3 Abstract

The text data which is unstructured one, is transformed to analysable form by various means like calculating count of words, most frequent words, tf-idf, etc. Getting insights from graphs and pictures is quite easy and also showing text information in these forms is foremost step in text analysis. So most of the explorations are ended up in tables, graphs and/or images.

The data is prepared in the form of **tidytext**, where each row is corresponding to single term and each column is its attribute. This form helps us to perform different analysis like sentiment analysis, topic modeling, etc

Most important part of news is to know the level of sentiment and also to compare it in different sections. The effect of same is expressed in graphical way.

Topic modeling is also performed to put news in different buckets(topics) then after they are associated to corresponding sections to validate the model accuracy.

The New York Times

4 Introduction

4.1 Motivation

The New York Times

- **The New York Times** is a global media organization dedicated to helping people understand the world through unrivaled, on-the-ground, expert and deeply reported independent journalism
- Founded as the *New-York Daily Times* on September 18, 1851, by journalist and politician Henry Jarvis Raymond and former banker George Jones
- It is owned by **The New York Times Company**
- It has got worldwide influence and readers. The paper has won 125 **Pulitzer Prizes**, more than any other newspaper.
- The New York Times is the second-most-circulated newspaper in the US (approximately 1,865,318 average circulation)
- **Slogan:** *"All the News That's Fit to Print"*
- **Reputation :** The Times has developed a national and international "reputation for thoroughness" over time.
  - Among journalists, the paper is held in high regard; a 1999 survey of newspaper editors conducted by the Columbia Journalism Review found that the Times was the "best" American paper, ahead of The Washington Post, The Wall Street Journal, and Los Angeles Times.
  - The Times also was ranked #1 in a 2011 "quality" ranking of U.S. newspapers by Daniel de Vise of The Washington Post; the objective ranking took into account the number of recent Pulitzer Prizes won, circulation, and perceived Web site quality.
  - A 2012 report in WNYC called the Times "the most respected newspaper in the world."
- The NYTimes, unparalleled source of news and information, has an option to access articles information by requesting API keys, which are available at The New York Times Developer Network
- URL: https://www.nytimes.com/

4.2 Project Focus

The main focus of project is text analyzing news of NYT which are in text format and those are pertaining to the massive **Parkland School Gun Schooting**, printed in between 22 - February 2018 and 22 - April 2018. Totally 500 such articles are selected for exploration and analysis.

5 Getting data

5.1 Article search

First of all we are searching all the articles from *12 February 2018* to *22 April 2018* having the keyword **parkland** by using the function `as_search()` with required arguments in it. Then we get the metadata like *url, headline, snippet, keywords, pub_date, word_count*, etc of all the articles containing the specified keyword. Then after to access the full content of each article, we need to scrap by using the article url(which is in metadata) with the help of packages like `curl`,`xml2` and other.

Our interest is to focus on text articles only, so the scrapping of content is done only for those articles, which are in textual format and not other multimedia formats like image, video, etc.

```
a <- as_search(q="Parkland", # search query term
               key = NYTIMES_AS_KEY, # the secret key
               begin_date = '20180212', # Start date
               end_date = '20180422', # End date
               all_results = TRUE,
               fl = c('web_url','snippet','lead_paragraph',
                      'abstract','blog','source',
                      'headline','keywords','pub_date',
                      'document_type','news_desk','byline',
                      'type_of_material','word_count')
               # fields to return
               )
# str(a)
```

```
[1] "The size of metadata is : 2.031024 MB"
```

The metadata is containing following variables

```
 [1] "web_url"                 "snippet"
 [3] "source"                  "keywords"
 [5] "pub_date"                "document_type"
 [7] "new_desk"                "type_of_material"
 [9] "word_count"              "score"
[11] "headline.main"           "headline.kicker"
[13] "headline.content_kicker" "headline.print_headline"
[15] "headline.name"           "headline.seo"
[17] "headline.sub"            "byline.original"
[19] "byline.person"           "byline.organization"
```

The dimension of metadata is

```
[1] 514  20
```

The view of metadata for any single article is as below. The left side column contains various attributes and right one is just the attribute value.

```
                        [,1]
web_url                 "https://www.nytimes.com/2018/02/26/us/parkland-florida-shooting.html"
snippet                 "Emergency responders had to check for signs of life when they found the 17-yea
source                  "The New York Times"
keywords                List,4
pub_date                "2018-02-26T19:41:02+0000"
document_type           "article"
new_desk                "National"
type_of_material        "News"
word_count              534
score                   0.9067013
headline.main           "Maddy Wilford, Shot 3 Times in Parkland, Is ‘So Grateful to Be Here’"
headline.kicker         "NA"
headline.content_kicker "NA"
headline.print_headline "Shot 3 Times, and Nearly Ready for Class"
headline.name           NA
headline.seo            NA
headline.sub            NA
byline.original         "By JESS BIDGOOD"
byline.person           List,8
byline.organization     "NA"
```

5.2 Scrapping each text article

Here we are considering 500 text news articles, which contain the word *parkland*, in the specified time period. First of all, the full text content of each article is collected by using the `web_url` available in metadata separately for each article. Then these extracted news are appended serially one after one in a string vector along with their serial numbers.

6 Tidytext format

Unstructured data analysis involves converting unstructured data to some structured format, so that it would become easy to analyse/explore. The *tidytext* format comes handy in such cases.

Tidy text is nothing but each row represents the unit fragmant of text(word/sentence/paragraph/article/section) and each column is its attribute. This form is convenient for analysis with the dplyr, tidytext and ggplot2 packages.

Here we are converting the extracted data to tidy format, so that it will become easy for text analysis with the help of some suiatble packages like' `tidyr`,`tidytext`.

We already have a metadata of all 500 articles and then required attributes from metadata are appended to its corresponding news. Thus we will have a full news along with its all necessary attributes as below.

```
Observations: 500
Variables: 6
$ new_desk     <chr> "National", "National", "Culture", "National", "N...
$ artcle_no    <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15...
$ article_body <chr> "By PATRICIA MAZZEIMARCH 15, 2018 Surveillance vi...
$ pub_date     <date> 2018-03-15, 2018-02-15, 2018-04-18, 2018-02-26, ...
$ score        <dbl> 0.9281394, 0.8754205, 0.8734122, 0.8654591, 0.828...
$ day          <chr> "Thursday", "Thursday", "Wednesday", "Monday", "F...
```

Now our data is ready, each row is article and each column is its attribute. Next we are tokenizing it in unigram and bigram then performing some graphical explorations and analysis

---

6.1 Single word tokenization

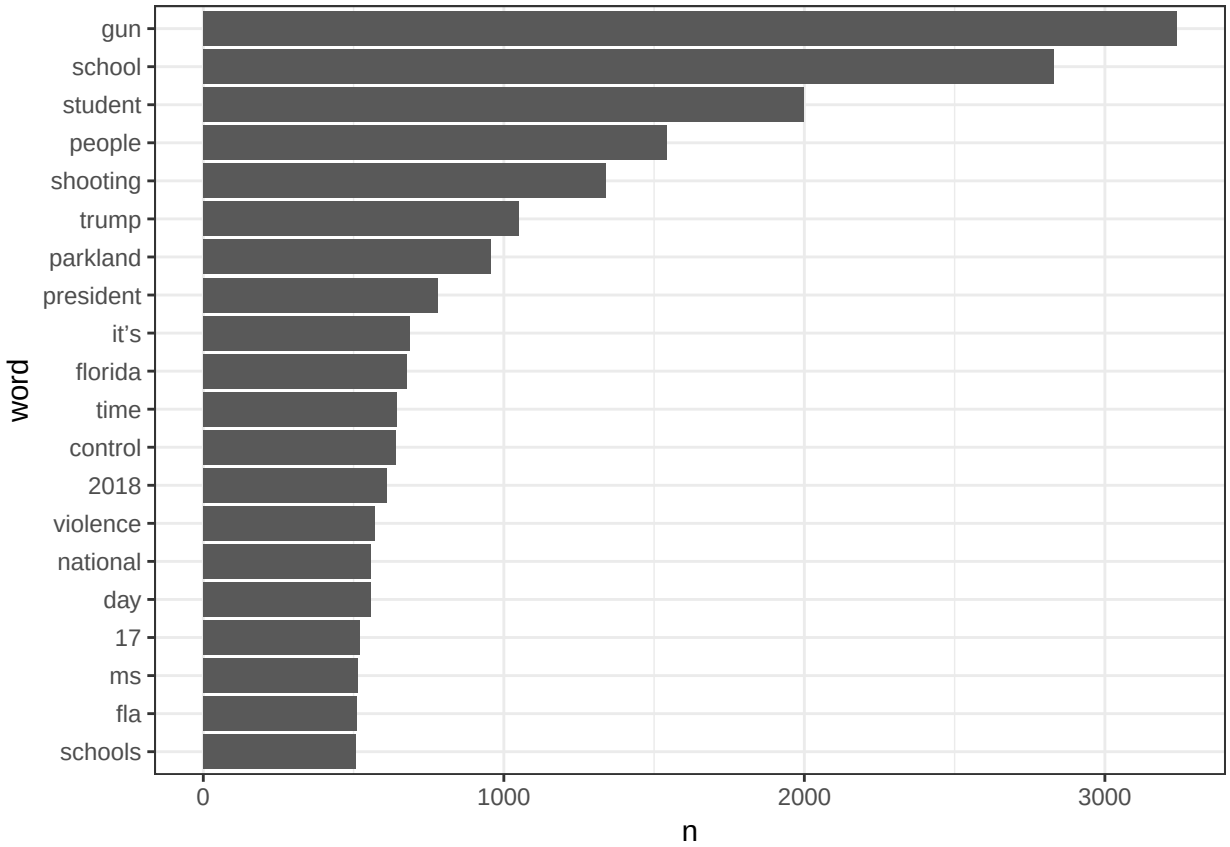
The whole text body of an article is split into single words. The other attributes like its section, corresponding publication date, article serial number and so on are retained.

new_desk	artcle_no	pub_date	score	day	word
National	1	2018-03-15	0.9281394	Thursday	by
National	1	2018-03-15	0.9281394	Thursday	patricia
National	1	2018-03-15	0.9281394	Thursday	mazzeimarch
National	1	2018-03-15	0.9281394	Thursday	15
National	1	2018-03-15	0.9281394	Thursday	2018
National	1	2018-03-15	0.9281394	Thursday	surveillance
National	1	2018-03-15	0.9281394	Thursday	video
National	1	2018-03-15	0.9281394	Thursday	released
National	1	2018-03-15	0.9281394	Thursday	thursday
National	1	2018-03-15	0.9281394	Thursday	showed

- *new_desk:* section of article containing this word(unigram)
- *article_no:* serial number of article from 1 to 500
- *pub_date:* date published the article containing this word
- *score:* the score of article in its metadata
- *day:* day of publication
- *word:* the unigram

6.2 Frequent words in all articles

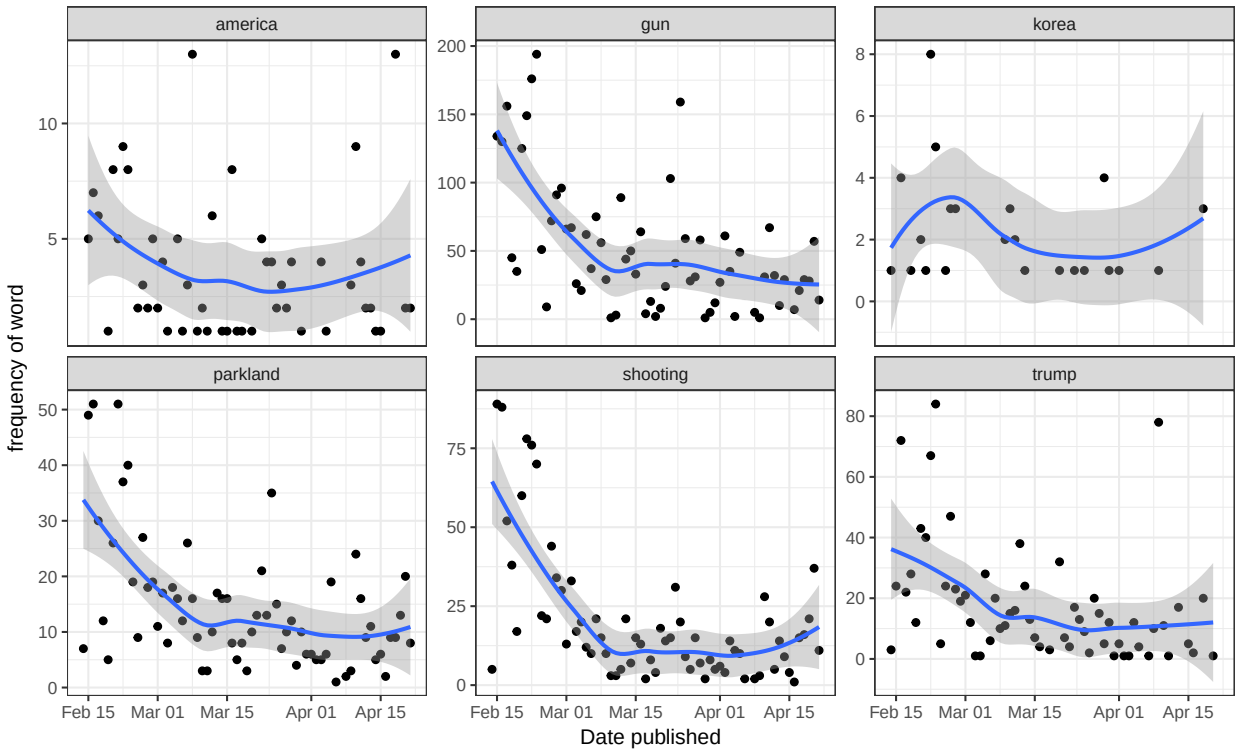
Let us see what are different words appeared most frequently in all 500 articles after removing all english stopwords and some custom stopwords like *new,york,times,subscribe,headline,article,edition.*



No surprise, as we scrapped data pertaining to **parkland shooting**, the frequent words whch appeared are also related to it.

