Categorizing Newspaper Articles

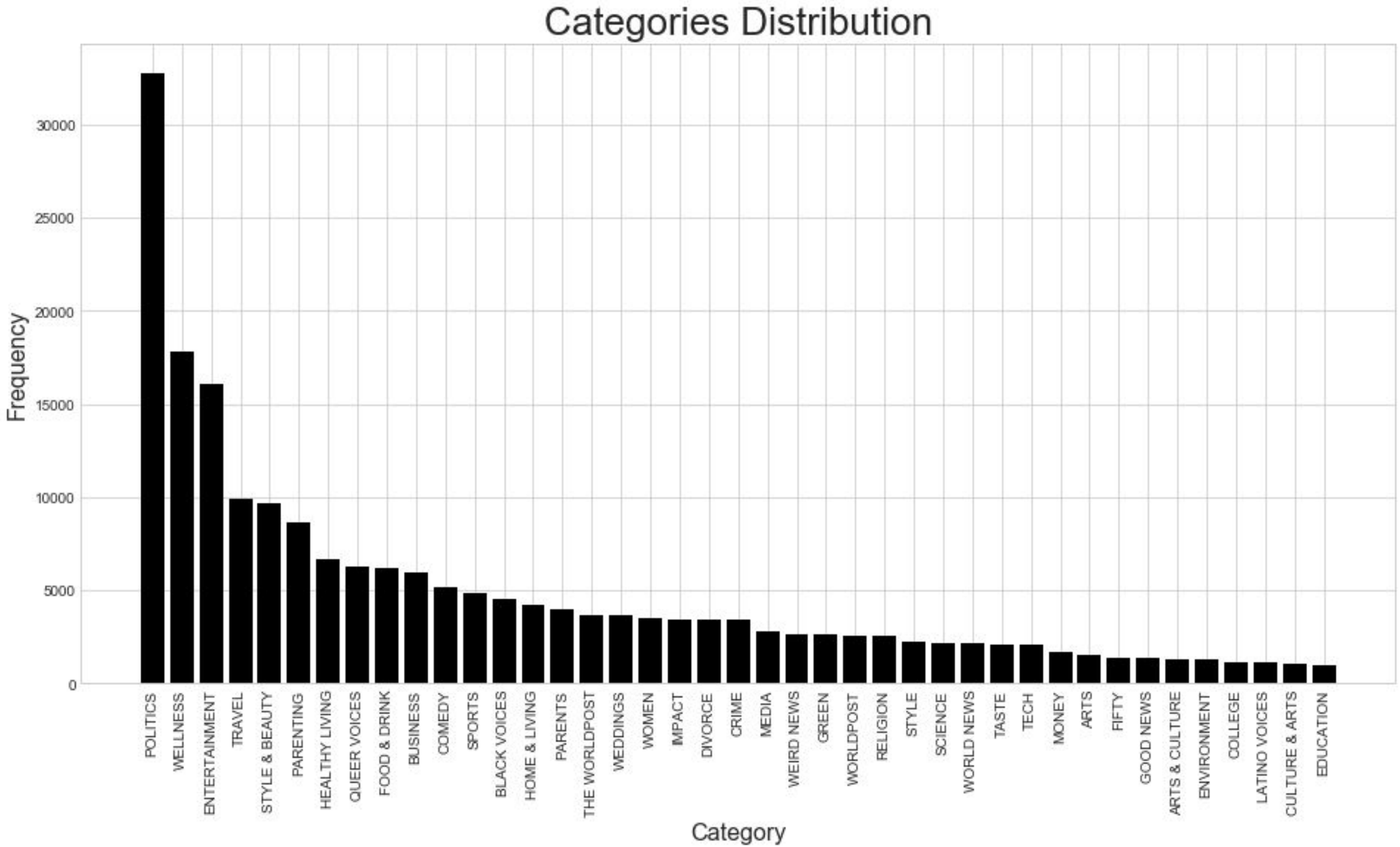
Michael Hodel, Andrin Rehmann

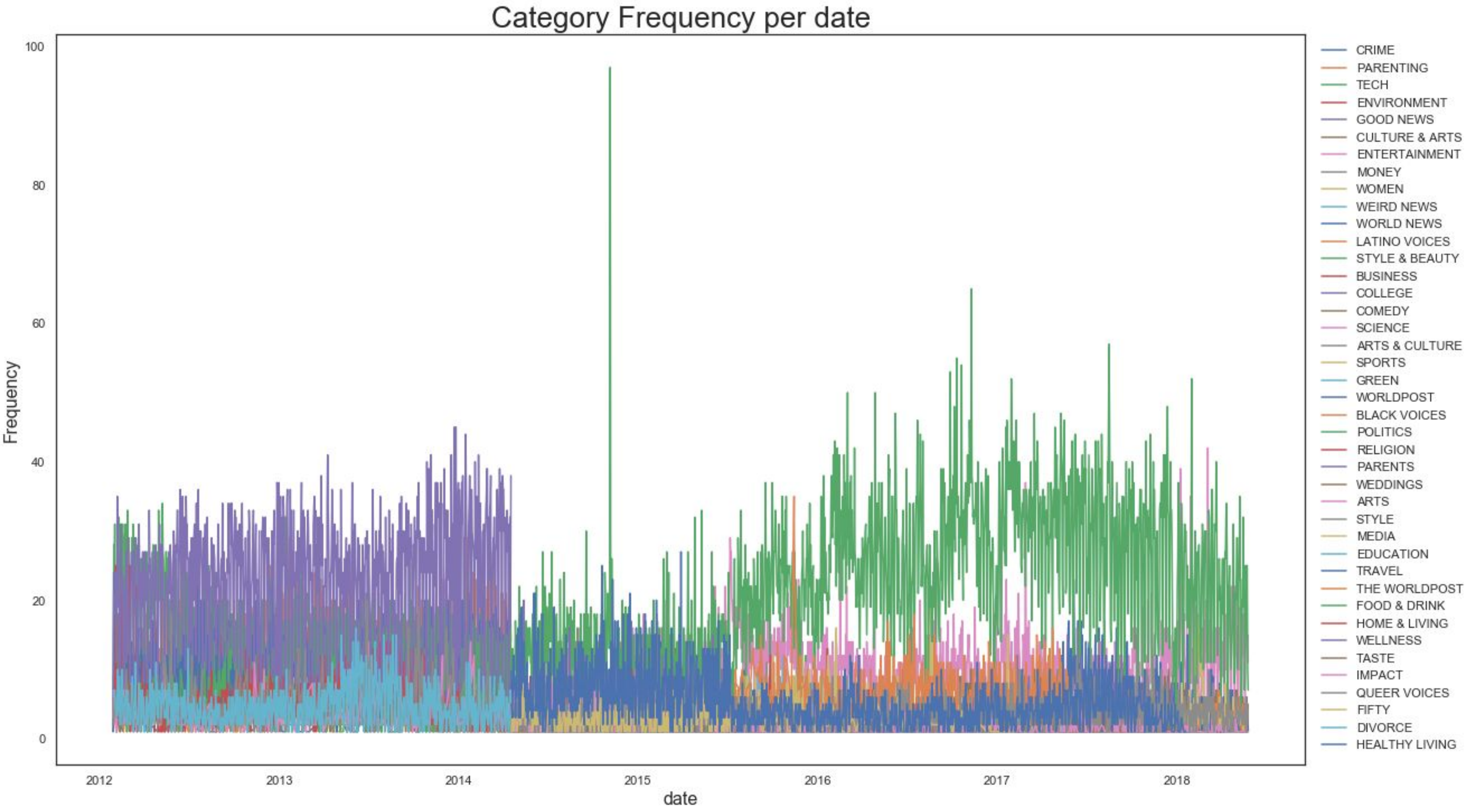
dataset

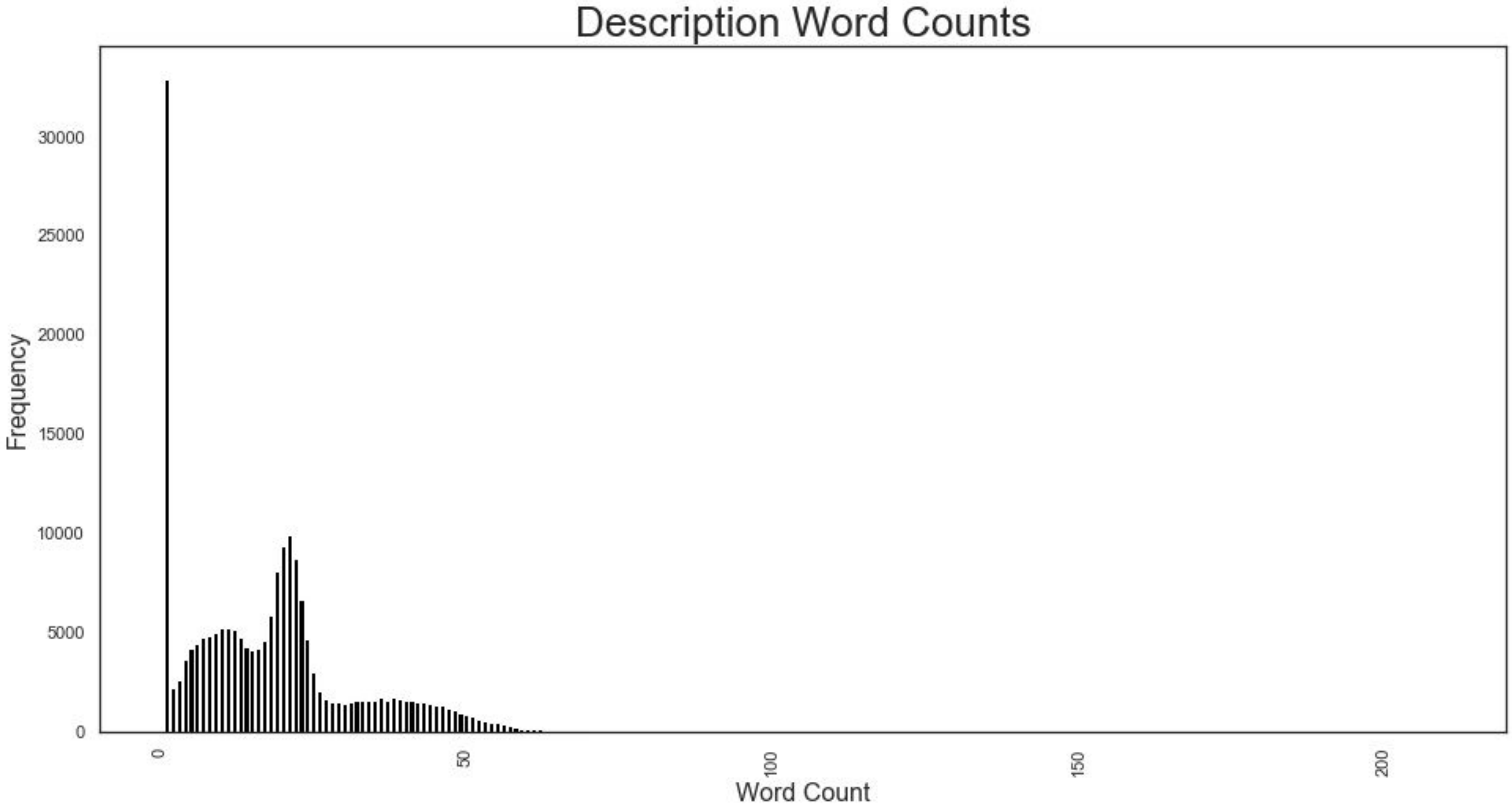
> 200’000 articles from huffpost.com

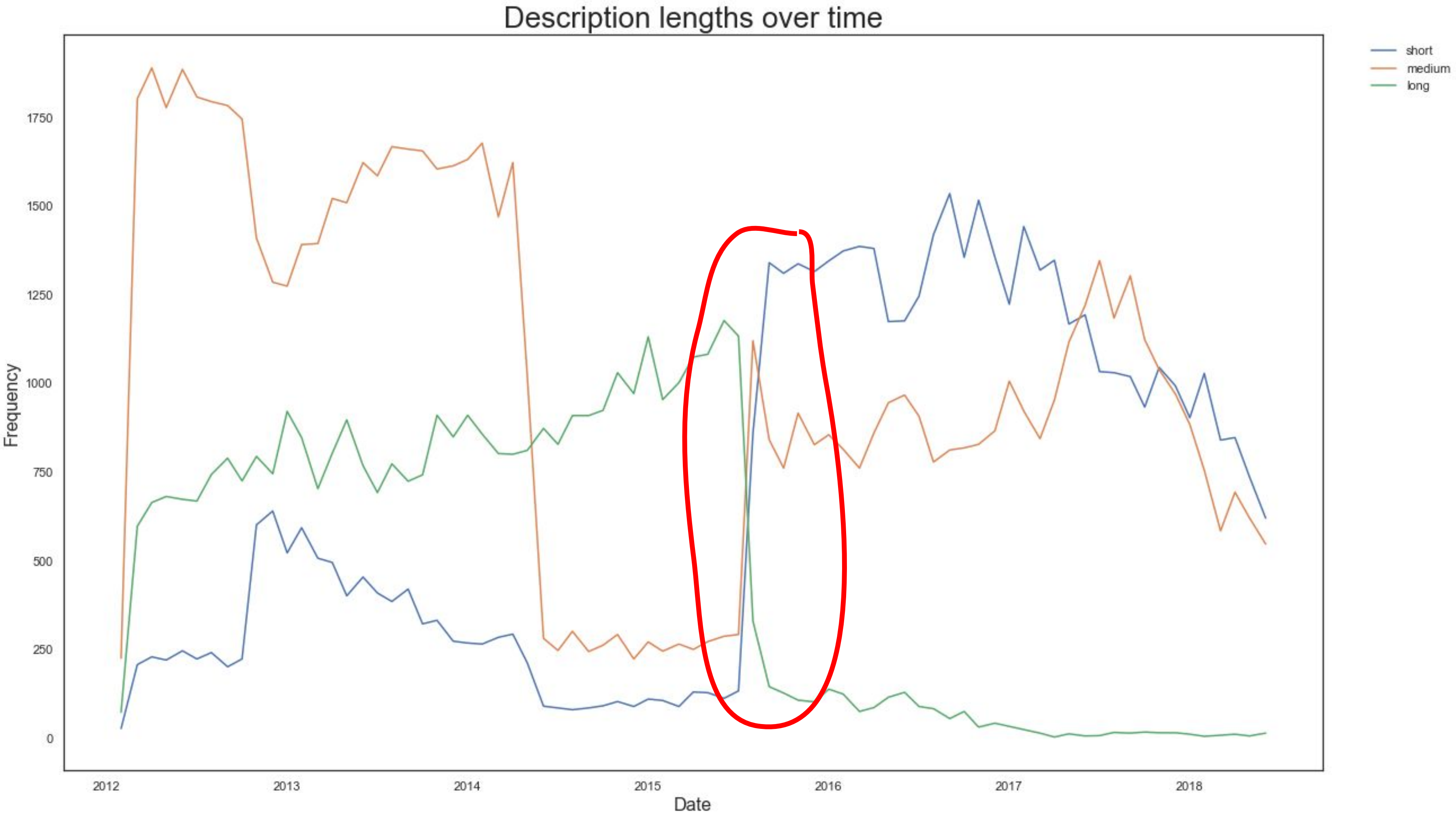
> 40 categories

```
{
   category:       ENTERTAINMENT
   headline:       Oprah Reacts To Trump's Tweet Calling Her 'Very Insecure'
   authors:        Cole Delbyck
   link:           https://www.huffingtonpost.com/entry/oprah-reacts-to-...
   description:    \"I don\u2019t like giving negativity power.\"
   date:           2018-02-22
}
```

