Comparing US Presidential Speeches.
A Multi-Method Data Analysis of Barack Obama and Donald Trump

Bachelor-Thesis
Autor: **Julian Lemmerich**
Erstgutachter: **Prof. Dr. Marcus Müller**
Zweitgutachter: **Prof. Dr. Thomas Weitin**

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Content

1. Introduction

With Barack Obama and Donald Trump two presidents very different from one another led the United States of America back-to-back in the last decade. Barack Obama was the first black president in the history of the nation and introduced progressive policies like the American Health Care Act during his two terms in office. Donald Trump on the other hand followed very conservative and nationalist politics with harsh immigration policies propagating the motto "America First". The popular consensus is that there are stark differences in their corpora. I want to find out if there is objective support for these felt differences.

In two previous papers I laid the groundwork for comparing the texts of these two presidents, creating corpora of their public speeches.[1]

The goal of this thesis is to use a multitude of digital methods to show differences between the two corpora. These methods include simple digital analyses like type-token-ratio, word and sentence length, part of speech tagging and word frequencies, and more complex methods like readability scoring, topic modeling, sentiment analysis and authorship attribution with machine learning. This thesis will not include the political or societal impacts and interpretations of the data based results. The main emphasis will hereby lie on the methodology and generating data.

2. Biographies

2.1. Barack Obama

Barack Obama was the 44th president of the United States of America from 2009 until 2017. He was born on the 4th of August 1961 in Hawaii. In 1979 he started studying at the *Occidental College* in Los Angeles. In his two years there he held his first public speech about the College's participation in disinvestment from South Africa. He studied another two years at the *Columbia University* in New York City, an Ivy League University. After his Bachelor's degree in political science with an emphasis on international relations and in English literature in 1983 he worked for a year at the *Business International Corporation*, where he was a financial researcher and writer. Even though Obama was offered a scholarship for the *Northwestern University School of Law*, he started studying at *Harvard Law School* in 1988. At the end of his first year, he started writing for the *Harvard Law Review*, then was selected as president of the journal in the following year. In 1991 he graduated as a Juris Doctor from *Harvard Law School*.[2]

In 1996 Obama was elected to the Illinois Senate. In 2004 he was elected to the US Senate. In 2007 he started his ultimately successful run for president at the age of 46. He was president until Trump took over in 2017.

Throughout his political career, Obama was supported by a Team of speechwriters: Jon Favreau, Adam Frankel, Ben Rhodes, Jon Lovett, David Litt and Kyle O'Connor. Many of them

[1] Lemmerich, Julian (2020): *President Obama's Speeches. Corpus. Quantitative Analysis and Authorship Attribution*, Technische Universität Darmstadt, unpublished manuscript

Lemmerich, Julian (2021a): *Trump Speech Corpus*, Technische Universität Darmstadt, unpublished manuscript

[2] Wikipedia: *Barack Obama*, https://en.wikipedia.org/wiki/Barack_Obama (retrieved 10.08.2021)

studied at renowned universities, mostly with degrees in language or political sciences. It was not a secret that he had speechwriters and advisors.

Biographies of Obama's speechwriters can be read in the appendix chapter 9.1.1.

2.2. Donald Trump

Donald Trump was the 45th president of the United States from 2017 till 2021. Before his presidential campaign he was CEO of the *Trump Organization* and host of the casting show *The Apprentice*. At the age of 13 his father sent him to *New York Military Academy*. After graduating with a high school degree, he studied economic sciences first at *Fordham University*, then at *Wharton School* in Philadelphia.[3]

Unlike Obama, Trump tried to hide his speechwriters from the public. One reason for that is, that he is very proud of his free and independent speech style. He heavily criticized his predecessor Obama and opponent Hillary Clinton multiple times for using speech writers and teleprompters. He even went as far as repeatedly saying that teleprompters should not be allowed if you are running for president.[4] He himself claimed to think of the speeches himself. »I think about my speeches a lot. Essentially, I don't use notes and I definitely don't read the speeches. [...] I do a lot of things by myself. [...] People are shocked at how smart I am.«[5]

Trump wants to present the image of independence in his speeches. He thus has to hide any assistance he gets for writing the speeches. Requests by journalists about the speechwriting process have been declined by administration officials.[6] I was able to find four members of his staff, that assisted with speechwriting: Stephen Miller, Vincent M. Haley, Ross P. Worthington and Ryan Jarmula.

Biographies of Trump's speechwriters can be read in the appendix chapter 9.1.2.

[3] White House Historical Association: *Donald Trump. THE 45TH PRESIDENT OF THE UNITED STATES*, https://www.whitehouse.gov/about-the-white-house/presidents/donald-j-trump/ (retrieved 25.03.2021)

[4] Hains, Tim: *Trump: If You're Running For President You Shouldn't Be Allowed To Use A Teleprompter*, in: *RealClear Politics* (25.08.2015), http://www.realclearpolitics.com/video/2015/08/25/trump_i_write_my_own_tweets_if_youre_running_for_president_you_should_be_allowed_to_have_teleprompters.html?jwsource=cl (retrieved 22.03.2021)

Nussbaum, Matthew: *Trump and the teleprompter: A brief history*, in: *POLITICO* (06.07.2016), https://www.politico.com/story/2016/06/donald-trump-teleprompter-224039 (retrieved 22.03.2021)

Wolf, Zach Byron: *Trump breaks his own rule, uses teleprompter.*, in: *CNNPolitics* (22.03.2016), https://edition.cnn.com/2016/03/21/politics/trump-teleprompter-aipac-speech/index.html (retrieved 22.03.2021)

[5] Hains (2015)

Mango News (17.08.2015): *Donald Trump: Obama is Teleprompter Guy. We Dont Want Scripted President*, https://www.youtube.com/watch?v=hs5woj5Ae48 (retrieved 22.03.2021), cited in Lemmerich 2021a

[6] Rogers, Katie: *The State of the Union Is Trump's Biggest Speech. Who Writes It?*, in: *The New York Times* (02.03.2020), https://www.nytimes.com/2020/02/03/us/politics/trump-state-of-the-union.html (retrieved 22.03.2021)

3. Previous Research

3.1. Corpora

The Obama corpus was sourced from American Rhetoric.[7] The first version of this corpus was provided by Sabine Bartsch, which was also sourced from American Rhetoric. It was created in 2014 and thus only contained texts from 2009 till 2014. By the time of writing Lemmerich 2020, Obama had concluded his second term and held more speeches. For that paper I thus decided to create a new and complete Obama corpus, including speeches from 2004 till 2017. American Rhetoric provides 467 speeches by Obama. The only metadata included was date and title of the speech, which was encoded in the file name. The contents were encoded in plaintext for easier handling. After filtering 377 speeches remained.[8]

The Trump corpus was sourced from the UCSB American Presidency Project (APP).[9] American Rhetoric, where the Obama corpus was sourced from, did not have any texts by Trump yet as of February 2021. Since the APP does not only collect speeches but also other forms of texts, it was filtered to only the following categories: Interviews, Miscellaneous, Remarks, News Conferences, Spoken Addresses and Remarks, Farewell Addresses, Inaugural Addresses, Oral Address, Saturday Weekly Addresses, State Dinners, State of the Union Addresses, Campaign Documents, Convention Speeches, Presidential Nomination Acceptance Addresses, Statements. This creates a corpus of 2968 texts from the APP database. After filtering 1797 speeches remained.[10]

As a reference corpus the *Corpus of Contemporary American English* (COCA) will be used. It was created by Mark Davies from *Brigham Young University* and consists of over one billion words.[11] Access to the full corpus is not possible for free, but access to a smaller dataset of the 5000 most frequent words is free.[12]

[7] Eidenmuller, M. E.: *Obama Speeches*, https://www.americanrhetoric.com/barackobamaspeeches.htm (retrieved 27.03.2019)

[8] Lemmerich (2020), p. 8

[9] The American Presidency Project, https://www.presidency.ucsb.edu/about

[10] Lemmerich (2021a), pp. 9-11

[11] Davies, M. (2010): *The Corpus of Contemporary American English as the first reliable monitor corpus of English*, in: *Literary and Linguistic Computing* 25 (4), pp. 447–464. DOI: 10.1093/llc/fqq018

Available at https://www.english-corpora.org/coca/ (retrieved 15.08.2021)

[12] https://www.corpusdata.org/formats.asp (retrieved 09.07.2021)

3.2. Previous Paper Results

3.2.1. Barack Obama[13]

The initial aim of my paper about President Obama's speeches was using stylometric analysis for authorship attribution of different speeches to the different speechwriters in Obama's staff. A difficulty would be that the speechwriters obviously try to emulate the style of the speaker. A paper by Jonathan Herz and Abdelghani Bellaachia from George Washington University was however successful in identifying the speechwriters on a small corpus of 37 speeches.[14] A similar classification of presidential speechwriters was tried in *Who Wrote Ronald Reagan's Radio Addresses?* by Airoldi, Anderson, Fienberg and Skinner in 2006. Using a mix of different classifiers they were able to identify authors in 207 out of the 312 speeches with unknown authorship.[15]

The results from my stylometric analysis were a lot less promising. With the Burrow's Delta it was not possible to distinguish between different authors. There could have been several reasons for this: The speechwriters are trying to emulate the style of the speaker. Making it indistinguishable is their job and they did it really well. Obama was also involved in the writing process himself. Drafts would bounce back and forth between the speechwriters and Obama. Working together over a decade also unifies the writing style. Other methods, like with tagged data trained machine learning algorithms could also have been more successful than the Burrow's Delta.

3.2.2. Donald Trump[16]

Donald Trump's public communication style is very different from presidents before him and other politicians. It is a characteristic that he is very proud of and that finds good resonance with his target voters. After his reelection campaign was unsuccessful in 2020, his presidency concluded on the 20th of January 2021 and a comprehensive corpus of his time as president could be created. This was the aim of my 2021 paper *Trump Speech Corpus*.

Even though Twitter was Trump's main form of communication with the public, it was not included in the corpus, as the corpus was supposed to be comparable to the already existing Obama Speech Corpus. Nevertheless, there are a few interesting things that can be learned from this: Trump believed in the repetition of a simple message.[17] Twitter is the perfect medium for this. Classic political speeches are different from that. His style of speeches does however reflect this. Other researchers found Trump to use shorter sentences, shorter words and have a lower

[13] Lemmerich (2020)

[14] Herz, J.; Bellaachia, A. (2014): *The Authorship of Audacity: Data Mining and Stylometric Analysis of Barack Obama Speeches*, in: Stahlbock, R., Weiss, G. M., Abou-Nasr, M. & Arabnia, H. R.: *DMIN 2014 : proceedings of the 2014 International Conference on Data Mining*, http://worldcomp-proceedings.com/proc/p2014/DMI8024.pdf (retrieved 27.03.2019)

[15] Airoldi, E. M.; Anderson, A. G.; Fienberg, S. E.; Skinner, K. K. (2006): *Who Wrote Ronald Reagan's Radio Addresses?*, in: Bayesian Analyst, pp. 289-320, https://projecteuclid.org/download/pdf_1/euclid.ba/1340371064 (retrieved 30.06.2020)

[16] Lemmerich (2021a)

[17] Milbank, Dana: *Trump's fake-news presidency*, in: *The Washington Post* (18.11.2016), https://www.washingtonpost.com/opinions/trumps-fake-news-presidency/2016/11/18/72cc7b14-ad96-11e6-977a-1030f822fc35_story.html (retrieved 18.03.2021)

lexical density and lower Moving-Average-Type-Token-Ratio than other politicians.[18] His Part of Speech (POS) usage was also very distinct: he had the highest percentage of verbs and adverbs in comparison to other presidential candidates.[19]

3.3. Does Complex or Simple Rhetoric Win Elections?[20]

This paper by Conway et al. includes two studies on the correlation of complexity of a candidate's speech and his success.

The first study is about the 2004 democratic primaries. The study found out that there was no overall difference of complexity between the winners and losers. The average complexity was similar. Over time however the complexity changed. Near the election date the complexity lowered.

The second study dealt with the 2008 presidential election. This study was done on a smaller scale: Participants were asked their likeliness to vote for either candidate and then presented a paragraph by that candidate of varying complexity. For McCain, more complexity actually increased the favorability of the participants.

This contradicts the "simplicity sells"-view, a common assumption in political psychological theory.

[18] Savoy, Jacques (2018a): *Analysis of the style and the rhetoric of the 2016 US presidential primaries*, in: *Digital Scholarship in the Humanities* 33, pp. 143–159, https://academic.oup.com/dsh/article/33/1/143/2993886 (retrieved 06.03.2021), doi: 10.1093/llc/fqx007

Savoy, Jacques (2018b): *Trump's and Clinton's Style and Rhetoric during the 2016 Presidential Election*, in: *Journal of Quantitative Linguistics* 25, pp. 168–189, doi: 10.1080/09296174.2017.1349358, p. 8

Vrana, Leo; Schneider, Gerold (2017): *Saying Whatever It Takes: Creating and Analyzing Corpora from US Presidential Debate Transcripts* 2017, doi: 10.5167/uzh-145668

[19] Savoy (2018a), p. 150; Savoy (2018b), p. 8

[20] Conway, L. G.; Gornick, L. J.; Burfeind, C.; Mandella, P.; Kuenzli, A.; Houck, S. C.; Fullerton, D. T. (2012): *Does Complex or Simple Rhetoric Win Elections? An Integrative Complexity Analysis of U.S. Presidential Campaigns*, in: *Political Psychology* (33), pp. 599-618, doi:10.1111/j.1467-9221.2012.00910.x

4. Analysis

4.1. Basic Analysis

4.1.1. Type-Token-Ratio (TTR)

Type-Token-Ratio is an indicator of lexical diversity. The lower the TTR is, the less diverse the vocabulary of a corpus is.

The Obama corpus contains 377 texts. It has 1.552.078 tokens and 23.195 types for a type-token-ratio of 1,49%.

The Trump corpus contains 2251 texts. It has 6.070.332 tokens and 32.314 types for a type-token-ratio of 0,54%.

This is much lower than the Obama corpus, but the Trump corpus is also much larger in size, so a direct comparison is not very meaningful. Comparing the Moving-Average Type-Token-Ratio may give better insight into the diversity in vocabulary of the two presidents.

4.1.2. Moving-Average Type-Token-Ratio (MATTR)

Moving-Average Type-Token-Ratio (MATTR) for both corpora was calculated with MATTR2.0 with a window size of 500.[21]

The MATTR of the Obama corpus is 0,504.

The MATTR of the Trump corpus is 0,435.

A lower MATTR means less lexical diversity in a text. The Trump corpus is lower than the Obama corpus here, but not with as big of a margin as the normal TTR was.

4.1.3. Big Words

‘Big words’, as defined by Jacques Savoy, are words that are equal to or longer than 6 letters. A text with a high percentage of ‘big words’ tends to be more complex to understand.

The Obama corpus has 333.000 tokens equal to or longer than 6 letters for a rate of 21,4%.

The Trump corpus has 1.414.612 tokens equal to or longer than 6 letters for a rate of 23,3%.

The 5000-word excerpt from COCA has 194.051.881 tokens equal to or longer than 6 letters for a rate of 23,0%.

These results are very similar to each other, with the Trump corpus even having slightly more ‘big words’ than the Obama corpus. Similar results were found by Savoy in 2018a and 2018b, with the Trump corpus having an even higher percentage of big words at 29%.[22]

This result would indicate a higher complexity for the Trump corpus.

[21] Covington, Michael A; McFall, Joe D: *MATTR 2.0*, Institute for Artificial Intelligence. University of Gerorgia, http://ai1.ai.uga.edu/caspr/ (retrieved 20.03.2021)

[22] Savoy (2018a); Savoy (2018b)

4.1.4. Mean Sentence Length (MSL)

The Stanford tagger tags the end of sentences with _. , which allows to calculate MSL.

The Obama corpus contains 64.213 sentences and 1.552.078 tokens for a mean sentence length of 24,17 tokens per sentence.

The Trump corpus contains 578.255 sentences and 6.070.332 tokens for a mean sentence length of 10,49 tokens per sentence. This differs from the result in Lemmerich 2021a because the Corpus was adjusted. For this thesis I used the `_.` end tag, , which also includes questions and exclamations, and not `._.` , like in Lemmerich 2021a.

These findings also confirm Savoy 2018b.[23] Trump uses much shorter sentences than typical politicians. The difference is even more drastic than found by Savoy. Clinton had a MSL between 18,6 and 20,1. Obama's MSL of 24 is even higher than that.

4.1.5. Parts of Speech (POS)

To calculate the part of speech (POS) distribution, as well as the mean sentence length (MSL) the Stanford tagger[24] with the english-left3words-distsim model[25] was used to tag both corpora. This also tags sentence ends with `_.`, which allows to calculate MSL.

[23] Savoy (2018b), p. 8

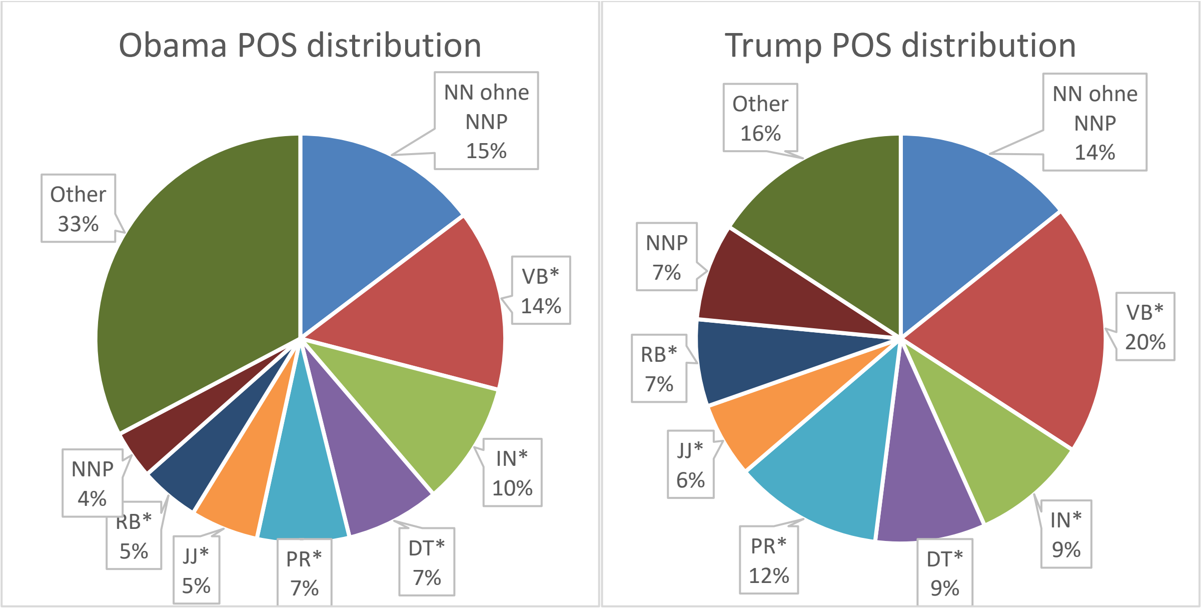
[24] Toutanova, Kristina; Manning, Christopher D. (2020): *Stanford Part-Of-Speech Tagger*, https://nlp.stanford.edu/software/tagger.shtml (retrieved 20.03.2021)

Toutanova, Kristina; Klein, Dan; Manning, Christopher D.; Singer, Yoram: *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*, in: *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics 2003*, pp. 252–259, https://www.aclweb.org/anthology/N03-1033, doi: 10.3115/1117794.1117802

Toutanova, Kristina; Manning, Christopher D. (2000): *Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger, in: Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13* (= EMNLP '00), pp. 63–70, doi: 10.3115/1117794.1117802

[25] english-left3words-distsim.tagger: Trained on WSJ sections 0-18 and extra parser training data using the left3words architecture and includes word shape and distributional similarity features. Penn tagset. UDv2.0 tokenization standard.