6.3 How word frequency changed day by day?

The appearance pattern of any word which may be place/person/event in news articles is related to some eventual happening about that particular word. Here considering some words like *"gun","shooting","parkland","students","trump","korea"* then plottong the frequency versus the date thay appeared.



As we see here, the words like *gun,parkland,shooting,students* appeared more frequently on the day of 14 February because the **parkland school shooting** happened on this day, and then the frequency decreased gradually.

We can see different frequency pattern for the word *korea*, which is irrelevent to this shooting.

7

6.4 Frequent words in different sections

There are many news sections in New York Times like *World*,*U.S.*,*Politics*,*National* and so forth. Below is comparison of which word appeared frequently in which section(s) and also what are different common words in pair of sections.

- Y-axis is the proportion of words in Business section and X-axix is proportion of words in *National*, *NYTNow*, *Washington* and *Opinion* respectively clockwise from upper-left facet.
- Words with clear view appeared frequently
- Word near to diagonal line appeared almost equally in both news sections (e.g the word *news* appeared equally in *Business* as well as *washington* sections)
- Word appearing off-diagonal or near to one of axes is frequent word in that section only. (e.g *israel* occured more in *National* but rarely in other news sections)
- The word **gun** is appeared in all *Business*,*National*,*Opinion* and *Washington* more or less same times
- Most of the words in *Business* and *Natioanl* are same (may be with more or less frequency), as the area is almost fully occupied. In contrast the gap at lower frequency in *Business* versus *NYTNow* indicates, there are many different word sets in these two sections.

7 Sentiment Analysis

Sentiment Analysis (sometimes known as opinion mining or emotion AI) refers to the use of natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify and study affective states and subjective information.

What sentiment we expect from a news which is totally related to terrific shooting? Letus see what is there in data and also compare sentiment for different sections as well.

The **bing** lexicon categorizes words in a binary response that is positive and negative categories. Means it has collection of English words and also type of their sentiment (positive/negative). `tidytext` package provides a function called `get_sentiments`, which is used to get these sentiments.

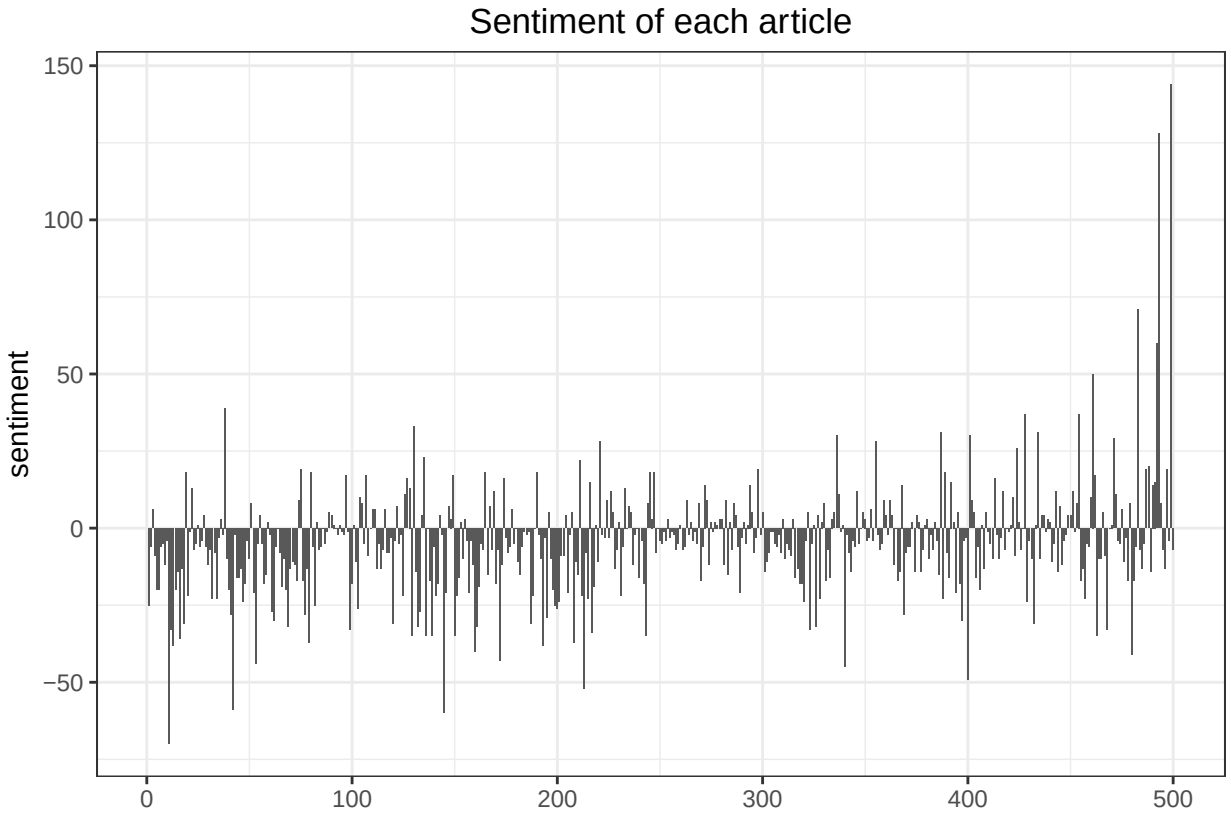
Note that there are other lexicons also available like -

- **AFINN** is an affective lexicon by Finn Årup Nielsen. It is a list of English words rated for valence with an integer between minus five (negative) and plus five (positive). The words have been manually labeled by Finn Årup Nielsen in 2009-2011. For more details please refer: **AFINN**
- The **NRC** Emotion Lexicon is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). The annotations were manually done by crowdsourcing. For more details please refer: **nrc**

Below the greph shows the difference in total negative sentimented terms from positive sentimented terms for each article, thus giving overall level of its sentiment.

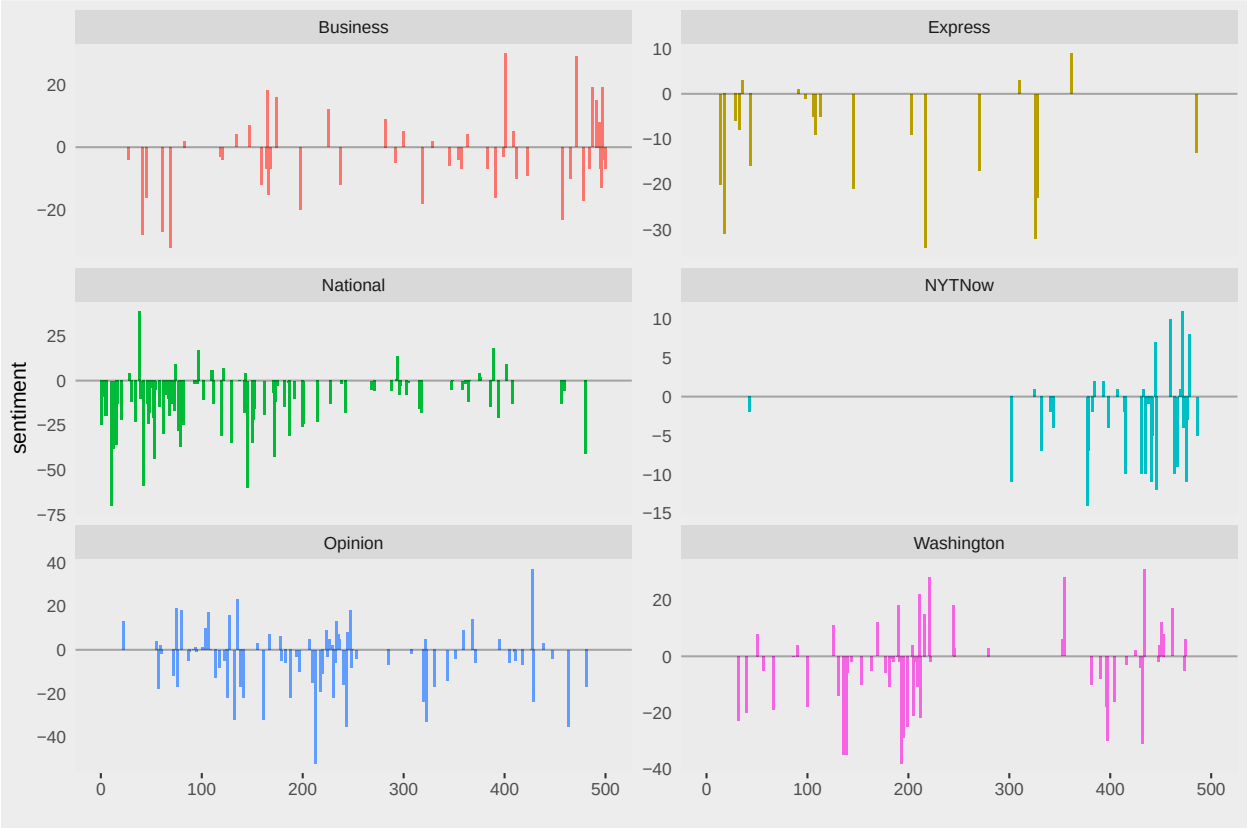
7.1 Sentiment of each article

We can divide the sentiments in above graph among different sections, so that we can compare these news sections on their sentiment level.



This is what sentiment of all articles in 2 months having the specified word *parkland*, clearly indicating the dominance of negative sentiment level. There is some high positive sentimented articles which appeared recently.

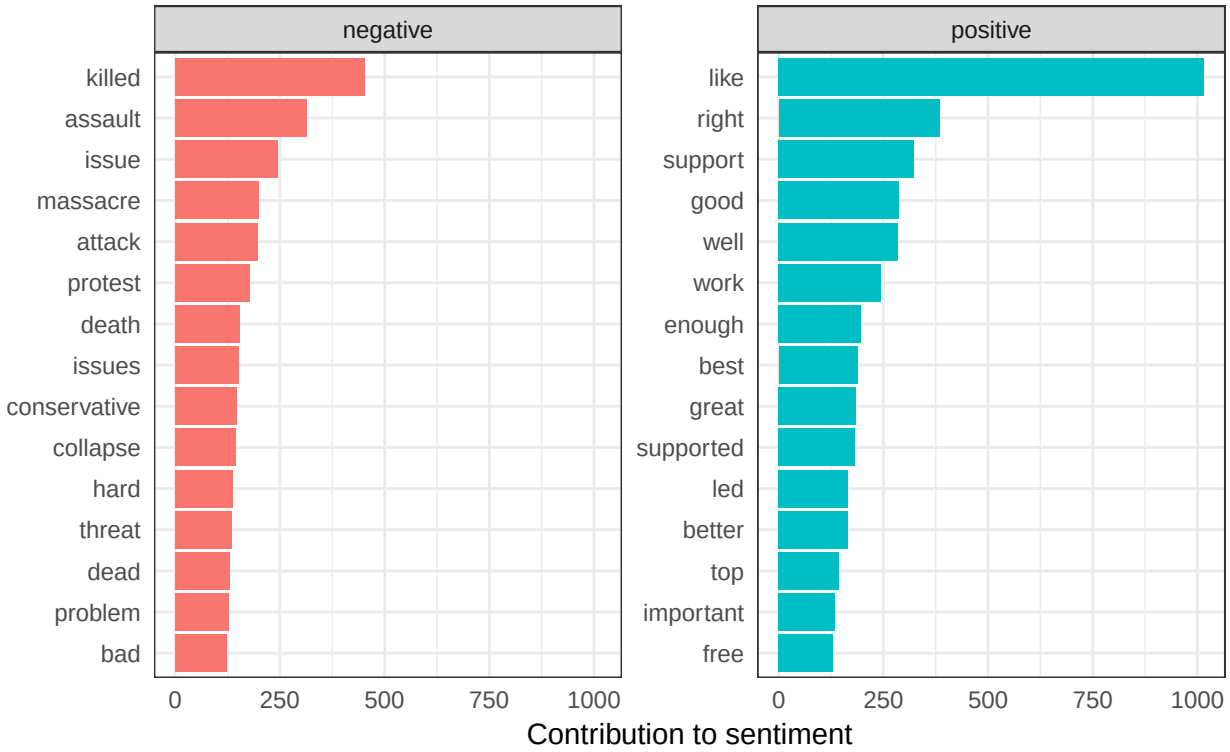
7.2 Sentiment for selective news sections



As we see most of the articles are under *Business*,*National*,*Opinion* and *Washington* sections. The important observation from all facets is that the number as well as the level (height) of negative seniment is more than that of positive in all of the sections.

