Understanding Relationship Dynamics Through Natural Language Processing of Instant Messaging Data

Michael A. Alcorn

Introduction

Michael (the author) is in a long-distance relationship with a girl, Katherine, who lives in Chicago. Because their relationship is long-distance, much of their communication occurs via instant messaging through the Google Hangouts platform. Further, as their relationship has been long distance from the start (March 2014), they have built up a considerable log of conversations up to this point, which, while not ideal from a relationship perspective, is perfect for gathering data for a natural language processing project. For my project, I quantitatively analyzed various aspects of the relationship between Michael and Katherine using a number of different natural language processing techniques, including: text classification, sentiment analysis, and document clustering (Katherine, who is a PhD student in evolutionary biology and also enjoys data analysis, has given me her permission to carry out the study).

Data Collection

I was able to download all of Michael's Hangouts data using the Google Takeout service. These data are provided as a single JSON file, which contains all of the messages sent to and from Michael. I then used Hangouts Reader to extract only those messages sent between Michael and Katherine. The output of Hangouts Reader is formatted such that each line represents a single message and each line includes: ***(1)*** a timestamp of when the message was sent and ***(2)*** the sender of the message.

Messaging Tendencies

For the first part of my analysis, I investigated the message sending tendencies of Michael and Katherine. In total, Michael and Katherine exchanged 36,423 messages between March 16th, 2014 and November 3rd, 2014, with 20,530 of those messages being sent by Michael and 15,893 messages being sent by Katherine. Michael sent a total of 162,114 words

for an average of 7.90 words per message, while Katherine sent a total of 150,614 words for an average of 9.48 words per message. The relative distribution of their respective message lengths can be seen in **Figure 1**.

