Exploring sentiment and emotion in two-party collections calls

Caitlin Hilverman, Insight Fellowship Project, Summer 2019

For my Insight project, I worked with data from Prodigal Technologies. They use AI and machine learning to provide services for collections companies with the goal of enhancing borrower compliance and agent performance. This is a market that is ripe for the introduction of AI and machine learning analytics and techniques; the traditional model for assessing compliance and performance is sampling, listening to, and scoring calls by hand. This is time intensive and does not give a clear picture of how agents and companies are performing. Prodigal fills that role by using automated speech recognition and analytical models to generate alerts and reports to provide their customers with.

I worked towards the goal of creating a “sentiment track” to integrate into their call flow. Sentiment analysis is typically done at a whole document level and captures sentiment with a broad brush. I extended sentiment analysis in the following ways: I (1) used VADER to assign a sentiment score to a corpus of data, with a score assigned to each turn (agent or borrower) and generated plots to demonstrate on the level of a single call how sentiment changed for both the Agent and Borrower, and (2) designed a labeling scheme that captured and labeled different emotions for both Agents and Borrowers. I then used supervised machine learning to train learning algorithms on the labeled codes for Borrower emotions for classification purposes.

1.1. Using VADER for sentiment analysis of dialogue

After poking around for different options regarding sentiment analysis, I converged on the VADER sentiment analysis tool (https://github.com/cjhutto/vaderSentiment). VADER is a pre-trained lexicon that is specifically attuned to social media data. Although not social media data, the call transcripts that I worked with for this project were similar in a number of ways: each turn in the conversation is relatively short - an average of just 12.6 words per exchange - and the exchanges lack grammatical information (because they are talk-to-text without any punctuation). VADER also takes degree modifiers (intensifiers) into account when calculating sentiment score. For example, "I really really do not want to pay you" is rated as more negative than, "I do not want to pay you". Given that I was implementing this on naturalistic text, it was important that intensifiers were considered in determining valence.

Further, VADER is pre-trained so it doesn't require any training data (the transcripts were initially unlabeled for any sort of emotion). And because it doesn't require training data, VADER is fast and computationally cheap. Prodigal is interested in one day integrating a sentiment track in real-time, and VADER would be able to do so with almost zero lag. As such, I calculated a VADER valence score for each turn in the corpus, separately for Agents and Borrowers.

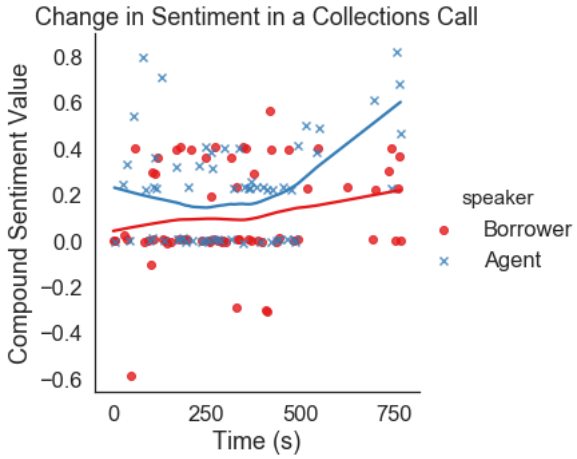
1.2. Calculating sentiment change over time

For the current purposes, I focused on the compound score. The compound score is a normalized weight composite score of the sentiment for the each turn. It's calculated by summing the valence scores of each word in the lexicon and is normalized to be between -1 (negative) and +1 (positive). VADER’s creators call this the "most useful metric if you want a single unidimensional measure of sentiment for a given sentence".

When looking at the distributions of the compound score for both Agents and Borrowers, I found that both were skewed towards the positive end of the scale. This could be due to the nature of the conversations - conversations between strangers likely incorporate more positive sentiment than less formal conversations - and could also be due to that lexicons in general tend to be better at capturing positive emotion.

After using VADER to find a compound score for each turn, I was left with a variable with a compound score for each speaker (Agent and Borrower) for each turn in each conversation. This is what I used to explore and characterize different kinds of sentiment flow in these conversations. Each turn had a time stamp for when each utterance was produced during a

conversation (start time and end time; I used end time for the visualizations). This allowed me to create visualizations of how sentiment changed over time for each conversation partner.



1.3. Characterizing different types of sentiment flows

From these visualizations, it is possible to identify different "types" of sentiment flows in conversations. For example, this figure demonstrates what would be characterized as a "warm up"; the borrower is initially defensive and wary, but after some positive cajoling from the agent they eventually provide their credit card number and make a payment (leading to increasing sentiment scores in the agent).

1.4. Sentiment analysis/things to note

- The sentiment score derived by VADER appears to become less accurate with increasing number of words. The average length of a turn in this dataset was 12.6, but the range extended past 500 words. It might be worth cutting long turns down first before calculating a VADER score.
- I have not normalized the time variable for the visualizations, but will likely be important for cluster analyses to assess differences in sentiment flow.
- The visualizations were generated with lmplot from the seaborn package (with loess = True for smoothing). scipy's interpolate package might also be a good route for these.