7.3 Positive and negative words

Letus see what are frequently appeared positive and negative words which are contribted to the sentiments of articles depicted in previous plots.



The word *like* is somewhat like outlier, which changes its meaning situationally. Because it is preposition, conjuction, noun, adverb and adjective as well. But in our analysis it is reason for high skewness in positive sentiment.

One drawback of these unigrams is that they consider the single word only, no matter what was its previous word. Because some negative words like *no, never, not, without, don't* totally reverse the meaning of positive word to negativ and vise-versa. Such bigrams with negative first word are seen in bigram analysis (9.2).

7.4 Wordcloud

All frequent positive and negative words are put graphically together in wordcloud.

---

8 tf-idf

[Term frequency and inverse document frequency]

tf–idf or TFIDF is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling.

tf measures the frequency or proportion of terms in each document. If particular term is appeared many times in all documents, then that term is not at all important. So **idf** reduces the weight of such words which are common in most of documents. so to have a measure which quantifies the relative measure of each term(word) with respect to each document, we need to consider the product of both, i.e. *tf-idf*

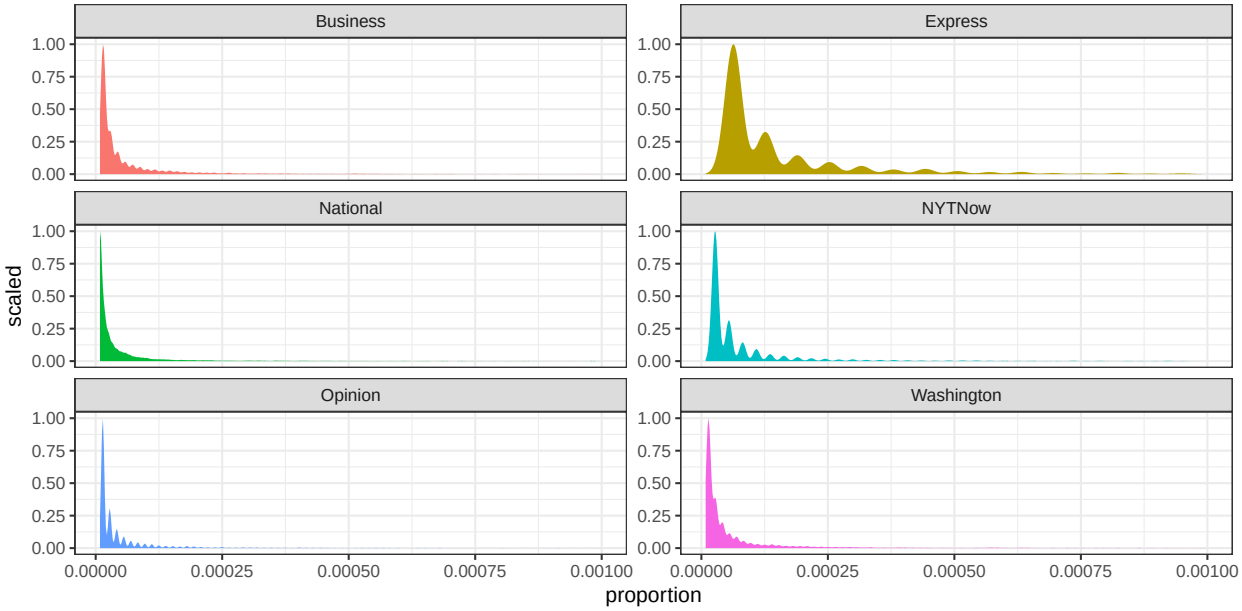
$tf(t,d) = \frac{\text{number of times term t appeared in document d}}{\text{total number of terms in document d}}$

$idf(t) = log\left[\frac{N}{n}\right] = log\left[\frac{\text{Total number of documents}}{\text{number of documents containing the term t}}\right]$

tf-idf(t,d) $= tf(t,d) \times idf(t)$

new_desk	word	n	total
National	the	6441	118180
Washington	the	4206	71732
Business	the	3760	68378
Opinion	the	3729	72132
National	a	3301	118180
National	to	3237	118180

8.1 Distribution of proportion of words

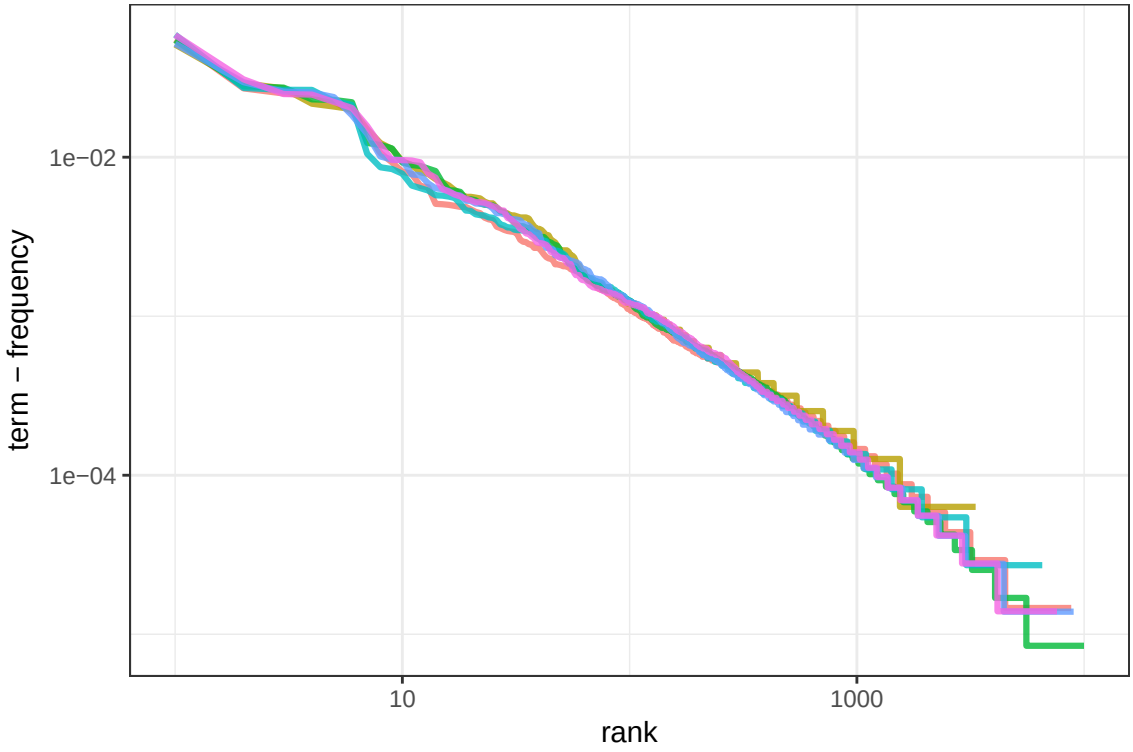


This is distribution of proportion of words in each section separately. We see that words with less frequency have high density in almost all of the sections. There is discrete pattern of proportion in *Express and NYTNow*

8.2 Zipf's law

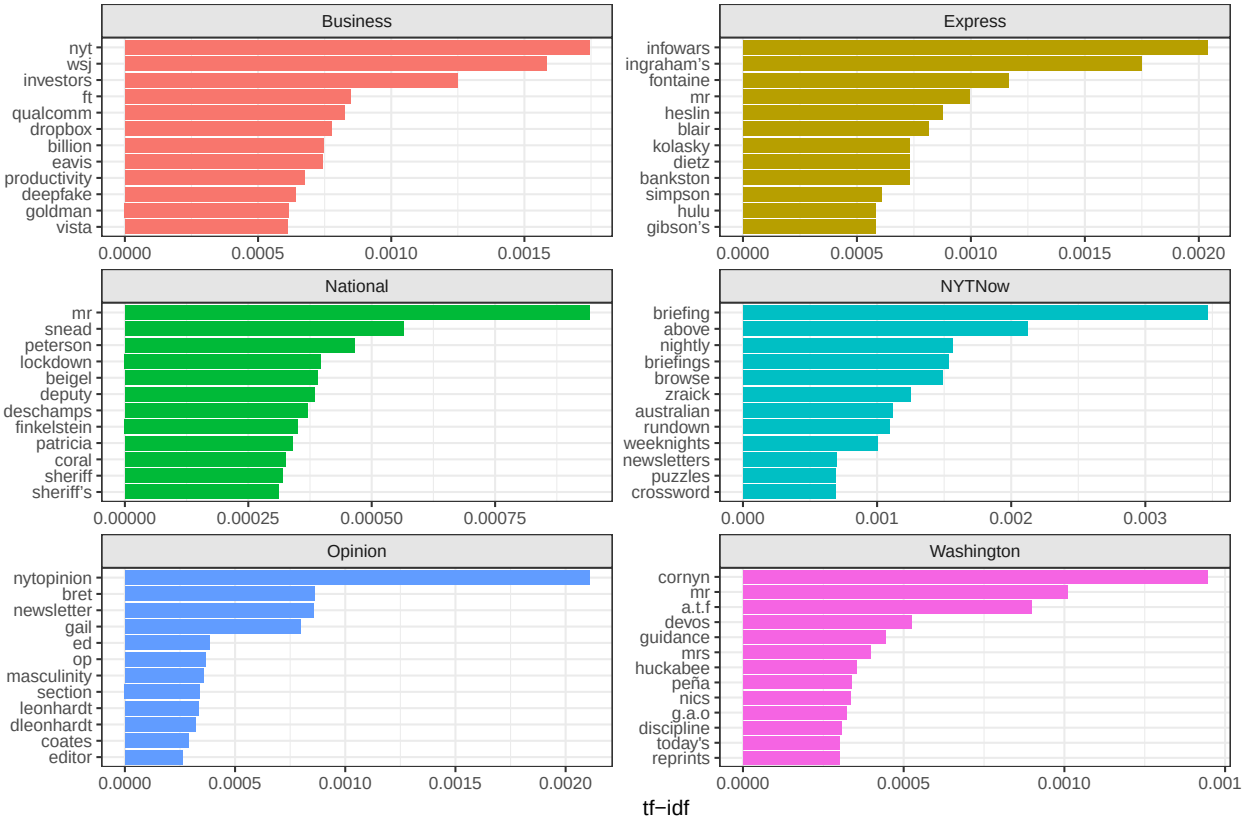
- *Zipf's law states that the frequency that a word appears is inversely proportional to its rank*

Within a section, all words are given the rank based on their frequency of appearance, that is *rank*. Whereas term-frequency is ratio of frequency of term in particular section to total words in that section. There exists some inverse relationship between these two factors, which is stated by Zip.



From the graph, we see that Zipf's law is valid throughout different sections.

8.3 Top words interms of tf-idf across sections



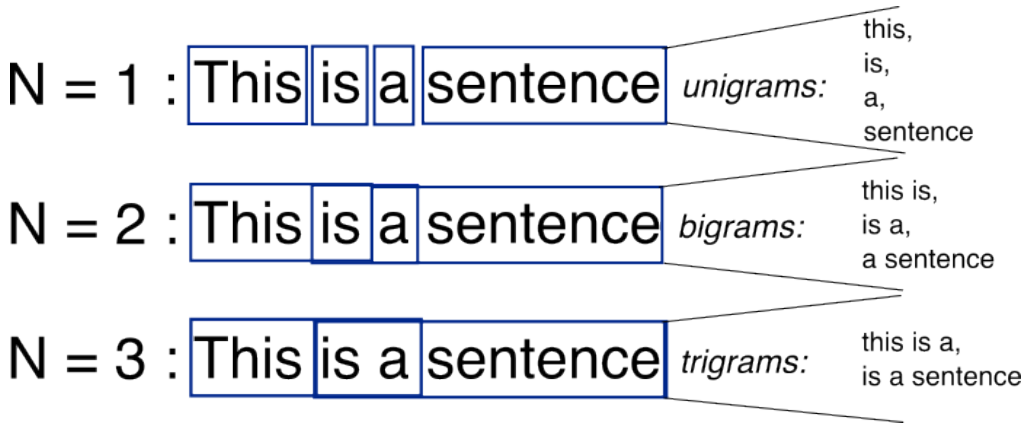
---

9 Relationships between words

[N-grams and correlations]

An N-gram is a contiguous sequence of N items from a given piece of text. The items can be phonemes, syllables, letters, words or base pairs according to the application. The N-grams typically are collected from a text or speech corpus.

An N-gram of size 1 is referred to as a "unigram"; size 2 is a "bigram" (or, less commonly, a "digram"); size 3 is a "trigram" and so on.



N-grams can also be used for efficient approximate matching. By converting a sequence of items to a set of N-grams, it can be embedded in a vector space, thus allowing the sequence to be compared to other sequences in an efficient manner.

However, we know empirically that if two strings of real text have a similar vector representation (as measured by cosine distance) then they are likely to be similar.

