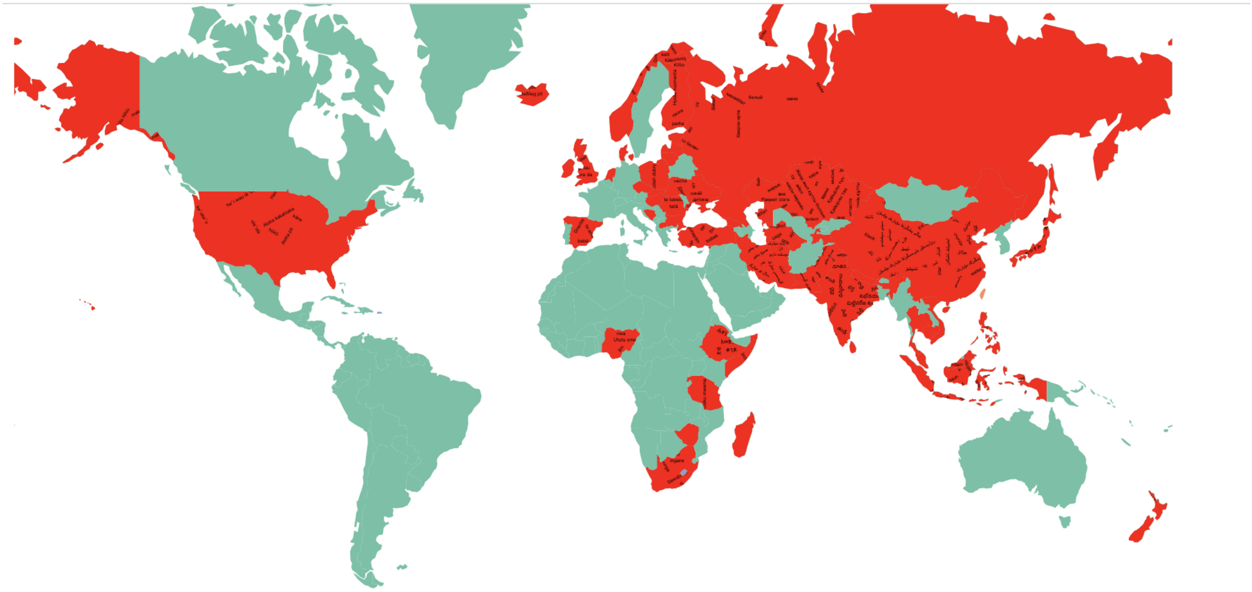


Fig. 7 Word Cloud Texture Language Map with d3

A colorful map is initialized when we load the page. With the mouse cursor move over different areas on the map, the touched country will either change to green color or red color with a word cloud texture. (Check the attached **video**).

In the map, the red areas show the countries that we find a good relationship with their spoken language. And the green areas are countries that lack data. In those red areas, a word cloud representing some common words written in that language works as a texture attached to them, only show on their areas.

For the current version, there is a lot of error data presented, eg. the word cloud in China is not correct. In addition, the position of some word clouds are not perfectly aligned, eg. the word cloud in Russia. There are still many future works that should be done.

Future work

First of all, we need to find solutions to all the problems mentioned above. We need to find a more reliable dataset to build the relationship between countries and popular words groups. For the current result, we are so lacking in correctly paired data and leave most of the map empty (in green). This greatly hinders us from discovering some phenomena. What’s more, we also need to search for a dataset that can contain the language area in a multilingual country. Then we can apply the same strategy. One way to show the word cloud in a deformed narrow area is to generate the word cloud more densely and repeatedly applied to the whole bounding area. Even though it might be hard to read, with the help of zoom, readers can still gain the sense. It’s always better to have something than empty.

After, when we finally reach a completed word cloud texture language map, we can improve it with any related knowledge. By showing extra information directly on the same map or other place in the page or in a dialogue or animation, there are many existing opportunities. The map will be a solid milestone for people to explore more in the language study field.