2.1. Labeling emotions in collections calls transcripts

To capture emotion on a more fine-grained level, I decided to do hand label the emotions from the transcripts to train a supervised learning algorithm. After poking around in the transcripts and listening to some of the recordings, it became clear that Agents and Borrowers have different sets of emotions that they exhibit in the phone calls that should be classified separately. I focused on only the Borrower's emotions. I converged on five main emotion categories: *polite*, *angry*, *avoidant*, *confused*, and *no emotion*. I was slightly concerned about training the model on the *no emotion* category; these made up a vast majority of turns and were incredibly variable. I tried to capture as much variability as I could in this category. I labeled roughly 100 in each category.

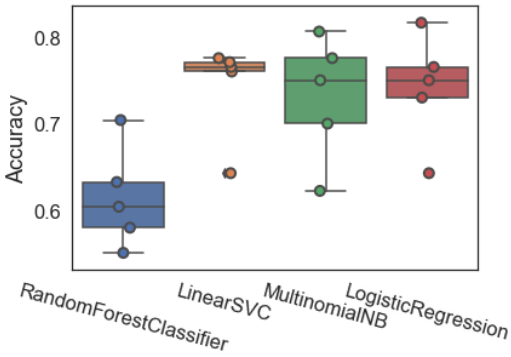
2.2. Converting transcripts to tf-idf vectors

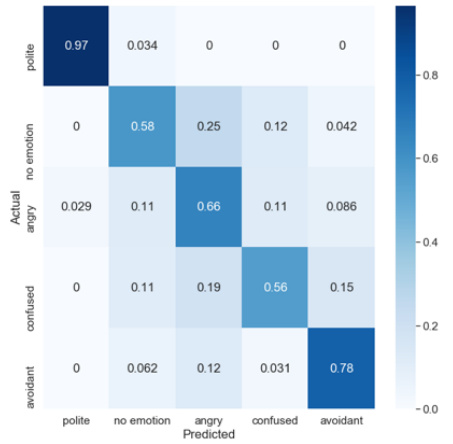
I chose to calculate tf-idf rather than using a different approach because the *relevance* of the words in a category were more important for classification than the frequency. Since the lexicon is fairly constrained with respect to calls (e.g., most documents likely contain words like "bill" and "payment" regardless of class), I needed to use a method to hone in on words that were both frequent and specific to the class that the algorithm would learn to categorize the turns into. For example, the bigram "thank you" was present across all categories; indeed, it was the most frequent bigram for three of the five categories. Because this bigram was present across documents, it would be down-weighted relative to other words in the corpus.

2.3. Comparing different supervised learning models.

Exploring sentiment and emotion in two-party collections calls
Caitlin Hilverman, Insight Fellowship Project, Summer 2019

I tested four different models: Logistic Regression, Multinomial Naïve Bayes, Linear SVC, and Random Forest. Three of the four models performed fairly well; accuracy exceed ed 70%. I moved forward with the One vs. All Logistic Regression model. The model had variable performance on the categories; it excelled at identifying the *polite* class. Unsurprisingly, the worst performance was in the *no emotion* class. To mitigate this issue I integrated the VADER scores into the data frame and excluded all turns that had a compound score of 0. This removed around 30% of turns, and it increased model performance by 10%. This decision was based on the assumption that turns with 0 sentiment likely also have no emotion being expressed.





2.4. Emotion classification: things to note

- The model was fast and accurate with just 100 labels per class. Increasing performance might be as trivial as doing some additional labeling, expanding to include more categories, or expanding the range for VADER scores that are excluded.

Future Directions

- *Examining sentiment in real time.* Because VADER's lexicon can be accessed very quickly, it seems possible that the documents can be fed into a sentiment analyzer as they are being transcribed with very little lag. Introducing this into the call flow would allow for fast and easy access to past conversations that could be quickly visualized.
- *Characterizing and predicting changes in sentiment flow.* Although it is useful to have access to individual call sentiment flows, categorizing a call by the "type" of flow that is exhibited could be useful for assessing agent performance and spotting issues. This could be done using a cluster analysis. Similarly, it might also be useful – particularly in real-time - to be able to predict when the sentiment is going to change based on the words that are being used.
- *Assessing lexical entrainment as a measure of social rapport.* When conversation partners are having a good social interaction, they tend to use and re-use the same words (called lexical entrainment). I would suspect that agents that have higher rates of similar word use with borrowers also tend to have more positive experiences/are more likely to make collections. This could be an interesting avenue to explore down the line.
- *Visualizing differences in emotion with sentiment scores.* Since we now have sentiment scores and emotion labels for each turn, it is possible to visualize them both together. An easy way to do this would be to plot sentiment scores as is done above, and then have the point specify which emotion is being expressed in that turn.

If you have any questions or comments about the above document, please contact me at caitie.hilverman@gmail.com.