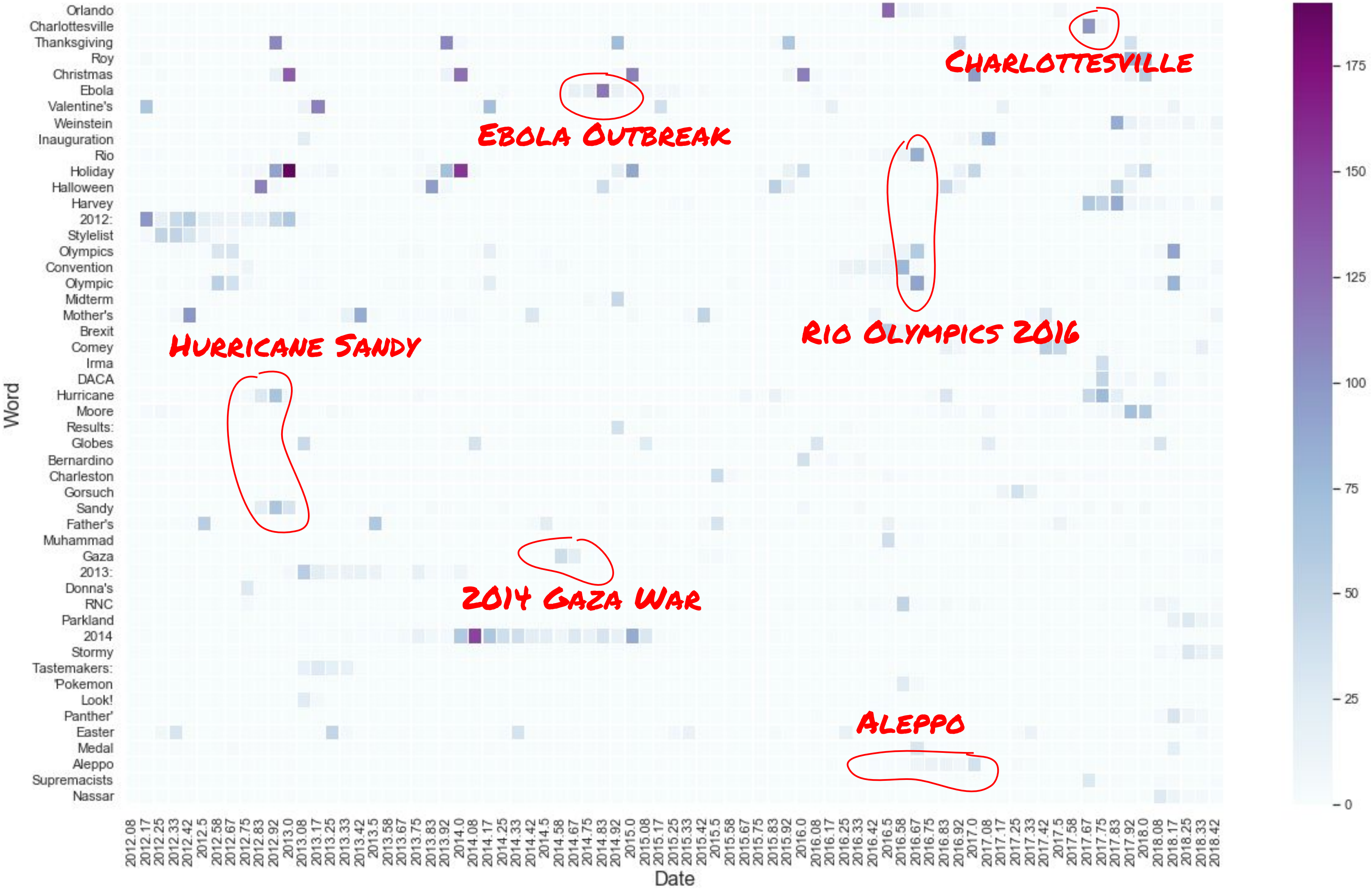

Categories Distribution
Frequency
30000
25000
20000
15000
10000
5000
0
POLITICS
WELLNESS
ENTERTAINMENT
TRAVEL
STYLE & BEAUTY
PARENTING
HEALTHY LIVING
QUEER VOICES
FOOD & DRINK
BUSINESS
COMEDY
SPORTS
BLACK VOICES
HOME & LIVING
PARENTS
THE WORLDPOST
WEDDINGS
WOMEN
IMPACT
DIVORCE
CRIME
MEDIA
WEIRD NEWS
GREEN
WORLDPOST
RELIGION
STYLE
SCIENCE
WORLD NEWS
TASTE
TECH
MONEY
ARTS
FIFTY
GOOD NEWS
ARTS & CULTURE
ENVIRONMENT
COLLEGE
LATINO VOICES
CULTURE & ARTS
EDUCATION
Category


Category Frequency per date
Frequency
100
80
60
40
20
0
2012
2013
2014
2015
2016
2017
2018
date
CRIME
PARENTING
TECH
ENVIRONMENT
GOOD NEWS
CULTURE & ARTS
ENTERTAINMENT
MONEY
WOMEN
WEIRD NEWS
WORLD NEWS
LATINO VOICES
STYLE & BEAUTY
BUSINESS
COLLEGE
COMEDY
SCIENCE
ARTS & CULTURE
SPORTS
GREEN
WORLDPOST
BLACK VOICES
POLITICS
RELIGION
PARENTS
WEDDINGS
ARTS
STYLE
MEDIA
EDUCATION
TRAVEL
THE WORLDPOST
FOOD & DRINK
HOME & LIVING
WELLNESS
TASTE
IMPACT
QUEER VOICES
FIFTY
DIVORCE
HEALTHY LIVING


Description Word Counts
Frequency
30000
25000
20000
15000
10000
5000
0
Word Count
0
50
100
150
200


Description lengths over time
Frequency
Date
0
250
500
750
1000
1250
1500
1750
2012
2013
2014
2015
2016
2017
2018
short
medium
long