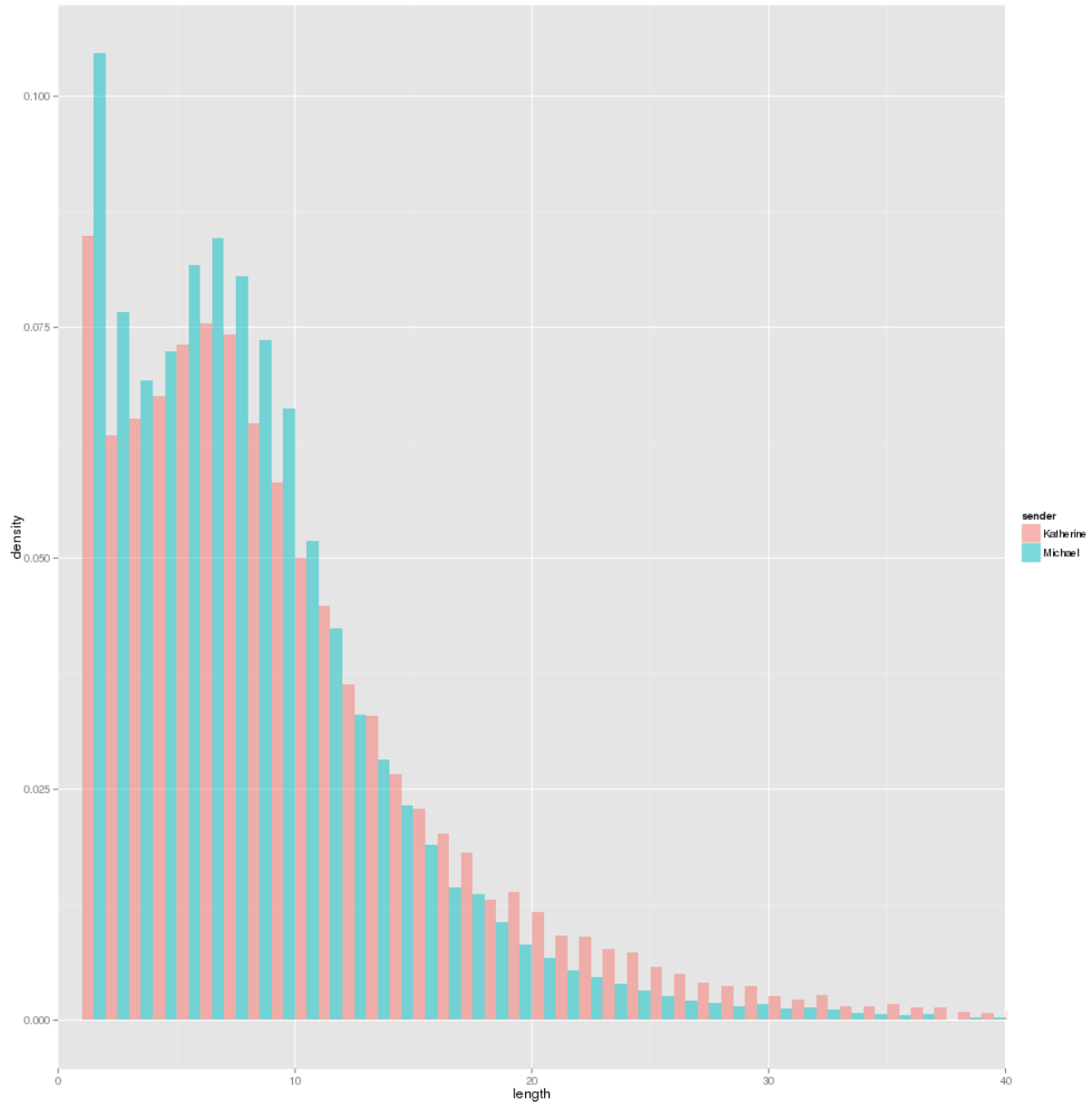

Figure 1. The relative distribution of message lengths for each sender.

As you can see in **Figure 1**, a much larger proportion of Michael's messages are extremely short, while Katherine's messages tend to be longer. This trend is supported by intuition, as

Michael typically sends multiple messages to express a single idea (e.g., by breaking a sentence into multiple messages at points where a comma would go).

The temporal messaging tendencies of Michael and Katherine can can be seen in **Figure 2**. The trends here are mostly unsurprising. Starting in the morning, there is a gradual increase in the number of messages sent until approximately lunchtime, whereafter there is a decline in the number of messages sent as the “grind” of the day sets in. As the end of the workday approaches, there is a steady increase in the number of messages, which continues until the two parties go to sleep at around midnight, which is clearly demarcated by the sudden drop in number of messages sent. Last of note are the vocabulary sizes of Michael and Katherine, of which Katherine had a vocabulary of 11,792 unique tokens, whereas Michael had a vocabulary of 10,733 unique tokens.

