PoliTwit

Examining political speech on social media

Tyler Nevell and Kyle Staub

Dataset

- Origin:
  - Crowdflower Data for Everyone dataset: https://www.crowdflower.com/data-for-everyone/ (https://www.crowdflower.com/data-for-everyone/)
  - https://www.kaggle.com/crowdflower/political-social-media-posts/data (https://www.kaggle.com/crowdflower/political-social-media-posts/data)
- Meta-data:
  - _unit_id: a unique id for the message
  - _golden: always FALSE; (presumably whether the message was in Crowdflower's gold standard)
  - _unit_state: always "finalized"
  - _trusted_judgments: the number of trusted human judgments that were entered for this message; an integer between 1 and 3
  - _last_judgment_at: when the final judgment was collected
  - audience: one of national or constituency
  - audience:confidence: a measure of confidence in the audience judgment; a float between 0.5 and 1
  - bias: one of neutral or partisan
  - bias:confidence: a measure of confidence in the bias judgment; a float between 0.5 and 1
  - message: the aim of the message. one of:
    - attack: the message attacks another politician
    - constituency: the message discusses the politician's constituency
    - information: an informational message about news in government or the wider U.S.
    - media: a message about interaction with the media
    - mobilization: a message intended to mobilize supporters
    - other: a catch-all category for messages that don't fit into the other
    - personal: a personal message, usually expressing sympathy, support or condolences, or other personal opinions
    - policy: a message about political policy
    - support: a message of political support

  - message:confidence: a measure of confidence in the message judgment; a float between 0.5 and 1
  - orig__golden: always empty; presumably whether some portion of the message was in the gold standard
  - audience_gold: always empty; presumably whether the audience response was in the gold standard
  - bias_gold: always empty; presumably whether the bias response was in the gold standard
  - bioid: a unique id for the politician
  - embed: HTML code to embed this message
  - id: unique id for the message WITHIN whichever social media site it was pulled from
  - label: a string of the form "From: firstname lastname (position from state)"
  - message_gold: always blank; presumably whether the message response was in the gold standard
  - source: where the message was posted; one of "facebook" or "twitter"
  - text: the text of the message

Goal

Seeking to predict relationships between certain words and certain message types as determined by qualified human assessors.

Data provenance and trustworthiness considerations

This dataset contains 5000 entries of tweets or Facebook posts by members of Congress, that are labelled by "contributors". Contributors from Crowdflower (https://www.crowdflower.com/data-for-everyone-full-library/#!/ (https://www.crowdflower.com/data-for-everyone-full-library/#!/)) go through each post and classify its contents on the basis of the audience, bias, and message components listed in the metadata. It is an interesting dataset because it contains confidence measures, but is also subject to persuasion by the contributors themselves, pointing to a central problem in trusting human labeled datasets. We believe that this dataset will largely conform with the average person's intuitions about the labelling of tweets according to a binary statement of bias, or a categorical attribution of message type.

Data cleaning

For the purposes of our scope, many of the columns listed in the metadata are superfluous. We will be using only four columns: Message, Bias, Label, and Text. Text will be split into a vector of words, on which to classify contribution to a partisan designation. Similarly, we will attempt to classify a message as one of:

- attack
- constituency
- information
- media
- mobilization

- other
- personal
- policy
- support

In [19]:

```
%matplotlib inline
import numpy as np
import scipy as sp
import scipy.stats as stats
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns; sns.set()
from collections import Counter
```

In [20]:

```
import scipy.stats
import sklearn.linear_model
import sklearn.discriminant_analysis
import sklearn.preprocessing
import sklearn.model_selection
import sklearn.neighbors
```

In [21]:

```
twt = pd.read_csv('political_statements_raw.csv', encoding='utf8')
```

In [22]:

```
# Removing columns irrelevant to the current analysis
tdf = twt.drop(['_unit_id', '_golden', '_unit_state', '_trusted_judgments', '_last_judgment_at', 'audience', \
                'audience:confidence', 'bias:confidence', 'message:confidence', 'orig__golden', 'audience_gold',
               'bias_gold', 'bioid', 'embed', 'id', 'message_gold', 'source'], axis=1)

# Creating a new column, wordvec, containing a vector of all words in the text field
tdf['wordvec'] = [i for i in tdf.text.str.split(" ")]

# Converting the bias and message fields to categorical
tdf.bias = pd.Categorical(tdf.bias)
tdf.message = pd.Categorical(tdf.message)

# Taking an initial look at our dataframe
tdf.head()
```

