Fan Speak: A Language Analysis of NFL Fan Comments

Chad Harness[1], Rob Mulla[1], Keri Wheatley[1]
University of California, Berkeley, CA, School of Information

Abstract

Sporting events evoke strong reactions among fans and thus provide a rich exploration of emotive language. The rising popularity of online message boards and increased availability of computational tools provide new ways to examine emotive language during game play. This paper provides an analysis of the language used in comments posted to Reddit, a community-determined online content aggregator, by fans during American NFL football games in 2017. We explore different computational methods to better understand the language used in this corpus.

1. Introduction

When the Eagles defeated the Patriots in the 2018 Super Bowl, thousands of Eagles fans crowded the streets of Philadelphia in celebration. A few days later, an estimated 670,000 fans showed up to participate in the Eagles Super Bowl victory parade, according to a team at Manchester Metropolitan University [1]. Neilson reported 103.4 million television viewers and 170.7 million social media interactions for this game [2]. Sporting events have a history of garnering intense interest from fans in a structured, repeatable way and the introduction of real-time community-based content websites creates the opportunity to analyze the language and emotion of these fans. Through the process of data collection, corpus analysis, hypothesis generation, and classification, our goal is to identify key differences in language between fans and casual observers.

2. Data Collection

Gratch et al. used Twitter to identify patterns in fan sentiment during the 2014 World Cup games demonstrating the possible uses of social media data to better understand human emotion [3]. We based our research on data from Reddit. Over 2 million user posts were scraped from Reddit using the PRAW API. At the highest level, Reddit content is organized into topic-specific forums called subreddits. Threads can be created in each subreddit to discuss events within the broader topic. User posts were scraped from 298 "In-Game" threads of the /r/nfl/ subreddit. Each thread contains live discussions during an NFL game.

Data points collected include: (1) the game thread of the post, (2) the UTC timestamp of the post, (3) the post's "Reddit score" as voted on by other Reddit users, (4) the author's username (5) the author's associated "team flair", and (5) the comment body of the post. Each user in the /r/nfl/ subreddit can optionally tag their account with NFL team affiliation called "team flair". We found that more than 97%[2] of the posts contain team flair, thus enabling us to identify specific team fans.

3. Corpus analysis

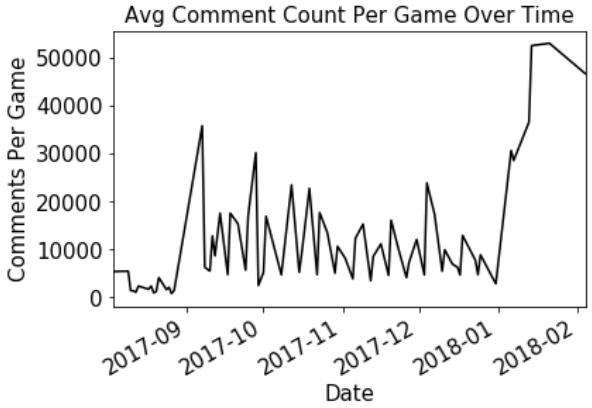
In the corpus, we identified posts made by "casual observers" as those with no author flair or author flair unassociated with either playing team of the current game thread. The 298 game threads

[1] Equal contribution Correspondence to: Chad Harness <chadharness@berkeley.edu>, Rob Mulla <robmulla@berkeley.edu>, Keri Wheatley <keriwheatley@berkeley.edu>

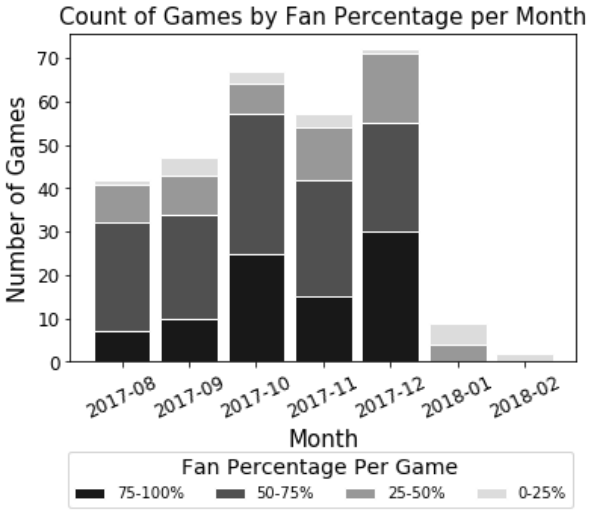
[2] Of a total 2,018,028 comments collected, 52,322 of them did not have an associated team flair.

represent comments from 298 football games across the 2017 NFL season, including preseason, regular season, wildcard, playoff, and championship games. Authors average 5.72 posts per game with a right-skewed distribution. Variations exist between game threads in the number of comments per game and the ratio of fans versus casual observers.

