Predicting Messaging Response Time in a Long Distance Relationship

Meng-Chen Shieh

m3shieh@ucsd.edu

I. Introduction

The key to any successful relationship is communication, especially during times when hundreds of miles separate two people apart. With each person leading his or her own busy life, Facebook Messenger has provided a way to share our lives with each other instantly, almost as though we were in each other's company.

A less intended and used feature of Facebook Messenger is the chat history that comes as a product of digitalized communication. Especially in the case of long distance relationships, the dialogue is dense and in abundance enough that it has the potential of showing patterns and providing insight into.

The most grueling part of a long distance relationship is probably the waiting. In this paper, aside from showing general trends in the exploratory analysis portion, I will aim to estimate the response time from both his and her perspective.

II. Dataset

I have used the Chrome extension Facebook Chat Downloader to download the complete chat log history as an .htm file. The raw file contains messages from November 13, 2013 to November 29, 2015, totaling 69,653 messages, where each time the Send button is pressed counting as one message.

I then parsed the raw file into a JSON-like format, where each object contained four fields:

a. Sender
b. Date and Time
c. Message
d. Response Time

Examining user habits, I realized that in a conversation one user would press Send multiple times, breaking the sentence up so as to be easier to understand. Because of this, in the case where there are subsequent messages from the same sender spaced less than 2 minutes apart, I concatenated the messages and set the time of the first message as the Date and Time.

After that, I subtracted the Date and Time of each message with the Date and Time that same user last received a message to get a response time, represented in seconds.

Lastly, I eliminated the first 470 JSON objects, since starting from 471 is when the long distance relationship began. This amounted to a total of 32,349 data points from March 13, 2014 to November 29, 2015.

One limitation of this dataset is since Facebook only records the message time up to the minute, although response times are calculated in seconds, they show up as 60 second increments.

III. Exploratory Analysis

A. Distribution of Messages

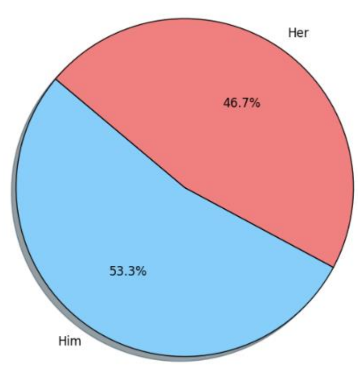

Figure 1

Her	Him
45.09007365	51.82210906

Figure 2: Average character count in message

Figure 1 shows the proportion of messages in the dataset sent by Him and Her. Combined with Figure 2 below, which shows that, on average, His messages are six characters longer, I conclude that in the context of chat, He is a bit more talkative.

B. Response time

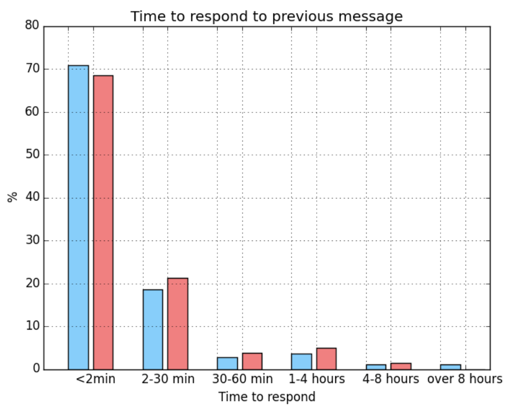

Figure 3

	Her	Him
Mean	1927.61514	1446.192493
Median	60	60

Figure 4: Response Time Statistics in seconds

Figure 3 shows the about 70% of messages sent from both Him and Her are responded to within two minutes. This makes sense because messaging often happens like a real-life conversation, with immediate response times. She currently lives in a city that requires more driving than He, so that could account for her slightly longer response times. Messaging during the night when one person has gone to bed could account for the >4 hour response time, which leads to the next analysis looking at the time of day messages are sent.

Figure 4 illustrates a huge difference between mean and median response times. From this statistic the conclusion can be made that at least half of all messages sent are responded within a minute.

C. Messages at different times of the day

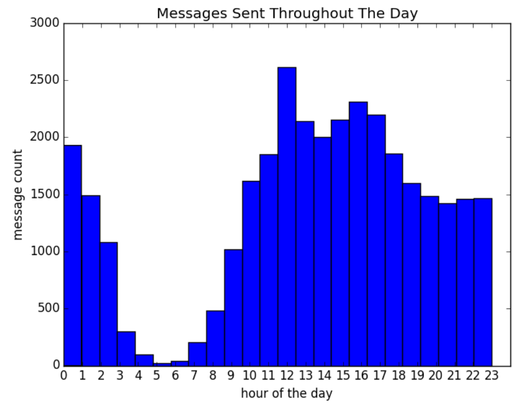

Figure 5

In Figure 5, messages are grouped by the hour they are sent and presented as a histogram. Even though He and She are in a long distance relationship, they are fortunate enough to be in the same time zone. The significant drop at 2am and rise at 9am is due to sleep, and the later than usual nature could be due to their current college lifestyle.

IV. Predictive Task

As instantaneous as Facebook Messenger has made long-distance communication, it is inevitable that sometimes, all you can do is wait. In this project, I am interested in building a model, given a sent message, predict how long the response time will be from the other end.

I started out trying to predict exactly how many seconds the next message's response time would be, but due to the nature of the dataset, I was unable to produce meaningful results (more in Section VII). This led me to adjust my predictive task to predict whether a message would be responded to within a certain period of time.

I measured the performance of models using the mean-squared-error (MSE) and the coefficient of determination R^2 metrics. The formula is as follows:

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

$$R^2 = \frac{MSE}{variance\ in\ true\ ratings}$$

Where N is the number of samples, (Y hat) is a vector of predictions and Y is a vector of observed values.