Out[22]:

	bias	message	label	text	wordvec
0	partisan	policy	From: C.A. Dutch Ruppersberger (Representative...	.@DeltaDiva3 One idea is to allow students to ...	[.@DeltaDiva3, One, idea, is, to, allow, stude...
1	neutral	media	From: Jeb Hensarling (Representative from Texas)	Joining the @MarkDavis show on @660KSKY this m...	[Joining, the, @MarkDavis, show, on, @660KSKY,...
2	neutral	information	From: Cory Booker (Senator from New Jersey)	RT @HeraldNews See photos from the ‰Û÷Run With...	[RT, @HeraldNews, See, photos, from, the, ‰Û÷R...
3	neutral	support	From: John McCain (Senator from Arizona)	Headed to #Scottsdale for event honoring Marsh...	[Headed, to, #Scottsdale, for, event, honoring...
4	neutral	information	From: JosÌ© Serrano (Representative from New Y...	Today at 10:30 ribbon cutting ceremony at @sou...	[Today, at, 10:30, ribbon, cutting, ceremony, ...

In [56]:
```
# Answering some initial questions about our dataset:

# 1. What are all of the unique words in wordvec?
allwords = Counter()
tdf.wordvec.apply(allwords.update)
print('50 most common words:\n', allwords.most_common(50))
print('\n\n')
print('50 of the least common words:\n', allwords.most_common()[-250:-200])
```

```
50 most common words:
 [('the', 6979), ('to', 5844), ('and', 3604), ('of', 3360), ('in', 2712), ('a', 2384), ('', 2250), ('for', 204
4), ('on', 1657), ('I', 1407), ('is', 1356), ('with', 1098), ('that', 1096), ('our', 1061), ('at', 979), ('thi
s', 920), ('my', 740), ('be', 707), ('will', 689), ('are', 669), ('The', 660), ('you', 659), ('from', 650), ('b
y', 604), ('have', 569), ('about', 548), ('we', 531), ('their', 508), ('as', 452), ('was', 446), ('who', 428),
('it', 411), ('&amp;', 409), ('more', 408), ('House', 397), ('has', 379), ('an', 362), ('can', 354), ('all', 33
5), ('your', 334), ('not', 317), ('This', 296), ('his', 293), ('today', 291), ('We', 274), ('out', 253), ('the
y', 249), ('-', 247), ('or', 238), ('great', 229)]



50 of the least common words:
 [('"Public', 1), ('Servants', 1), ('Dinner."', 1), ('Parenting', 1), ('Galveston', 1), ('‰ÛÏEvery', 1), ('chil
d,', 1), ('aborted,', 1), ('Lord,', 1), ('birth,', 1), ('born,', 1), ('rejection', 1), ('world.‰Û\x9d-', 1),
('interesting.', 1), ('Index', 1), ('reception,', 1), ('beyond.', 1), ('hearty', 1), ('applauds', 1), ('range
r', 1), ('memorial.', 1), ('terror.', 1), ('Brenda', 1), ('commits', 1), ('Organizations', 1), ('(VSO).', 1),
('14?', 1), ('slavery.', 1), ('420', 1), ('95%,', 1), ('Include:', 1), ('*Provide', 1), ('*Ability', 1), ('appl
es', 1), ('*Independent,', 1), ('*Reduces', 1), ('*Increases', 1), ('last.', 1), ('River.', 1), ('Hemmer', 1),
('10:25', 1), ('Monica', 1), ('filmed', 1), ('Ensuring', 1), ('https://www.youtube.com/watch?v=2jE7kzv5hzc',
1), ('exceeds', 1), ('dependents', 1), ('in-state', 1), ('Post-9/11GI', 1), ('items', 1)]
```

It appears that there are some very clear outliers in this data, as you would suspect. The grammatical connective words ('the', 'to', 'and', etc.) are extremely common. Similarly, there are some outright unique words, largely due to an one of an encoding issue, hyperlink, etc. We intend to monitor this skew in the word frequency data for possible correction in the future.

In [ ]:
```
```