N-gram Models Unigram model: $P(w_1)P(w_2)...P(w_n)$

Bigram model: $P(w_1)P(w_2|w_1)P(w_3|w_2)...P(w_n|w_{n-1})$

Trigram model: $P(w_1)P(w_2|w_1)P(w_3|w_1,w_2)...P(w_n|w_{n-1},w_{n-2})$

N-gram model: $P(w_1)P(w_2|w_1)...P(w_n|w_{n-1},w_{n-2},...,w_{n-N+1})$

N-gram models: refer

9.1 Analyzing bigrams

bigram	n
of the	3048
in the	2505
to the	1272
on the	970
in a	923
at the	846
for the	821
high school	814
new york	743
to be	740

As we see the most occuring words are stopwords, so it is better to remove such bigrams where either or both of its words are stopwords.

word1	word2	n
gun	control	566
stoneman	douglas	485
parkland	fla	421
gun	violence	352
school	shooting	306
marjory	stoneman	281
president	trump	244
ar	15	241
white	house	241
york	times	237

Now this makes sense, as all stopwords are removed. Letus reunite those two words of bigram

new_desk	artcle_no	pub_date	score	day	bigram
Arts&Leisure	492	2018-04-18	0.0008733	Wednesday	heart attack
Arts&Leisure	492	2018-04-18	0.0008733	Wednesday	question mark
Arts&Leisure	492	2018-04-18	0.0008733	Wednesday	dream role
Arts&Leisure	492	2018-04-18	0.0008733	Wednesday	role he's
Arts&Leisure	492	2018-04-18	0.0008733	Wednesday	childhood picasso
Arts&Leisure	492	2018-04-18	0.0008733	Wednesday	natgeo's genius
Arts&Leisure	492	2018-04-18	0.0008733	Wednesday	taffy brodesser
Arts&Leisure	492	2018-04-18	0.0008733	Wednesday	brodesser aknerapril
Arts&Leisure	492	2018-04-18	0.0008733	Wednesday	aknerapril 18
Arts&Leisure	492	2018-04-18	0.0008733	Wednesday	18 2018

What are words with `parkland` and `gun`?

	new_desk	word1	word2	n
1	National	2018	parkland	7
2	Business	suspected	parkland	2
3	National	killed	parkland	2
4	National	vegas	parkland	2
5	None	e.d.t	parkland	2
6	U.S./Politics	post	parkland	2
7	Business	included	parkland	1
8	Business	vegas	parkland	1
9	Business	vocal	parkland	1
10	Culture	piece	parkland	1
11	Express	2018	parkland	1
12	Foreign	recent	parkland	1

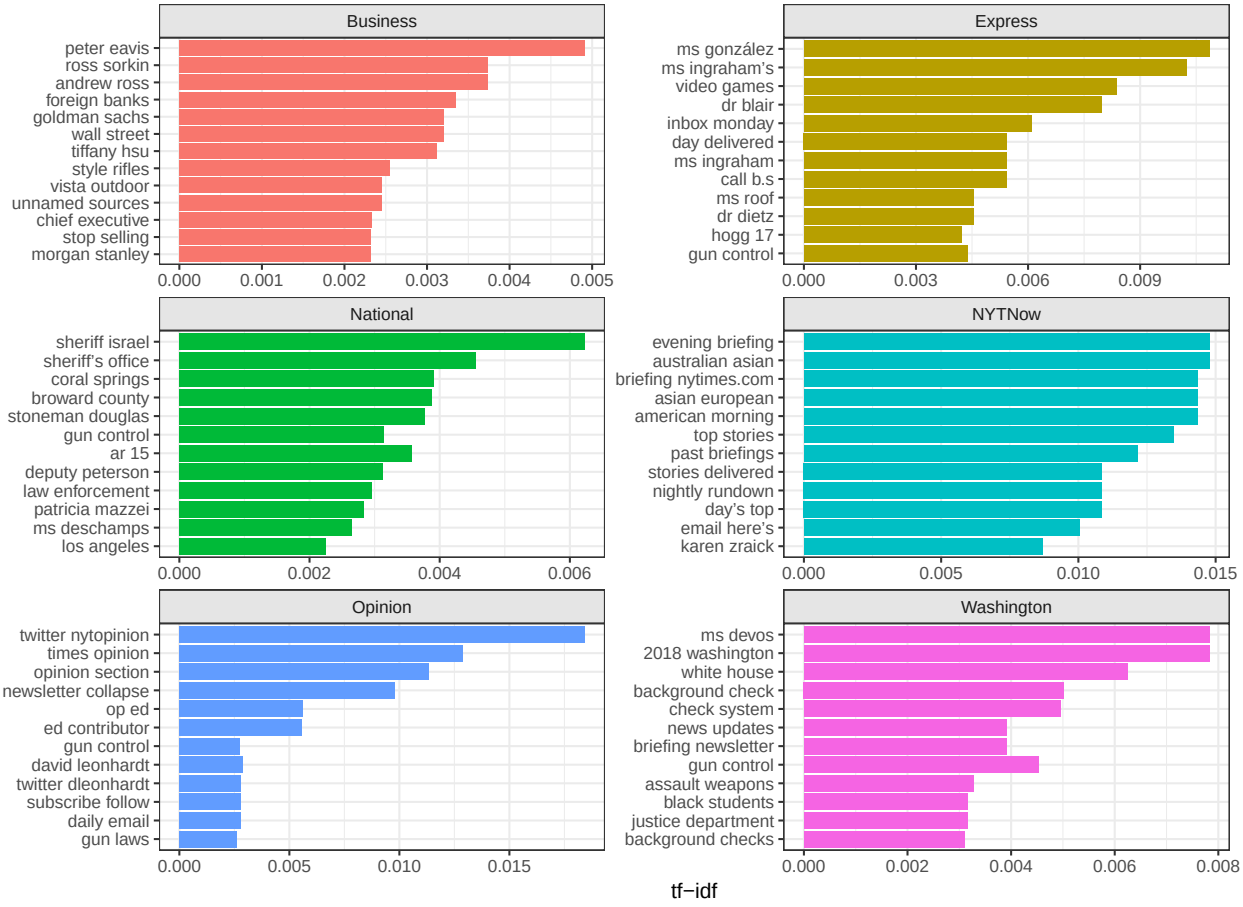
	new_desk	word1	word2	n
1	National	stricter	gun	19
2	National	pro	gun	12
3	None	stricter	gun	10
4	Opinion	sense	gun	9
5	National	prevent	gun	8
6	Opinion	anti	gun	8
7	Business	biggest	gun	7
8	National	anti	gun	7
9	None	tougher	gun	7
10	Opinion	reduce	gun	7
11	Opinion	stricter	gun	7
12	Washington	anti	gun	7

Bigram tf-idf score

Like unigrams, we can compute **tf-idf** score for bigrams also it is possible to calculate **tf-idf** score.

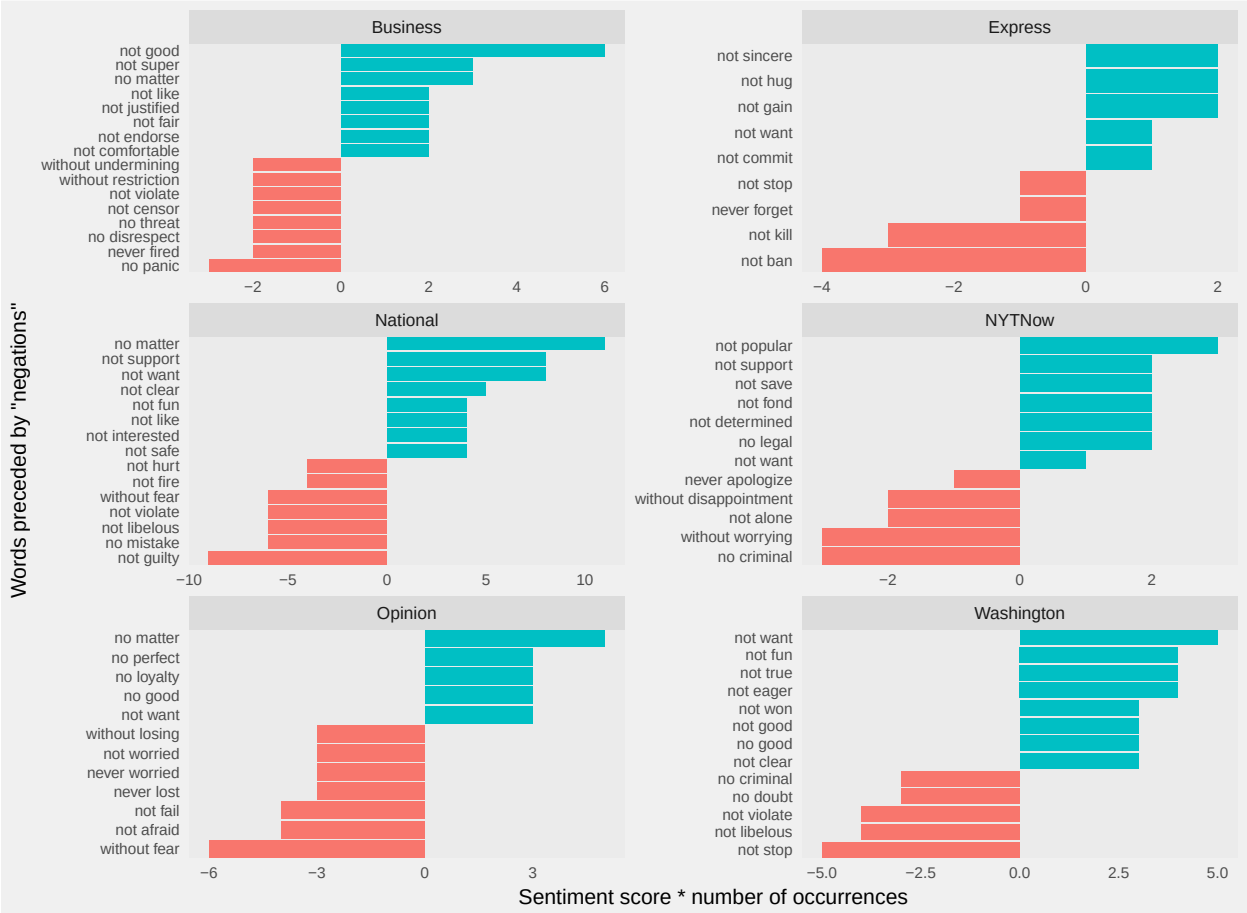
new_desk	bigram	n	tf	idf	tf_idf
Summary	page a3	3	0.0491803	3.433987	0.1688846
SpecialSections	pop ups	8	0.0349345	3.433987	0.1199646
Summary	headline quote	2	0.0327869	3.433987	0.1125897
Summary	quote appears	2	0.0327869	3.433987	0.1125897
Weekend	dear evan	6	0.0273973	3.433987	0.0940818
Weekend	evan hansen	6	0.0273973	3.433987	0.0940818
Podcasts	flash briefing	24	0.0247678	3.433987	0.0850523
Podcasts	mobile device	24	0.0247678	3.433987	0.0850523
Climate	climate change	18	0.0600000	1.354546	0.0812727
NewsDesk	school received	9	0.0273556	2.740840	0.0749774

The td_idf is high for these words, which appeared in rare sections of newspaper (i.e. Summary, SpecialSection, Weekend).


Business
peter eavis
ross sorkin
andrew ross
foreign banks
goldman sachs
wall street
tiffany hsu
style rifles
vista outdoor
unnamed sources
chief executive
stop selling
morgan stanley
0.000
0.001
0.002
0.003
0.004
0.005
Express
ms gonzález
ms ingraham's
video games
dr blair
inbox monday
day delivered
ms ingraham
call b.s
ms roof
dr dietz
hogg 17
gun control
0.000
0.003
0.006
0.009
National
sheriff israel
sheriff's office
coral springs
broward county
stoneman douglas
gun control
ar 15
deputy peterson
law enforcement
patricia mazzei
ms deschamps
los angeles
0.000
0.002
0.004
0.006
NYTNow
evening briefing
australian asian
briefing nytimes.com
asian european
american morning
top stories
past briefings
stories delivered
nightly rundown
day's top
email here's
karen zraick
0.000
0.005
0.010
0.015
Opinion
twitter nytopinion
times opinion
opinion section
newsletter collapse
op ed
ed contributor
gun control
david leonhardt
twitter dleonhardt
subscribe follow
daily email
gun laws
0.000
0.005
0.010
0.015
Washington
ms devos
2018 washington
white house
background check
check system
news updates
briefing newsletter
gun control
assault weapons
black students
justice department
background checks
0.000
0.002
0.004
0.006
0.008
tf−idf

9.2 Negative words contributed to positive sentiment and vice-versa -> unigram vs bigram

There are many cases where a bigram with first word being **negative** (e.g **not good, never killed**, etc). In such cases the unigram analysis seems to be week approach (mentioned in (7.3)). So here are some bigrams with first word being negative and to what extent they would contributed to reverse sentiment if unigram approach is used is shown in below figure.



