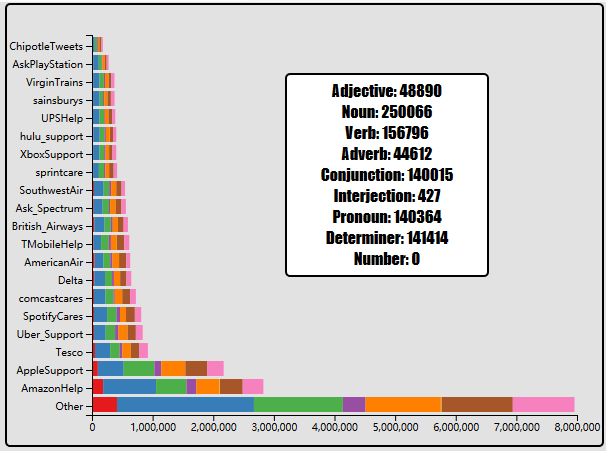
Visualizing Customer Support on Twitter – Jiayu Tang

**1. Description**

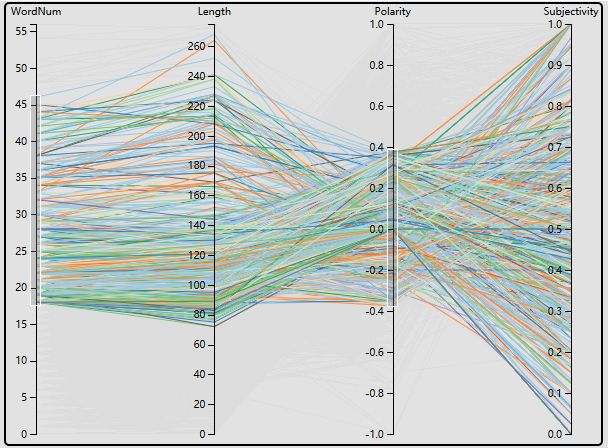
In this project, I chose the Customer support on Twitter dataset and textblob library for python to complete the visualization. The dataset includes over 2 million tweets between customers and support groups on twitter, including the text, id of author, the response, etc.

The visualization includes three views:

The first view is a histogram to present the frequency of different categories of word based on the tweets of these support groups. There are 10 categories, including noun, verb, adjective, adverb, conjunction, interjection, pronoun, determiner and number. I used different color for each category added a mouseover function to present the details of usage. This view can also play as a filter. When clicking on one of the bars, user can choose whether or not to show that support group’s result in two other views.



The second view is a parallel coordinates view. This view is based on the sentimental analysis provided by textblob library, which analysis the polarity and subjectivity of certain texts. This view shows the relationship among the number of words, the number of characters, the polarity and subjectivity of tweets of support groups. User can select on axis to see the desired range of certain axis only.



The third view is a word cloud. It focuses on single words in these tweets. The color for each category is consistent with such color in the first view. And the size of the word is determined by its frequency. When move mouse over one of the word, there will be a tip showing its category and its appearance among tweets of selected support groups. I limited the number of words in word cloud to 300 by choosing words that appeared top-300 frequently.



**2. NLP Library**

Textblob library for python is used for preprocessing. The script used for preprocessing is included in the zip file. I kept the tweets from customer support groups only and removed the topics and @ in tweets. I used this library for:

Part-of-Speech Tagging: Analysis the texts and make use of the tagging information. The library tags words into POS tags, while I manually separated different tags into 10 categories mentioned above. Then words are counted and outputted into csv file to play as the data file for first view and third view. Note that same words with different category are counted as different words.

Sentimental Analysis: The library provided a function to analyze the sentiment of tweets. I recorded the result and used the result as the data for my second view. I only kept the analysis result of 5000 tweets when visualizing the second view on the consideration of performance.

**3. Insights**

The appearance ratio of different categories of words from different support groups’ tweets are close. Most tweets tend to have medium polarity. Also the number of words used and the length of tweets vary a lot.

The only interjection with top-300 appearance is “oh”.

Tweets by AppleSupport are all over 20 characters and 5 words. While other support groups like AmazonHelp has tweets that’s really short.

**4. User Guide**

To interact with this visualization, one can move mouse over the bars in the first view and the words in the third view. This will show the detail information.

User can also click on bars in the first view to do the filtering on the support groups. If one group is not chosen, the color of corresponding bar will fade. At the same time, paths in the second view of that group will disappear. The word cloud in the third view will redraw to exclude words from that customer support group.

For the second view, user can drag on the axis to filter the range of data. Then only the paths whose value in that axis is in the range will be kept highlight, otherwise only a grey line will be left in the second view.

**5. Student Review**

Review by Yingdi Liu:

“His visualization is intuitive and easy to explore, and all the plots are well designed. But generating and updating the plots take too much time. Also there should be one color legend indicating the corresponding color for each word type. And in the word cloud, simple words like you, I, a should be filtered out since they don't have too much meaning.”

The performance issue is actually a big issue I didn’t solve, especially for the second view. The original parallel coordinates example I found was working on only 407 lines of data, while after I complete the preprocessing, there still are over 1 million lines of data. In the end I have to keep only 5000 lines of data to stop it from not even loaded up. One way to solve this is to process the data further like, calculating the average length and sentiment for tweets from same customer support group, but in that way the lines of data will be too little.

Also on the performance issue, the dragging for second view works poorly. Since the example I found is in d3 version 3 and the brush function in d3 version 4 is much different than version 3, I ended up with implementing a very slow brush filtering function.

For the color legend, my consideration is that since when moving mouse over the words, one can see the detail information including its category, it’s easy for users to understand meaning of colors. But surely having a color legend will be better.

For simple words thing, I think that it’s hard to tell what kind of words can be regarded as a simple word. In fact, such words are only a small portion of the word cloud. Thus I kept these simple words.

**6. Reference**

[1] Parallel Coordinates - bl.ocks.org <https://bl.ocks.org/jasondavies/1341281>

[2] jasondavies/d3-cloud: Create word clouds in JavaScript. <https://github.com/jasondavies/d3-cloud>

[3] d3/API.md at master <https://github.com/d3/d3/blob/master/API.md>