The following chart displays the average number of comments per game thread by date in the 2017 NFL season. This chart shows a lack of activity for preseason games (games played prior to 2017-09) and a large increase in activity for wildcard and championship games (games played after 2018-01).



The chart below shows a categorization of each game by month and percentage of commentators who are "fans". This chart shows a change in fan patterns for games played in JAN-18 and FEB-18 to majority casual observers.



When comparing the two charts, it appears wildcard and championship games garner more attention from the casual observer than preseason and regular season games. When including all games in this corpus, the total ratio of fans to casual observers is 45.4%. Because wildcard and championship game threads contain many more comments with the vast majority made by casual observers, these games can be considered "irregular" in the corpus. If the irregular games are removed from the corpus, then the total ratio of fans to casual observers becomes 52.8%.

4. Hypothesis

Our hypothesis is ***the type of language used during sporting events differs between fans and casual observers.*** In this paper, we explore nuances in language by preprocessing the corpus and training logistic regression and LSTM neural network classification models using the two classes: fan and casual observer. By training models on this classification task, we believe we can use the models' predictions and learned feature weights (or approximations thereof) to surface linguistic differences between these two groups. The idea is straightforward: if we can learn a model that performs well at differentiating between fans and casual observers, we believe that model is extracting linguistic patterns characteristic to each group. We then analyze the

weighted features to determine some major differences in language between the two groups.

5. Preprocessing

These steps were taken to prepare the data for model training:

a. Remove irregular game threads to prevent model bias; wildcard and championship games threads are "irregular" because they contain many more comments than regular season games, a vast majority of those made by casual observers.
b. Convert "bandwagon fans", as denoted by author flair, to "fan" designation (less than 1% of corpus).
c. Convert strings to lowercase.
d. Convert digits to string "DG".
e. Convert URL hyperlinks to string "postedhyperlinkvalue".
f. Remove posts with comment body "[deleted]" or "[removed]".
g. Remove posts by bots; comment body contains "^This ^message ^was ^created ^by ^a ^bot".
h. Remove posts by moderators; comment body contains "**Please review the rules for".
i. Remove stop words using English TfidfVectorizer; as directed by Hughes et al., in order to retain all relevant information for the neural network, stop word removal was only performed for logistic regression [4].

After preprocessing, the final corpus contained 1.5 million comments with 52.8% classified as fan and 47.2% classified as casual observer[3].

6. Logistic Regression Language Evaluation

Based on background knowledge, we chose two variants of regularized logistic regression to serve as our linear classifiers (see performance charts that follow).

Model	Accuracy	# Features
(0) Most common class baseline	0.528	1
(1) SGDClassifier(loss="log", penalty="L1", alpha=0.0001)	Train 0.555 Test 0.558	101
(2) SGDClassifier(loss="log", penalty="L2", alpha=0.000005)	Train 0.612 Test 0.587	115,473

(1) Penalty="L1"				
	precision	**recall**	**f1-score**	**support**
casual observer	0.57	0.21	0.3	71887
fan	0.56	0.86	0.68	82317
avg/total	0.56	0.56	0.5	154204

(2) Penalty="L2"				
	precision	**recall**	**f1-score**	**support**
casual observer	0.58	0.44	0.5	71887
fan	0.59	0.72	0.65	82317
avg/total	0.59	0.59	0.58	154204

While logistic regression with L2 regularization (2) has better performance, the logistic regression with L1 regularization (1) demonstrates a larger separation between word groups in the two classes.

The following charts provide comparisons of top weighted features for each class and model.

[3] The resulting corpus contained 1,542,049 total posts; 814,293 "fan" posts and 727,756 "casual observer" posts

Top Features "Fan" (1) Penalty="L1"		Top Features "Fan" (2) Penalty="L2"	
2.35 we're	0.78 fucking	2.7 we're	1.87 shula
1.3 defense	0.75 3rd	2.3 we've	1.85 fouts
1.17 we've	0.7 fuck	2.06 haley	1.84 downing
0.83 offense	0.65 today	2.06 ifedi	1.8 dola

Top Features "Casual Observer" (1) Penalty="L1"		Top Features "Casual Observer" (2) Penalty="L2"	
-1.24 romo	-.85 mcadoo	-2.46 tuned	-1.97
-1.2 chiefs	-.84	-2.21 rex	skycam
-1.08 nfl	commercial	-2.01	-1.9 sergio
-.92 gruden	-.81 football	mcdonough	-1.89 yahoo
-.87 ravens		-1.98 mcadoo	-1.83 chiefs

For each class, certain patterns emerge. In the "fan" class, top features: (1) refer to the collective "we", (2) focus on game play, and (3) show immense emotion. In the "casual observer" class, top features: (1) refer to subjects outside of game play, (2) focus on commercials, (3) refer to popular or unpopular coaches and commentators.