Guidelines from Santorini, Beatrice (06.1990): *Part-of-Speech Tagging Guidelines for the Penn Treebank Project*



Obama			
POS	**TAG**	**TOKENS**	**%**
Noun	NN*[26]	228 309	14.71%
Name	NNP	58 934	3.80%
Pronoun	PR*	112 949	7.28%
Adjective	JJ*	83 264	5.36%
Verb	VB*	222 233	14.32%
Adverb	RB*	72 893	4.70%
Determiner	DT*	114 768	7.39%
Preposition/ Conjunction	IN*	150 756	9.71%
Other		507 972	32.73%

Trump			
POS	**TAG**	**TOKENS**	**%**
Noun	NN*	867 506	14.29%
Name	NNP	465 351	7.67%
Pronoun	PR*	711 845	11.73%
Adjective	JJ*	359 068	5.92%
Verb	VB*	1 205 766	19.86%
Adverb	RB*	416 729	6.87%
Determiner	DT*	528 683	8.71%
Preposition/ Conjunction	IN*	554 717	9.14%
Other		960 667	15.83%

Diagrams 1 & 2 & Table 1 & 2: Part of speech distribution for Obama and Trump.

Trump uses more verbs and adverbs than Obama, making it more action oriented. The corpus also contains more pronouns and names, which hints towards his more direct style of communication.

[26] _NN ohne NNP

4.1.6. Speech Length

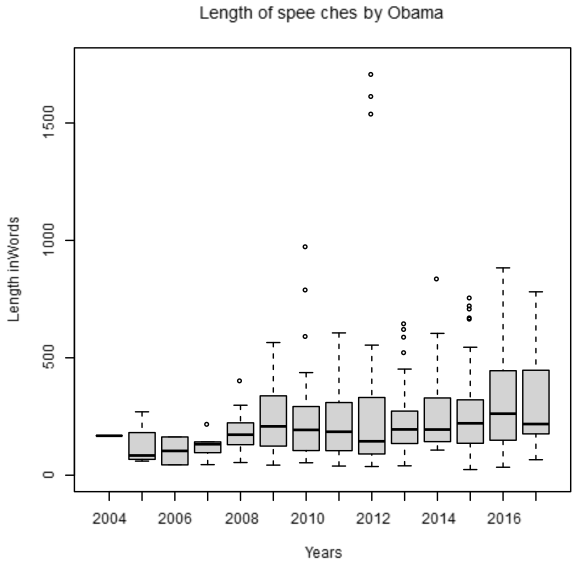
The length of speeches could be very dependent on the occasion of a speech. Usually a longer text would indicate a higher complexity.

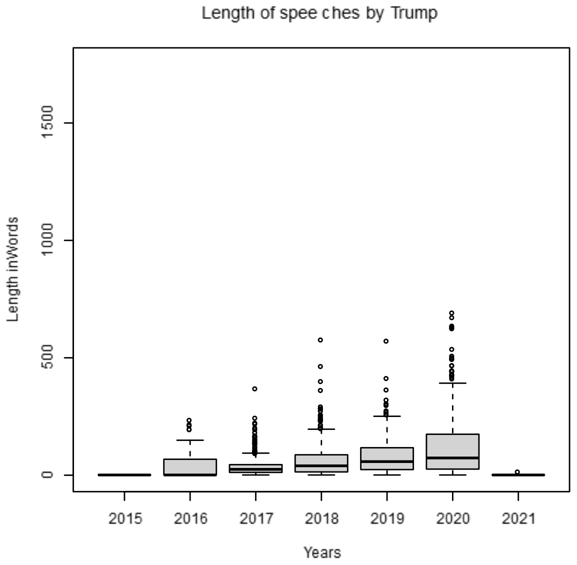
Code

```
# Julian Lemmerich
# Thesis
# Textlängengraphen

## Obama

obama.years <- c(2004:2017)
obama.files <- list()
obama.length <- list()

for (i in 1:length(obama.years)) {
  obama.files[[i]] <- list.files(path=paste0('C:/Users/julian.lemmerich/OneDrive/User Data/Uni/Semester 8/Thesis/Corpora/obama by year/', obama.years[i]), pattern="*.txt", full.names=TRUE, recursive=FALSE)
}

for (j in 1:length(obama.files)) {
  obama.length[[j]] <- list()
  for (h in 1:length(obama.files[[j]])) {
    text <- readLines(file(paste0(obama.files[[j]][h])))
    obama.length[[j]][h] <- length(text)
  }
  obama.length[[j]] <- unlist(obama.length[[j]]) #converts the list into a vector, which makes it readable by the boxplot function
}

boxplot(obama.length, names=obama.years, ylim=c(0, 1750), main="Length of speeches by Obama", ylab="Length in Words", xlab="Years")
boxplot(obama.length, names=obama.years, ylim=c(0, 500), main="Length of speeches by Obama", ylab="Length in Words", xlab="Years")


## Trump

trump.years <- c(2015:2021)
trump.files <- list()
trump.length <- list()

for (i in 1:length(trump.years)) {
  trump.files[[i]] <- list.files(path=paste0('C:/Users/julian.lemmerich/OneDrive/User Data/Uni/Semester 8/Thesis/Corpora/trump 2-3 by year/', trump.years[i]), pattern="*.txt", full.names=TRUE, recursive=FALSE)
}

for (j in 1:length(trump.files)) {
  trump.length[[j]] <- list()
  for (h in 1:length(trump.files[[j]])) {
    text <- readLines(file(paste0(trump.files[[j]][h])))
    trump.length[[j]][h] <- length(text)
  }
  trump.length[[j]] <- unlist(trump.length[[j]]) #converts the list into a vector, which makes it readable by the boxplot function
}

boxplot(trump.length, names=trump.years, ylim=c(0, 1750), main="Length of speeches by Trump", ylab="Length in Words", xlab="Years")
boxplot(trump.length, names=trump.years, ylim=c(0, 500), main="Length of speeches by Obama", ylab="Length in Words", xlab="Years")
```

This code reads all the files from the corpus directory which contains subfolders sorted by year and takes the files length into a list. The list of one year is then again put into a vector to create a boxplot, where each box is a year of texts and the deviation is from the different texts within that year.

Graphs

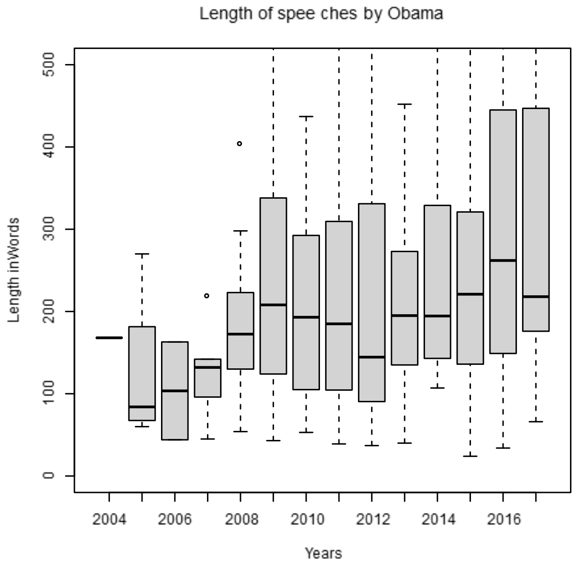


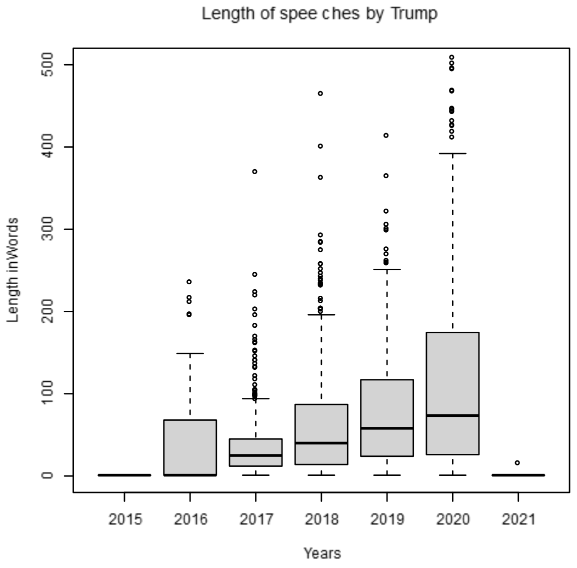


Plot 1: Boxplot of the length of speeches in the Obama corpus.

Plot 2: Boxplot of the length of speeches in the Trump corpus.

Both scaled to fit all of Obama's speeches.





Plot 3: Boxplot of the length of speeches in the Obama corpus.

Plot 4: Boxplot of the length of speeches in the Trump corpus

Both scaled closer for the Trump corpus.

The three longest texts in the Obama corpus, the extremes in the year 2012, are debates with Mitt Romney.

Generally, Obama has much longer speeches. This would indicate more complicated speeches. It could also be from a difference in genre. There are quite a lot of interviews and press conferences in the Trump corpus, that might lead to this drastic difference in text length.

4.2. Word Frequencies

Code: Creating Wordclouds

```
# Julian Lemmerich
# Thesis
# Word cloud creation

library("tm")[27]
library("SnowballC")[28]
library("wordcloud")[29]
library("RColorBrewer")[30]

#read the text
text <- readLines(file.choose())

#reading stopwords from file
stopwords <- readLines(file.choose())

#load the data as a corpus
docs <- Corpus(VectorSource(text))

#cleanup/stopword removal etc
toSpace <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs <- tm_map(docs, toSpace, "/")
docs <- tm_map(docs, toSpace, "@")
docs <- tm_map(docs, toSpace, "\\|")

#convert the text to lower case
docs <- tm_map(docs, content_transformer(tolower))
#remove numbers
docs <- tm_map(docs, removeNumbers)
#remove your own stop word
#specify your stopwords as a character vector
docs <- tm_map(docs, removeWords, stopwords)
#remove punctuations
docs <- tm_map(docs, removePunctuation)
#eliminate extra white spaces
docs <- tm_map(docs, stripWhitespace)
```

[27] Feinerer, Ingo; Hornik, Kurt; Meyer, David (2008): *Text Mining Infrastructure*, in: *R. Journal of Statistical Software* 25(5), pp. 1-54, https://www.jstatsoft.org/v25/i05/

Feinerer, Ingo; Hornik, Kurt (2020): *tm: Text Mining Package*, R package version 0.7-8, https://CRAN.R-project.org/package=tm

[28] Bouchet-Valat, Milan (2020): *SnowballC: Snowball Stemmers Based on the C 'libstemmer' UTF-8 Library*, R package version 0.7.0, https://CRAN.R-project.org/package=SnowballC

[29] Fellows, Ian (2018): *wordcloud: Word Clouds*, R package version 2.6, https://CRAN.R-project.org/package=wordcloud

[30] Neuwirth, Erich (2014): *RColorBrewer: ColorBrewer Palettes*, R package version 1.1-2, https://CRAN.R-project.org/package=RColorBrewer

```
wordcloud(docs
          , scale=c(5,0.5)     #set min and max scale
          , max.words=100      #set top n words
          , random.order=FALSE #words in decreasing freq
          , rot.per=0.35       #% of vertical words
          , use.r.layout=FALSE #use C++ collision detection
          , colors=brewer.pal(8, "Dark2"))
```

This code reads a single text file into R, cleans up the corpus and then creates a wordcloud with the wordcloud package.[31]

Code: Combining Multiple Text Files into One

To combine all the text files that make up the corpus into one large text file for analysis, this Powershell oneliner can be used:

```
Get-ChildItem "C:\---\Corpora\trump 2-3\corpus\*" -include *.txt | Get-
  Content -encoding UTF8 | out-file -Encoding UTF8 "C:\---\Corpora\trump 2-
  3\combined.txt"
```

COCA Stopword Filtering

Since COCA was not acquired as a whole corpus but only the word frequency tables, I could not use Antconc to filter stopwords. Instead, I used a script to clean them up. Since the data is delivered in xslx format I decided to use Visual Basic to remove the stopwords right in Excel.

```
Function IsInArray(ByVal VarToBeFound As Variant, ByVal Arr As Variant) As Boo
lean
    Dim Element As Variant
    For Each Element In Arr
        If Element = VarToBeFound Then
            IsInArray = True
            Exit Function
        End If
    Next Element

    IsInArray = False
End Function
```

This assistant function checks if a passed value is contained in a passed array, and if so, returns true.[32]

[31] STHDA: *Text mining and word cloud fundamentals in R: 5 simple steps you should know*, http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know (retrieved 18.07.2021)

Singham, Luke (17.01.2021): *How to Make a Wordcloud Using R*, https://lukesingham.com/how-to-make-a-word-cloud-using-r/ (retrieved 18.07.2021)

[32] JensS (05.09.2017): Answer to *EXCEL VBA compare cell values to an Array*, https://stackoverflow.com/a/46050159/9397749 (09.07.2021)

```
Sub Delete_Lines()

    arr = Range("A11470:A12140").Value
    ' the values here need to be adjusted to where the stopwordlist has been p
asted

    Dim lRow As Long
    Dim iCntr As Long
    lRow = 11465
    For iCntr = lRow To 1 Step -1
        If IsInArray(Cells(iCntr, 2).Value, arr) Then
            'cell.Interior.Color = RGB(0, 176, 80)
            Rows(iCntr).Delete
        End If
    Next
End Sub

' (ln26) Number “3” in the ‘If IsInArray(Cells(iCntr, 3).Value represents the
third column (C) and needs to be adjusted as such
' (ln24) lRow = 1000 means it will check the first 1000 rows.
```

Since this code is only meant to be run once it has not been optimized and made user friendly with the replaceable variables. The stopwordlist needs to be in the same Excel worksheet and the range of the stoplist should be entered for variable `arr`. The `lRow` variable should be set to the number of lines of data. It can be bigger than the amount of data, but should not be less. The commented line of `cell.Interior.Color` allows the cells to be colored instead of deleted to check for mistakes first. [33]

Code: Creating Text for Wordclouds from Word Frequency Table

```
wordfreqtable <- read.csv(file.choose())

text <- c()

for (i in 1:length(wordfreqtable$word.freq)) {
  for (j in 1:(wordfreqtable$word.freq[i]/10000)) {
    text <- c(text, wordfreqtable$word[i])
  }
}
```

Since I do not have the full text from the COCA, which would be needed for the wordcloud creation, I created a “text” placeholder for this. This is probably not the most efficient way to do this, since the wordcloud function internally will again have to create a word frequency table

[33] Scott, Mark (20.07.2015): *Remove All Rows Containing Certain Data*, http://excelzoom.com/remove-all-rows-containing-certain-data/ (retrieved 09.07.2021)

Dennis (05.09.2017): *EXCEL VBA compare cell values to an Array*, https://stackoverflow.com/questions/46049323/excel-vba-compare-cell-values-to-an-array (retrieved 09.07.2021) to calculate the size of the words, but it is the easiest way. The resulting `text` variable from this code snippet can be inserted into the previous wordcloud code snippet instead of `text <- readLines(file.choose())`.

Code: Creating Wordclouds v2

The wordclouds created with the previous code had one issue I only noticed later: Words shorter than 3 characters were simply dropped from the wordcloud.

But it turned out all this code is not really necessary and can be shortened. A vector of words and a vector of frequencies can be passed to the `wordcloud` function instead of a corpus to create the wordcloud. And since I already have csv's of word frequency tables I can simply load them into R and create the wordcloud directly from the word frequencies.

This also made the "creating text from word frequency table" code an unnecessary step. And the frequencies don't need to be given in absolute values but can be given in relative frequency.

```
library("wordcloud")[34]
library("RColorBrewer")[35]

wordfreqtable <- read.csv(file.choose())

wordcloud(wordfreqtable$diffwords, wordfreqtable$diffbetrag
          , scale=c(5,0.5)
          , max.words=100
          , random.order=FALSE #words in decreasing freq
          , use.r.layout=FALSE #use C++ collision detection
          , colors=brewer.pal(8, "Dark2"))
```

4.2.1. Whole Corpus Word Frequencies

Word frequency lists are one of the most common ways to find important words in a corpus. It simply counts the number of words. A number of interesting findings can be discovered with this.

In the COCA corpus I used the word forms not the lemmata, since the MFW lists I have from my own corpora are not lemmatized. This makes the list 11.461 words long.

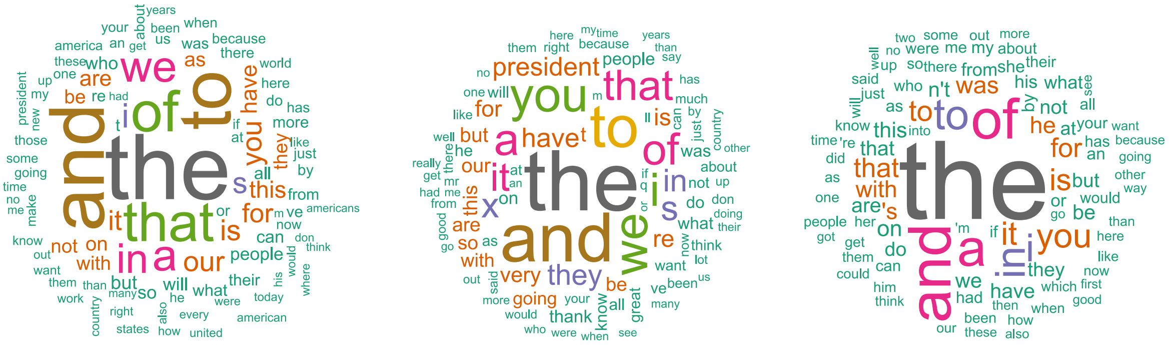
The wordlists I use in the following chapter will display percentages, because it makes comparison much easier. The corpora are very different in size, so displaying absolute frequencies would not lead to useful conclusions.