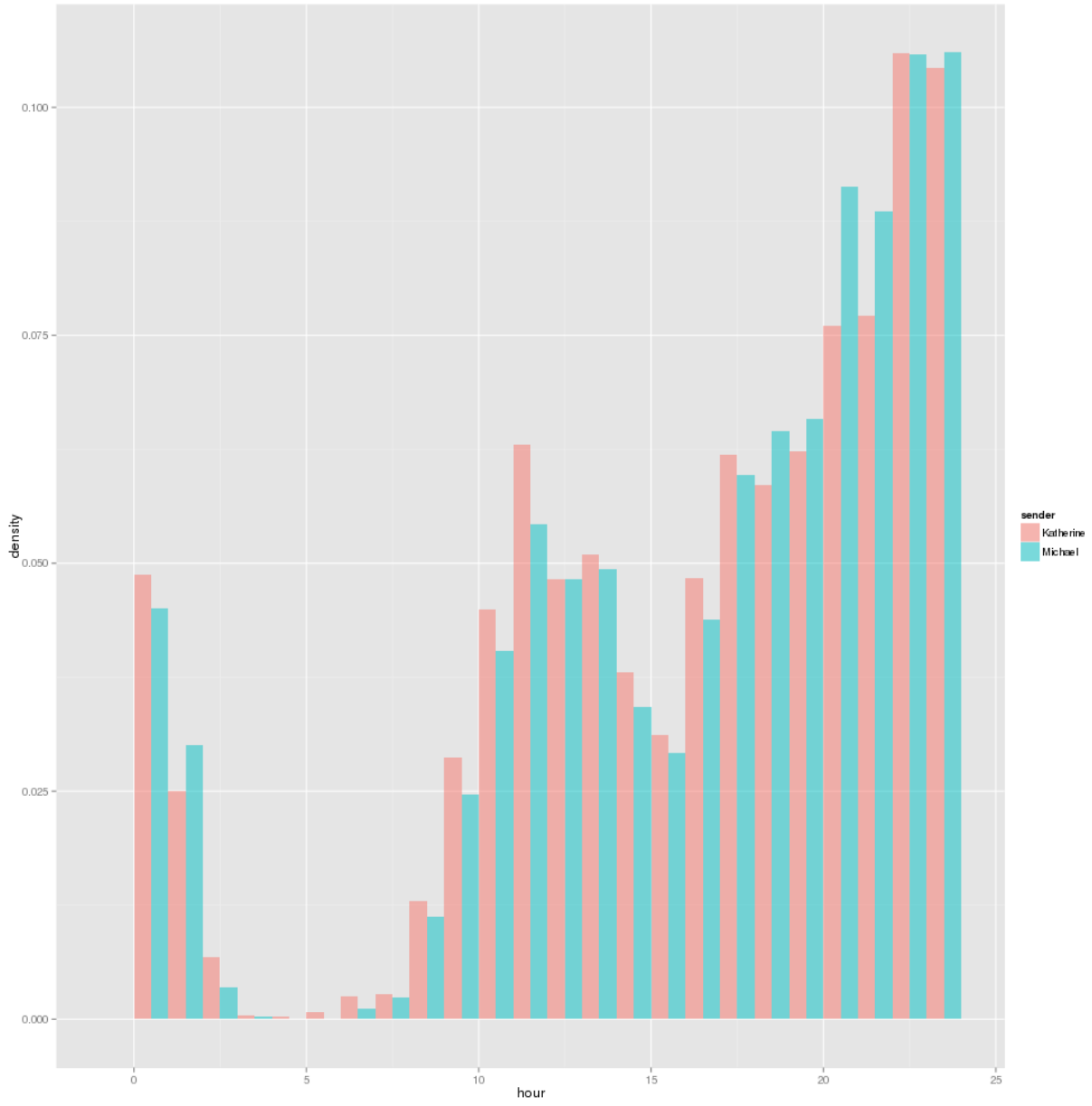

Figure 2. The relative distribution of message sending times for each sender.

Sender Classification

For the next part of my project, I implemented a naive Bayes classifier using the Natural Language Toolkit (NLTK) for Python to classify individual messages by sender. The dataset was trained on 75% of the messages sent by each party and then tested on the remaining 25% of messages for each party. Because Michael sent more messages than Katherine, classifying all messages as “Michael” would produce an accuracy of:

$$\frac{20{,}530}{15{,}893+20{,}530} = 0.563$$

which can be used as a baseline. The naive Bayes classifier achieved an accuracy of 0.692 with an F1 score of 0.704. This accuracy seems fairly respectable considering the extreme brevity of the majority of messages in the dataset. The ten most important features for distinguishing messages between Michael and Katherine can be seen in **Table 1**. The ratios are calculated as:

$$\frac{p(word|Katherine)}{p(word|Michael)}$$

So, for example, Katherine is 37.5 times more likely to use “ppl” than Michael. Many of the features shown in **Table 1** are quite intuitive, as using abbreviations is fairly characteristic of Katherine’s messaging style, while Michael frequently uses terms like “Word.” and “aw”. Other interesting features in the top 100 include: several different emoticons, which Michael frequently uses in his messages, “Red” and “Hat”, which is where Michael has been interning since May 2014 (Red Hat, Inc.), “Daniel”, who is Katherine’s brother, “grant”, which is something Katherine seems to be constantly working on, “wine”, which is something Katherine enjoys, and “Auburn”, which is Michael’s undergraduate alma mater.

Table 1. The ten most important features for distinguishing messages.

Word	Katherine : Michael Ratio
ppl	37.5 : 1.0
aw	1.0 : 36.4
min	28.8 : 1.0
Also	21.4 : 1.0
With	20.2 : 1.0
certainly	19.4 : 1.0

Word.	1.0 : 19.4
:-*	1.0 : 18.7
Truth	17.7 : 1.0

Katherine hypothesized that the couple's language may be converging as time passes. To test this hypothesis, I divided the dataset into five "phases" of the relationship and then trained a naive Bayes classifier on each phase. I then examined the accuracy and F1 score for each subset. If the couple's language was converging, you would expect the accuracy and F1 scores to decrease over time (because their messages would be more difficult to distinguish). In fact, the opposite was true, as can be seen in **Table 2**.

Table 2. The accuracy and F1 scores of different naive Bayes classifiers trained on different phases of the couple's relationship.

Relationship Phase	Accuracy	F1 Score
2014-03-16 to 2014-05-03	0.588	0.622
2014-05-04 to 2014-07-12	0.684	0.685
2014-07-13 to 2014-09-04	0.715	0.701
2014-09-05 to 2014-10-02	0.712	0.700
2014-10-03 to 2014-11-03	0.684	0.695

Comparison and commonality word clouds (an aesthetically pleasing way to visualize word usage trends) for the different phases of the relationship can be found in the appendix. The word clouds were created using the R package 'wordcloud', and the comparison clouds are described by the creators with the following:

"Let $p_{i,j}$ be the rate at which word i occurs in document j, and p_j be the average across documents $\left(\frac{\sum p_{i,j}}{ndocs}\right)$. The size of each word is mapped to its

maximum deviation (max_i ($p_{i,j}$ - p_j), and its angular position is determined by the document where that maximum occurs."

