Determining If A Coffee Chat Has Been Scheduled

Dept. of CIS – Senior Design 2015-2016

Hong Kim
honki@seas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

Minsu Kim
minski@sas.upenn.edu
Univ. of Pennsylvania
Philadelphia, PA

ABSTRACT
We schedule to meet people everyday, whether it is to pick up keys from an Airbnb host or to go on a date with your match on OkCupid. Service providers such as Airbnb or OkCupid have great incentive to know if the two parties involved have scheduled to meet because failure to do so often indicates a fail rate of the service itself. However, there is no easy way to automatically determine this because scheduling often happens over free-form text messages. The goal of our project was to build a classifier that would determine if a meeting has been scheduled given a text message exchange as input.

We ran a service called FreeForCoffee where users were paired with other people to schedule coffee via SMS. Messages that were exchanged on this platform were used as the data source of our project. We built an anonymized labeling web interface where we labeled text messages with four labels: time, location, agreement, and cancel. With the labeled data, we worked on label prediction and ultimately trained a decision tree classifier that could accurately predict whether a coffee chat was scheduled given a text conversation.

1. INTRODUCTION
Interest for this problem stemmed from a personal side project of one of the authors called FreeForCoffee. It is a service used on Penn’s campus to help students meet other students within the same student group. Every week, it asks users of their availability, then connects two people who are both free via text message (SMS). Currently, it is being used by over 30 groups and 1000 students at Penn. Up to a 100 matches are made on FreeForCoffee each week. The natural question that stemmed from that fact was how many of those matches actually schedule coffee and meet.

There are a few ways to go about answering that question. The most obvious one is to survey the users, and FreeForCoffee actually does this. It sends a survey text asking whether you’ve met and how the meeting was a week after the initial match goes out. Around 60% of the users reply to this survey text. We could calculate an average scheduling rate based on the survey result and expect that this scheduling rate is representative of the entire user base. However, this approach fails on grounds of selection bias. For example, it is not hard to imagine that people who actually met with their match are more likely to respond to the survey text, because they have more good will towards the service.

Another approach might be to sample a portion of the data set and manually determine the percentage of matches that actually schedule. This would eliminate the kind of selection bias mentioned above. The limitation of this approach is that it is costly to repeat. If you want to know how the schedule rate changes over time, you would have no other option but to manually determine this every week. This shortcoming points to a strong need for an automated classifier that would accurately predict if a match has been scheduled given its text message exchange. This is the goal of our research.

![Figure 1: Goal](image-url)
Any service that involves bringing people to meet, whether it is dating applications such as Tinder and Okcupid or others like Airbnb, there is the need to automatically classify if an objective has been achieved via a text message exchange. Our hope is that our solution could be useful for this broader set of services as well.

2. APPROACH

2.1 Data Collection

Our data comes entirely from the aforementioned service, FreeForCoffee. We used all the data that was collected over the course of this academic year. Below is the number of cumulative matches (actual SMS conversation threads) we have collected over time.

![Graph showing cumulative matches over time]

A total of 1359 matches were collected. In terms of individual text messages, it is 13,103, so on average about 10 messages were exchanged per match. In the graph above you can also notice that at early parts of the semester there are steeper growth. It is because there is most desire to meet new people early-mid semester.

2.2 Data Labeling

First thing we did, after ensuring that we were on track to collect enough data, was to build a web interface to label anonymized data for future supervised learning. Below is a screenshot of the interface that we had built.

![Screenshot of labeling interface]

2.3 Machine Learning

2.3.1 Label Prediction

After we had labeled some of our data. We started working on label prediction, because ultimately the solution should not rely on human beings having to label messages. For a while we struggled with
training a logistic regression classifier for each label. We tried unigram and bigram features along with rule-based features such as the existence of a question mark in a message, but we ended up going with a regex type approach that later further simplified to a whitelist of keywords. The issue with logistic regression classifiers was that since we had a relatively small data set, and the signal was low (for example, the percentage of cancel messages is less than one percent), precision always turned out to be very low. In other words, the classifier would under predict labels. Using a simple whitelist of keywords to predict labels would lead to over prediction in some cases, but as we will explain later, it returned better results for the ultimate scheduled/not scheduled decision.

The following are the whitelists of keywords that we settled on for each label. These were mostly adopted from the top features extracted from the logistic regression classifier.