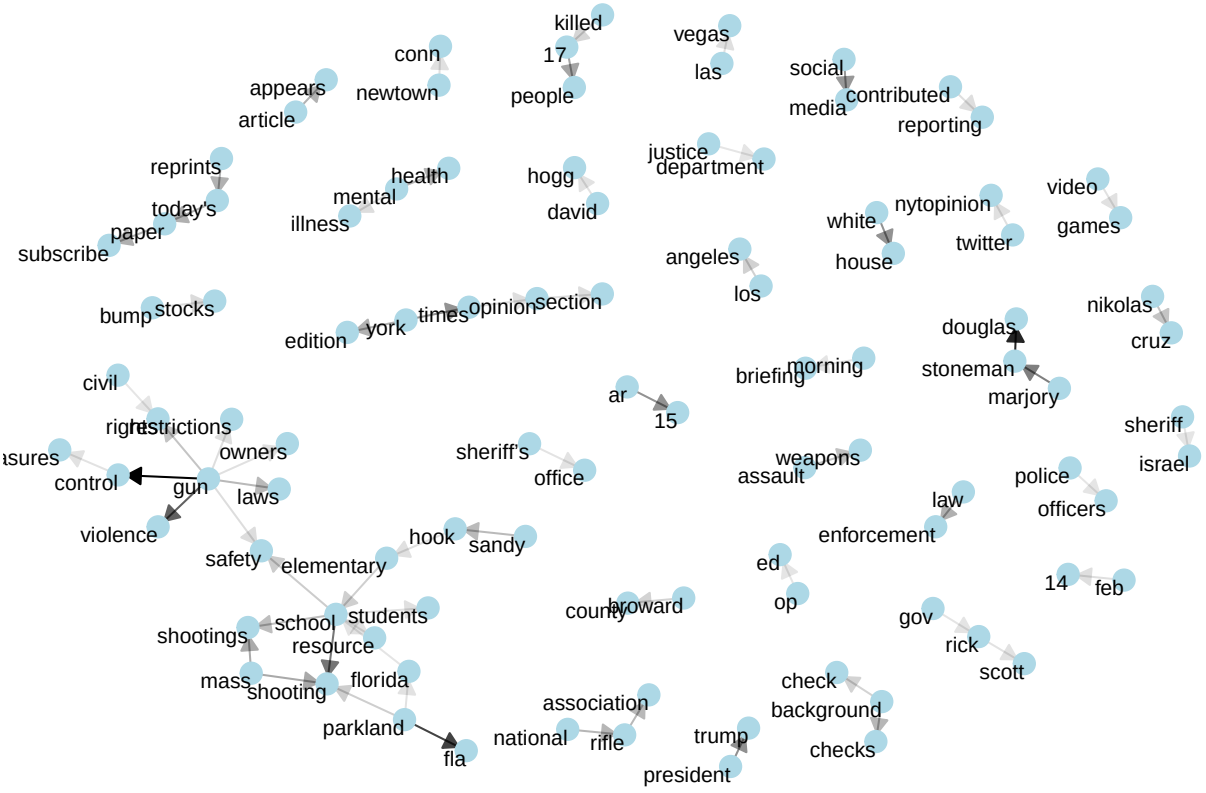
These bigrams when considered in unigrams (without those negations) contributed as reverse sentiment to the news in different sections

21

9.3 Network of bigrams

In a bigram, out of two words, sometimes one or both may be appeared with some other words. So it would be interesting to know if there is network of links between many words along with their strength (frequency).

```
IGRAPH 85b7e8a DN-- 97 66 --
+ attr: name (v/c), n (e/n)
+ edges from 85b7e8a (vertex names):
 [1] gun       ->control     stoneman  ->douglas
 [3] parkland  ->fla         gun       ->violence
 [5] school    ->shooting    marjory   ->stoneman
 [7] president ->trump       ar        ->15
 [9] white     ->house       york      ->times
[11] social    ->media       17        ->people
[13] mass      ->shooting    mass      ->shootings
[15] paper     ->subscribe   today's   ->paper
+ ... omitted several edges
```

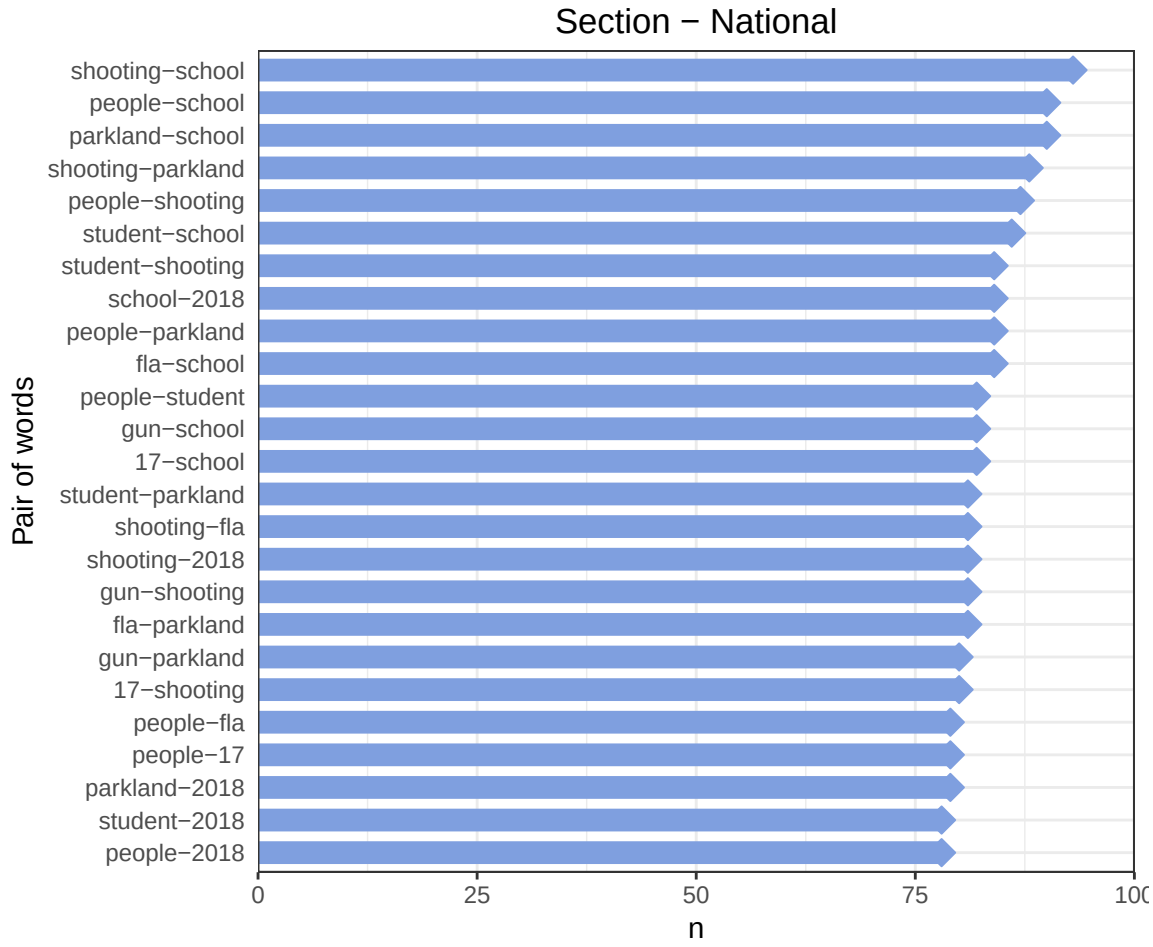


For example, if we take the word **gun**, which is linked to many other words in respective bigrams with different nuber of times. Most of the times it appeared with **control** then after **violence, laws, owners, restrictions, rights, safety**. Again **control** is connected to **measures** in another bigram. Like this we can see some other patterns in above network diagram.

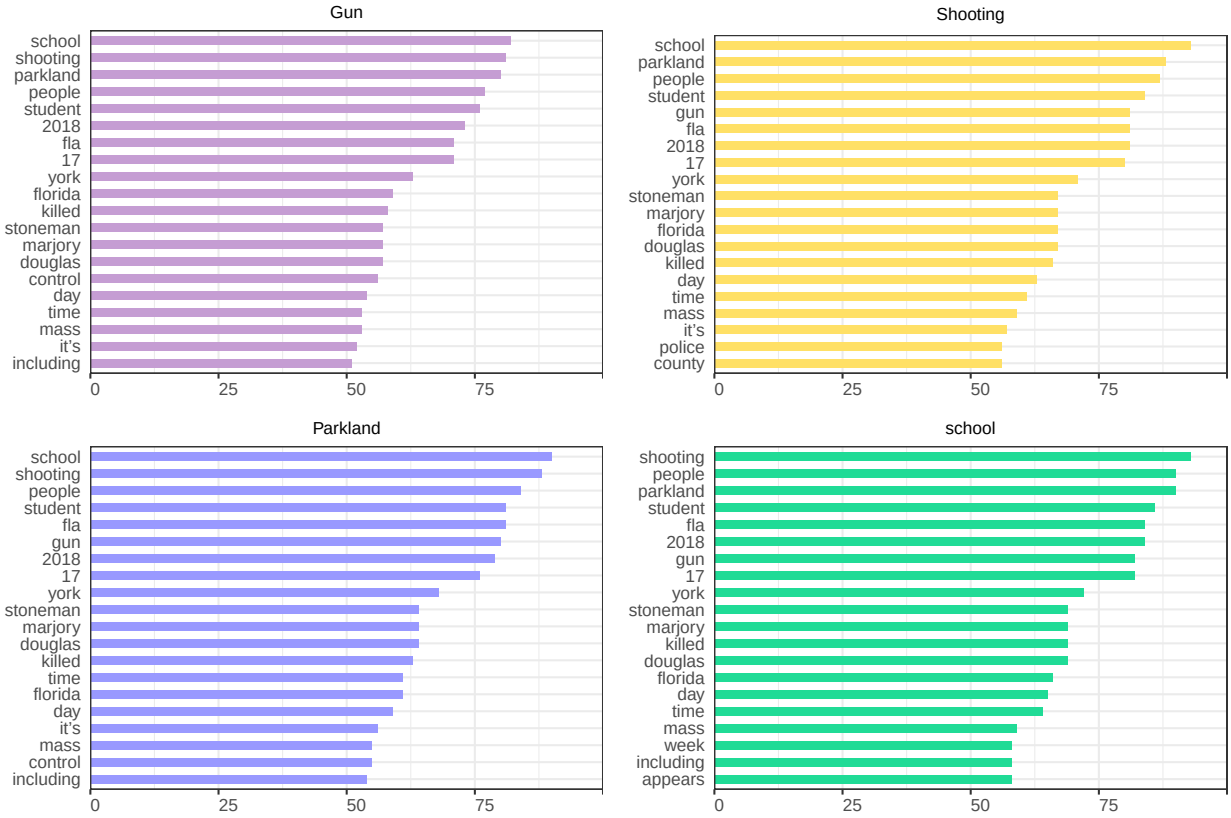
9.4 Count of word pairs

The `widyr` package makes operations such as computing counts and correlations easy, by simplifying the pattern of "widen data, perform an operation, then re-tidy data"

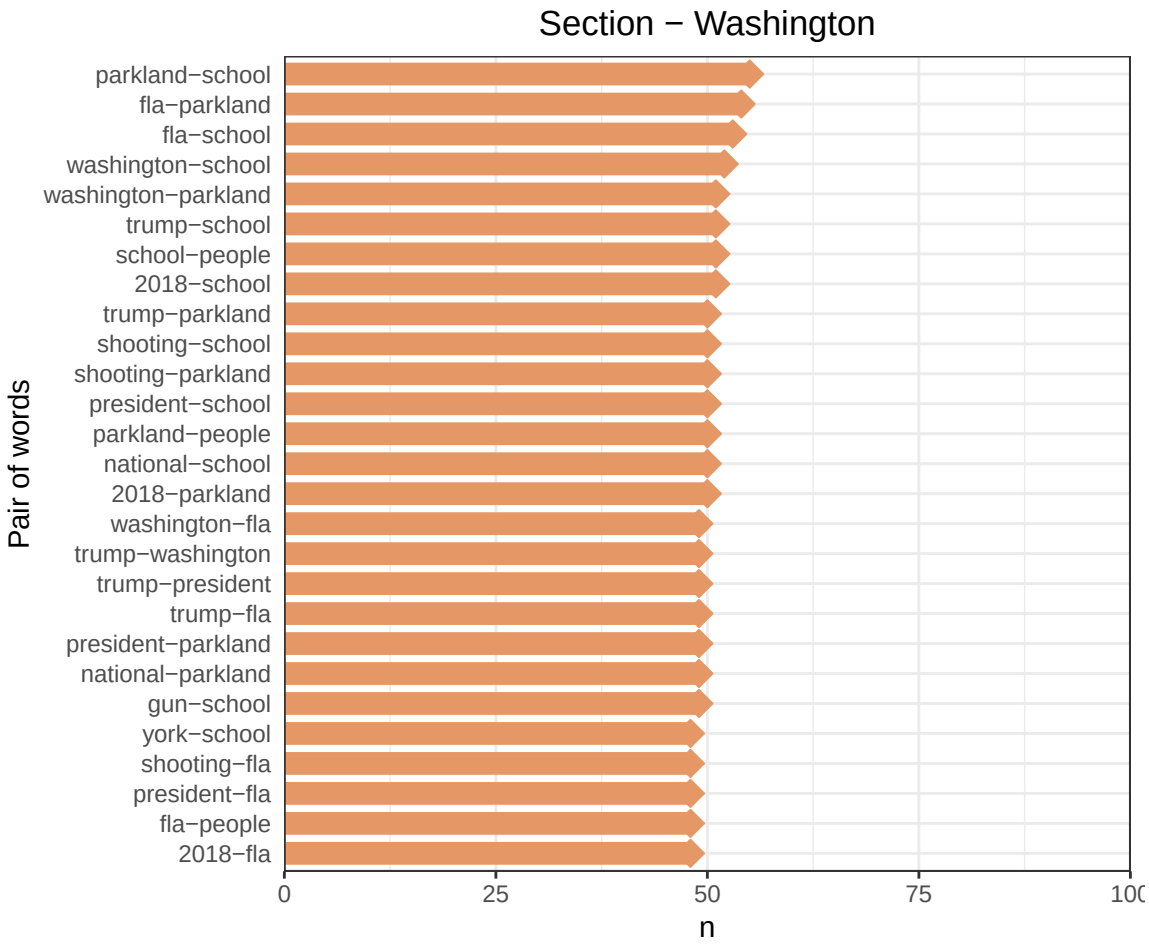
Now we are going to see which two words (need not be adjacent) appeared frequently in each article of *National* and *Opinion* sections respectively.