Sentiment Analysis

My sentiment analysis of the couple's messages consisted of two parts: ***(1)*** looking at the sentiment of all messages and ***(2)*** looking at how message sentiment changed over time. To determine the sentiment score for a message, I again used a naive Bayes classifier implemented in NLTK, but trained on the Sentiment140 corpus. The Sentiment140 corpus contains 1,600,000 tweets that have been annotated as either positive or negative (evenly split) using a maximum entropy classifier (Go, Bhayani, and Huang, 2009). Once the naive Bayes classifier was trained, I simply determined the probability of each message being positive and used that as the sentiment score. The relative distribution of message sentiments can be seen in **Figure 3**. The average sentiment per message for Michael was 0.454 while Katherine had an average sentiment of 0.442 per message. These results were somewhat surprising to the couple, as Michael is generally regarded as the more negative of the two (by a considerable margin). However, as you can see in **Figure 3**, Katherine has a markedly higher proportion of messages with an extremely negative score, which is likely driving down her average. The spikes for Michael's messages around 0.7 are being driven by his frequent use of "haha" and its variants, which is likely inflating his average sentiment score.

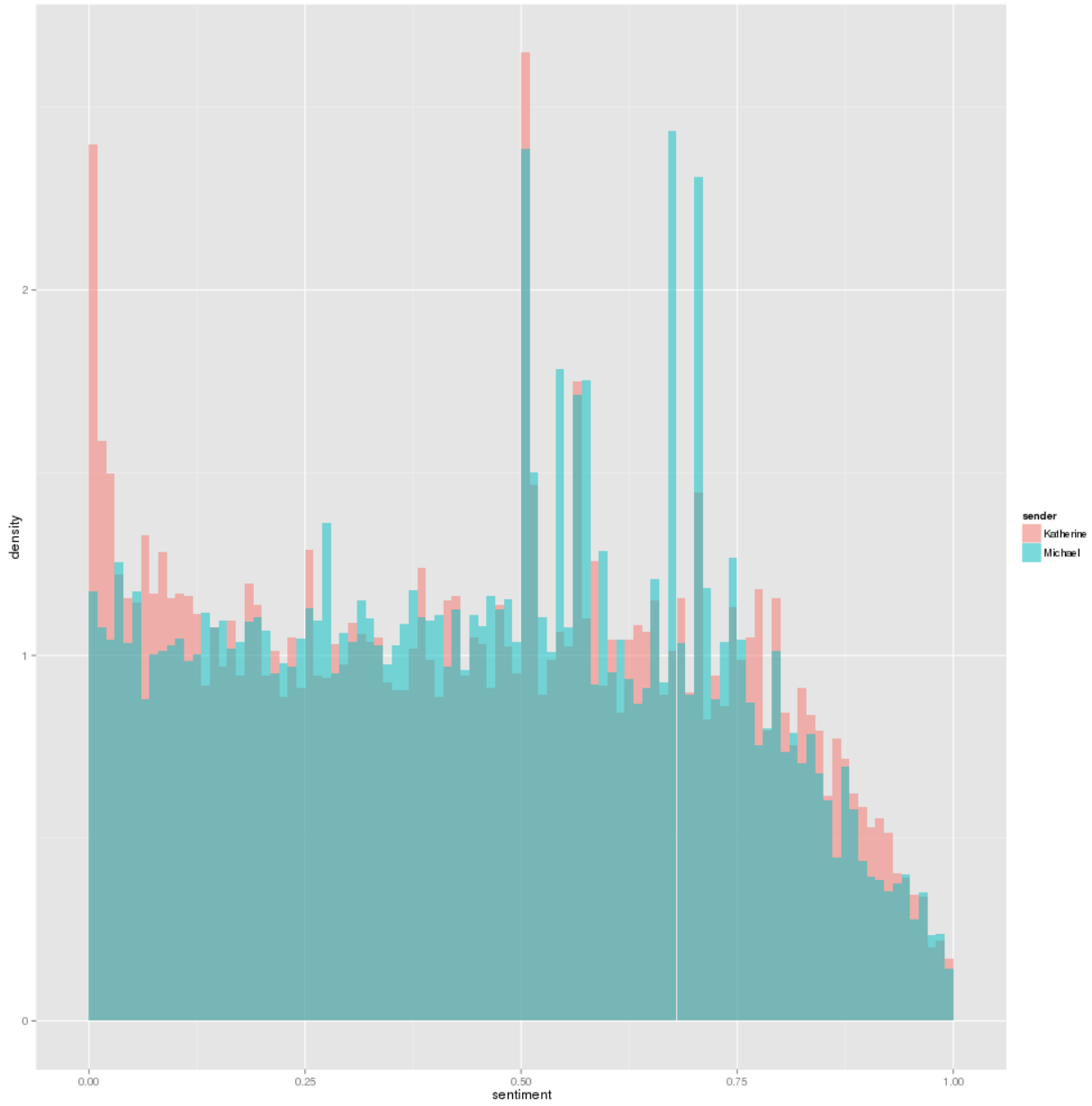

Figure 3. The relative distribution of message sentiment scores for each sender.

In general, the message sentiment scores seemed to support intuition. Examples can be seen in **Table 3**. However, it is worth noting that, in the training data, many sexually explicit terms and swear words have a higher probability of being present in negative tweets than they do of being in positive tweets, which is likely deflating the sentiment scores of certain messages, as these probabilities do not reflect the way Michael and Katherine communicate.

Of course, ensuring that the training set accurately represent the test set is a vital part of conducting successful machine learning work, and this is an area I would like to approve upon if given more time.

Table 3. Example messages and their sentiment scores.

Message	Sentiment Score
Now who's giving out too many compliments?! But seriously, you make me laugh and smile a lot too. And it's refreshing for someone other than myself to appreciate my lame sense of humor	0.998
Lemme just say again real quick how handsome you are. On top of innumerable other amazing qualities :)	0.997
Good morning, beautiful. I hope you have a nice breakfast!	0.992
well that's a nice way of putting it! haha. I'm glad you're into it. I like how you have a lot of things to say too.	0.963
I'll bring a swimsuit	0.498
I forwarded you the reservation.	0.495
With most flights over 400, so we should buy sooner rather than later	0.490
Fucking TWO different times people were giving away their home run derby tickets today and both times they were gone by the time I got there.	0.00254
Yeah I had a headache when I woke up, but it feels worse now and I'm starting to feel a little nauseous :(	1.84e-05