Differences to COCA can be found in Appendix chapter 9.2.

[34] Fellows, Ian (2018): *wordcloud: Word Clouds*, R package version 2.6, https://CRAN.R-project.org/package=wordcloud

[35] Neuwirth, Erich (2014): *RColorBrewer: ColorBrewer Palettes*, R package version 1.1-2, https://CRAN.R-project.org/package=RColorBrewer

4.2.1.1. Word Frequencies Including Stopwords



Plot 5-7: Wordcloud of the most frequent words of the Obama and Trump corpora and COCA

Obama	
words	**percent**
the	4.4431%
and	3.6625%
to	3.3817%
of	2.6031%
that	2.4726%
a	1.9361%
we	1.9127%
in	1.7435%
i	1.3513%
s	1.1493%
is	1.0909%
our	1.0584%
you	1.0505%
it	1.0228%
for	0.9550%
this	0.8040%
have	0.7427%
are	0.7189%
as	0.6131%
on	0.5987%
with	0.5665%
be	0.5659%
not	0.5595%
they	0.5583%
but	0.5456%
so	0.4979%
who	0.4836%
will	0.4812%
people	0.4714%
can	0.4473%

Trump	
word	**percent**
the	4.1596%
and	3.2952%
to	2.7896%
you	2.2270%
we	2.2197%
i	2.0803%
a	2.0392%
of	1.9651%
that	1.8588%
it	1.7038%
x	1.5044%
s	1.4838%
in	1.2741%
they	1.1417%
have	1.0162%
president	0.9709%
re	0.8780%
is	0.8694%
for	0.8039%
very	0.7337%
but	0.6692%
t	0.6564%
our	0.6516%
this	0.6407%
are	0.6170%
so	0.6118%
be	0.6008%
with	0.5895%
going	0.5316%
on	0.5184%

COCA	
word	**percent**
the	5.0033%
and	2.4778%
of	2.3159%
a	2.1166%
to	1.6258%
in	1.5671%
i	1.4218%
you	1.2053%
it	1.1042%
is	1.0094%
to	0.9233%
that	0.8320%
for	0.8195%
was	0.6849%
he	0.6467%
with	0.6443%
's	0.6304%
on	0.6080%
this	0.5541%
n't	0.5285%
we	0.5181%
be	0.5047%
have	0.5023%
that	0.5003%
are	0.4983%
not	0.4656%
but	0.4523%
they	0.4504%
do	0.4501%
at	0.4024%

all	0.4370%	thank	0.5066%	what	0.3808%
t	0.4211%	people	0.5028%	his	0.3719%
what	0.4014%	was	0.4815%	from	0.3711%
re	0.3845%	what	0.4680%	or	0.3420%
their	0.3813%	do	0.4504%	by	0.3372%
more	0.3744%	he	0.4442%	she	0.3188%
ve	0.3716%	know	0.4331%	my	0.3107%
from	0.3595%	all	0.4189%	an	0.3059%
or	0.3506%	will	0.3962%	as	0.2946%
was	0.3437%	great	0.3840%	had	0.2724%
by	0.3354%	ve	0.3824%	if	0.2710%
do	0.3244%	not	0.3708%	me	0.2639%
there	0.3140%	as	0.3651%	your	0.2578%
us	0.3077%	want	0.3474%	can	0.2516%
at	0.2987%	think	0.3456%	all	0.2504%
just	0.2986%	at	0.3380%	who	0.2493%
has	0.2869%	about	0.3352%	has	0.2444%
here	0.2818%	can	0.2983%	about	0.2428%
because	0.2807%	been	0.2976%	their	0.2417%

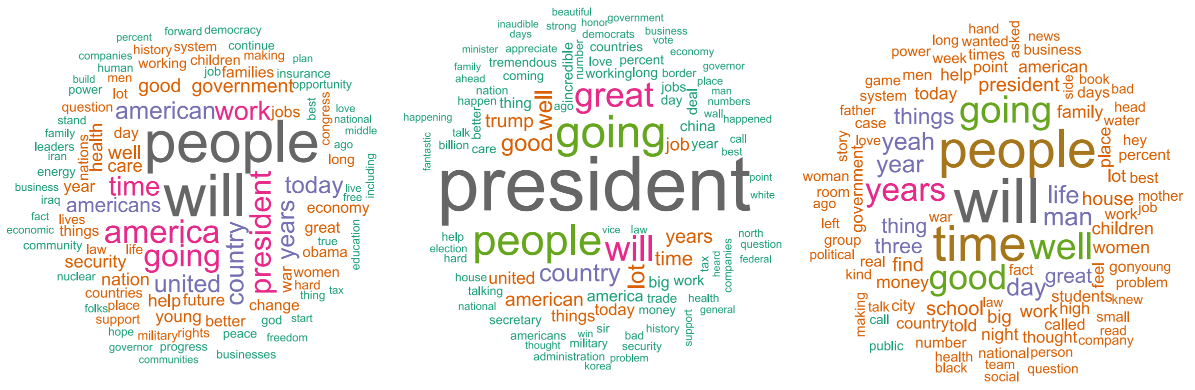
Table 3-5: Relative most frequent words in the Obama and Trump corpora and COCA

Both the Trump and the Obama Corpus have “the”, “and” and “to” at the top of their word frequency list. While “the” and “and” are the two top words in the American language too, “to” is only on position 5. After that though, the Trump corpus and the Obama corpus differentiate quite a bit too.

Trump uses a lot of pronouns, which could already be seen in the part-of-speech distribution in chapter 4.1.5, starting off with “you”, directly speaking to the audience, then “we”, which is only two ranks lower in the Obama corpus but on position 21 in the American language. Politicians like to use “we”, because its ambiguous who is included in the “we”: the government, the president and his team or the people of the nation. “I” is also used more often in the Trump corpus than in the other two corpora. Obama uses “I” about the same amount as in the COCA.

4.2.1.2. Word Frequencies Excluding Stopwords

The full list of stopwords used in this chapter can be seen in appendix 9.3.



Plot 8-10: Wordcloud of the most frequent words of the Obama and Trump corpora and COCA excluding stopwords

Obama	
word	**percent**
will	1.3076%
people	1.2809%
america	0.6514%
going	0.6240%
president	0.5879%
time	0.5201%
work	0.5028%
country	0.4801%
united	0.4545%
years	0.4543%
american	0.4516%
today	0.4399%
americans	0.3570%
well	0.3267%
government	0.3256%
good	0.3252%
care	0.2960%
health	0.2937%
security	0.2917%
nation	0.2899%
help	0.2829%
young	0.2699%
future	0.2564%
war	0.2535%
year	0.2499%
jobs	0.2434%
better	0.2380%
great	0.2335%

Trump	
word	**percent**
president	2.9815%
going	1.6324%
people	1.5440%
will	1.2167%
great	1.1792%
country	0.7462%
well	0.7012%
good	0.6959%
lot	0.6949%
time	0.5452%
years	0.5124%
american	0.4999%
trump	0.4974%
job	0.4511%
united	0.4160%
things	0.3875%
today	0.3818%
america	0.3604%
big	0.3572%
china	0.3159%
thing	0.3156%
work	0.3031%
incredible	0.2873%
deal	0.2852%
year	0.2702%
working	0.2598%
jobs	0.2590%
long	0.2536%

COCA	
word	**percent**
will	0.2154%
people	0.1783%
time	0.1669%
going	0.1218%
well	0.1189%
good	0.1112%
years	0.1032%
man	0.0742%
life	0.0719%
day	0.0716%
yeah	0.0704%
year	0.0698%
things	0.0630%
thing	0.0572%
three	0.0570%
great	0.0552%
school	0.0526%
president	0.0519%
find	0.0512%
house	0.0494%
big	0.0470%
work	0.0456%
women	0.0443%
children	0.0442%
family	0.0440%
money	0.0437%
lot	0.0436%
today	0.0431%

day	0.2320%	day	0.2497%	told	0.0427%
families	0.2320%	percent	0.2479%	night	0.0422%
economy	0.2315%	trade	0.2415%	place	0.0420%
change	0.2308%	countries	0.2375%	help	0.0412%
things	0.2178%	tremendous	0.2272%	american	0.0411%
obama	0.2153%	better	0.2262%	government	0.0396%
nations	0.2064%	coming	0.2232%	thought	0.0390%
children	0.2061%	care	0.2207%	students	0.0383%
women	0.2048%	secretary	0.2203%	high	0.0383%
countries	0.2010%	love	0.2183%	feel	0.0383%
working	0.1985%	sir	0.2137%	country	0.0377%
lot	0.1978%	money	0.2037%	point	0.0376%
long	0.1972%	americans	0.1962%	city	0.0374%
system	0.1873%	talking	0.1953%	called	0.0365%
question	0.1868%	military	0.1927%	percent	0.0357%
law	0.1857%	billion	0.1873%	work	0.0357%
life	0.1857%	nation	0.1839%	gon	0.0354%
place	0.1837%	tax	0.1836%	days	0.0353%
lives	0.1803%	border	0.1834%	times	0.0349%
support	0.1794%	number	0.1820%	men	0.0348%
congress	0.1787%	help	0.1816%	real	0.0348%

Table 6-8: Relative most frequent words in the Obama and Trump corpora and COCA excluding stopwords

The top two words used in the Obama corpus, “will” and “people”, which I interpreted as forward looking in my paper about the creation of the Obama corpus,[36] are actually the two most often used words in COCA as well, meaning this is not specific to Obama, but normal in the American language.

“president” is very prominent in the Trump corpus, but not necessarily because Trump himself uses it. Instead, looking at the concordances shows, that often reporters repeatedly start their sentences with “Mr. president”. It is also paratext to mark when the president starts speaking in interviews or in press conferences.

“great” is one of Trump’s very distinct words. He used it 5 times more often than Obama. It appears in many clusters that are distinct to him, like “great job”, his election slogan “Make America Great Again” and “great country”.

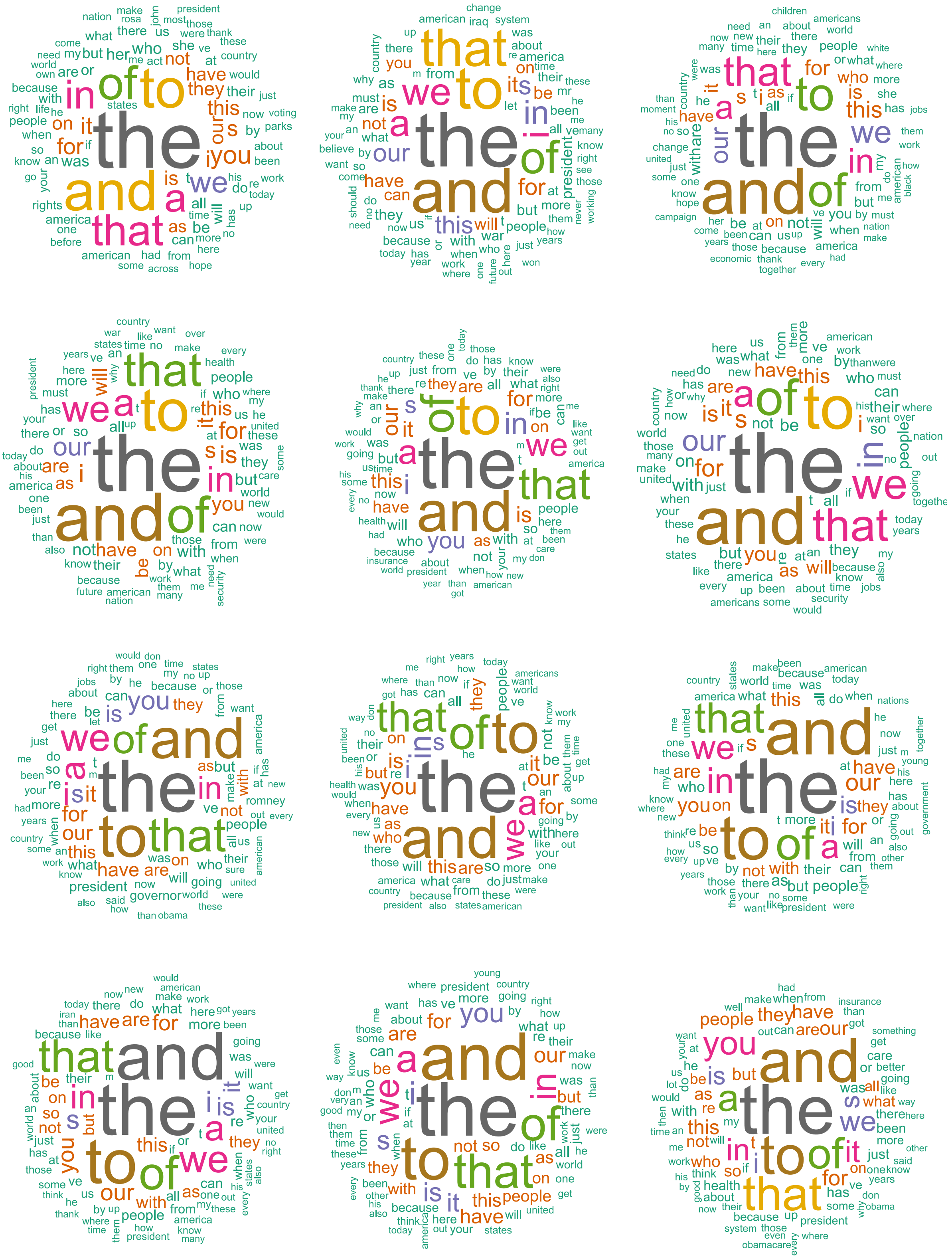
“health” and “care” are very frequent words for Obama, but they don’t appear in these top 50 from the Trump corpus.

Both presidents, as well as the COCA, though not as much, have a lot of different variations of “America” in their corpus. This shows the American patriotism and pride in their country. Obama does not mention any more countries, Trump mentions “china”, one of the biggest scapegoats and global opponents of his presidency.

[36] Lemmerich (2020), p. 13

4.2.2. Word Frequencies by Year

Obama unfiltered

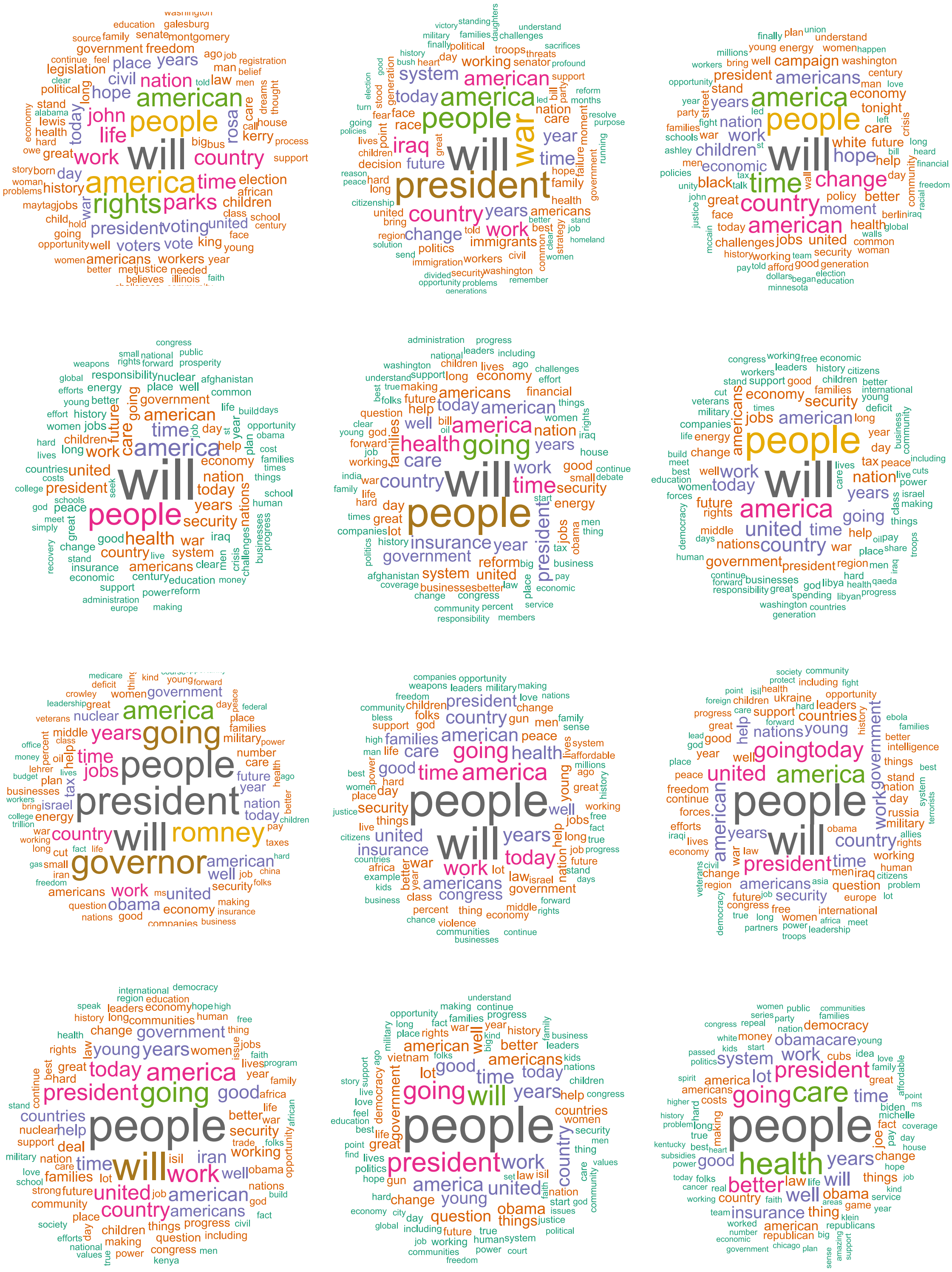


Plots 11-22: Wordcloud from 2004 to 2017 of the Obama corpus without filtering stopwords (2004 to 2006 have been condensed into one wordcloud due to corpus size of these years)

The full tables can be seen in the file-attachments 10.