MSE is good for comparing different models to each other, but since MSE depends on the variance in the data it is measuring, it is not a good measure to determine whether a model is good at predicting the data. R^2 hence takes into account fluctuations in MSE due to the variance in the data itself by dividing MSE by variance, and thus is a better representation of the accuracy of the model.
In the classification task, I used percentage to represent accuracy of prediction.

I used the following models for prediction:

Baseline: Predict next response time solely from previous response time
When two people are in dialogue, it is likely that most of the time, the response time will be similar. This model doesn't account for ending and initiating of conversations, but this model assumes those are relatively fewer occurrences.

Logistic Regression:
Logistic Regression is a binary classifier that essentially draws a line through the sand.
Using a set of crafted features, we can write a linear model to map how each feature influences the response time as follows:

$$Y = X \cdot \theta$$

Where X is the feature matrix θ is the learned vector representing each feature's influence. The trained θ is the line in the sand, but error can be further minimized if we don't necessarily need a linear decision boundary. This is where SVM comes in.

Support Vector Machine
Support Vector Machines train a classifier that focuses on "difficult" examples by minimizing misclassification errors.

V. Feature selection

From the exploratory analysis, there are several features that could potentially have an impact on response time.

1. Sender: It is worth noting that this is information about the sender of the current message, and we're trying to estimate how long it will take the other person to respond. Since She generally takes longer to respond, if the Sender is Him, θ value corresponding to the feature should be larger than if She were waiting for His message. This feature will be represented as one binary value, He as 1 and She as 0.

2. Time of day: As seen in the exploratory analysis, times where people are expected to be sleeping have the least messages sent, and messaging peaks around noon. Thus messages sent in the middle of the night should expect to have a longer response time then those sent during the day. By dividing the day up into two hour increments, this feature is represented by 11 binary values, where if the message falls in the (04:00-06:00) timeslot, all 11 values are 0.

3. Number of characters: Intuitively, when talking about a topic, there will be more characters per message to convey an idea via text, compared to if the messages are meant to exchange pleasantries. Also, in the middle of a conversation, there tends to be shorter response time. This feature is represented by one column, where the value is the character count in the message.

VI. Related Work

As instant messaging grows more and more important in daily communication, adopted not only in personal life, but also widely used in the workplace, the resulting records has presented itself to be a subject of study for human computer interaction researchers.

In *Responsiveness in Instant Messaging: Predictive Models Supporting Inter-Personal Communication* by Daniel Avrahami and Scott E. Hudson, the researchers examine instant messaging in the workplace using features such as tasks being executed at the same time and status displays.

In *Didn't You See My Message? Predicting Attentiveness to Mobile Instant Messaging* by Martin Pielot et al, the researchers predicted whether a user will view a message within a few minutes using a random forest classifier.

This project was originally inspired by a blog post by Dadadly on iwoaf.com.

VII. Prediction Results

In conducting the predictions, I split the data 80%-20%, with the latter used for testing. Of the First 80%, I did another 80-20 split, 80 being used for training and 20 for validation.

Initially, the goal was to predict the response time for the next received message. Using linear regression with the features described in section V, the MSE was 72984504.4525 and R^2 value was 0.9925. The large MSE was mostly consisted of variance in the dataset, and the predictor didn't do much predicting at all. Taking into account messages can be grouped into conversations, where each consecutive message is less than a few minutes apart, I limited the prediction to only the messages that had response time of over 2 minutes. This made the MSE 72984504.4525 and R^2 0.96448: slightly better, but still largely trivial. At this point I decided to change strategies. Instead of trying to predict exactly how many seconds it would take to get a response, I decided to turn this into a classification question: Given a message, predict whether a response will be received in a certain amount of time. I used three time intervals for prediction: 2 minutes, 5 minutes, 10 minutes.

1**. Baseline: Predict next response time solely from previous response time:**

2 minutes	5 minutes	10 minutes
63.83%	72.71%	75.37%

It can be inferred that messaging response ties have a strong temporal effect. How long the sender took to respond to the previous message can be an indicator as to how long it will take to receive the next message. As the threshold grows higher, so does the accuracy; this is because the 72% that got a response within 5 minutes includes the 63% that got a response within 2 minutes.

2. Logistic Regression

	2 minutes	5 minutes	10 minutes
Test	65.58%	70.88%	77.11%
Train	65.98%	71.41%	79.02%
Validate	65.07%	71.41%	79.01%

In the 2 minute and 10 minute category, the logistic regression predictor does only slightly better than the baseline, while in the 5 minute category, it does worse, but also not by much. Based on the results from the test set, I conclude the logistic regression model and the baseline have the same performance, where slight variance is negligible.

3. Support Vector Machine

	2 minutes	5 minutes	10 minutes
Test	64.78%	76.55%	82.43%
Train	87.72%	90.91%	92.61%
Validate	66.28%	72.98%	73.01%

Except for the two minute category, whose performance is about the same as the baseline and logistic regression, SVM does significantly better than the other two models.

VIII. Conclusion

One of the first challenges that were encountered was the difficulty of predicting an exact response time. A linear regression model probably would have performed better with a more fine-grained feature-vector, but due to the high variance and volatility of human factors, such as schedules that change every quarter, in-person visits that are not accounted for in the message log, etc.

It proved to be a much easier task to estimate in time intervals. Instead of asking, "How long will it take to get a response?" asking, "Will I get a response in x minutes?" is a question the models in this paper can answer much better. Because the users choose to use instant messenger to hold real conversations, the predictor can achieve relatively high accuracy simply by mirroring the response time of the current message.

Between logistic regression and support vector machine models, SVM achieves a high accuracy, with the added benefit of being relatively resistant to overfitting, it would be the model of choice.