$$score_{word} = std(freq_{word}) - 0.03 \times sum(freq_{word})$$



CHARLOTTESVILLE

EBOLA OUTBREAK

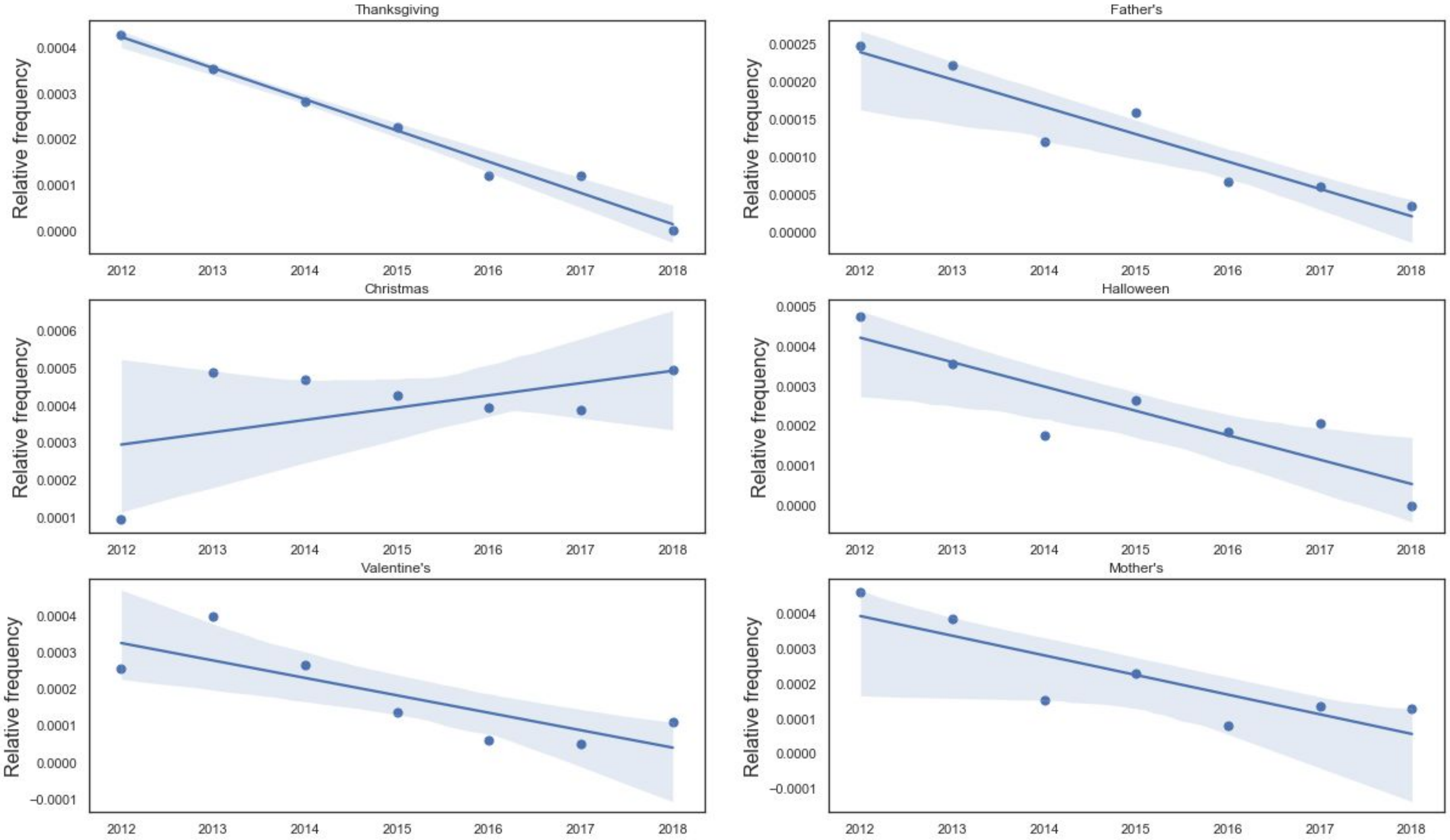
RIO OLYMPICS 2016

HURRICANE SANDY

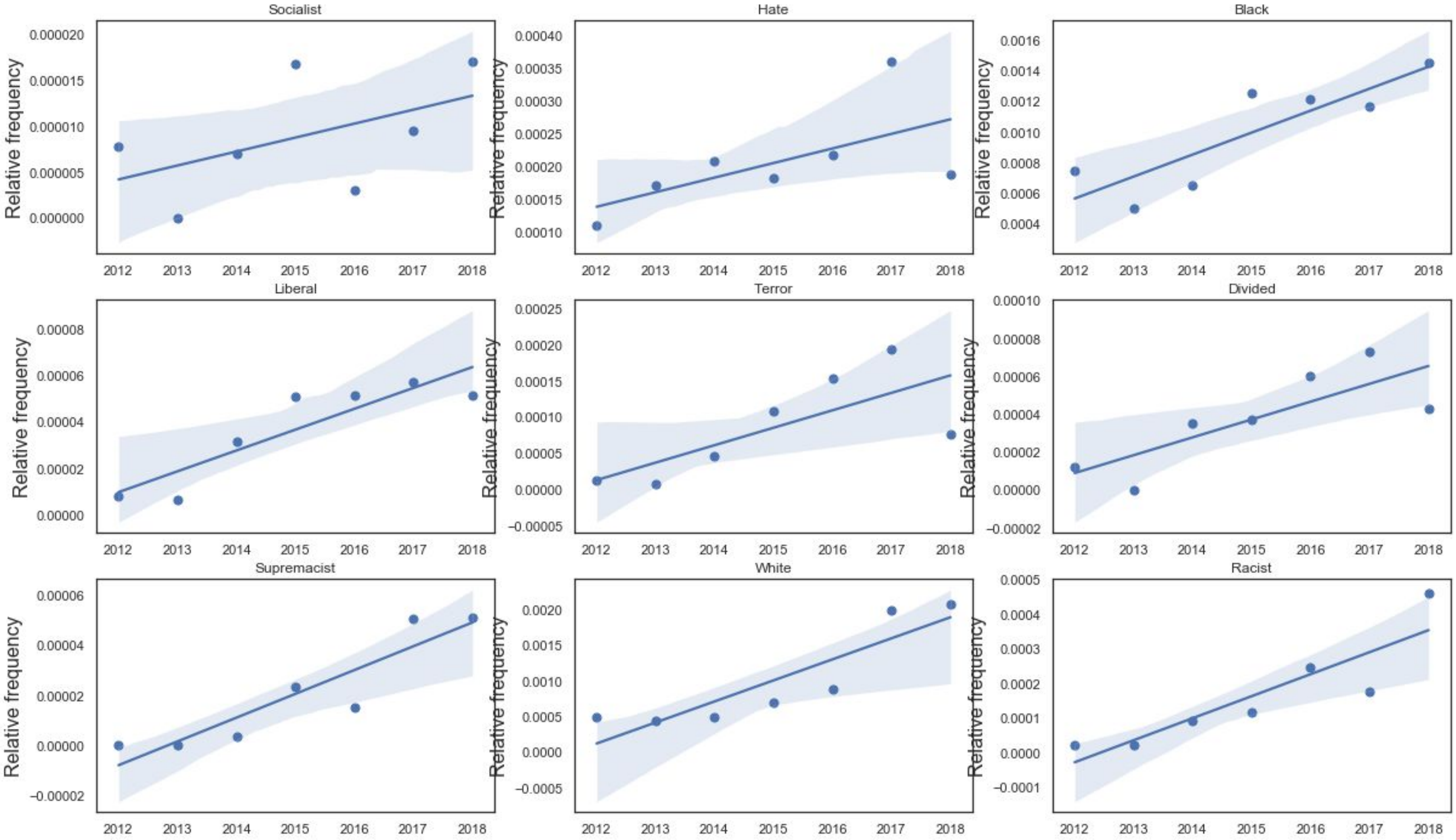
2014 GAZA WAR

ALEPPO

Regressions for normalized word frequencies per year for seasonal holiday words



Regressions for normalized word frequencies per year for strongly polarising political terms