The dominating words don’t change much, when no stopword filtering is used. There is variation, but none of significance. The more interesting words appear, when filtering stopwords.

Obama filtered



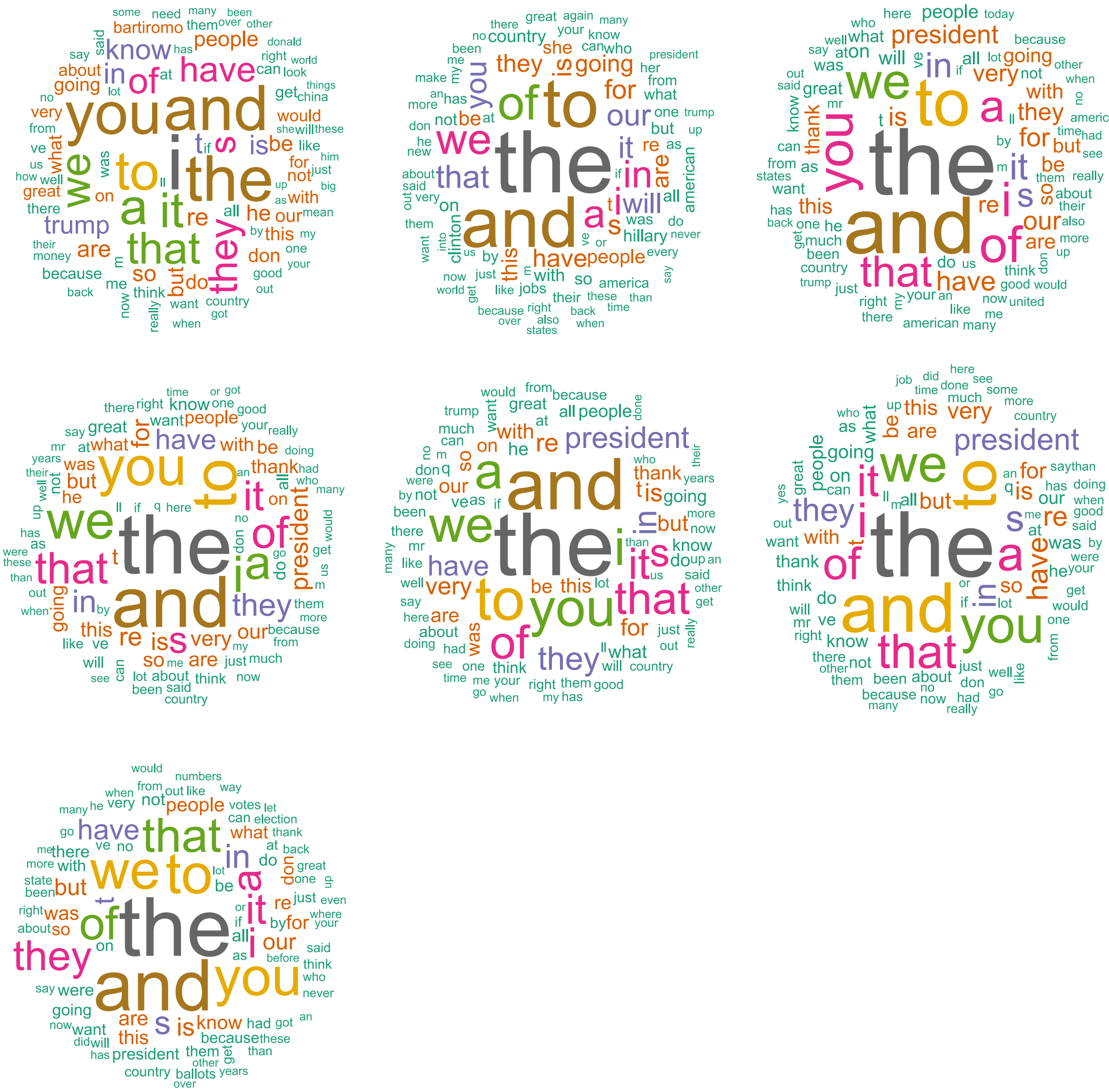
Plots 23-34: Wordcloud from 2004 to 2017 of the Obama corpus excluding stopwords (2004 to 2006 have been condensed into one wordcloud due to corpus size of these years)

A table of the top 30 most frequent words for each year can be seen in appendix 9.4.1. The full tables can be seen in file-attachments 10.

The development over time with the Obama corpus shows that “people” grows in importance. In 2012, during the presidential election campaigns, “governor” and “president” gains importance. In 2017 “health” and “care” are the second and third most often used words.

In 2012 his opponent “Romney” gets mentioned a lot. But in 2008 there is no similar occurrence of his opponent.

Trump unfiltered



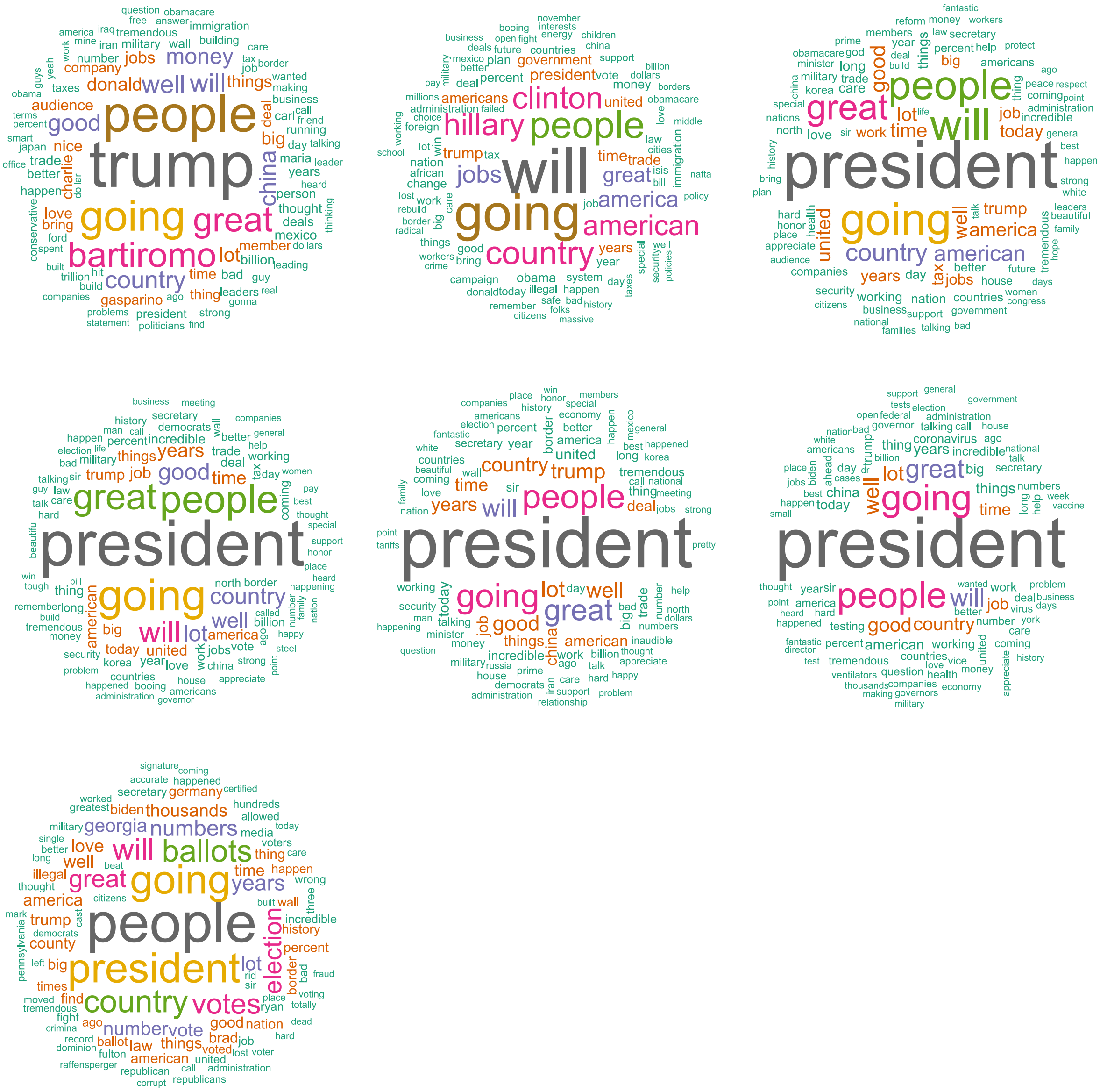
Plots 35-41: Wordcloud from 2015 to 2021 of the Trump corpus without filtering stopwords

The full tables can be seen in the file-attachments 10.

The most noticeable here is, that “I” is the most used word in 2015 in the Trump corpus. There are only two texts from 2015 in this corpus, which can influence this result a lot. One of them is the *Announcement for Candidacy* in New York.

The rest of the years have a similarly hardly changing word set of “the”, “and”, “to”, “you” and “we”. The more interesting words show when filtering stopwords.

Trump filtered



Plots 42-48: Wordcloud from 2015 to 2021 of the Trump corpus excluding stopwords

A table of the top 30 most frequent words for each year can be seen in appendix 9.4.2. The full tables can be seen in file-attachments 10.

In 2015 and 2021 “people” is at the top of the frequency list, which is position three in the overall most frequent words, and the most frequent word of the COCA.

Like in the overall corpus, “president” is the most common word between 2017 and 2020.

“Trump” is the most frequent word in 2015. The reason for this is most probably paratextual. One of the two texts in this corpus from 2015 is an interview. Every time Trump speaks, it is led by “TRUMP:”. As already discussed, it is hard to filter these paratextual entries from the corpus, so they show up here. “Bartiromo” in pink is the Interviewer and shows up here for a very similar reason.

In 2016 “Hillary Clinton”, Trump’s main opponent in the presidential race, is among the most frequent words. A similar occurrence happens with Obama in 2012.

The 2021 wordcloud is the most interesting. Like 2015 there are not many texts in the 2021 corpus, but 10 more than 2015. Many words are related to the election. By 2021 the results of the 2020 election were known: Trump had lost. He did however not want to accept the results. Trump tried to overturn the results in the state of Georgia, which is the 14th most frequent word. “votes”, “election”, “ballots” are all very distinct for the Trump corpus of that year.

4.3. Readability

To make the readability easier to compare I will use the Flesch-Kincaid readability test. In Lemmerich 2021a I already compared a few parameters for readability. Flesch-Kincaid is not hugely different to that, but it uses a formula to standardize these scores.[37]

The formula for the Flesch reading ease is:

$$206{,}835 - 1{,}015\left(\frac{total\ words}{total\ sentences}\right) - 84{,}6\left(\frac{total\ syllables}{total\ words}\right)$$

The calculation of a grade is:

$$0{,}39\left(\frac{total\ words}{total\ sentences}\right) + 11{,}8\left(\frac{total\ syllables}{total\ words}\right) - 15{,}59$$

I will use the python library textstat to calculate these scores.[38]

Code

```
import textstat
import os
import csv

## Obama

file = open('C:\\Users\\julian.lemmerich\\OneDrive\\User Data\\Uni\\Semester 8
\\Thesis\\Corpora\\obama\\combined.txt', 'r', encoding='utf8')
obamatext = file.read()

print("Obama")
print("ease: " + str(textstat.flesch_reading_ease(obamatext)))
print("grade: " + str(textstat.flesch_kincaid_grade(obamatext)))

obamanames = []
obamaeasescores = []
obamagradescores = []

for filename in os.listdir('C:\\Users\\julian.lemmerich\\OneDrive\\User Data\\
Uni\\Semester 8\\Thesis\\Corpora\\obama\\corpus\\'):
    file = open(str('C:\\Users\\julian.lemmerich\\OneDrive\\User Data\\Uni\\Se
mester 8\\Thesis\\Corpora\\obama\\corpus\\' + filename), 'r', encoding='utf8')
    text = file.read()

    obamanames.append(filename)
```

[37] Kincaid, J.P.; Fishburne, R.P.; Rogers, R.L.; Chissom, B.S. (1975): *Derivation of new readability formulas (automated readability index, fog count, and flesch reading ease formula) for Navy enlisted personnel*, in: *Research Branch Report*, pp. 8–75, Chief of Naval Technical Training: Naval Air Station Memphis

[38] Bansal, Shivam; Aggarwal, Chaitanya: *Textstat*, Python package version 0.7.2, https://pypi.org/project/textstat/ or https://github.com/shivam5992/textstat

```
    obamaeasescores.append(textstat.flesch_reading_ease(text))
    obamagradescores.append(textstat.flesch_kincaid_grade(text))

obamaframetemp = [obamanames, obamaeasescores, obamagradescores]
obamaframe = zip(*obamaframetemp) #flips the table from rows to columns
with open('C:\\Users\\julian.lemmerich\\OneDrive\\User Data\\Uni\\Semester 8\\
Thesis\\Data\\readability_kincaid_obama.csv', "w", newline='') as f:
    writer = csv.writer(f)
    writer.writerows(obamaframe)

## Trump

file = open('C:\\Users\\julian.lemmerich\\OneDrive\\User Data\\Uni\\Semester 8
\\Thesis\\Corpora\\trump 2-3\\combined.txt', 'r', encoding='utf8')
trumptext = file.read()

print()
print("Trump")
print("ease: " + str(textstat.flesch_reading_ease(trumptext)))
print("grade: " + str(textstat.flesch_kincaid_grade(trumptext)))

trumpnames = []
trumpeasescores = []
trumpgradescores = []

for filename in os.listdir('C:\\Users\\julian.lemmerich\\OneDrive\\User Data\\
Uni\\Semester 8\\Thesis\\Corpora\\trump 2-3\\corpus\\'):
    file = open(str('C:\\Users\\julian.lemmerich\\OneDrive\\User Data\\Uni\\Se
mester 8\\Thesis\\Corpora\\trump 2-
3\\corpus\\' + filename), 'r', encoding='utf8')
    text = file.read()

    trumpnames.append(filename)
    trumpeasescores.append(textstat.flesch_reading_ease(text))
    trumpgradescores.append(textstat.flesch_kincaid_grade(text))

trumpframetemp = [trumpnames, trumpeasescores, trumpgradescores]
trumpframe = zip(*trumpframetemp) #flips the table from rows to columns
with open('C:\\Users\\julian.lemmerich\\OneDrive\\User Data\\Uni\\Semester 8\\
Thesis\\Data\\readability_kincaid_trump.csv', "w", newline='') as f:
    writer = csv.writer(f)
    writer.writerows(trumpframe)
```

4.3.1. Full Corpora Readability

The Obama corpus gets a reading ease score of -2,76 which equates to a reading grade of 36,0. The 36th grade of course does not exist but would indicate a reader's age of 41. This is not realistic, as it would mean the average master's graduate would not be able to understand Obama's speeches. While they may be more complex to understand than those of his successor Trump, they are not complex on this level.

The Trump corpus gets a reading ease score of 74,49 which equates to a reading grade of 6,3. This affirms what many other values have already suggested: Trump's speeches are easier to understand and appeal to a wider audience. While this is unusual for a politician, it is part of the appeal for many of Trump's supporters.

4.3.2. Single Speeches Readability

Obama hardest to read

Speech	**Ease**	**Grade**
2010_01_First Presidential State of the Union Speech.txt	-6879.58	2690.7
2009_02_State of the Nation Address to Congress Speech.txt	-5522.52	2169.3
2009_09_Joint Session of Congress Heath Care Speech.txt	-5242.38	2061.6
2009_09_United Nations 64th Session General Assembly Speech.txt	-4811.35	1893.9
2009_12_Nobel Prize for Peace Speech and Lecture.txt	-3951.3	1565.6
2009_05_Notre Dame University Commencement Speech.txt	-3312.87	1320.2
2016_06_United State of Women White House Summit Address.txt	-3248.93	1283.3
2016_06_Counter-ISIL Meeting Update Briefing.txt	-2984.36	1179.5
2013_09_United Nations 68th Session General Assembly Speech.txt	-2745.67	1085.7
2008_07_Speech to the People of Berlin.txt	-2732.96	1099.5

Table 9: Hardest to read speeches in the Obama corpus, calculated with Flesch-Kincaid

These scores are even higher than the average. The *First Presidential State of the Union Speech* from 2010 has a reading grade of 2690, which would equate to a reader's age of nearly 2700 years old. This is of course not a realistic score. (see 4.3.3 conclusion)

But there is no general pattern in these speeches. There are speeches to different audiences, for example to other politicians, to university students and to the general public in Berlin.

Obama easiest to understand

2009_05_White House Correspondents Dinner Speech.txt	74.42	8.4
2008_01_Ebenezer Baptist Church Speech.txt	73.92	8.6
2009_09_Back-to-School Speech to America's Schoolchildren.txt	73.1	8.9
2011_04_White House Correspondents Dinner Speech.txt	71.14	7.6
2011_05_Commencement Speech at Booker T. Washington HS.txt	70.94	7.6
2017_01_Farewell Remarks at Andrews to Staff and Supporters.txt	68.44	10.7
2016_11_Remarks on the U.S. Presidential Election Outcome.txt	67.69	8.9
2011_05_Speech to Troops at Fort Campbell.txt	66.57	9.3
2011_10_Speech on the Death of Muammar Qaddafi.txt	66.57	9.3
2010_04_Eulogy for Upper Big Branch Miners.txt	66.37	9.4

Table 10: Easiest to read speeches in the Obama corpus, calculated with Flesch-Kincaid

These scores are also interesting. The easiest speeches in the Obama corpus have a higher reading grade than the average speech in the Trump corpus.

The White House Correspondents Dinner speeches are mostly humorous, which can explain the low score. The speech to school children makes sense to be written in a way that makes it easier to understand.

These speeches are mostly addressed to the public, some to the military.