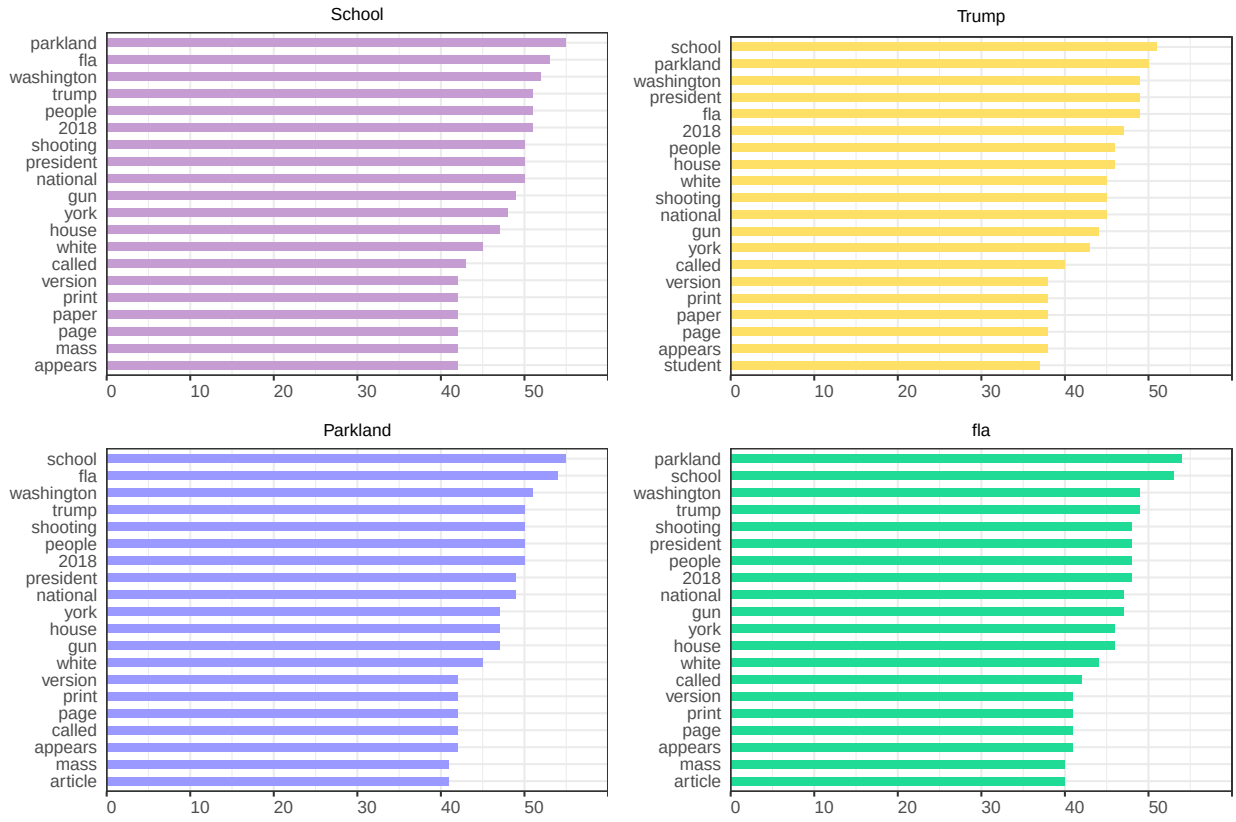
As we see here, most of the words in bigram are among *shooting*,*school*,*gun* and *parkland*. Let us see which are the other words appeared with these dominant words in **National** section


Gun
school
shooting
parkland
people
student
2018
fla
17
york
florida
killed
stoneman
marjory
douglas
control
day
time
mass
it's
including
0
25
50
75
Shooting
school
parkland
people
student
gun
fla
2018
17
york
stoneman
marjory
florida
douglas
killed
day
time
mass
it's
police
county
0
25
50
75
Parkland
school
shooting
people
student
fla
gun
2018
17
york
stoneman
marjory
douglas
killed
time
florida
day
it's
mass
control
including
0
25
50
75
school
shooting
people
parkland
student
fla
2018
gun
17
york
stoneman
marjory
killed
douglas
florida
day
time
mass
week
including
appears
0
25
50
75

24

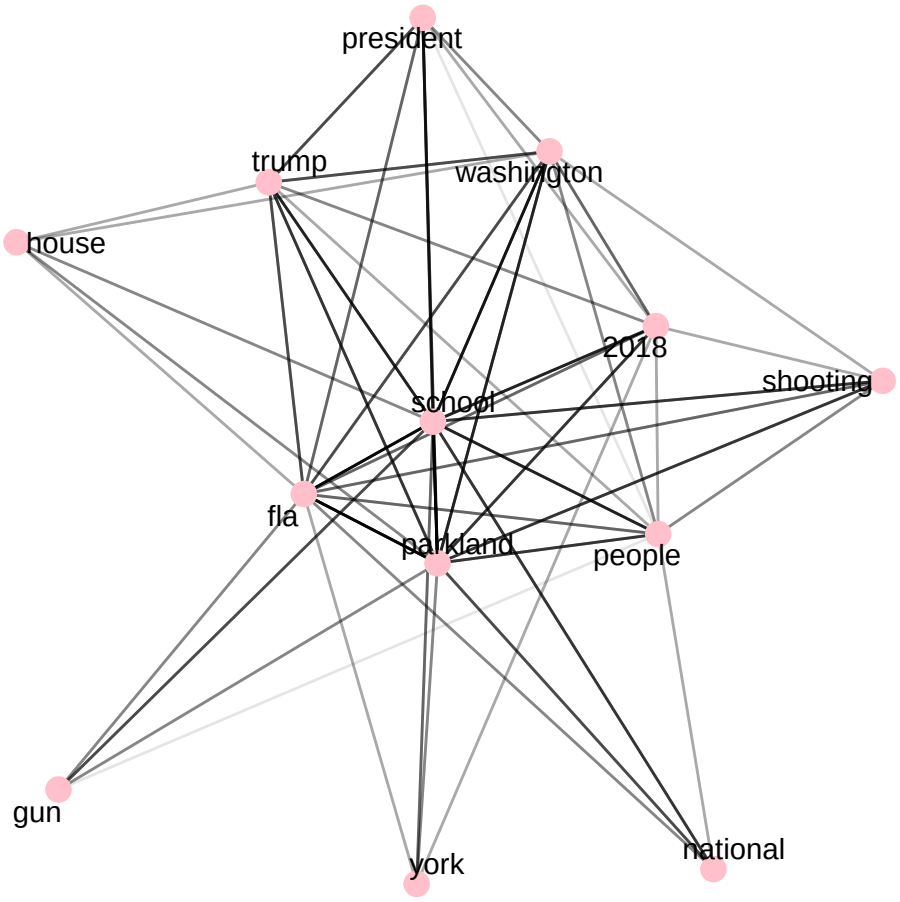


As we see here, most of the words in bigram are among *school*,*fla*,*trump* and *parkland*. Let us see which are the other words appeared with these dominant words in **Washington** section


School
parkland
fla
washington
trump
people
2018
shooting
president
national
gun
york
house
white
called
version
print
paper
page
mass
appears
0
10
20
30
40
50
Trump
school
parkland
washington
president
fla
2018
people
house
white
shooting
national
gun
york
called
version
print
paper
page
appears
student
0
10
20
30
40
50
Parkland
school
fla
washington
trump
shooting
people
2018
president
national
york
house
gun
white
version
print
page
called
appears
mass
article
0
10
20
30
40
50
fla
parkland
school
washington
trump
shooting
president
people
2018
national
gun
york
house
white
called
version
print
page
appears
mass
article
0
10
20
30
40
50

26

Word-pair : Network diagram of frequently appeared pairs



This network again tells how two words appeared frequently in *Washington* section.

9.5 Coefficient of association (correlation)

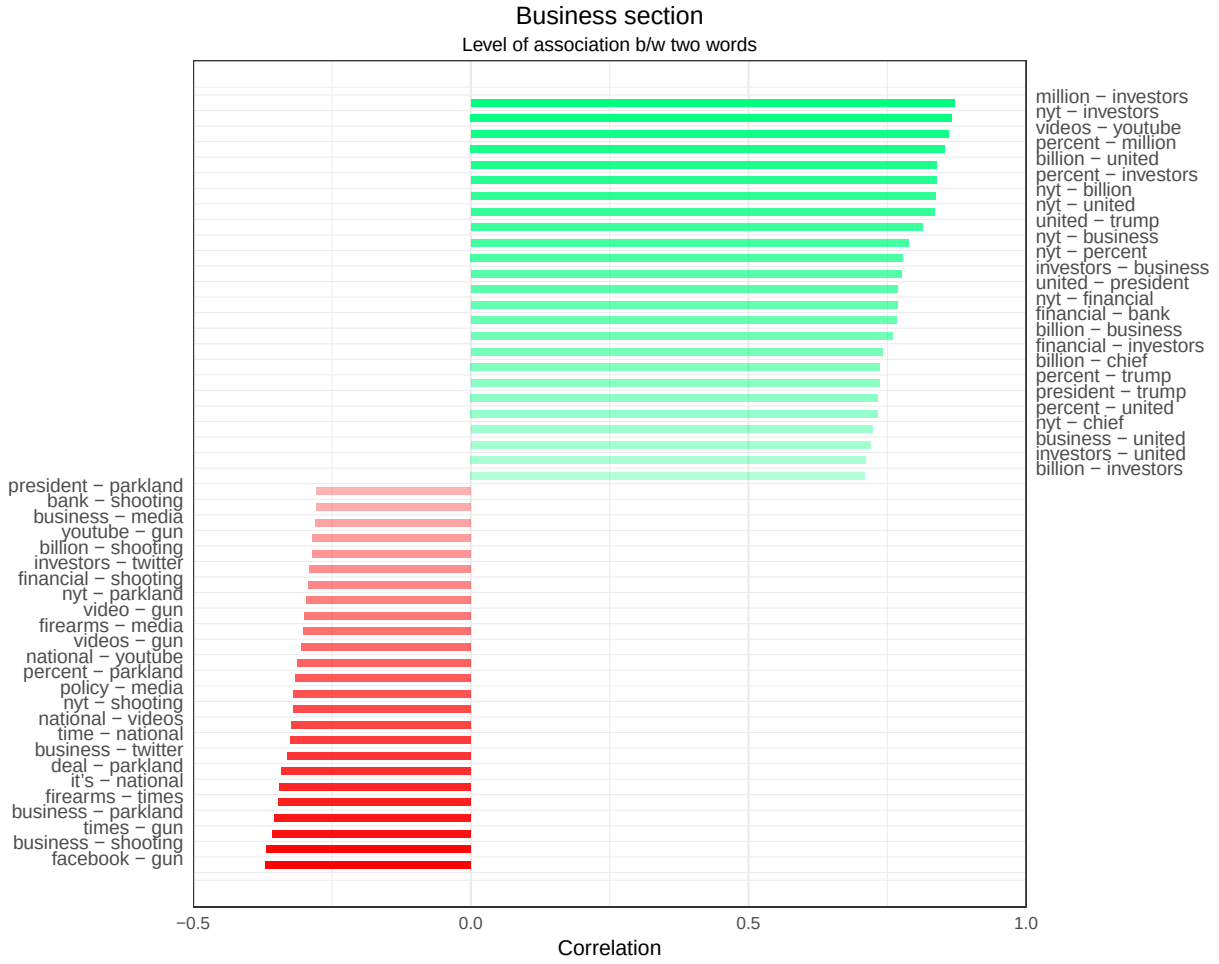
Here we'll focus on the ϕ coefficient, a common measure for binary correlation. The focus of the phi coefficient is how much more likely it is that either both word X and Y appear, or neither do, than that one appears without the other in each article

This ϕ coefficient is actually a statistical formula for measuring association between two attributes. Same formula here we are using to measure the level as well as direction of association between two words in each news article of a particular section(s).