7. LSTM Neural Network Language Evaluation

Purpose and performance drove our choice of neural network architecture. Our purpose was to detect and describe differences in the language usage of two different groups of people during the live game threads at r/nfl/. Lacking linguistic training, we believed that a classification sub-task would assist our understanding of the language differences evidenced by the two groups. While Convolutional Neural Network (CNN) architectures have been shown to perform well on sentence classification tasks [5], we found a Long Short-Term Memory (LSTM) architecture performed better against our corpus (see tables below). Both models were implemented using Keras and TensorFlow.

Model	Accuracy
(0) Most common class baseline	0.528
(1) Convolutional Neural Network	Train 0.583 Test 0.590
(2) Long Short-Term Memory Network	Train 0.585 Test 0.591

(1) LSTM				
	precision	**recall**	**f1-score**	**support**
casual observer	0.57	0.53	0.54	71887
fan	0.61	0.65	0.63	82317
avg/total	0.59	0.59	0.59	154204

(2) CNN				
	precision	**recall**	**f1-score**	**support**
casual observer	0.57	0.49	0.53	71887
fan	0.60	0.68	0.64	82317
avg/total	0.59	0.59	0.59	154204

Code/Results:https://github.com/chadharness-mids/w266_final_project_merged/blob/master/submissions/NeuralNet_Models_Classification_Analysis.ipynb

While model results were comparable between architectures, we observed continued oscillation in the performance of our CNN beginning at epoch 10 and continuing at least to epoch 17, when we terminated training. Contrasted with the steady convergence of our LSTM parameters, this instability ultimately swayed us towards an LSTM.

Along with our chosen architecture, we pre-trained our corpus using word2vec with skip-grams and negative sampling, for speed. The resulting 100-dimensional embedding vectors (one for each "word" in our vocabulary) provided a 1-2 pp lift in accuracy over vectors initialized as integers and trained by our neural network. More importantly, this drastically reduced the trainable parameters of our model, resulting in tremendous efficiency gains, e.g. epoch times decreased as much as 50%, in some cases.

Word2vec class from Gensim's free Python library:https://github.com/RaRe-Technologies/gensim

Code:https://github.com/chadharness-mids/w266_final_project_merged/blob/master/submissions/train_embeddings.ipynb

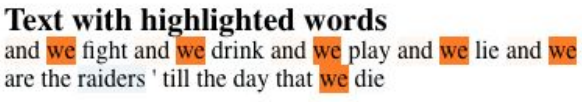
Results:https://github.com/chadharness-mids/w266_final_project_merged/blob/master/submissions/model_nsg_pp2.bin

Because neural nets are much more difficult to interpret than linear classification models, we also used an explanation technique called LIME (Local Interpretable Model-Agnostic Explanations). At a high level, the LIME algorithm learns a linear function that minimizes the difference between the value of that function and the value of the model you are attempting to approximate in the vicinity of specific values in your network's feature-space. This function then learns parameters for those features that allow linear combinations of the parameters and the features to approximate the behavior of the more complicated model within a bounded area of the model space. A more in-depth treatment of the technique is out-of-scope here but we have included references here and in our bibliography that we encourage our readers to explore.

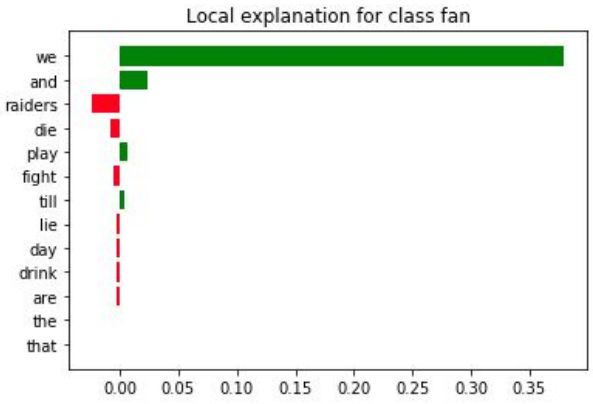
LIME code and documentation may be found at: https://github.com/marcotcr/lime

Like the parameters of a linear model, LIME provides weights for each input feature that you can use to better understand how each feature influences the classifier's predictions.