Trump hardest to read

2016-08-30_remarks-the-xfinity-arena-everett-washington_ascii.txt	20.35	25
2020-07-08_joint-declaration-president-trump-and-president-andres-manuel-lopez-obrador-mexico_ascii.txt	26.27	18.6
2020-10-30_excerpts-from-president-donald-j-trumps-remarks-make-america-great-again-peaceful-protest_ascii.txt	26.41	20.6
2020-05-01_trump-campaign-honors-asian-pacific-american-heritage-month_ascii.txt	28.17	15.8
2020-10-13_excerpts-from-president-donald-j-trumps-remarks-tonights-make-america-great-again-rally_ascii.txt	30.91	20.9
2016-08-08_remarks-the-detroit-economic-club-1_ascii.txt	32.94	20.2
2017-11-09_remarks-members-the-press-with-president-xi-jinping-china-beijing-china_ascii.txt	34.7	15.4
2017-01-28_the-presidents-weekly-address-163_ascii.txt	35.68	19.1

2018-03-10_the-presidents-weekly-address-442_ascii.txt	35.71	15
2017-06-02_the-presidents-weekly-address-421_ascii.txt	35.88	19

Table 11: Hardest to read speeches in the Trump corpus, calculated with Flesch-Kincaid

All of these speeches are addressed to the public, like Obama's "easiest to understand" speeches. This is different from Obama's "hardest to read" speeches, which are also addressed to university students or other politicians.

Trump easiest to read

2021-01-20_remarks-reporters-prior-departure-for-palm-beach-florida_ascii.txt	96.99	1.8
2021-01-06_videotaped-remarks-during-the-insurrection-the-united-states-capitol_ascii.txt	96.69	1.9
2019-11-28_remarks-during-engagement-with-united-states-troops-bagram-airfield-afghanistan_ascii.txt	95.27	2.4
2018-12-15_remarks-the-congressional-ball_ascii.txt	93.44	3.1
2020-05-09_remarks-reporters-during-meeting-with-senior-military-leadership-and-members-the-national_ascii.txt	91.71	3.8
2020-09-18_remarks-exchange-with-reporters-bemidji-minnesota_ascii.txt	89.65	2.5
2018-05-04_remarks-exchange-with-reporters-aboard-air-force-one-while-en-route-dallas-texas_ascii.txt	88.84	2.8
2019-10-17_remarks-exchange-with-reporters-during-tour-the-louis-vuitton-rochambeau-ranch-keene-texas_ascii.txt	88.33	3
2017-08-29_remarks-annaville-fire-station-5-corpus-christi-texas_ascii.txt	88.13	3.1
2019-04-11_remarks-meeting-with-world-war-ii-veterans-and-exchange-with-reporters_ascii.txt	87.92	3.2

Table 12: Easiest to read speeches in the Trump corpus, calculated with Flesch-Kincaid

Many of the "easier to read"-speeches seem to be addressed to the military, like with Obama.

Interestingly, the Trump texts have an overall much smaller span in the Kincaid score.

4.3.3. Readability: Conclusion

The Flesch-Kincaid-Readability does not give useful results for Obama. The scores for Trump are in a more realistic realm, but given the unrealistic Obama scores, their meaningfulness may be doubtful for political speeches. Other factors might be more useful to consider, when deciding about the readability of a text, like "big words" by Savoy, part of speech distribution, lexical density and Moving-Average Type-Token-Ratio.

4.4. Topic Modeling

Topic modeling is a method of clustering documents into groups by discovering overarching themes between these documents.[39]

The goal of topic modeling in this Thesis is to create subcorpora that can be analyzed and compared against each other with other methods. To create the topics I decided to use the *Dariah Topics Explorer*[40] because of the ease of use and my familiarity with the software.

The base stopwordlist (v1) and explanations for the extended stopwordlists v2 and v3 can be seen in appendix 9.3.

First Try with Stopwords v2

Topic 1	going, it's, people, we're, president, that's, they're, great, lot, i'm, good, country, years, he's, well
Topic 2	president, we're, it's, going, people, that's, well, lot, they're, good, great, i'm, job, will, you're
Topic 3	president, trump, it's, will, united, we're, well, going, great, people, trade, lot, good, deal, china
Topic 4	will, american, jobs, country, america, going, people, tax, great, united, years, americans, workers, percent, companies
Topic 5	great, people, will, today, president, job, incredible, that's, american, years, honor, love, nation, america, god

Table 13: Topic modeling topics in the Trump corpus. Filtered stopwords v2 and 400 iterations

It is clear, that still too many words end up in the topics, that I do not want there, because they don't carry enough meaning to form a topic, even with the extended stopwordlist. So I created another iteration of stopwords: version 3.

[39] Blei, David M. (2012): *Probabilistic topic models*, in: *Commun. ACM* 55 (4), pp. 77–84, doi: 10.1145/2133806.2133826

[40] Simmler, Severin; Vitt, Thorsten; Pielström, Steffen (2019): *Topic Modeling with Interactive Visualizations in a GUI Tool*, in: *Proceedings of the Digital Humanities Conference*, https://dev.clariah.nl/files/dh2019/boa/0637.html

Second Try with Stopwords v3

Topic 1	going, people, president, well, it's, obama, good, we're, lot, health, care, i'm, things, question, work
Topic 2	people, united, will, young, america, countries, rights, nations, change, democracy, country, today, future, freedom, history
Topic 3	day, that's, time, americans, god, american, country, families, today, life, years, will, men, love, lives
Topic 4	will, jobs, economy, america, american, energy, americans, that's, year, years, businesses, work, companies, tax, country
Topic 5	will, security, war, people, military, american, united, nuclear, iraq, international, forces, iran, government, afghanistan, troops

Table 14: Topic modeling topics in the Obama corpus. Filtered stopwords v3 and 400 iterations

Topic 1	going, people, president, great, lot, country, well, good, thing, border, years, wall, democrats, time, things
Topic 2	president, going, people, well, lot, good, great, will, job, things, time, secretary, working, country, work
Topic 3	president, trump, will, going, well, united, people, great, lot, deal, good, trade, china, minister, korea
Topic 4	will, going, american, great, jobs, country, people, tax, america, years, time, percent, companies, big, trade
Topic 5	great, will, today, american, people, america, nation, incredible, job, god, honor, years, love, day, president

Table 15: Topic modeling topics in the Trump corpus. Filtered stopwords v3 and 400 iterations

Topic 1	going, people, will, great, country, years, jobs, american, lot, tax, good, time, america, percent, big
Topic 2	president, going, people, well, lot, great, good, will, job, things, secretary, country, time, working, work
Topic 3	president, going, trump, well, people, will, great, lot, good, deal, united, china, trade, things, time
Topic 4	will, people, america, united, american, work, today, that's, it's, future, time, government, security, americans, nations
Topic 5	great, president, people, today, will, american, job, years, good, love, god, incredible, day, honor, time

Table 16: Topic modeling topics in both corpora combined. Filtered stopwords v3 and 400 iterations

Third Try with 100 mfw as Stopwords

The *Dariah Topics Explorer* also has the option to exclude the most frequent words of the corpus as the stopwords. As a baseline to compare my stopwordlists against I also used this option.

Topic 1	jobs, got, back, tax, never, way, she, make, america, care, over, we've, percent, ever, you're
Topic 2	trump, deal, we'll, look, china, did, united, things, yes, trade, we've, he's, done, him, something
Topic 3	yes, we've, done, job, you're, we'll, secretary, things, back, working, where, please, over, work, could
Topic 4	united, world, america, those, new, make, work, that’s, it’s, must, security, its, today, together, government
Topic 5	his, today, every, first, job, day, god, love, come, honor, nation, incredible, never, america, those

Table 17: Topic modeling topics in both corpora combined. Filtered 100 mfw as stopwords and 400 iterations

Topic 1	sure, good, then, well, i’m, very, obama, go, things, lot, it's, everybody, we’ve, don’t, question
Topic 2	she, day, her, families, god, lives, nation, men, him, home, life, women, never, after, made
Topic 3	own, rights, together, must, change, countries, believe, young, future, see, human, democracy, progress, freedom, come
Topic 4	health, care, jobs, economy, insurance, businesses, companies, year, energy, tax, governor, plan, system, workers, why
Topic 5	security, war, nuclear, military, its, iraq, must, international, iran, forces, against, government, nations, including, support

Table 18: Topic modeling topics in the Obama corpus. Filtered 100 mfw as stopwords and 400 iterations

Topic 1	got, did, he's, never, you're, way, look, back, didn't, tell, him, how, big, thing, ever
Topic 2	yes, job, we'll, you're, things, secretary, these, also, over, back, please, how, where, could, working
Topic 3	united, states, deal, trade, china, we'll, also, yes, minister, korea, countries, things, prime, look, u.s
Topic 4	jobs, america, tax, new, border, make, states, also, every, united, americans, these, back, into, again
Topic 5	today, his, first, honor, incredible, america, also, every, job, world, nation, god, day, love, united

Table 19: Topic modeling topics in the Trump corpus. Filtered 100 mfw as stopwords and 400 iterations

10.000 Iterations

Since the corpora are not that big the number of iterations can be increased drastically, for example 10000.

Topic 1	will, america, today, country, time, day, life, americans, god, american, years, people, children, men, lives
Topic 2	going, president, people, it’s, obama, i’m, well, we’re, good, lot, things, question, that’s, don’t, work
Topic 3	will, health, jobs, care, economy, america, insurance, american, businesses, work, americans, year, time, people, years
Topic 4	people, united, will, nations, countries, peace, human, young, future, rights, america, president, progress, democracy, work
Topic 5	will, security, war, military, american, iraq, united, forces, iran, nuclear, people, afghanistan, troops, america, we’re

Table 20: Topic modeling topics in both corpora combined. Filtered stopwords v3 and 10.000 iterations

The topics from these different settings were not convincing to create subcorpora from, so I will not be pursuing this direction further in this paper.

4.5. Sentiment Analysis

When creating the Obama speech corpus I noticed that there are many positive words in the most frequent words list (mfw). I used the Obama speech corpus and the Trump speech corpus to analyze and compare emotions in the speeches of these two presidents in a second paper in 2021. This chapter is an extension of the previous research.[41]

4.5.1. Previous Research

"Trump uses an informal, direct, and provoking communication style to construct and reinforce the concept of a homogeneous people and a homeland threatened by the dangerous other."[42] This informal stlye of communications can be observed through different finds of digital text analysis.

The main focus of a paper by Dilai, Onukevych and Dilay[43] was the creation of a Ukrainian speech sentiment dictionary from speeches by the then president of the Ukraine Petro Proschenko. A corpus of speeches by Donald Trump was used to compare. The analysis of Trump's texts showed that he uses many positive words. Both presidents used mostly positive words, but in comparison to Poroschenko, Trump used a lot more.

Jacques Savoy found a noticeable negativity in Trump's speeches, but his analysis was not of quantitative but of qualitative nature. Trump used more negative words than Hillary Clinton in the presidential debates for the 2016 election.[44]

The corpus of Liu and Lei[45] consisted of pre-written speeches, not transcripts of live speeches, of both 2016 presidential candidates, Donald Trump and Hillary Clinton. This choice could have an effect on the results, since Trump often deviated from the written speeches and rather spoke freely. The results showed that Trump has more negative sentences than his opponent, but also used overall more emotional sentences than Hillary Clinton.

The results of Lemmerich 2021b show, that the corpus of Trump is much more positive than Obama's, even though he was known for his demagogic politics. But this positivity in his speeches may explain why many people thought of him as a good president. Sentiment analysis cannot find nuances, exaggerations or lies in speech. But there are also different ways to describe politics and Trump often described his politics from the most positive side.

My paper also shows that Trump speaks much more emotional than Obama. Both in the positive as well as negative direction his sentiment values are bigger. This fits Trump's speech characteristic as an informal, impulsive and provocative speaker.

[41] Lemmerich, Julian (2021b): *Sentimentanalyse. Barack Obamas und Donald Trumps Reden im Vergleich*, Technische Universität Darmstadt, unpublished manuscript

[42] Kreis, Ramona (2017): *Right-Wing Populism in Europe & USA*, in: *JLP 16*, https://www.jbe-platform.com/content/journals/10.1075/jlp.17032.kre (retrieved 22.03.2021), cited in Liu & Lei (2018)

[43] Dilai, Marianna; Onukevych, Yuliya & Dilay, Iryna (2018): *Sentiment analysis of the US and Ukrainian presidential speeches*, http://ena.lp.edu.ua:8080/handle/ntb/42572 (retrieved 03.03.2021)

[44] Savoy (2018a)

[45] Liu, Dilin; Lei, Lei (2018): *The appeal to political sentiment: An analysis of Donald Trump's and Hillary Clinton's speech themes and discourse strategies in the 2016 US presidential election*, in: *Discourse, Context & Media 25*. p. 143–152, doi: 10.1016/j.dcm.2018.05.001

4.5.2. Tools for Sentiment Analysis

There are different tools for sentiment analysis. The main differentiation is between dictionary-based and machine-learning-based tools.

For dictionary-based-tools a sentiment dictionary is needed. That is a list of words, where every entry has sentiment information saved to it. This sentiment information is usually a number on a scale between -1 and +1. The closer to +1, the more positive the word is, the closer to -1, the more negative. A word with sentiment-information is called a *sentiment bearing word* (sbw).[46] There are also sentiment-dictionaries, where words are categorised in more emotional dimensions than just binary positive and negative. The values of each sbw are then added up for a tally of sentiment of a text.

Machine-learning-tools use annotated training-datasets to determine the sentiment of a text. The downside is the much larger amount of work to create these annotated training-datasets. But the results can be more accurate.[47]

Like in Lemmerich 2021b before, I again decided to use dictionary-based methods, since they have been researched more and are also much easier to realize. There are also many existing sentiment-dictionaries for the English language, for example by Tausczik and Pennebaker,[48] Jockers with the Syuzhet tool[49] and Liu and Lei.[50]

Dictionary-based Tools

Dilai, Onukevych and Dilay[51] use among others *SentiStrength* by the University of Wolverhampton, UK. The tool is free to use for academic research.[52]

Jockers created an R-package with the name *Syuzhet*.[53] One of the big advantages of *Syuzhet* is that it has an integrated dictionary. The analysis in this thesis, like in the previous paper of sentiment analysis, will thus be using *Syuzhet*.

[46] Schmidt, Thomas; Burghardt, Manuel; Dennerlein, Katrin (2018): *Kann man denn auch nicht lachend sehr ernsthaft sein? Zum Einsatz von Sentiment Analyse-Verfahren für die quantitative Untersuchung von Lessings Dramen*, in: *Book of Abstracts, DHd 2018*

[47] Liu & Lei (2018)

[48] Tausczik, Yla R.; Pennebaker, James W. (2010): *The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods*, in: *Journal of Language and Social Psychology 29*, pp. 24–54, doi: 10.1177/0261927X09351676

[49] Jockers, Matthew L. (05.06.2014): *A Novel Method for Detecting Plot*, https://www.matthewjockers.net/2014/06/05/a-novel-method-for-detecting-plot/ (retrieved 07.03.2021).

[50] Liu, Bing; Hu, Minqing; Cheng, Junsheng (2005): *Opinion observer*, in: von,Allan, Ellis: *Proceedings of the 14th international conference on World Wide Web - WWW '05*, p. 342, doi: 10.1145/1060745.1060797

[51] Dilai; Onukevych; Dilay (2018)

[52] Thelwall, M.; Buckley, K.; Paltoglou, G.; Cai, D.; Kappas, A. (2010): *Sentiment strength detection in short informal text*, in: *Journal of the American Society for Information Science and Technology* 61, pp. 2544–2558, http://www.scit.wlv.ac.uk/~cm1993/papers/SentiStrengthPreprint.doc (retrieved 25.03.2021)

[53] Jockers, Matthew L. (2015): *Syuzhet: Extract Sentiment and Plot Arcs from Text*, https://github.com/mjockers/syuzhet (retrieved 26.03.2021)

Problems with Syuzhet

After the release of *Syuzhet*, Annie Swafford released a post with the problems of Syuzhet.[54] These Problems are:

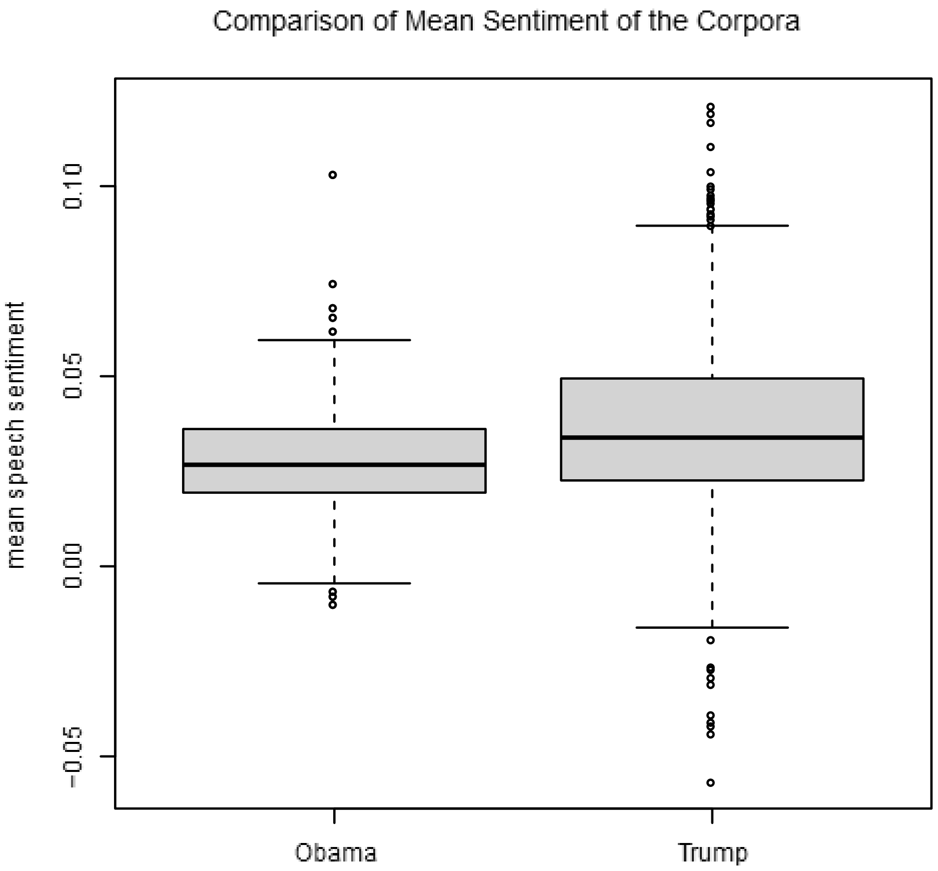
Splitting the text into sentences has problems, especially around quotation marks. It often groups more than one sentence into one “sentence”. This problem does not affect the next chapters of this paper, since I will not be going into such detail on the texts.