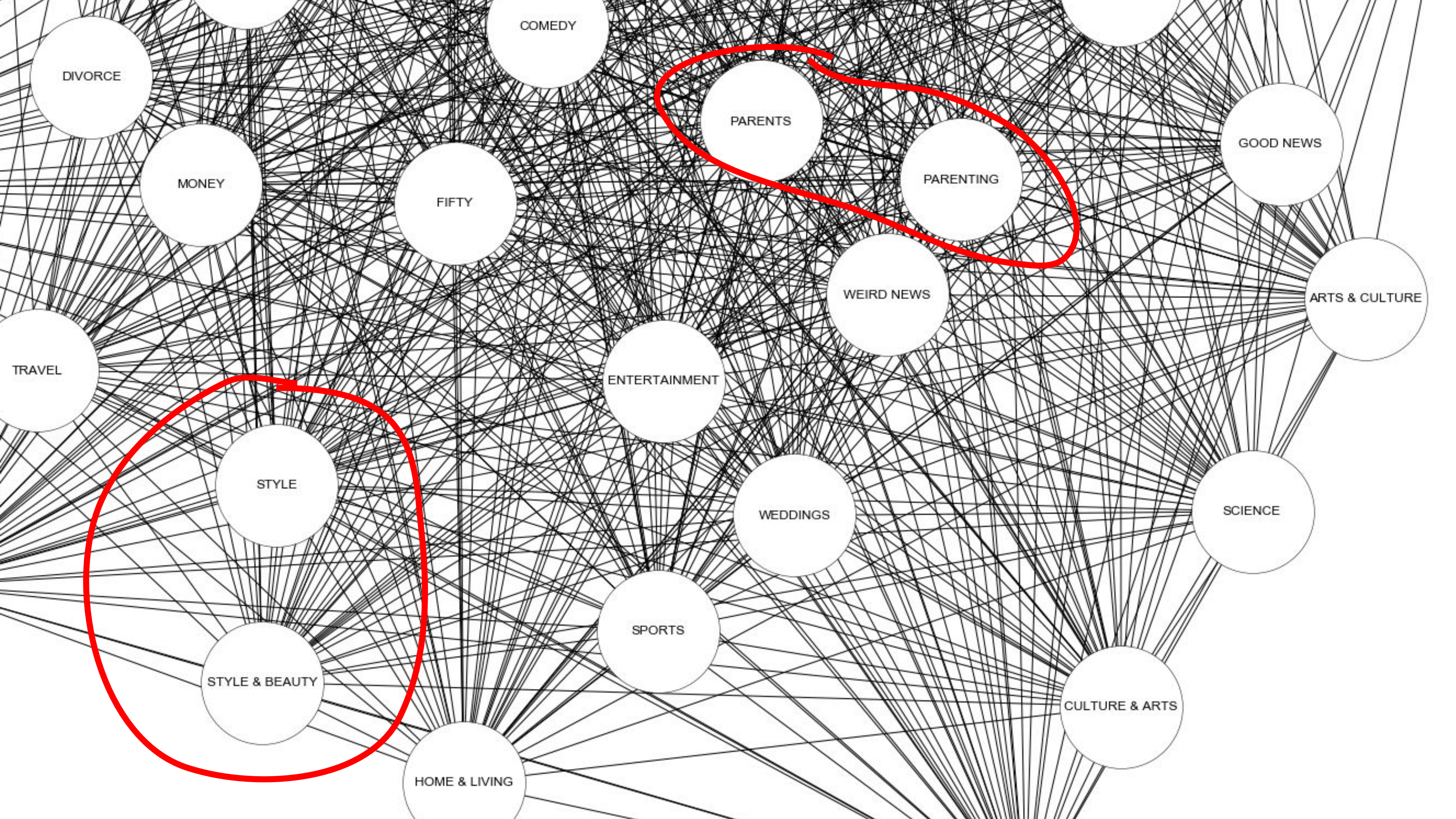
$$G = \{1000\ most\ common\ English\ Nouns\}$$

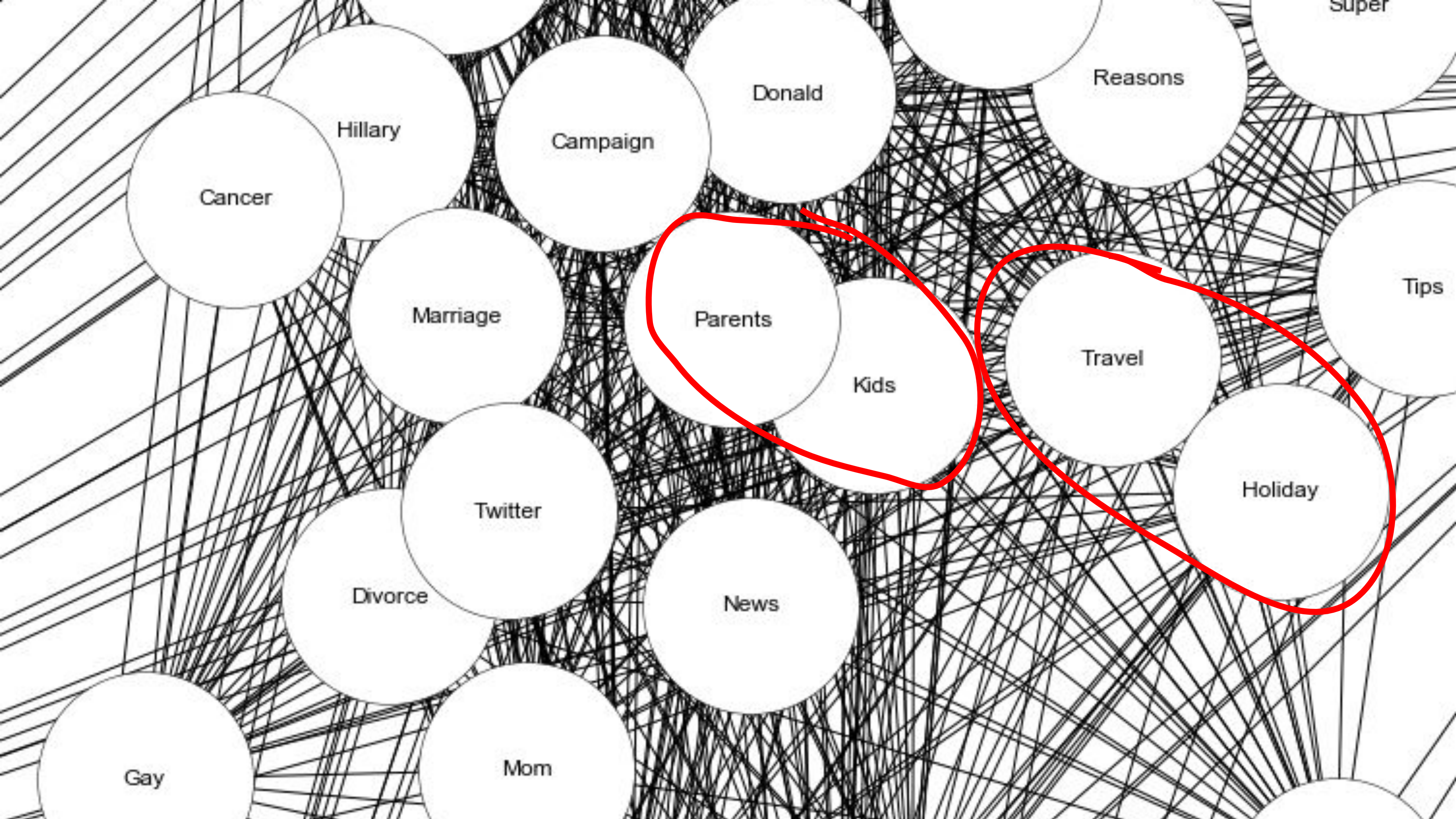
$$N_i = \{Most\ popular\ Nouns\ in\ Category\ i\}\ \text{-}\ G$$

$$sim(a,b) = \frac{|N_a \bigcap N_b| - min_{i,j}\{\left|N_i \bigcap N_j\right|\}}{max_{i,j}\{\left|N_i \bigcap N_j\right|\} - min_{i,j}\{\left|N_i \bigcap N_j\right|\}}$$