Consider the following comment from our corpus (most influential words are highlighted in the color representing their class, i.e. blue for "casual observer" and orange for "fan"):



In this example, our LSTM model assigns a probability of 0.84 that this comment belongs to a fan (indeed it does). As shown here, the personal pronoun "we" or some form thereof, is highly predictive of fan authorship of a comment in our corpus. This is consistent with our linear model. The remaining words comprising this comment received relative weights according to the following graph:

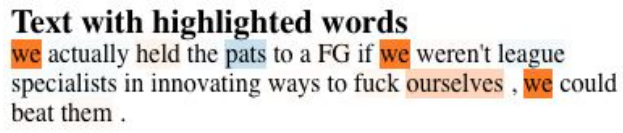


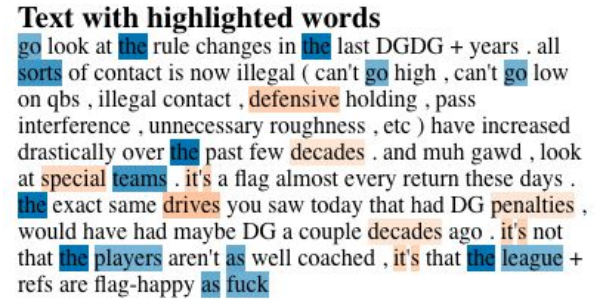
Feature weights expressed in green indicate a relative magnitude that increase the predicted probability that a given comment belongs to a "fan" while those in red indicate the same idea for the "casual observer" classification.

Although LIME did not always provide consistent approximations, we found those inconsistencies useful. Specifically, they provided insight into the behavior of our model and ***how the type of language used during sporting events***

differs between fans and casual observers up to our model's fidelity to real linguistic phenomena.

In the comments below, notice the color of the highlightings of each instance of the word "fuck". This seems to signal a departure from how our linear model treats individual features. In a linear model, the word has a single weight corresponding to a particular target class of that model. The LSTM appears to pick-up on some difference in usage, which becomes clearer upon a closer reading of both comments (fourth word from the right on the second line of the first comment, very last word in the last comment). This manifests in the LSTM model as a completely new direction for the word's influence on the classification probability. Notably, these comments have separate authors, the first of whom is a "fan" and the latter of whom is a "casual observer".





8. Conclusion

This project investigated linguistic differences between fans and casual observers during a live sporting event. By training classification models to distinguish between "fans" and "casual observers", we learned feature weights and made predictions using the comments themselves as our models' only features.. By examining both the predictions and the feature weights (or approximations) we demonstrated tangible linguistic differences between these two groups. The use of a classifier to learn how language differs between two groups is a fruitful way to explore linguistic phenomena, without necessarily being a linguist. However, the somewhat artificial and fluid distinction between classes leads to many instances that should belong to one group actually being labeled as another. This limits a classifier's ability to make accurate predictions. Additional features, such as POS-tagging, to better differentiate between the groups' usages of certain words, similar to the previous example, might provide additional information with which to learn how self-identification as a sports fan influences language usage.

References

[1] Crowd science team confirms 670,000 fans watched Super Bowl parade. (2018, February 8). Retrieved from https://www2.mmu.ac.uk/news-and-events/news/story/7157/

[2] SUPER BOWL LII DRAWS 103.4 MILLION TV VIEWERS, 170.7 MILLION SOCIAL MEDIA INTERACTIONS. (2018, February 5). Retrieved from http://www.nielsen.com/us/en/insights/news/2018/super-bowl-lii-draws-103-4-million-tv-viewers-170-7-million-social-media-interactions.html

[3] Gratch, Jonathan; Lucas, Gale; Malandrakis, Nikolaos; Szablowski, Evan; Fessler, Eli; Nichols, Jeffrey. "GOAALLL!: Using Sentiment in the World Cup to Explore Theories of Emotion." 2015 International Conference on Affective Computing and Intelligent Interaction (ACII), 2015, doi:10.1109/acii.2015.7344681.

[4] Hughes, Mark; Li, Irene; Kotoulas, Spyros; Suzumura, Toyotaro. "Medical Text Classification using convolutional Neural Networks." Studies in Health Technology and Informatics, 2017, doi: 10.3233/978-1-61499-753-5-246.

[5] Kim, Yoon. (2014, September 3). "Convolutional Neural Networks for Sentence

Classification.” Retrieved January 13, 2018 from arXiv:1408.5882v2.

[6] Ribeiro, Marco Tulio; Singh, Sameer; Guestrin, Carlos. (2016, August 9). “‘Why Should I Trust You?’ Explaining the Predictions of Any Classifier.” Retrieved April 17 2018 from arXiv:1602.04938.