The dictionary approach has multiple drawbacks. Each word is scored in isolation, so modifiers and negations have no effect. The sentences “I am not happy” and “I am extremely happy” have the same score. It can also not interpret multiple meanings of the same word. Additionally, a word is counted only once per sentence, so the sentence “I am happy, so happy, so happy” has the same score as “I am happy”. The dictionaries are also missing nuance, only giving scores of -1, 0 or +1. These are issues that will affect the scores in the following chapters of this paper.

[54] Swafford, Annie (02.03.2015): *Problems with the Syuzhet Package*, https://annieswafford.wordpress.com/2015/03/02/syuzhet/ (retrieved 25.08.2021)

4.5.3. Comparing Both Corpora at Full

To compare both corpora at full, I used code from chapter 4.5.4. to generate a boxplot of the mean speech sentiment of both corpora.



Plot 49: Boxplot of the mean sentiment of both corpora compared

This graph shows that Trump has a much higher variance of sentiment in his speeches. Even though Trump's overall mean at 0.034 is higher than Obama's 0.027, his lower outliers are lower than even Obama's lowest speech, and his upper outliers are higher than all but one of Obama's speeches. Obama has a standard deviation of 0,1393 while Trump has a standard deviation of 0,2108.

4.5.4. Sentiment Over Time

Code

```
## Julian Lemmerich
## 02.08.2021
## Thesis sentiment analysis

## Start of Code

library(syuzhet)[55]
library(readr)[56]
library(ggplot2)[57]


# Obama Corpus
setwd("C:/Users/julian.lemmerich/OneDrive/User Data/Uni/Semester 8/Thesis/Corp
ora/obama/corpus")
# Trump Corpus
setwd("C:/Users/julian.lemmerich/OneDrive/User Data/Uni/Semester 8/Thesis/Corp
ora/trump 2-3/corpus")


## Obama Corpus

setwd("C:/Users/julian.lemmerich/OneDrive/User Data/Uni/Semester 8/Thesis/Corp
ora/obama/corpus")

obama.filelist <- list.files()
obama.sumlist <- c() #sum of sentiment values of each text
obama.meanlist <- c() #mean sentiment of each text
obama.sdlist <- c() #standard deviation of each text
obama.veclist <- list() #list of text sentiment vectors


for (i in 1:length(obama.filelist)) {
  t <- read_file(obama.filelist[i])

  poa_word_v <- get_tokens(t, pattern = "\\W")
  syuzhet_vector <- get_sentiment(poa_word_v, method="syuzhet")

  obama.veclist[[i]] <- syuzhet_vector

  obama.sumlist <- c(obama.sumlist, sum(syuzhet_vector))
```

[55] Jockers (2015)

[56] Wickham, Hadley; Hester, Jim (2021): *readr: Read Rectangular Text Data*, R package version 2.0.0, https://CRAN.R-project.org/package=readr

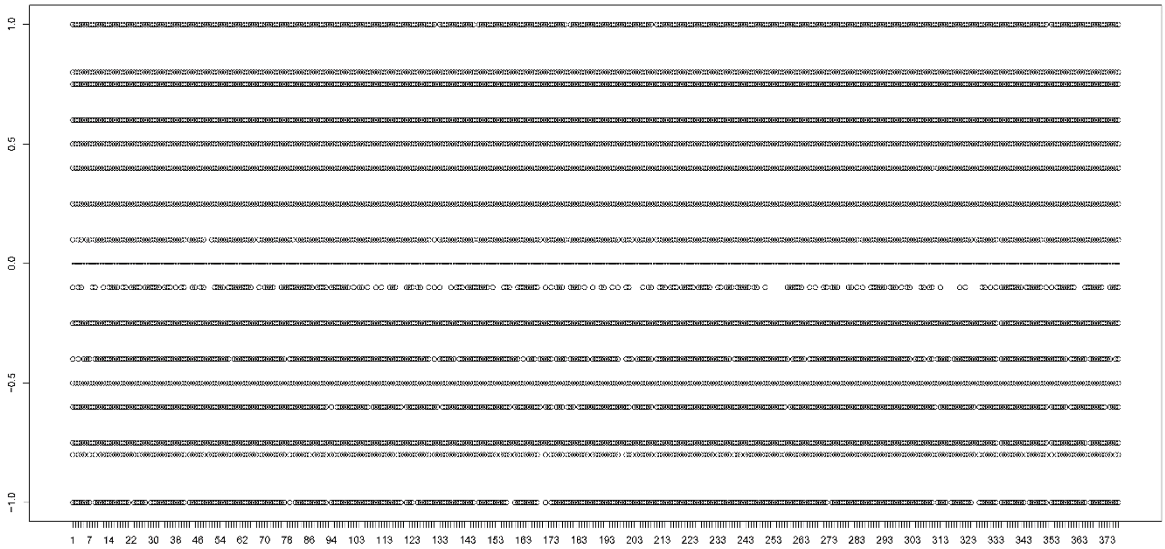
[57] Wickham, Hadley (2016): *ggplot2: Elegant Graphics for Data Analysis*

```
  obama.meanlist <- c(obama.meanlist, mean(syuzhet_vector))
  obama.sdlist <- c(obama.sdlist, sd(syuzhet_vector)) #calculates standard dev
iation
}

obama.sdmean <- mean(obama.sdlist)

oqplot(x=c(1:length(obama.meanlist)), obama.meanlist,
      ylab="Mean Sentiment and Standard Deviation", xlab="Speeches, sorted by 
time"
      )+geom_errorbar(aes(x=x, ymin=obama.meanlist-
obama.sdlist, ymax=obama.meanlist+obama.sdlist), width=0.25)

## Trump

setwd("C:/Users/julian.lemmerich/OneDrive/User Data/Uni/Semester 8/Thesis/Corp
ora/trump 2-3/corpus")

trump.filelist <- list.files()
trump.sumlist <- c() #sum of sentiment values of each text
trump.meanlist <- c() #mean sentiment of each text
trump.sdlist <- c() #standard deviation of each text
trump.veclist <- list() #list of text sentiment vectors


for (i in 1:length(trump.filelist)) {
  t <- read_file(trump.filelist[i])

  poa_word_v <- get_tokens(t, pattern = "\\W")
  syuzhet_vector <- get_sentiment(poa_word_v, method="syuzhet")

  trump.veclist[[i]] <- syuzhet_vector

  trump.sumlist <- c(trump.sumlist, sum(syuzhet_vector))
  trump.meanlist <- c(trump.meanlist, mean(syuzhet_vector))
  trump.sdlist <- c(trump.sdlist, sd(syuzhet_vector))
}

trump.sdmean <- mean(trump.sdlist)

trump.meansd <- sd(trump.meanlist)

x <- c(1:length(trump.meanlist))
qplot(x, trump.meanlist,
      ylab="Mean Sentiment and Standard Deviation", xlab="Speeches, sorted by 
time",
      )+geom_errorbar(aes(x=x, ymin=trump.meanlist-
trump.sdlist, ymax=trump.meanlist+trump.sdlist), width=0.25)
```

```
## Vergleich

sumcomparelist <- list(obama.sumlist, trump.sumlist)
boxplot(sumcomparelist, names=c("Obama", "Trump"), main="Comparison of Sum of Sentiment of the Corpora", ylab="sum of speech sentiment")

meancomparelist <- list(obama.meanlist, trump.meanlist)
boxplot(meancomparelist, names=c("Obama", "Trump"), main="Comparison of Mean Sentiment of the Corpora", ylab="mean speech sentiment")
```

Wrong Graphing Style

At first I wanted to use a boxplot to display the sentiments of individual speeches. But the boxplot is a dysfunctional type of plot for this situation, as can be seen from this example plot for all speeches by Obama.

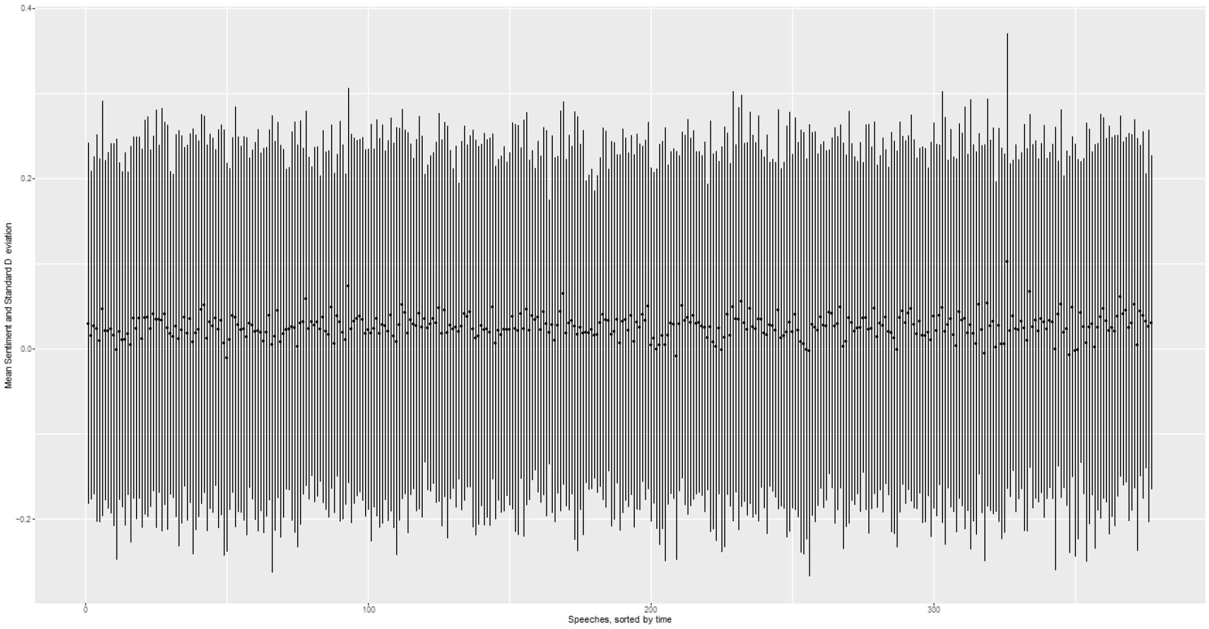


Plot 50: Boxplot of mean sentiment of all speeches in the Obama corpus over time

The reason for this is that more than 50% of the words in any given text do not have a sentiment value in either direction. This makes the boxplot only appear at 0.00 and outliers at every step of sentiment evaluation.

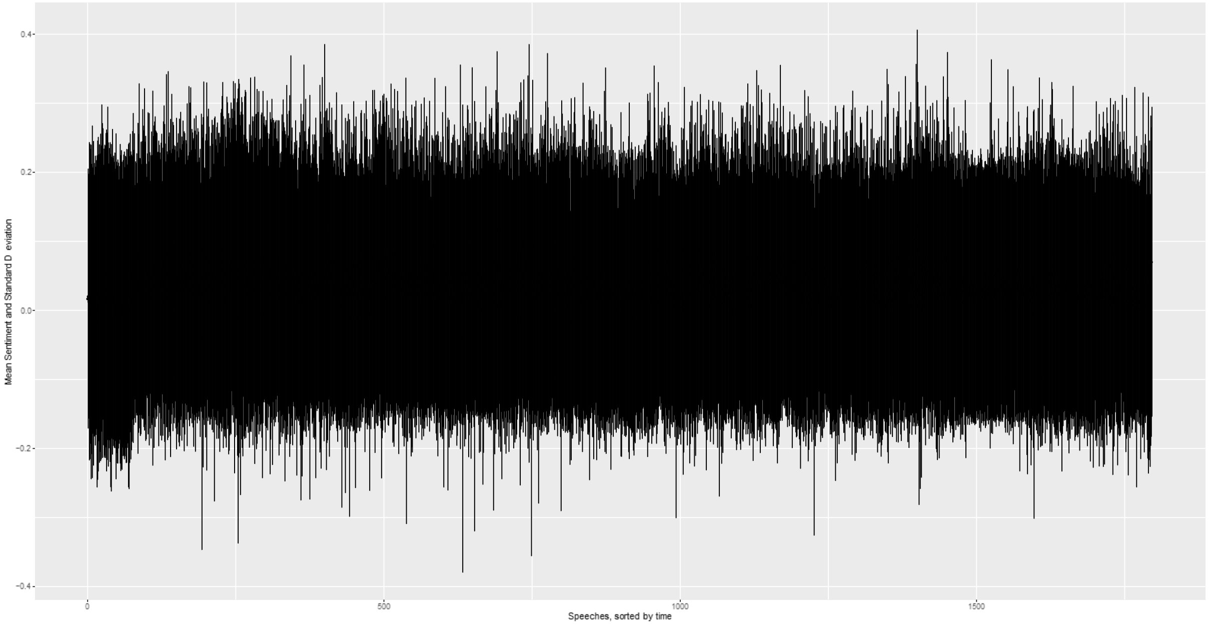
A much more useful plot is the standard deviation plot.

Obama



Plot 51: Standard deviation plot of all speeches in the Obama corpus over time

Trump



Plot 52: Standard deviation plot of all speeches in the Trump corpus over time

Trump has an average standard deviation of 0,2072 while Obama has an average standard deviation of 0,2166. This means that in the speeches, both presidents are rather consistent with the sentiment. But since Trump has a much higher deviation in the overall sentiment mean, this means that his speeches vary a lot more from each other than Obama's.

All in all there is no trend in the data over time.

4.5.5. Sentiment in Proximity to Certain Words

The sentiment is not only interesting over the span of a whole speech, but also in closer inspection, when changing in the proximity of certain words. These changes can indicate the sentiment towards a certain topic of the speaker. The comparison will be done between the excerpt and the whole corpus, since the difference is much higher and obviously comparing the proximity sentiment of for example “war” to a speech about war leads to a small difference.

Code

```
## Julian Lemmerich
## 03.08.2021
## Thesis
## The goal is to calculate sentiment of snippets in proximity to a certain wo
rd or word group.

library(syuzhet)[58]

library(quanteda)[59]
library(dplyr)[60]
library(stringr)[61]
library(knitr)[62]
library(kableExtra)[63]

#reading the files and converting them to a tokenized corpus
obama.corpus <- c()
setwd("C:\\Users\\julian.lemmerich\\OneDrive\\User Data\\Uni\\Semester 8\\Thes
is\\Corpora\\obama\\corpus")
files <- list.files(pattern=".txt", full.names=TRUE)
for (f in files) {
  text <- paste(readLines(f, encoding="UTF-8"), collapse=" ")
  obama.corpus <- c(obama.corpus, text)
}
```

[58] Jockers (2015)

[59] Benoit, K; Watanabe, K; Wang, H; Nulty, P; Obeng, A; Müller, S; Matsuo, A (2018): *quanteda: An R package for the quantitative analysis of textual data*, in: *Journal of Open Source Software*, **3** (30), doi: 10.21105/joss.00774, https://quanteda.io

[60] Wickham, Hadley; François, Romain; Henry, Lionel; Müller, Kirill (2021): *dplyr: A Grammar of Data Manipulation*, R package version 1.0.7, https://CRAN.R-project.org/package=dplyr

[61] Wickham, Hadley (2019): *stringr: Simple, Consistent Wrappers for Common String Operations*, R package version 1.4.0, https://CRAN.R-project.org/package=stringr

[62] Xie, Yihui (2021): *knitr: A General-Purpose Package for Dynamic Report Generation in R*, R package version 1.33

Xie, Yihui (2015): *Dynamic Documents with R and knitr*, 2nd edition, Chapman and Hall/CRC, ISBN 978-1498716963

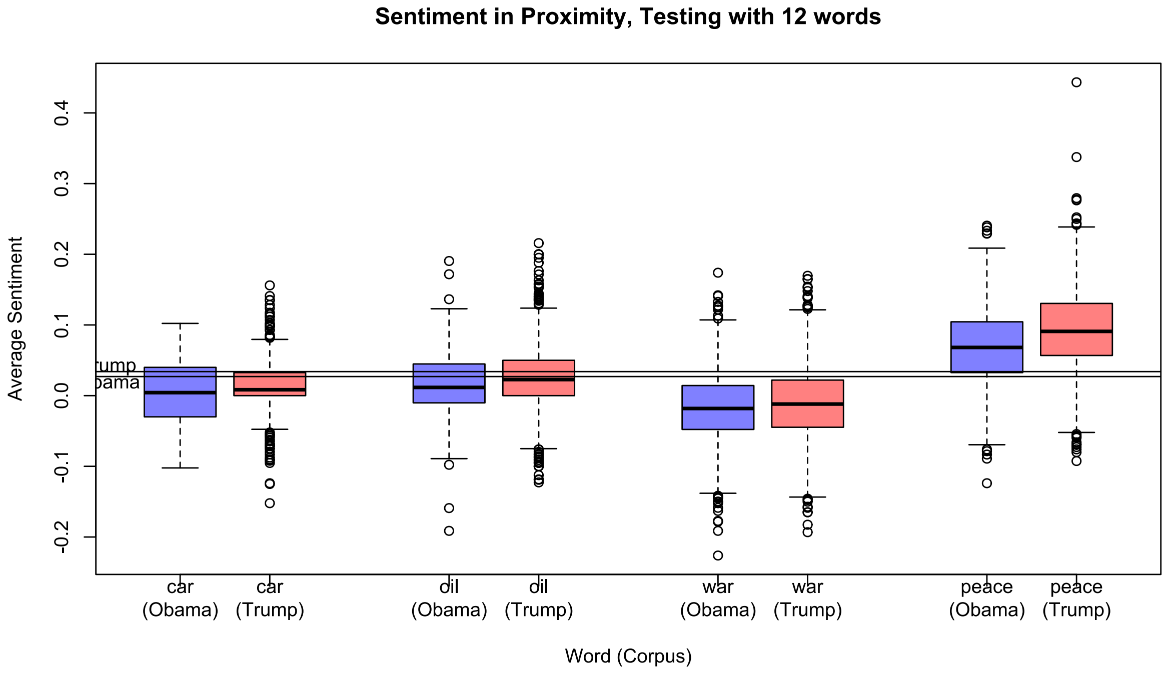
Xie, Yihui (2014): *knitr: A Comprehensive Tool for Reproducible Research in R*, in: Victoria Stodden, Friedrich Leisch and Roger D. Peng: *Implementing Reproducible Computational Research*, Chapman and Hall/CRC, ISBN 978-1466561595