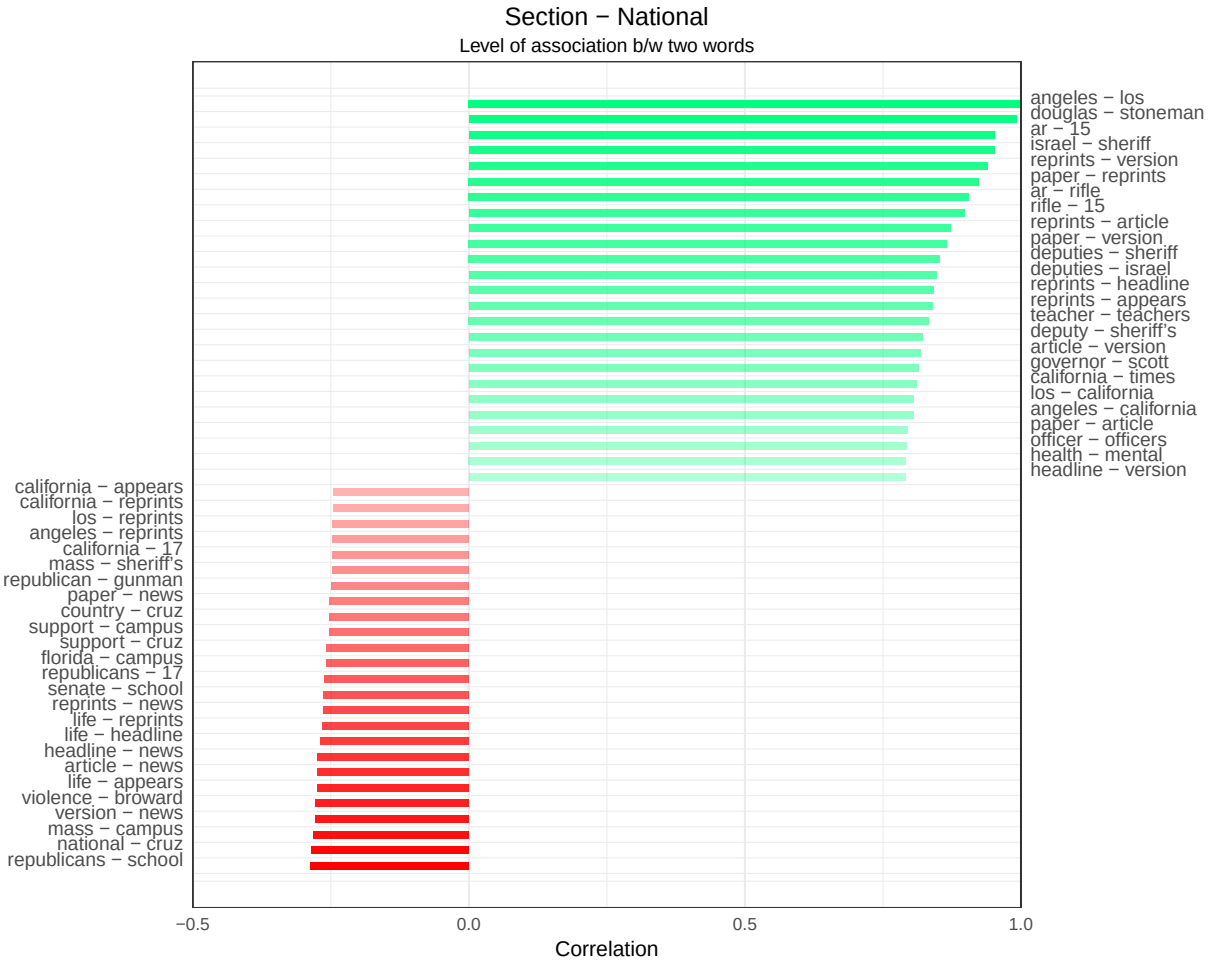
	d has word Y	d has no word Y	Total
d has word X	n11	n10	n1.
d has no word X	n01	n00	n0.
Total	n.1	n.0	n

$$\phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_1.n_0.n_{\cdot}0n_{\cdot}1}}$$

Pairs with high correlation(green) have high level of association and those with low correlation(red) have less association.

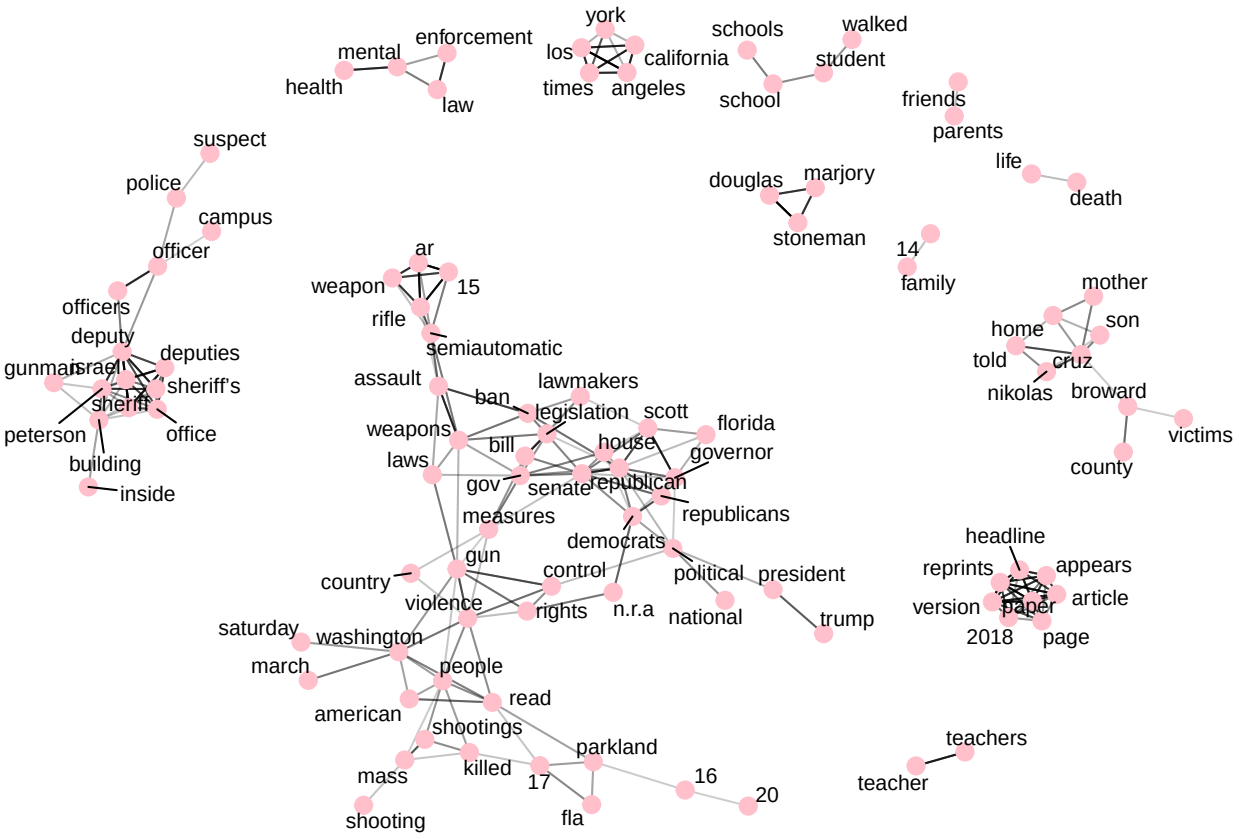


28


Section − National
Level of association b/w two words
angeles − los
douglas − stoneman
ar − 15
israel − sheriff
reprints − version
paper − reprints
ar − rifle
rifle − 15
reprints − article
paper − version
deputies − sheriff
deputies − israel
reprints − headline
reprints − appears
teacher − teachers
deputy − sheriff's
article − version
governor − scott
california − times
los − california
angeles − california
paper − article
officer − officers
health − mental
headline − version
california − appears
california − reprints
los − reprints
angeles − reprints
california − 17
mass − sheriff's
republican − gunman
paper − news
country − cruz
support − campus
support − cruz
florida − campus
republicans − 17
senate − school
reprints − news
life − reprints
life − headline
headline − news
article − news
life − appears
violence − broward
version − news
mass − campus
national − cruz
republicans − school
−0.5
0.0
0.5
1.0
Correlation

29

9.6 Association between pairs: network digram



Thus different words are connected to each other by a correlation. We see some words like *Washington, Sheriff, gun* are having maximum number of connections, meaning there is positive association between these words with most of some other words in news.

---

10 Topic modeling

- **A method for finding a group of words (i.e topic) from a collection of documents that best represents the information in the collection.** It can also be thought of as a form of text mining – a way to obtain recurring patterns of words in textual material.
- Topic modeling is a method for unsupervised classification of documents, similar to clustering on numeric data, which finds natural groups of items based on the similarity between their conents. It provides a simple way to analyze large volumes of unlabeled text.
- A "topic" consists of a cluster of words that frequently occur together. Using contextual clues, topic models can connect words with similar meanings and distinguish between uses of words with multiple meanings.
- There are many techniques that are used to obtain topic models.
  - Latent Dirichlet Allocation (LDA): a widely used topic modelling technique.
  - TextRank process: a graph-based algorithm to extract relevant key phrases.

10.1 LDA [Latent Dirichlet allocation]

Latent Dirichlet allocation (LDA) is a one of the popular method for fitting a topic model. Here each document is treated as a mixture of topics, and each topic as a mixture of words.

LDA is a mathematical method for finding the mixture of words that is associated with each topic, while also determining the mixture of topics that describes each document.

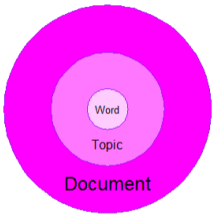
In LDA, we assume that there are k underlying latent topics according to which documents are generated, and that each topic is represented as a multinomial distribution over the $|V|$ words in the vocabulary. A document is generated by sampling a mixture of these topics and then sampling words from that mixture.

More precisely, a document of N words $W = (w_1, w_2, \cdots, w_N)$ is generated by the following process. First, θ is sampled from a Dirichlet $(\alpha_1, \alpha_2, \cdots, \alpha_k)$ distribution. This means that θ lies in the (k - 1)-dimensional simplex: $\theta_i \geq 0, \sum_i \theta_i = 1$. Then, for each of the N words, a topic $z_n \in \{1, 2, \cdots, k\}$ is sampled from a $Mult(\theta)$ distribution $P(z_n = i/\theta) = \theta_i$. Finally, each word w_n is sampled, conditioned on the z_nth topic, from the multinomial distribution $P(w/z_n)$. Intuitively, θ_i can be thought of as the degree to which topic i is referred to in the document. Written out in full, the probability of a document is therefore the following mixture:

$$p(w) = \int_\theta \left(\prod_{n=1}^{N} \sum_{z_n=1}^{k} P(w_n|z_n;\beta)P(z_n|\theta) \right) P(\theta;\alpha)d\theta$$

where $P(\theta;\alpha)$ is Dirichlet, $P(z_n|\theta)$ is a multinomial parameterized by θ, and $P(w_n|z_n;\beta)$ is a multinomial over the words. This model is parameterized by the kdimensional Dirichlet parameters $\alpha = (\alpha_1, \alpha_2, ..., \alpha_k)$ and a $k \times |V|$ matrix,β, which are parameters controlling the k multinomial distributions over words.

In our case each news article(500) is a single document and topics are particular sections like **National**,**Sports**,**Politics**, etc



The topic modeling is done on the news articles which are dominant interms of frequency. Thus *Opinion, Business, Washington, NYTNow* these are most sections we are refering in topic modeling. What all here done is taking all words from these four sections and then modeling them in four topics. At last we depict the effect of our modeling in pictorial way, where it is possible to know how well our modelling classified each word in different sections.

