Messenger Analysis

Table of Contents

Analysis

For some nice background music, as you peruse the results... https://www.youtube.com/watch?v=I-Tx763KKHM&list=PLf2BfRjbBfCkGwmVECi5T7Y3qga6CToHQ&index=4

```
FILES = {
    'msgs0.json'
    'msgs1.json'
    'msgs2.json'
    'msgs3.json'
    'msgs4.json'
    'msgs5.json'
    'msgs6.json'
    'msgs7.json'
    'msgs8.json'
    'msgs9.json'
    'msgs10.json'
    'msgs11.json'
    'msgs12.json'
    'msgs13.json'
    'msgs14.json'
    };
LINNEA = 100020242504829;
ALEX =   100001235194456;
TIMEOUT = minutes(45);
EMOTICONS = [";)", ":)", ":))", ":(", ":(("];

tic;
payloads = cell(1, numel(FILES));
for i = 1:numel(FILES)
    payloads{i} = jsondecode(fileread(FILES{i}))';
end
msgs = [payloads{:}];
msgs(cellfun(@isempty, {msgs.text})) = [];

dts = num2cell(datetime(round([msgs.dt] / 1000), ...
    'ConvertFrom', 'posixtime', ...
    'TimeZone', 'America/New_York'));
[msgs.dt] = dts{:};

[~, inds] = sort([msgs.dt]);
msgs = msgs(inds);
```

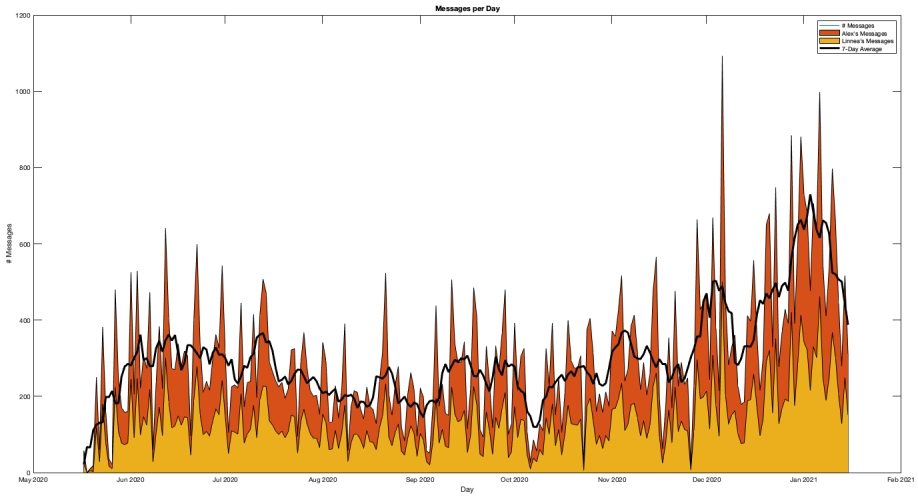
```
% Chains
chains = cell(size(msgs));
chainInd = 1;
for m = 2:numel(msgs)
    % am I TIMEOUT after last message?
    % If yes, start a new chain... if no, add to current chain
    if (msgs(m).dt - msgs(m - 1).dt) >= TIMEOUT
        chainInd = chainInd + 1;
        chains{chainInd} = msgs(m);
    else
        chains{chainInd} = [chains{chainInd} msgs(m)];
    end
end
chains = chains(1:chainInd);
solos = chains;
% clean the chains. If _all_ messages are from the same author... kill
 it?
for c = numel(solos):-1:1
    authors = [solos{c}.author];
    if ~all(authors == ALEX) || all(authors == LINNEA)
        solos(c) = [];
    end
end
% By Day
minDate = msgs(1).dt;
maxDate = msgs(end).dt;

dateRange = minDate:maxDate;
dd = yyyymmdd(dateRange);
numPerDay = zeros(size(dd));
alexPerDay = zeros(size(numPerDay));
linneaPerDay = zeros(size(numPerDay));
timePerDay = seconds(zeros(size(numPerDay)));
timePerDay.Format = 'hh:mm:ss';
chainInd = 1;
dailyMsgs = cell(size(dd));

authors = [msgs.author];
reps = yyyymmdd([msgs.dt]);
for d = 1:numel(numPerDay)
    mask = reps == dd(d);
    numPerDay(d) = sum(mask);
    alexPerDay(d) = sum(mask & (authors == ALEX));
    linneaPerDay(d) = sum(mask & (authors == LINNEA));
    % see if the current chain begins today
    while chainInd <= numel(chains) ...
            && yyyymmdd(chains{chainInd}(1).dt) == dd(d)
        timePerDay(d) = timePerDay(d) ...
            + (chains{chainInd}(end).dt - chains{chainInd}(1).dt);
        dailyMsgs{d} = [dailyMsgs{d} chains{chainInd}];
        chainInd = chainInd + 1;
    end
end
```

```
numPerHour = zeros(1, 24);
hrs = hour([msgs.dt]);
for h = 1:numel(numPerHour)
   numPerHour(h) = sum(hrs == (h - 1));
end

texts = {msgs.text};
words = cellfun(@split, {msgs.text}, 'UniformOutput', false);
textLens = cellfun(@length, words);
words = vertcat(words{:})';
words(cellfun(@isempty, words)) = [];
bag = bagOfWords(words);

fprintf('Analysis Complete in %0.4fs\n', toc);
```