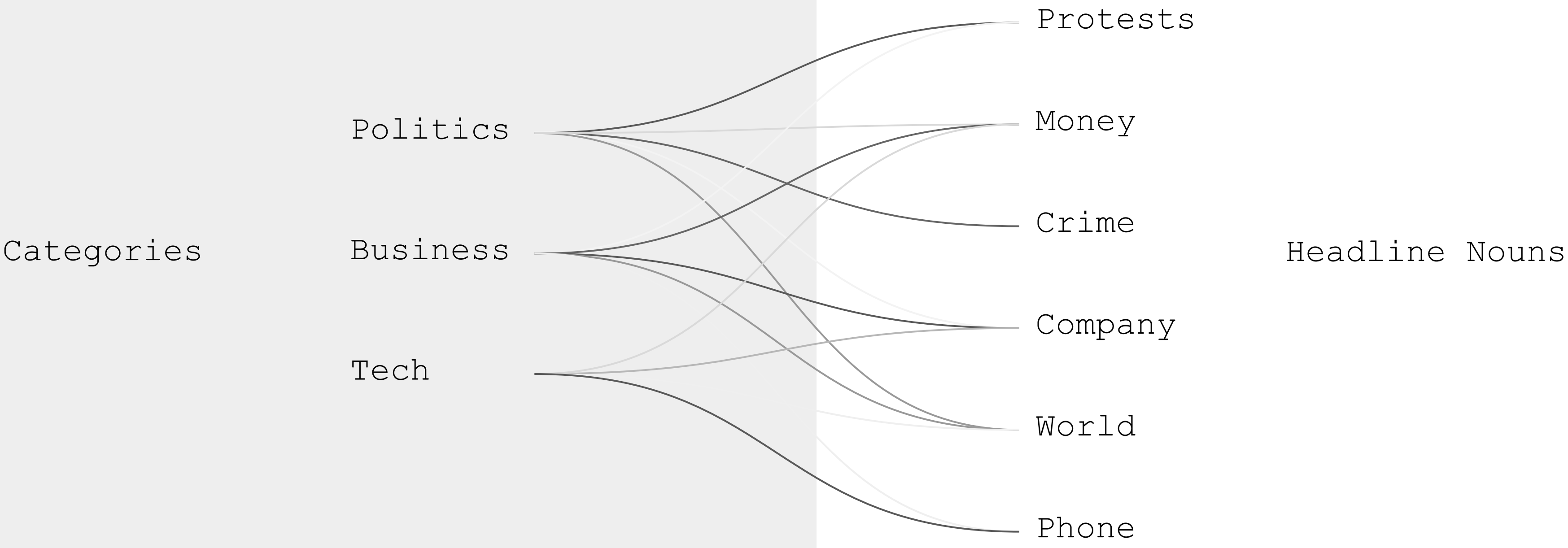
$$sim(Food\ \&\ Drink,\ Taste) = 1$$

$$sim(Religion,\ Arts) = 0$$


COMEDY
DIVORCE
PARENTS
GOOD NEWS
MONEY
FIFTY
PARENTING
WEIRD NEWS
ARTS & CULTURE
TRAVEL
ENTERTAINMENT
STYLE
WEDDINGS
SCIENCE
SPORTS
STYLE & BEAUTY
CULTURE & ARTS
HOME & LIVING


Super
Donald
Reasons
Hillary
Campaign
Cancer
Tips
Parents
Marriage
Travel
Kids
Holiday
Twitter
Divorce
News
Gay
Mom

Naive Classifier



Categories

Politics

Business

Tech

Protests

Money

Crime

Company

World

Phone

Headline Nouns

$$Prediction_{headline} = max_{score}\{Categories\}$$

$$Score_{headline,\ category} = \sum_{nouns} (rank\ of\ noun\ in\ category\ nouns)^{p}$$

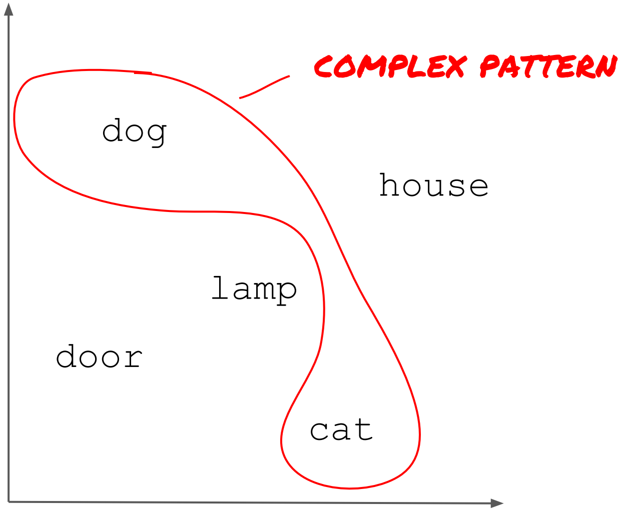
$$Best\ accuracy \approx 33\ \%$$

Word to vec

Given: vocabulary size = 4

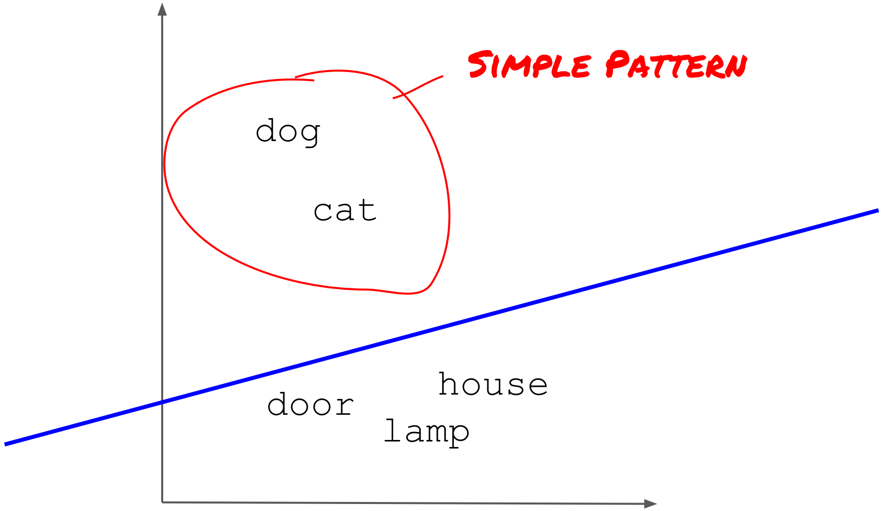
apple ->
(1,0,0,0)
cat ->
(0,1,0,0)
dog ->
(0,0,1,0)
frog ->
(0,0,0,1)

Task: classify animal or object



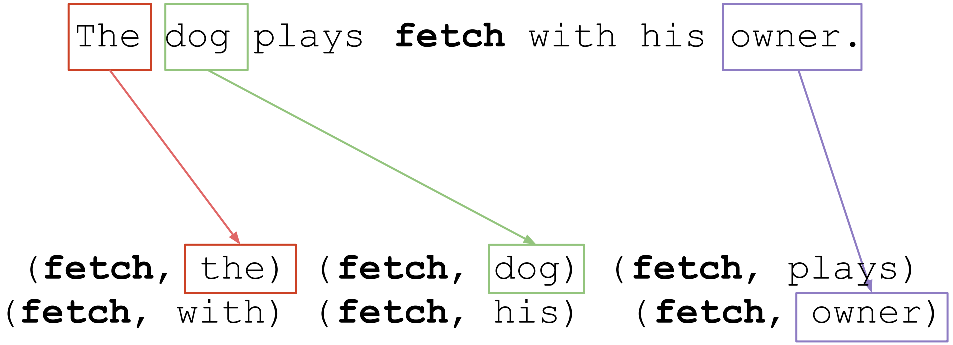
arbitrary vectors

Task: classify animal or object



word embeddings

Embedding method: Skip Gram



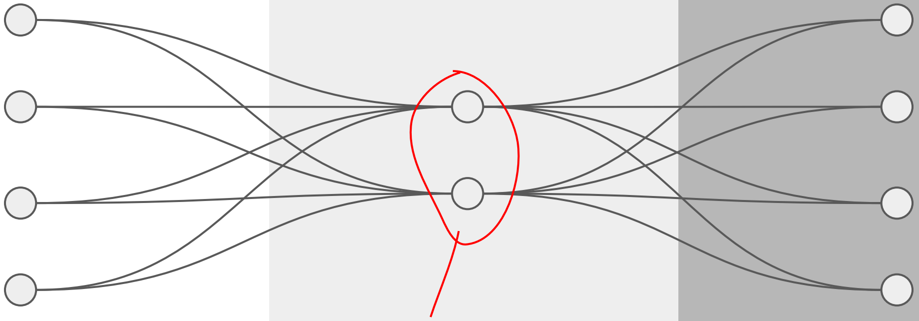
The dog plays **fetch** with his owner.