Ugh I slept terrible. At one point I woke up and I swear. I saw a mass of writhing dead hands hovering over the foot of my bed. Even when I blinked they stayed there. When I turned on my lamp it was just a towel over the fan	2.54e-06

The average weekly sentiment of messages can be seen in **Figure 4**. The two highest spikes occur in May and in August, right before the couple was to be physically reunited after extended periods away from each other. The three lowest spikes come at times of discord in

the relationship, but it is worth noting the scale in this figure, as the lowest average weekly sentiment does not go below 0.375.

Figure 4. The average weekly sentiment of messages.

Conversation Clustering

The clustering component of my project ended up being the most frustrating. My original hope was that the conversations between Michael and Katherine would cluster naturally into categories that might make sense, based on what is known about the couple

(e.g., “science”, “music”, “school”, “sex”, “movies”, “family”, “friends”), but I was not able to obtain particularly clean results. I think the problem lies in the fact that the conversations between Michael and Katherine are simply not one-dimensional, which results in multiple topics being covered in a single conversation, thereby muddying the clustering process.

Defining what constitutes an instant messaging conversation is, in itself, a nebulous task, as extended breaks between responses are quite common. For my study, I defined a gap between messages of one hour or greater as the end/start of a conversation, which resulted in 855 conversations being present in the dataset (an average of 3.69 conversations per day). The conversations were then stemmed using the Porter stemmer available through NLTK. I then created a term frequency-inverse document frequency (tf-idf) matrix of the stemmed conversations using the tf-idf tools available through scikit-learn. From here, I ran the scikit-learn implementation of k-means on the tf-idf matrix to cluster the documents into 50 clusters. The clusters and their most important terms can be viewed here.

Of note are several clusters relating to travelling, a cluster relating to schoolwork and Katherine’s research (oyster genetics), and a cluster about moving. There are several clusters that seem to cluster around a singular word or phrase, such as “eff yeah” and “mwah”, and there are a few catch-all clusters, which seem to attract long conversations that do not necessarily have anything in common.

Conclusion

The results of this study suggest natural language processing can be a useful tool for understanding relationship dynamics. Sender classification can help identify those things that are important to each party by extracting frequently used words over defined periods of time. Sentiment analysis could be valuable to couples in therapy, as it can pinpoint negativity that might not be inherently obvious to the couple. Lastly, conversation clustering can reveal common interests shared by a couple, which can then be explored further.