<table>
<thead>
<tr>
<th>Label</th>
<th>Whitelist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>digits from 1-9, noon, mon, tue, wed, thu, fri, sat, sun</td>
</tr>
<tr>
<td>Location</td>
<td>joe, starbucks, commons, hubbub, hubbub, capo, united, william, dunkin, tap, metro, sweetgreen, saxby, cosi, chattime, kiwi, sweet green, frontera, greek</td>
</tr>
<tr>
<td>Agreement</td>
<td>see you, sounds, good, works, perfect, great, yes, yeah, sure, yup, awesome, okay</td>
</tr>
<tr>
<td>Cancel</td>
<td>reschedule, fuck, forgot</td>
</tr>
</tbody>
</table>

Figure 5. Whitelists for labels

If the body of a message (taken in lowercase) included any of these keywords as a substring, we predicted that message to have the corresponding label.

2.3.2 Scheduled Prediction

We also tried a logistic regression to predict whether a match is scheduled for coffee given a sequence of labeled messages. We first attempted with the actual sequence of labels and then with the frequency of labels appearing in a match. In the end, this approach did not work well because with 1359 matches each sequence or frequency would only have one or two data points.

We had much more success training a decision tree classifier with boolean features. We experimented with different features but settled with the following three. We refer to a message with a certain label by just the name of the label for brevity. So for example, “location” refers to a message with a location label.

i. Location exists?
   Checks whether a location was ever mentioned.

ii. Agreement follows time and location?
   Checks if an agreement was reached after a time and location.

iii. Cancel follows time and location?
   Checks if cancellation occurred without rescheduling (reaching a new agreement).

With these three boolean features, we trained a decision tree classifier that would predict whether a coffee chat has been scheduled or not given a labeled message sequence.

3. RESULTS

3.1 Label Prediction

The results of individual label prediction based on the whitelist approach was as follows.

<table>
<thead>
<tr>
<th>Label</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>0.85</td>
<td>0.92</td>
<td>0.66</td>
</tr>
<tr>
<td>Location</td>
<td>0.96</td>
<td>0.67</td>
<td>0.93</td>
</tr>
<tr>
<td>Agreement</td>
<td>0.79</td>
<td>0.87</td>
<td>0.40</td>
</tr>
<tr>
<td>Cancel</td>
<td>0.98</td>
<td>0.30</td>
<td>0.34</td>
</tr>
</tbody>
</table>

Figure 6. Label prediction results

For the time and agreement labels our predictions have high precision and low recall. This means that our list of keywords was comprehensive but tended to have many false positives. For example, numbers that do not indicate time appear in messages occasionally. The location label on the other hand had lower precision and higher recall. This is because keywords in location’s whitelist tend to be names of coffee shops such as starbucks. There is a low chance that the word “starbucks” means something else than the coffee shop we imagine. However, we have lower precision because there is many ways people talk about a location that is not captured in the whitelist.
Predicting the cancel label is hard in general because it occurs much less frequently than the other labels and there are many forms in which it manifests in speech.

3.2 Scheduled Prediction

The result of training a decision three with the three boolean features mentioned previously was the following:

```
   1. Location
      False  True
      Not Scheduled  3. Cancel

   2. Agreement
      False  True
      Not Scheduled  Scheduled
```

Figure 7. Output decision tree

Once again, the three boolean features were
i. Location exists?
ii. Agreement follows time and location?
iii. Cancel follows time and location?

Qualitatively the decision tree makes sense. Following the “True” path, it reads that if a location message exists, if cancel message does not follow the last agreement message, and if an agreement message that follows time and location exists, then we deem the match to have been scheduled.

The quantitative results of this decision tree’s ability to predict whether a match has been scheduled was the following:

<table>
<thead>
<tr>
<th>Input Type</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labeled Data</td>
<td>0.96</td>
<td>0.94</td>
<td>0.97</td>
</tr>
<tr>
<td>Predicted Label</td>
<td>0.86</td>
<td>0.79</td>
<td>0.81</td>
</tr>
</tbody>
</table>

For manually labeled data (gold standard) the prediction was very accurate validating that the decision tree itself functions well. There is a noticeable drop in performance for the predicted label data due to the inaccuracy in the label prediction, but compared to the baseline of 55% (which is the percentage of not scheduled matches), 86% is significantly better.

4. PRIVACY CONSIDERATIONS

The approach of this solution involves manually labeling messages in conversations. Although most conversations are simple scheduling conversations, going back and forth about time and location, people could feel that their privacy is breached. We tried to take this concern into consideration by removing all personally identifiable information such as name or student group while labeling the data.

5. DISCUSSIONS

A decision tree trained with whitelist-based label prediction data is a simple and effective solution to this problem. Once the keywords in the whitelist has been determined, which can be achieved with some manual work upfront, the classification in the future is automatic and instantaneous with a reasonably high accuracy which was the goal of the project.

With more time we would have worked on further improving label prediction since that is the current bottleneck to our overall prediction accuracy. For example, we could have done a parts of speech analysis and verified that keywords we think to be locations are used as nouns. Similar verifications can be done for other labels as well.

6. REFERENCES