(**fetch**, the) (**fetch**, dog) (**fetch**, plays)
(**fetch**, with) (**fetch**, his) (**fetch**, owner)

input layer
size: total number of unique words

hidden layer
size: fixed

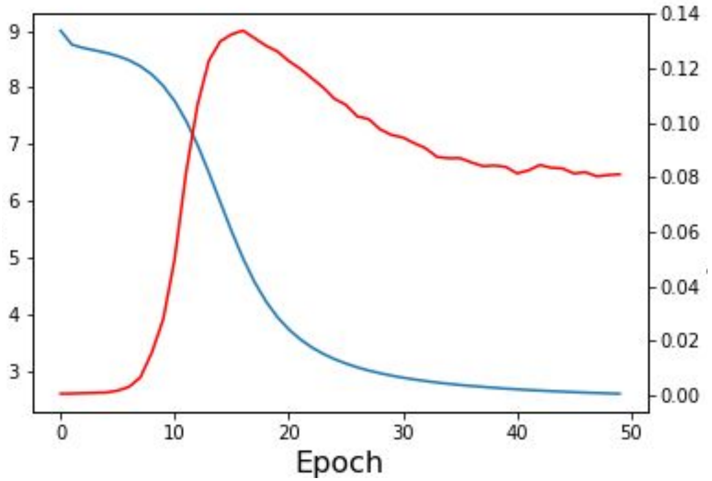
output layer
size: total number of unique words



IS USED AS
WORD VECTOR

Vocabulary size: ~10’000

Loss



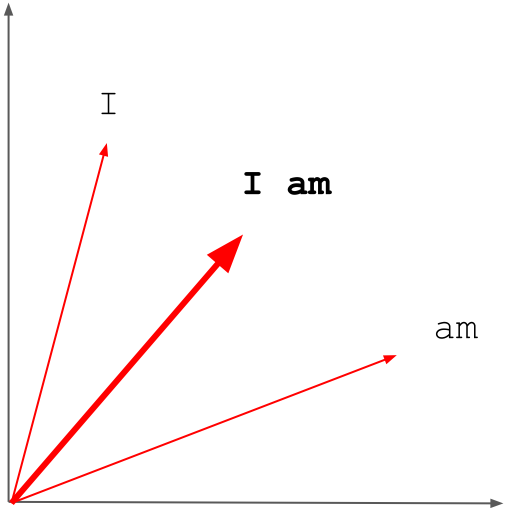
Accuracy

$$tf(term,\ category) = \frac{frequency\ of\ term\ in\ category}{number\ of\ words\ in\ category}$$

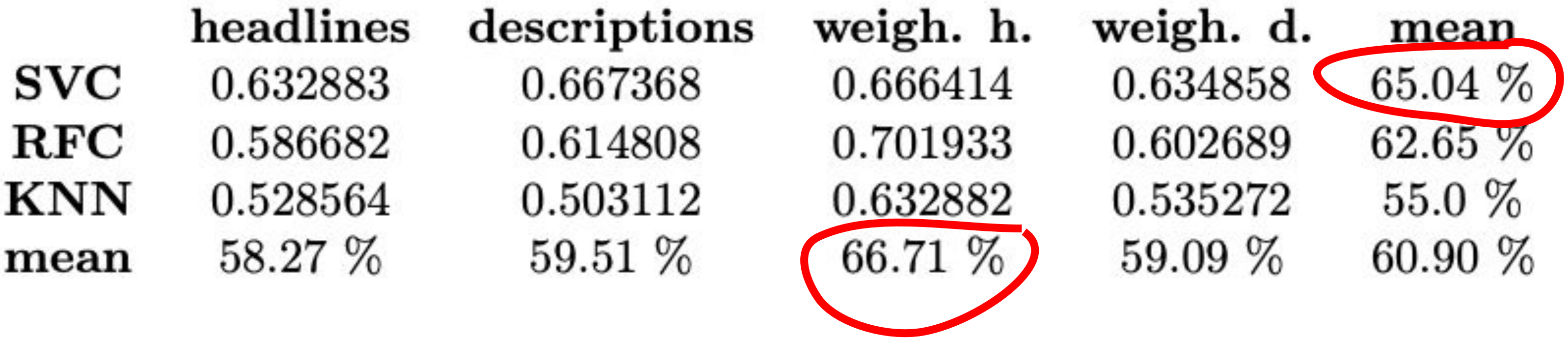
$$idf(term,\ corpus) = log\left(\frac{number\ of\ categories}{\#\ of\ categories\ containing\ term}\right)$$

$$tfidf(term,\ category,\ corpus) = tf(term,\ category) * idf(term,\ corpus)$$

Words to Sentence

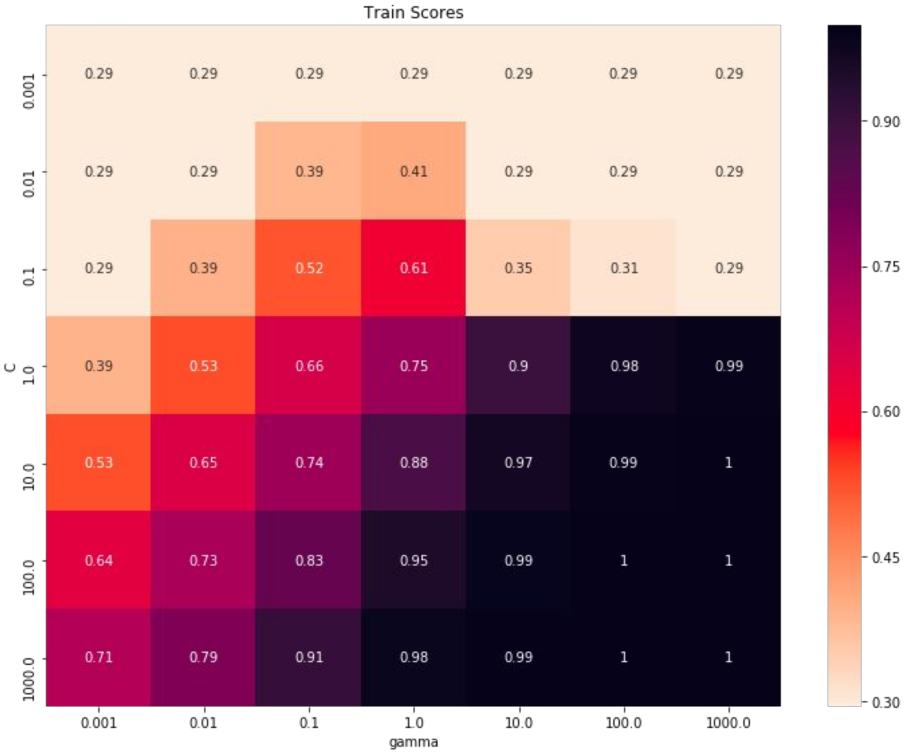


Null classifier accuracy: 15%



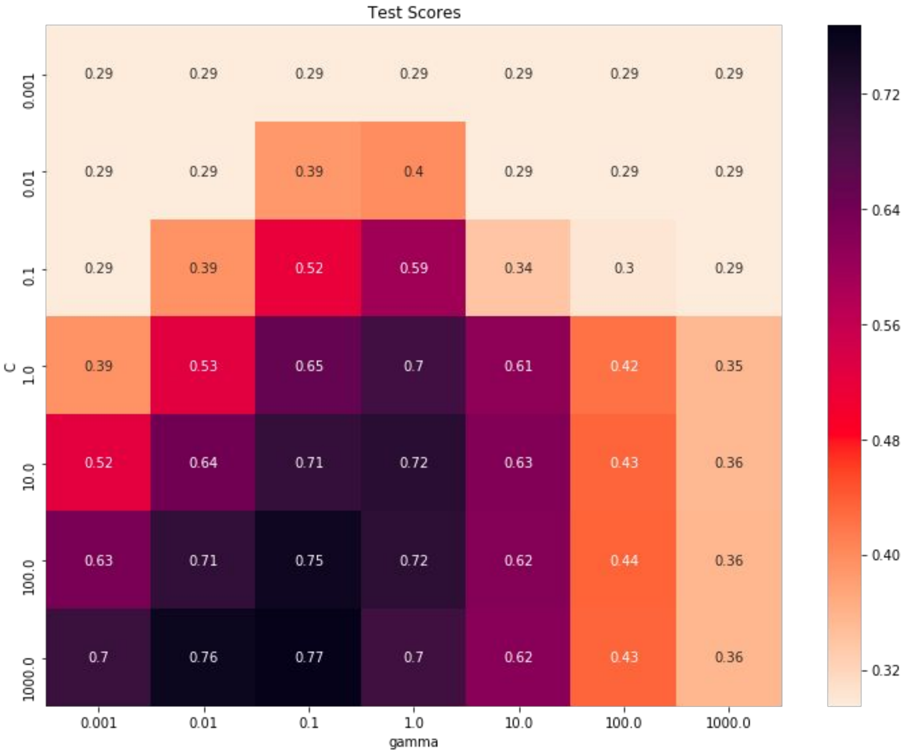
	headlines	descriptions	weigh. h.	weigh. d.	mean
SVC	0.632883	0.667368	0.666414	0.634858	65.04 %
RFC	0.586682	0.614808	0.701933	0.602689	62.65 %
KNN	0.528564	0.503112	0.632882	0.535272	55.0 %
mean	58.27 %	59.51 %	66.71 %	59.09 %	60.90 %

```
Classifier:               SVM
Validation accuracy:      75%
```