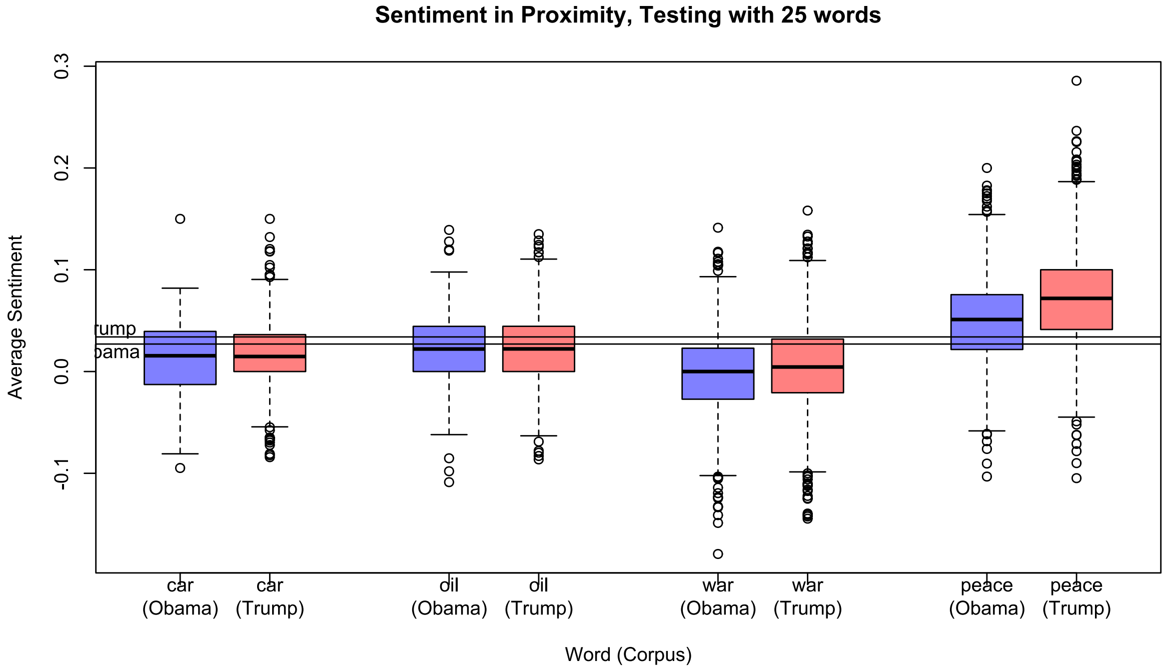
[63] Zhu, Hao (2021): *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax*, R package version 1.3.4, https://CRAN.R-project.org/package=kableExtra

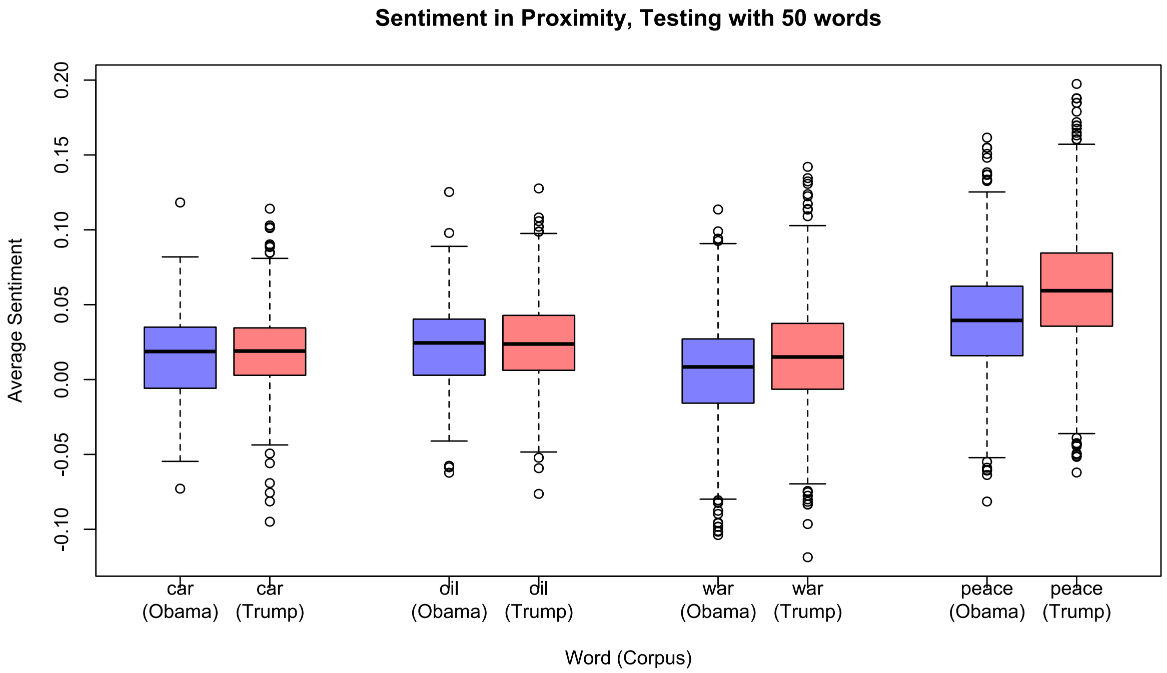
```r
obama.corpus <- tokens(obama.corpus)

trump.corpus <- c()
setwd("C:\\Users\\julian.lemmerich\\OneDrive\\User Data\\Uni\\Semester 8\\Thes
is\\Corpora\\trump 2-3\\corpus")
files <- list.files(pattern=".txt", full.names=TRUE)
for (f in files) {
  text <- paste(readLines(f, encoding="UTF-8"), collapse=" ")
  trump.corpus <- c(trump.corpus, text)
}
trump.corpus <- tokens(trump.corpus)

#mean of corpus
totalmean.obama <- 0.027 #obama
totalmean.trump <- 0.034 #trump

#setting variables
#phrases with space need to be wrapped in phrase(), * is a valid wildcard
findword <- c("Corona", "COVID", "pandemic", "virus", "vaccin*") #searched wor
d or list of words, when running with multiple corpora, the word needs to be l
isted twice
p <- 50 #number words before and after the searched that should be included in
 this analysis (a value of 50 means 101 words in total will be analysed)

#generating findword label, for plot later
findwordlabel <- findword

#one corpus
corpus <- trump.corpus
kwiclist <- list()
for (j in (1:length(findword))) {
  kwiclist[[j]] <- kwic(corpus, pattern=findword[j], window=p, case_insensitiv
e=TRUE)
}

#multiple corpora
#doubling findwordlist for multiple corpora
findwordtemp <- findword
findword <- c()
for (m in (1:length(findwordtemp))) {
  findword <- c(findword, findwordtemp[m], findwordtemp[m])
}
findwordlabel <- findword

#multiple corpora
#starting kwic analysis
kwiclist <- list()
l <- 0 #l is set to 0 to start with obama corpus, 1 for trump
for (j in (1:length(findword))) {
  if (l == 0) { #running with obama
    kwiclist[[j]] <- kwic(obama.corpus, pattern=findword[j], window=p, case_insensitive=TRUE)
    findwordlabel[j] <- paste0(findwordlabel[j], "\n(Obama)") #adding corpus name to label
    l <- 1
  }
  else if (l == 1) { #running with obama
    kwiclist[[j]] <- kwic(trump.corpus, pattern=findword[j], window=p, case_insensitive=TRUE)
    findwordlabel[j] <- paste0(findwordlabel[j], "\n(Trump)") #adding corpus name to label
    l <- 0
  }
}

## Calculating the Sentiment
#lists for a single kwic
veclist <- list()
meanvec <- c()
sdvec <- c()

#collection of all kwic lists
veclistlist <- list()
meanlist <- list()
sdlist <- list()

for (k in (1:length(kwiclist))){ #iterating over all kwiclists
  #clearing the lists for the next loop
  veclist <- list()
  meanvec <- c()
  sdvec <- c()

  if (length(kwiclist[[k]]$keyword > 0)) { #this if clause stops an error in execution, if the keyword does not appear in the corpus at all.

    for (i in (1:length(kwiclist[[k]]$keyword))) { #iterating over all elements in one kwic
      piece <- paste(kwiclist[[k]]$pre[i], kwiclist[[k]]$keyword[i], kwiclist[[k]]$post[i], sep=" ") #making the pre, kw and post into one string
      poa_word_v <- get_tokens(piece, pattern = "\\W") #tokenizing the piece
      syuzhet_vector <- get_sentiment(poa_word_v, method="syuzhet") #get sentiment from tokenized piece

      veclist[[i]] <- syuzhet_vector #adding the original syuzhet vector to a list for later analysis

      meanvec <- c(meanvec, mean(syuzhet_vector)) #calculating mean and sd of one text
      sdvec <- c(sdvec, sd(syuzhet_vector))
    }

  }

  #adding to kwic-lists
  veclistlist[[k]] <- veclist
  meanlist[[k]] <- meanvec
  sdlist[[k]] <- sdvec
}

#generating at and col values for boxplot
#colors are pretty simple, just alternating blue and red for the length of the list.
colours <- c()
for (n in 1:(length(findwordlabel)/2)) {
  colours <- c(colours, "#8080ff", "#ff8080")
}
#at values are a bit more complicated: the schema is c(1:2, 4:5, 7:8, ...)
atv <- c()
p <- 1
for (o in 1:(length(findwordlabel)/2)) {
  atv <- c(atv, p, p+1)
  p <- p+3
}

#if the labels are too large for the plot, the corpus association can be dropped here
findwordlabel <- findword

#creating the plot
boxplot(meanlist, names=findwordlabel,
        at=atv,
        col=colours,
        main="Sentiment in Proximity, Topic \"Corona\"", ylab="Average Sentiment", xlab="Word (Corpus)")
#adding abline for totalmean of obama
abline(h=totalmean.obama)
text(0.2, totalmean.obama-0.007, "Obama")
#adding abline for totalmean of trump
abline(h=totalmean.trump)
text(0.2, totalmean.trump+0.007, "Trump")
print("done")
```

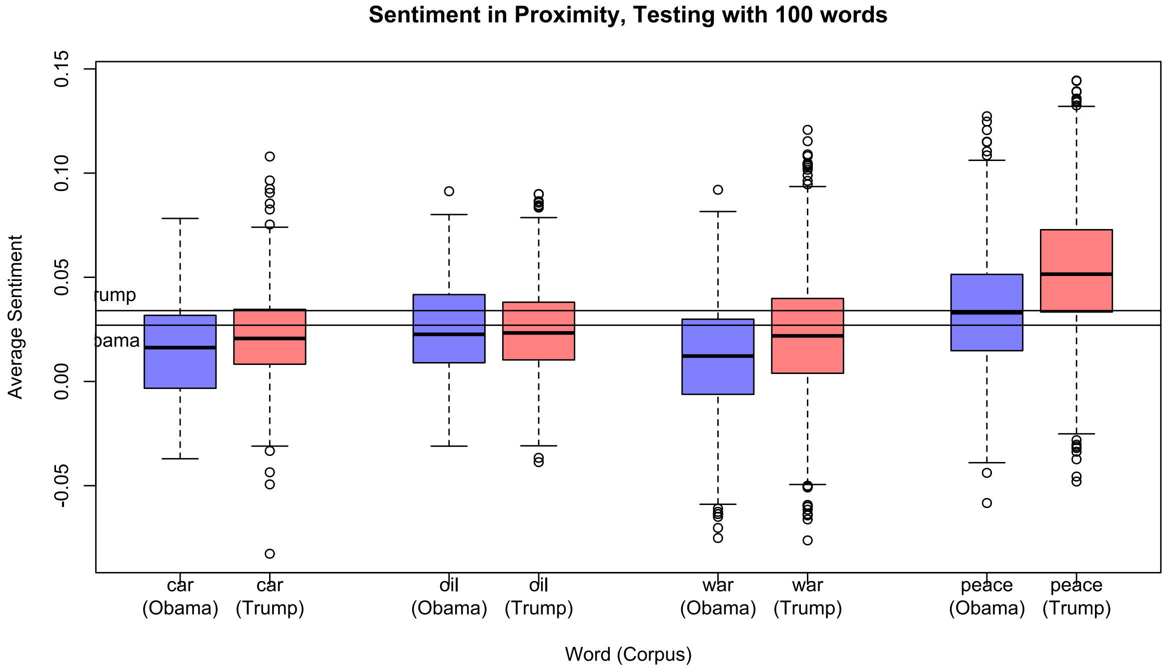
4.5.5.1. Testing KWIC Value

When extracting the key word in context (kwic) parts of the corpora, one needs to choose the number of words before and after the keyword that should be extracted. I tried four different values (12, 25, 50, 100) on a test set of words to see how this choice effects the results and which value will be used for the full analysis. The value is for the number of words per direction. So a value of 12 will yield an excerpt of 25 words (including the keyword), a value of 25 will yield an excerpt of 51 words, 50 -> 101, 100 -> 201.









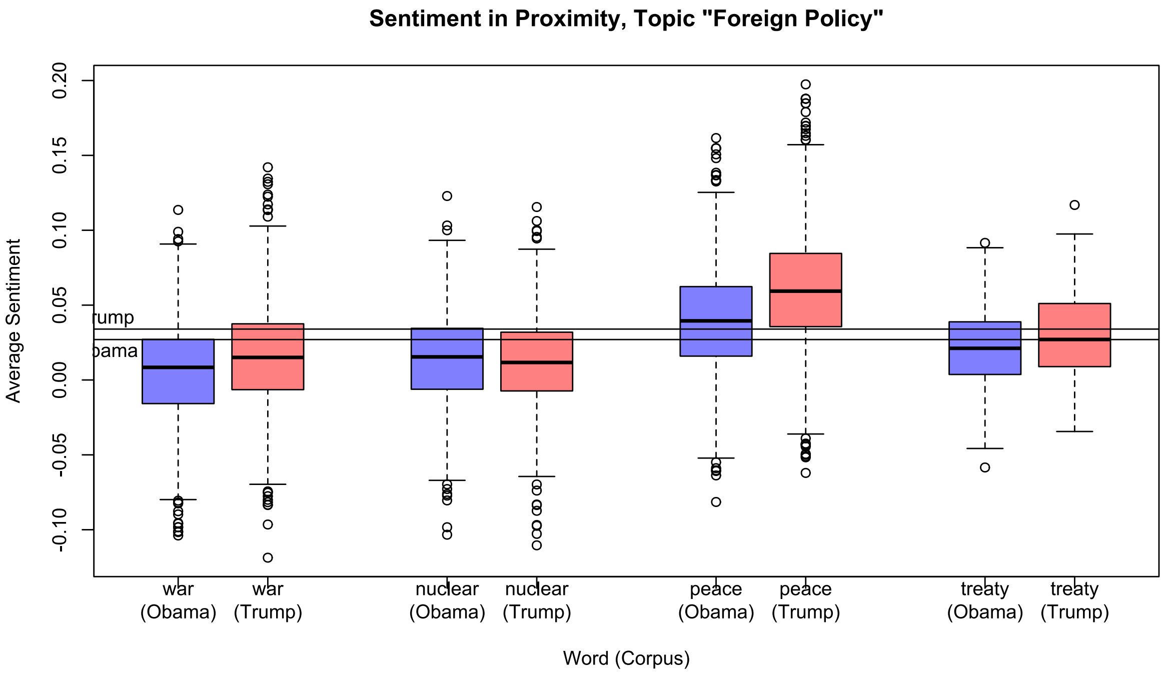
Plot 53-56: Boxplot of sentiment in proximity of testwords with a different scope of keywords-in-context. Additional horizontal line for the overall corpus average for the Trump and Obama corpora

With these four example boxplots it is clear, that the more context a keyword gets, the closer the average sentiment of the text excerpt gets towards the overall corpus average. This can be seen the most with “war” and “peace”. Since the sentiment value of “war” is -0,5 and the sentiment value of “peace” is +0,75 they will then have a larger impact on the average sentiment of the excerpt, the less words there are.

The average sentence length in the Obama corpus is 24,17 and the average sentence length of the Trump corpus is 10,49. This means that with a value of 12 one to two sentences will be included in the excerpt. With a value of 50 the surrounding five to eight sentences will be included. This is a suitable length for this analysis. It doesn't stray too far from the original keyword to become irrelevant, but it also does not give too much weight to a single word to disproportionally affect the sentiment score. I will thus be using a value of 50, resulting in 101 words per excerpt for this analysis.

4.5.5.2. Foreign Politics

The words for this topic are: *war, nuclear, peace, treaty*.



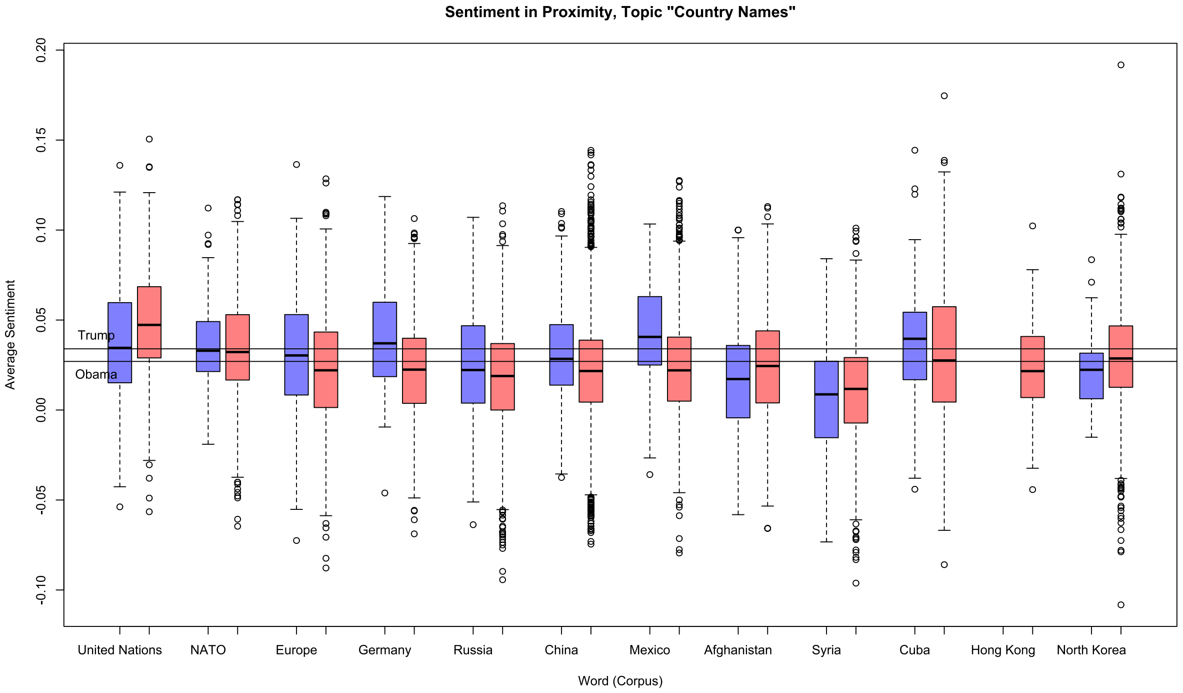
Plot 57: Boxplot of sentiment in proximity to the words *war, nuclear, peace, treaty*. Additional horizontal lines to display the overall corpus average for the Trump and Obama corpora

Like in Obama's and Trump's respective corpus sentiment average, Trump's sentiment is more positive in this topic than Obama's. The exception here is "nuclear", where Trump is slightly less positive than Obama. The most apparent but also least surprising result here is "peace" being overall more positive than "war", even more so with Trump than Obama.