Data Preparation

First of all, we are using `topicmodels` package for topic modeling, which needa a Document Term Matrix, and it is directly obtained by applying `cast_dtm()` function from `tidytext` package on the tidy data.

document	word	n
Business_478	news	49
Opinion_88	gun	43
Opinion_101	gun	42
Washington_40	gun	38
Washington_404	school	34
Business_282	school	33
Washington_212	student	33
Business_484	banks	32
Washington_185	gun	32
Washington_205	ar	32

```
<<DocumentTermMatrix (documents: 224, terms: 16738)>>
Non-/sparse entries: 84181/3665131
Sparsity           : 98%
Maximal term length: 22
Weighting          : term frequency (tf)
```

10.2 LDA on sections

As we considered 4 sections only, so the number of topics is also 4.

```
A LDA_VEM topic model with 4 topics.
```

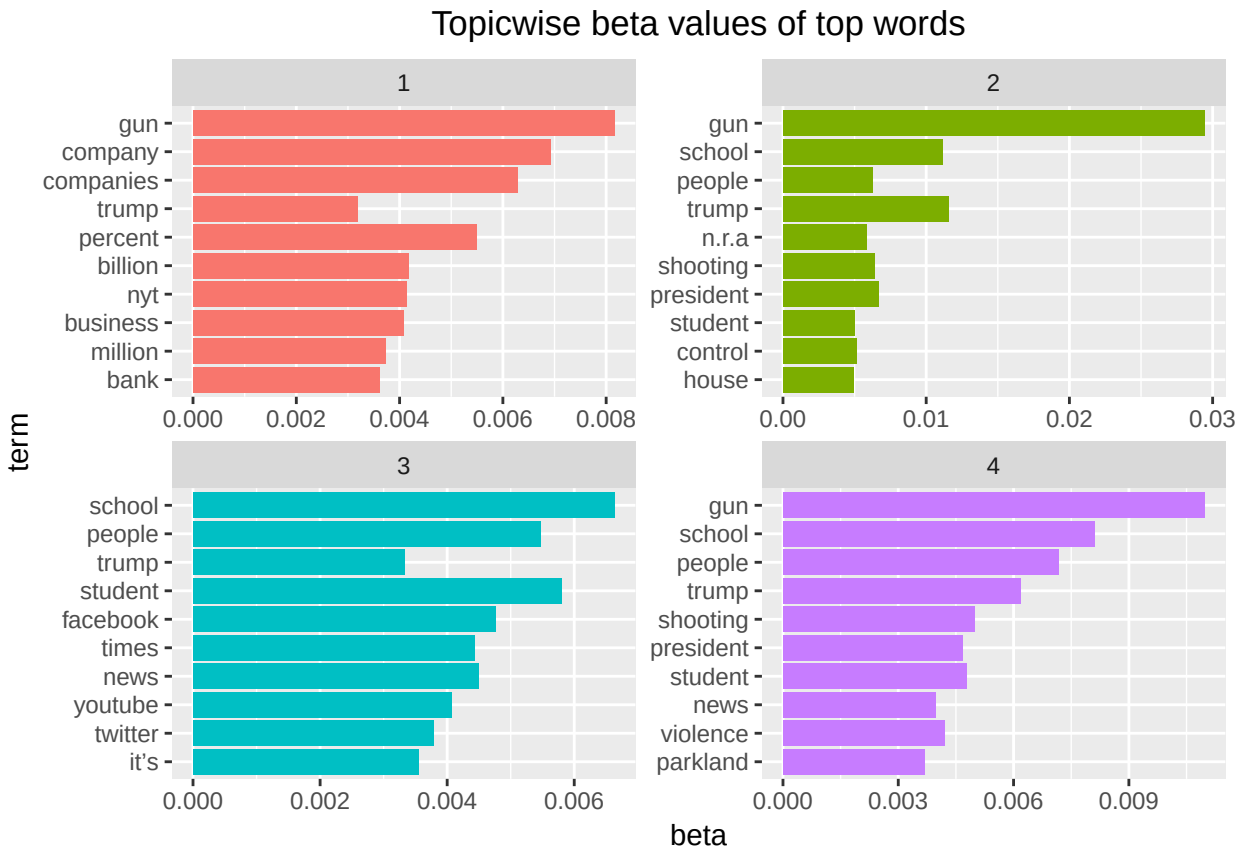
Beta values

- **Beta:** Which is a measure of probability of each word being classified in each topic. It is also called *per-topic-per-word probabily*

Document can be actually represented as the `Section_serial.number` for convenience.

topic	term	beta
3	towels	0.00e+00
4	towels	3.65e-05
1	trucker	0.00e+00
2	trucker	0.00e+00
3	trucker	0.00e+00
4	trucker	3.66e-05
1	uncross	0.00e+00
2	uncross	0.00e+00
3	uncross	0.00e+00
4	uncross	3.66e-05

The top 10 words in each topic interms of beta values are given in below graph.



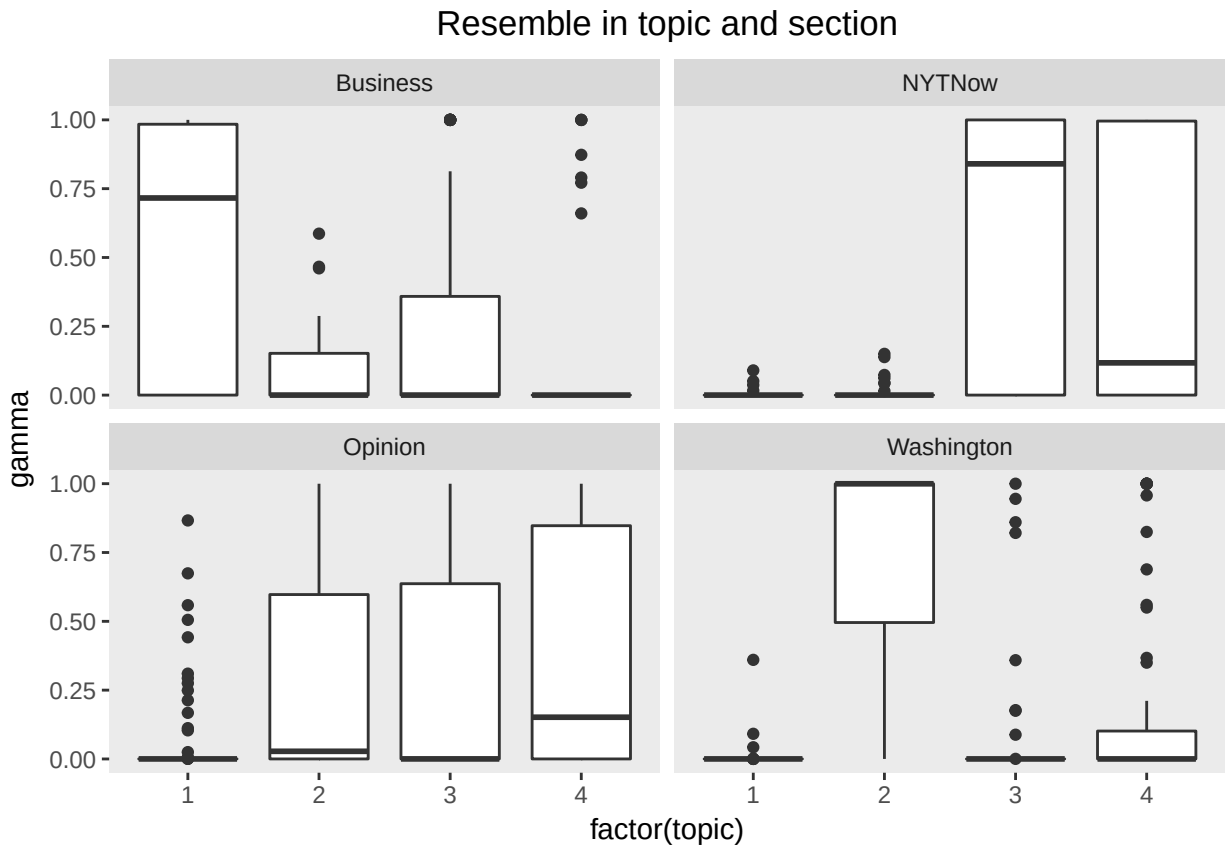
gamma values

Per-document classification

The document is each news article. As already mentioned, document name is combination of section and article serial number. Here the Document-topic probability is calculated (called **gamma**), which tells by what probability particular topic is related to particular document.

```
# A tibble: 10 x 3
   document     topic     gamma
   <chr>        <int>     <dbl>
 1 Business_119     1 0.336
 2 Business_121     1 1.000
 3 Business_134     1 0.825
 4 Business_147     1 0.0000979
 5 Business_159     1 0.740
 6 Business_164     1 0.534
 7 Business_165     1 0.955
 8 Business_166     1 0.0000699
 9 Business_168     1 0.764
10 Business_174     1 0.979
```

Each of these values is an estimated proportion of words from that particular document that are generated from that given topic.

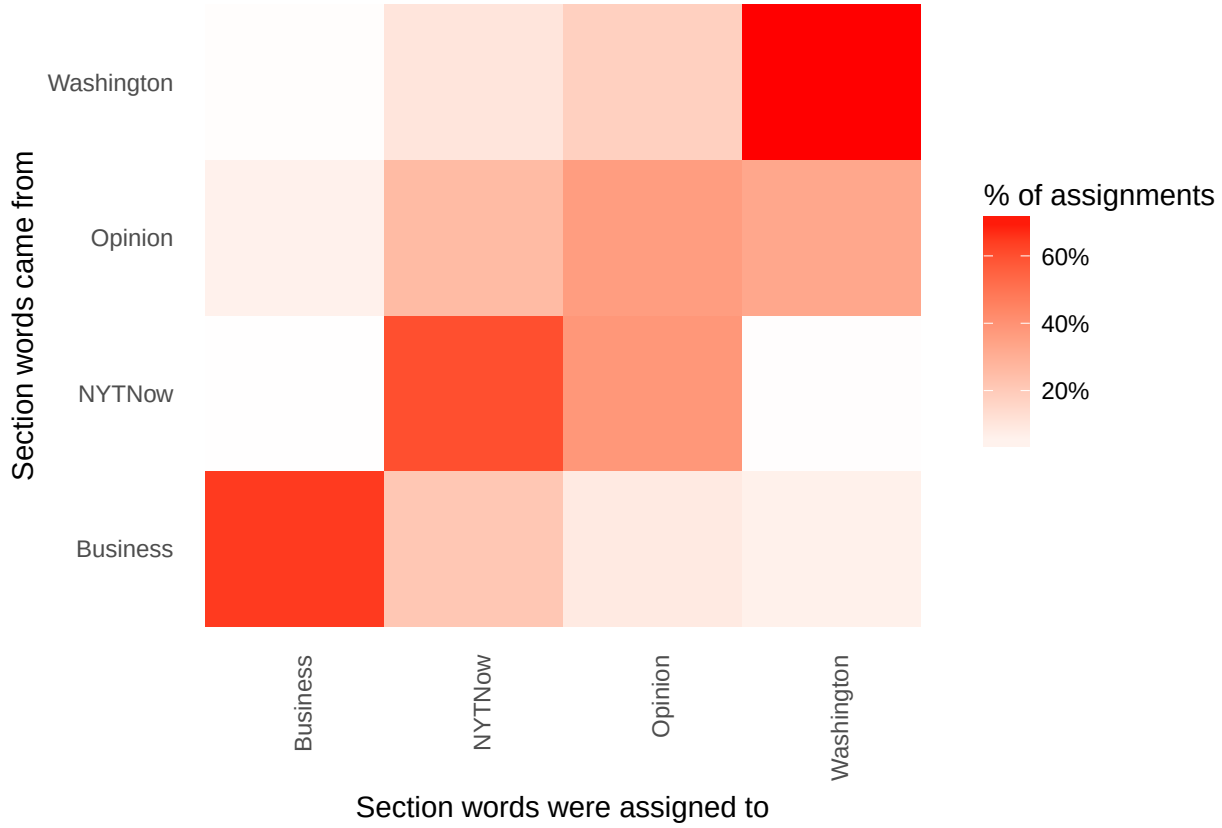


As we see, the topic 1 corresponds to *Business*, topic 2 corresponds to *Washington*, topic 3 and topic 4 are corresponding to *NYTNow* and *Opinion* sections respectively. But topic 2 which is analogous to *Washington* seems to have some contribution to *Business* and *Opininion* sections as well. Means there are some words in *Washington* (topic 2), which are also in other two mentioned sections (topics). Similarly the pattern of overlap of words is seen in other topics also. This will be clearly seen in below graph.

```
# A tibble: 12 x 4
# Groups:   topic [4]
   section    article_no topic gamma
   <chr>           <int> <int> <dbl>
 1 Business          471     1 1.000
 2 Business          495     1 1.000
 3 Business          497     1 1.000
 4 Washington        185     2 1.000
 5 Washington        205     2 1.000
 6 Washington         40     2 1.000
 7 Business          391     3 1.000
 8 Business          478     3 1.000
 9 Business          491     3 1.000
10 Opinion           428     4 1.000
11 Washington        397     4 1.000
12 Washington         51     4 1.000
```

10.3 Model assignments

Now we are assigning each topic to each sections and plotting the graph of classification, which will reveal how well the modeling is.



The graph tells us that most of the words which came from the section *Opinion* are classified into *NYTNow* and *Washington*. Anyhow there is sharing of some common words (must be related to parkland shooting) in almost all sections to some extent at least. But still *Business, Washington and NYTNow* sections seem to have better classification.

11 Conclusions

Two main important aspects of this project are **sentiment analysis** and **topic modeling**. We successfully analysed the sentiment of all articles through different sections. The sentiment of news turned to be negative overally, as all news were related to shooting.

In topic modeling, we took all news fromfour major news sections of NYT namesly *Washington, Opinion, Business and NYTNow*. Then blindly (like unsupervised learning) we modelled the news into four topics and associated them to corresponding sections considered. We have got nice classification except for *Opinion* section.

Beside of theses, many graphical explorations are also carried out throught the work, to comapre many factors among sections.

---

12 Reference

1) https://www.tidytextmining.com
2) https://www.kdnuggets.com/tag/text-analytics
3) https://www.tidyverse.org/packages/
4) https://rviews.rstudio.com/2017/06/08/what-is-the-tidyverse/
5) https://www.jstatsoft.org/article/view/v083b01/v83b01.pdf
6) https://developer.nytimes.com/
7) https://developers.google.com/chart/interactive/docs/gallery/sankey

---