Analysis Complete in 8.6136s

Plotting

```
close all;
plot(dateRange, numPerDay);
title('Messages per Day');
xlabel('Day');
ylabel('# Messages');
hold on;
area(dateRange, alexPerDay + linneaPerDay);
area(dateRange, linneaPerDay);
plot(dateRange, movmean(numPerDay, 7), 'k-', 'LineWidth', 3.0);
legend('# Messages', ...
    'Alex''s Messages', ...
    'Linnea''s Messages', ...
    '7-Day Average');
fig = gcf;
fig.Position(3:4) = [3 2].*fig.Position(3:4);
fig.Position(1) = 0;
```



Statistics

```
diary off;
diary ./results.txt
diary on;
numDays = sum(numPerDay ~= 0);
minMsg = min(numPerDay(numPerDay ~= 0));
[~, maxMsg] = max(numPerDay);
badDays = find([true numPerDay == 0 true]) - 1;
[longestStreak, dayInd] = max(diff(badDays));
streak = dateRange(badDays(dayInd) + 1):dateRange(badDays(dayInd+1) -
 1);

chainLengths = cellfun(@numel, chains);
chainDurations = cellfun(@(chain)(chain(end).dt - chain(1).dt),
 chains);

soloLengths = cellfun(@numel, solos);
soloDurations = cellfun(@(chain)(chain(end).dt - chain(1).dt), solos);

[~, longInd] = max(chainDurations);
fprintf('Messages sent: %d\n', numel(msgs))
fprintf('Number of days with messaging: %d\n', numDays);
fprintf('Number of days with no message: %d\n', sum(numPerDay == 0));
fprintf('Least messages in a day: %d\n', minMsg);
fprintf('Most messages in a day: %d on %s\n', ...
    numPerDay(maxMsg), datestr(dateRange(maxMsg), 'mmm dd, yyyy'));
fprintf('Words sent to each other: %d\n', full(sum(bag.Counts)));
fprintf('Average messages per day: %d\n', round(mean(numPerDay)));
fprintf('Average messages per day, per person:\n');
fprintf('\tAlex: %d\n\tLinnea: %d\n', ...
    round(mean(alexPerDay)), round(mean(linneaPerDay)));
fprintf('Longest streak: %s -> %s (%d days)\n', ...
    datestr(streak(1), 'mm/dd/YYYY'), ...
    datestr(streak(end), 'mm/dd/YYYY'), ...
    longestStreak);
fprintf('Number of distinct conversations: %d\n', numel(chains));
fprintf('Average number messages per conversation: %d\n', ...
    round(mean(chainLengths)));
fprintf('Average duration of conversation: %s\n', ...
    mean(chainDurations));
fprintf('Longest conversation: %s (%d messages)\n', ...
    chainDurations(longInd), chainLengths(longInd));
fprintf('Average number of words per message: %d\n', ...
    round(mean(textLens)));
fprintf('Average time per day: %s\n', mean(timePerDay));
fprintf('Most time in a day: %s\n', max(timePerDay));

fprintf('Popular emoji usage:\n');
for e = 1:numel(EMOTICONS)
    fprintf('\t%s - %d\n', EMOTICONS{e}, ...
        full(bag.Counts(bag.Vocabulary == EMOTICONS(e))));
end
```

```
fprintf('Hours (EST):\n');
for h = 1:24
    fprintf('\t%02d. %d\n', h - 1, numPerHour(h));
end
fprintf('Complete in %0.4fs\n', toc);
diary off;
```

```
Messages sent: 73429
Number of days with messaging: 243
Number of days with no message: 1
Least messages in a day: 10
Most messages in a day: 1092 on Dec 06, 2020
Words sent to each other: 685008
Average messages per day: 301
Average messages per day, per person:
 Alex: 163
 Linnea: 138
Longest streak: 05/19/2020 -> 01/15/2021 (243 days)
Number of distinct conversations: 1447
Average number messages per conversation: 51
Average duration of conversation: 00:52:52
Longest conversation: 08:44:52 (671 messages)
Average number of words per message: 9
Average time per day: 05:13:32
Most time in a day: 14:13:03
Popular emoji usage:
 ;) - 2339
 :) - 280
 :)) - 405
 :( - 290
 :(( - 31
Hours (EST):
 00. 7993
 01. 15586
 02. 12478
 03. 7101
 04. 2653
 05. 1016
 06. 259
 07. 29
 08. 77
 09. 152
 10. 143
 11. 359
 12. 2231
 13. 3003
 14. 2749
 15. 1780
 16. 1357
 17. 1340
 18. 1356
 19. 1147
```

```
 20. 1582
 21. 2187
 22. 3471
 23. 3380
Complete in 12.8533s
```

Published with MATLAB® R2020b