

Twitter Evidence for Language Variation Based on Culture

(Mark Kuether – DS710 Fall 2019)

Introduction

With modern media commonly visible across geographic boundaries, it is easy to see language variations that can be culturally based. One example observed at this time of year are the sentiments “Merry Christmas” and “Happy Christmas”. The former is commonly seen in American films, such as “It’s a wonderful life”, while the latter can be seen in British shows, such as “Dr. Who”. However, whether this is a genuine cultural difference is uncertain. This project uses twitter text and link text to provide evidence on this assertion. Because user location data is unreliable in Twitter, we compare the proportions of different sentiments to the number of users from different countries. If a correlation is found, this would benefit the marketing departments for aluminum Christmas tree manufacturers to better target their international customers.

Results Summary

The proportions of the two sentiments, “happy” vs “merry”, were calculated and compared to the proportions of the number of Twitter users for multiple countries. Those countries were the United States (48.35M), United Kingdom (13.7M), Australia (4.7M), and Canada (7.6M). The test performed was a proportion test with the hypothesis:

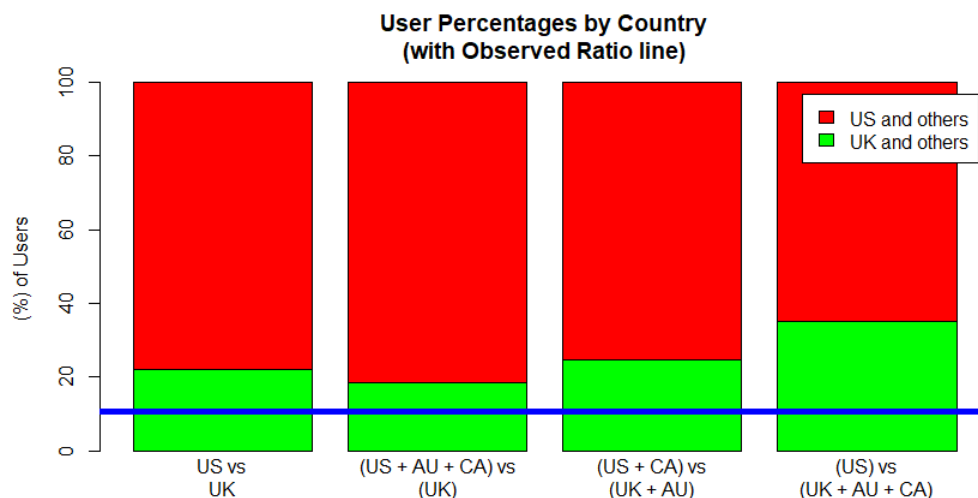
H0: $p(\text{“happy”}) = p(\text{UK based twitter users})$

H1: $p(\text{“happy”}) \neq p(\text{UK based twitter users})$

The four variations of the population proportions were:

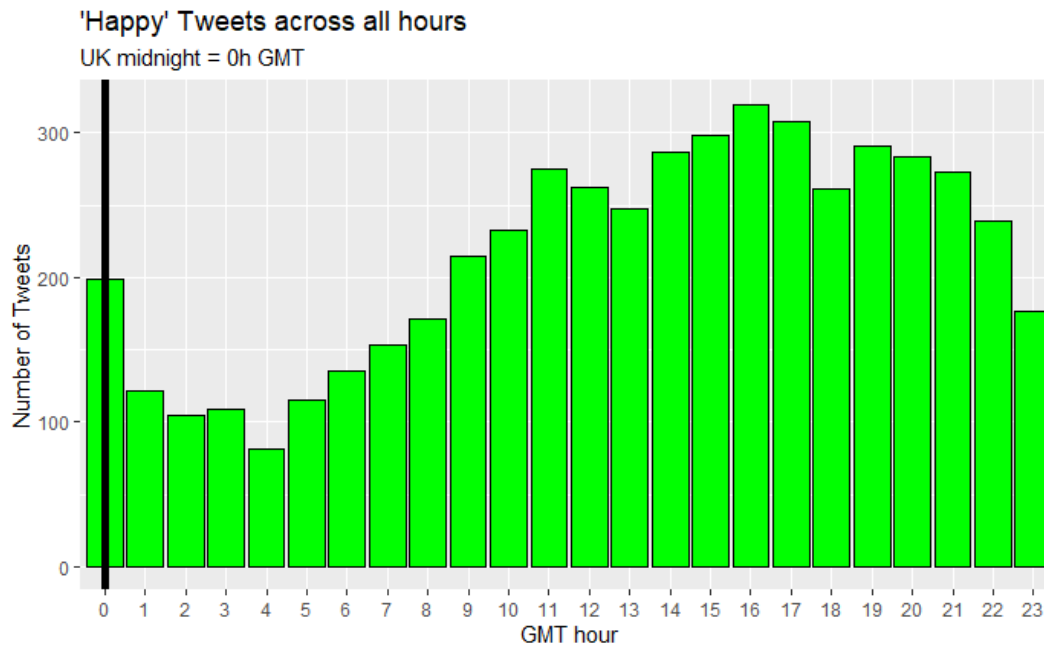
1. UK vs US only
2. UK vs US, Canada, and Australia
3. UK and Australia vs US and Canada
4. UK, Australia, and Canada vs US

The proportion for the number of “happy” sentiments observed was about 10.7% (blue line on chart). This was much lower than any of the population proportions considered, with a p value less than .0001 for all cases. The results do not support the assertion that the sentiment “Happy Christmas” is specific to English culture.



Results Conclusion

Despite the large statistical difference between the sentiment and the proportion of users, reviewing the times when “happy” was sent does suggest a stronger English or European usage.



One possible weakness of this analysis, as mentioned in this web site

(<https://www.grammarphobia.com/blog/2018/01/merry-happy-christmas.html>), is that the two sentiments of “happy” and “merry” may not be considered equivalent in some cultures. Using one or the other would not be a true substitution, and there would not be any clear division on the sentiment across different cultures.

Data Collection and Analysis

The data for this research was collected using a Python application accessing the Twitter REST API. The search was performed retroactively on Dec 8th for all times between Dec 1st and Dec 7th. This ensured both daily and weekly users were included. The search included the four terms “Merry Christmas”, “Merry Xmas”, “Happy Christmas”, and “Happy Xmas”, and limited tweets to only English text. The search phrase filtered out retweets and responses to ensure the sentiments captured were genuine and not repeated or parroted back. The Python application ignored case and tested text, url text, and dashed variants of the phrases.

The summarized results were written to a CSV file. An R script was used to open the CSV file, and further filter the data. Tweets by any users beyond the first sentiment were removed so statistics would not reflect specific user activity. In addition, tweets that were registered with both positive or both negative sentiments were also removed. This reduced the data set from 56,265 records to 48,187 records. That data was used to perform the four proportion tests in R as described above. The four tests had negligible p values < .0001. The tweet time graph was generated after these results to see if a pattern emerged.

Search Phrase Used:

```
q='"Merry Christmas" OR "Happy Christmas" OR "Merry Xmas" OR "Happy Xmas" AND -  
filter:retweets AND -filter:replies', tweet_mode='extended', count=200, lang='en',  
max_id=str(last_id - 1), since=the_date, until=str(next_day))
```