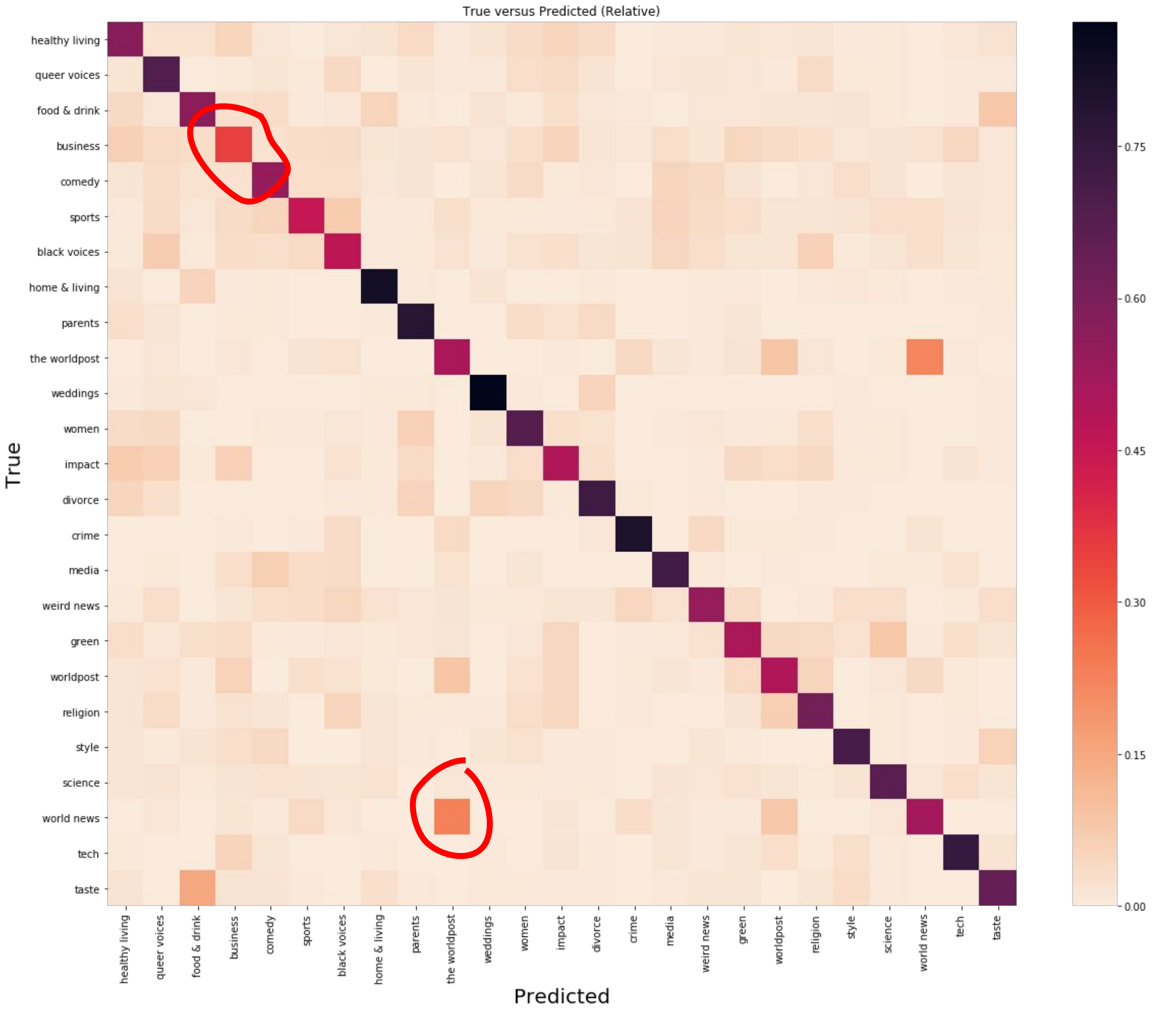
Train Scores

	0.001	0.01	0.1	1.0	10.0	100.0	1000.0
0.001	0.29	0.29	0.29	0.29	0.29	0.29	0.29
0.01	0.29	0.29	0.39	0.41	0.29	0.29	0.29
0.1	0.29	0.39	0.52	0.61	0.35	0.31	0.29
1.0	0.39	0.53	0.66	0.75	0.9	0.98	0.99
10.0	0.53	0.65	0.74	0.88	0.97	0.99	1
100.0	0.64	0.73	0.83	0.95	0.99	1	1
1000.0	0.71	0.79	0.91	0.98	0.99	1	1



Test Scores

	0.001	0.01	0.1	1.0	10.0	100.0	1000.0
0.001	0.29	0.29	0.29	0.29	0.29	0.29	0.29
0.01	0.29	0.29	0.39	0.4	0.29	0.29	0.29
0.1	0.29	0.39	0.52	0.59	0.34	0.3	0.29
1.0	0.39	0.53	0.65	0.7	0.61	0.42	0.35
10.0	0.52	0.64	0.71	0.72	0.63	0.43	0.36
100.0	0.63	0.71	0.75	0.72	0.62	0.44	0.36
1000.0	0.7	0.76	0.77	0.7	0.62	0.43	0.36

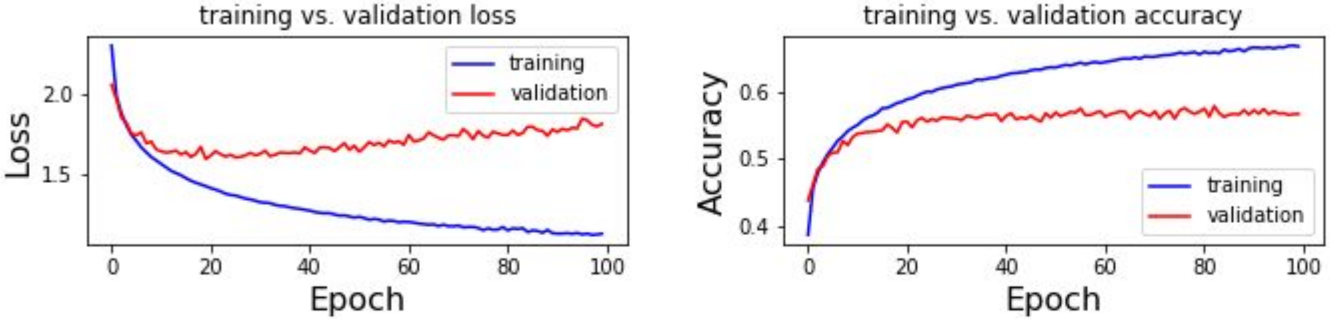

True versus Predicted (Relative)
True
Predicted
healthy living
queer voices
food & drink
business
comedy
sports
black voices
home & living
parents
the worldpost
weddings
women
impact
divorce
crime
media
weird news
green
worldpost
religion
style
science
world news
tech
taste
0.75
0.60
0.45
0.30
0.15
0.00

57 %

Classifier: Neural Network

Validation accuracy: 56%

Weighted descriptions with more layers


training vs. validation loss
Loss
Epoch
training
validation
training vs. validation accuracy
Accuracy
Epoch
training
validation