Notable is that no mean of these excerpts falls below 0, which means that these texts are still positive overall.

4.5.5.3. Countries

This topic is a subtopic of foreign politics. It includes the names of important foreign countries and associations: *UN (United Nations), NATO, Europe, Germany, Russia, China, Mexico, Afghanistan, Syria, Cuba, Hong Kong, North Korea*.



Plot 58: Boxplot of sentiment in proximity to the words *UN (United Nations), NATO, Europe, Germany, Russia, China, Mexico, Afghanistan, Syria, Cuba, Hong Kong, North Korea*. Additional horizontal lines to display the overall corpus average for the Trump and Obama corpora

For readability reasons the x-label has been shortened. The blue graphs show the Obama corpus, the red graphs the Trump corpus.

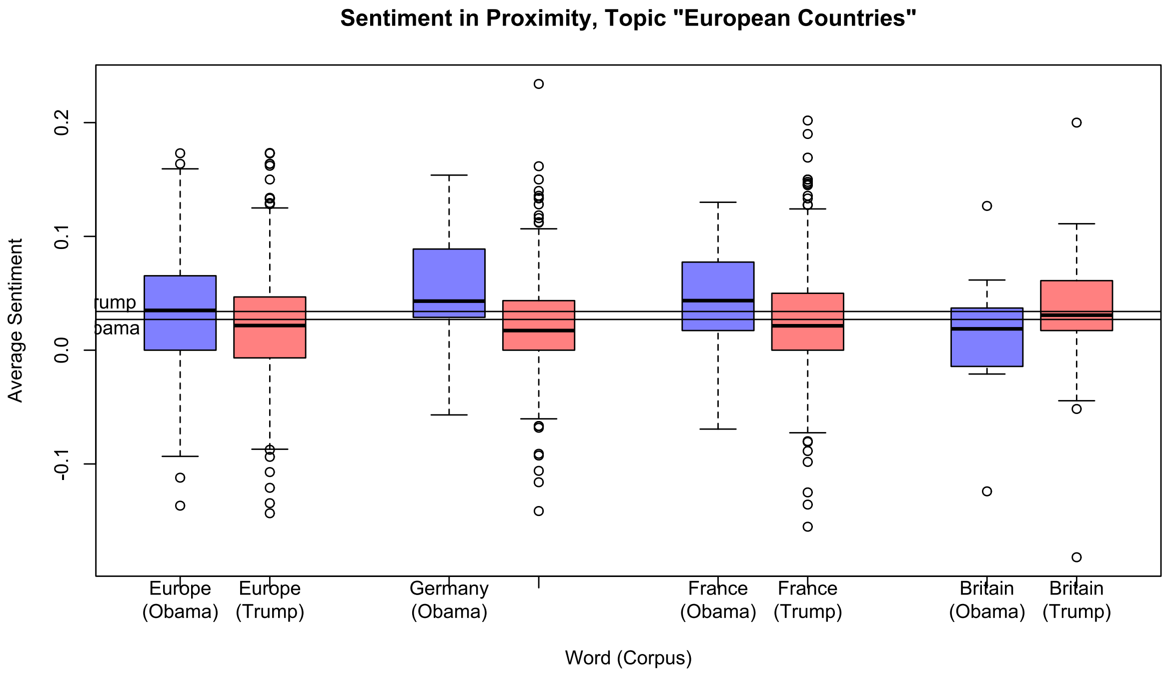
The "United Nations" have a much higher mean sentiment in Trump's corpus than in Obama's. In both cases it's higher than the corpus average.

NATO on average has about the same sentiment for both corpora, which is higher than corpus average for Obama, and a bit lower for Trump. Trump had a changing but overall not positive opinion of the NATO alliance. On the campaign trail he called NATO "obsolete"[64] and even threatened leaving the alliance, because in his opinion the other countries were not pulling their weights.[65]

[64] *Trump says NATO is obsolete but still 'very important to me'*, in: *Reuters* (15.01.2017), https://www.reuters.com/article/us-usa-trump-nato-obsolete-idUSKBN14Z0YO (retrieved 07.08.2021)

[65] *Nato will Donald Trump mit höheren Verteidigungsausgaben besänftigen*, in: *Zeit Online* (29.11.2019), https://www.zeit.de/politik/ausland/2019-11/nato-gipfel-donald-trump-verteidigungsausgaben-zahlen (retrieved 07.08.2021)

Trump wollte Nato angeblich mit Austritt der USA drohen, in: *Süddeutsche Zeitung* (23.06.2020), https://www.sueddeutsche.de/politik/regierung-trump-wollte-nato-angeblich-mit-austritt-der-usa-drohen-dpa.urn-newsml-dpa-com-20090101-200623-99-527321 (retrieved 07.08.2021)



Plot 59: Boxplot of sentiment in proximity to the words *Europe, Germany, France, Britain*. Additional horizontal lines to display the overall corpus average for the Trump and Obama corpora

Overall Trump was not as fond of Europe as his predecessor. This can be seen by the consistently lower sentiment scores for Europe and Germany. Calculating sentiment for the three central European countries as well as "europe" again, it shows Trump’s position in regards to Germany, France and Great Britain. France and Germany have a higher sentiment than the rest of Obama’s corpus, while they clearly fall lower with Trump. With Britain this relationship turns around. Trump was fond of British Prime Minister Boris Johnson and saw an ally in him.[66]

Returning to Plot 58, sentiment around "Russia" was lower with both Obama and Trump. For Trump relatively much lower in comparison to his corpus average. Trump's two 'scapegoats' on the international stage, China and Mexico unsurprisingly scored lower, but not much lower than Russia. Even greater is the contrast for Obama's positive sentiment for "Mexico". It shows the good US-Mexican relations in the Obama era.

"Afghanistan" and "Syria" both have very low sentiments. Given the US involvement in armed conflicts in both countries this is to be expected. Afghanistan may have a slightly higher sentiment with Trump since he started the pull-out shortly before the end of his term.[67]

Obama started the Cuban thaw in 2014, warming the relations with the country close to their south. In 2017 Trump stated that he was cancelling the Obama deals with Cuba and rolling back loosened travel restrictions. This explains their respective sentiment scores for Cuba.

[66] Lippman, Daniel; Toosi, Nahal (2019): *Boris and Donald: A very special relationship*, in: *Politico* (12.12.2019), https://www.politico.com/news/2019/12/12/trump-boris-johnson-relationship-083732 (retrieved 07.08.2021)

[67] Landwehr, Arthur (2020): *Trump schafft mit Truppenabzug Fakten*, in: *tagesschau.de* (17.11.2020), https://www.tagesschau.de/ausland/us-abzug-afghanistan-105.html (retrieved 07.08.2021)

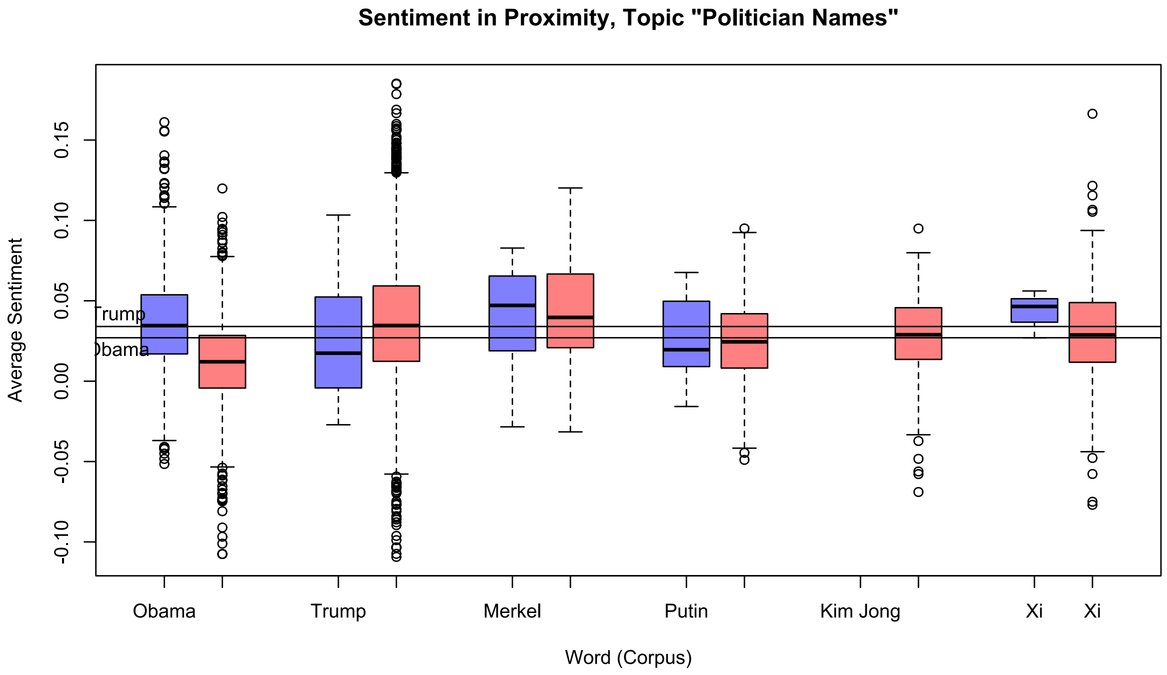
Another term where Trump’s sentiment is higher than Obama's is "North Korea". This can be explained by the restarted relations between the United States and North Korea during Trump's term. Trump's "North Korea" also has the excerpt with the highest and the lowest sentiment in this topic.

In conclusion, the sentiment in proximity to country names is a very good indicator of overall international relations with the respective country.

4.5.5.4. Names

From the larger topic of foreign politics, the names of heads of state of other countries, as well as Trump and Obama themselves are interesting to look at closely: *Obama, Trump, Merkel, Putin, Kim Jong, Xi*.

I decided against listing presidents or heads of states of other countries, since it is harder to compare them, when they did not interact with both Obama and Trump in their time in office.



Plot 60: Boxplot of sentiment in proximity to the words *Obama, Trump, Merkel, Putin, Kim Jong, Xi*. Additional horizontal lines to display the overall corpus average for the Trump and Obama corpora

The first thing that catches the eye is that Trump and Obama have a negative sentiment when mentioning each other. As already described in Lemmerich 2020 and Lemmerich 2021a their own names are seldomly used in texts, but more so a paratextual artifact left over, so this data can be disregarded.

"Merkel" has a more positive sentiment than the respective corpus average for both, for Obama higher than Trump. This is contrary to the lower sentiment of "Germany" in Trump's corpus, which show his differences with the German chancellor.

"Putin", like "Russia" in the previous chapter, has a lower than average sentiment score with both presidents. In both corpora "Putin" gets a lower score than "Merkel". Trump definitely respected Putin, maybe even wanted his recognition.[68]

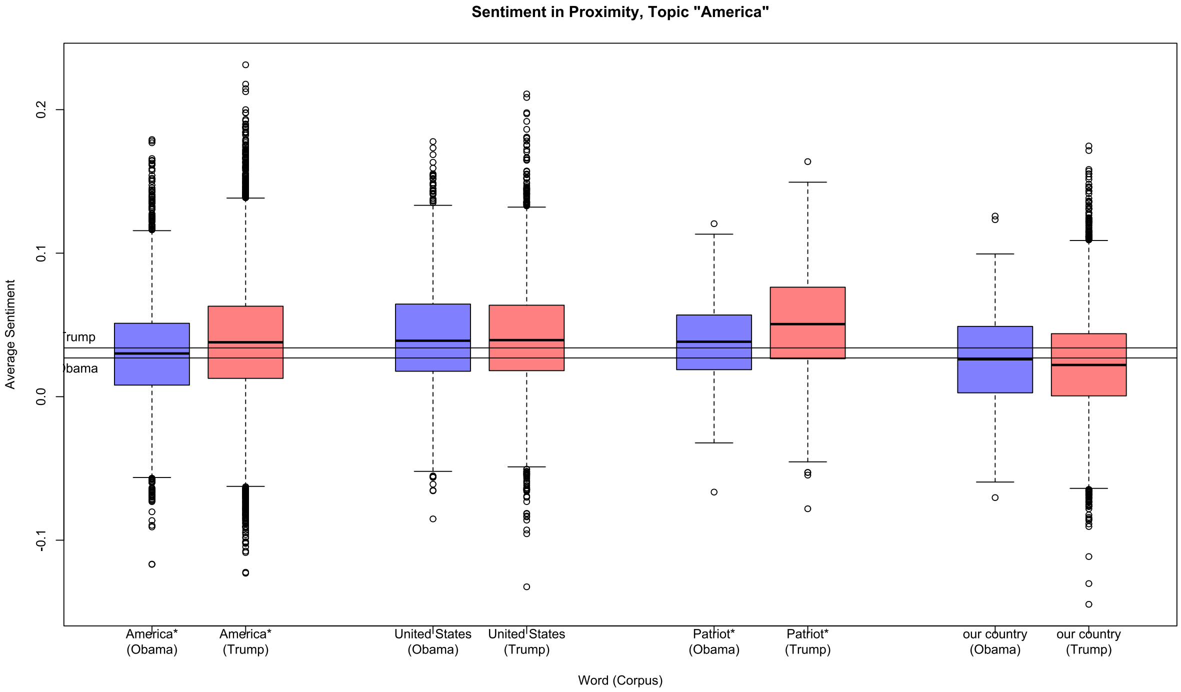
Neither Kim Jong-Il, until his death in 2011 the leader of North Korea, nor Kim Jong-Un, his son and predecessor, are mentioned in the Obama corpus. Even though US-North Korean relations were not warm in Obama's term, it is still a surprise, that neither leader was mentioned once in his corpus. Kim Jong-Un was mentioned 329 times in the Trump corpus. Just like "North Korea" in the previous chapter, the sentiment is slightly below the corpus average.

Like Trump's relationship with China, the sentiment for its president, Xi Jin-Ping is also lower than corpus average. Obama on the other hand has a more positive sentiment, but with only three mentions in the whole corpus the data is not really representative. Like Kim Jong-Il or Kim Jong-Un Obama does not mention this politician by name much.

4.5.5.5. America

Like already noticeable in chapter 4.2.1.2 in the Word Frequency lists, American presidents talk a lot about the United States of America. Patriotism is a very important topic in American politics, certainly a reason why the phrase "The United States of America" appears this often in the corpora.

The words in this topic are: *America*, United States, patriot*, our country*.



Plot 61: Boxplot of sentiment in proximity to the words *America*, United States, patriot*, our country*. Additional horizontal lines to display the overall corpus average for the Trump and Obama corpora

[68] Scheuermann, Christoph; Hebel, Christina (2018): *Ziemlich neue Freunde*, in: *Spiegel Online* (16.07.2018), https://www.spiegel.de/politik/ausland/donald-trump-und-wladimir-putin-anfang-einer-freundschaft-a-1218790.html (retrieved 07.08.2021)

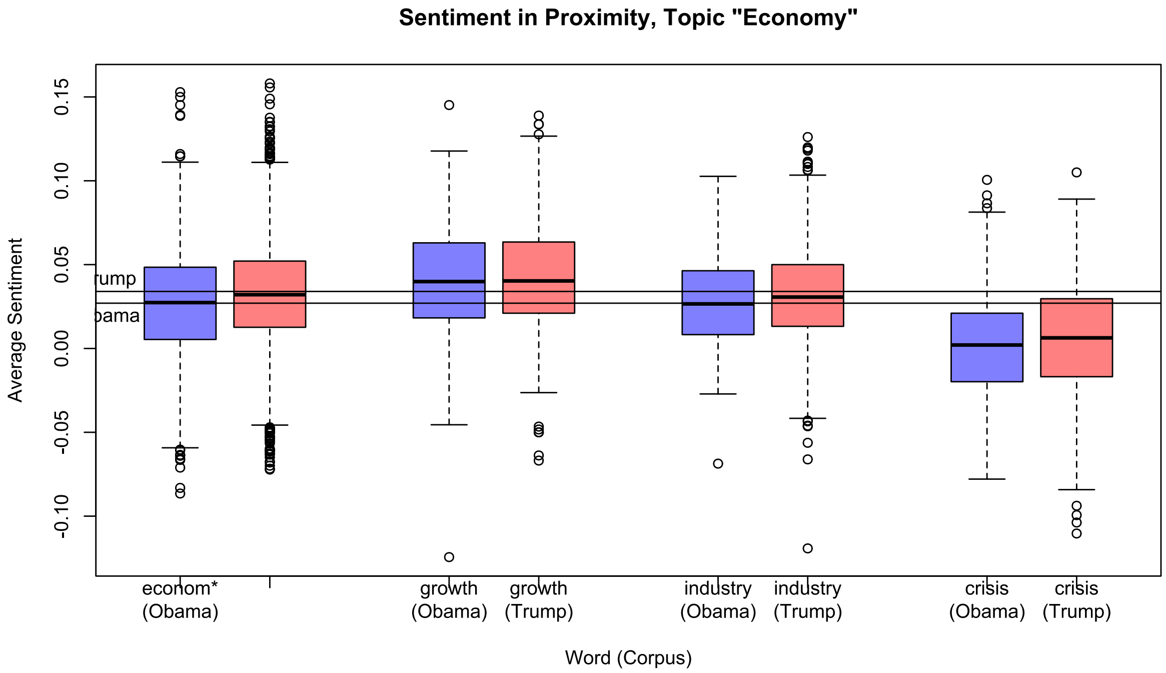
Since patriotism is such an important topic in the United States, it is not surprising to see that "patriot", "United States" and "America" all have higher sentiment than corpus average, there is a lower than average sentiment on "our country" though. Obama is just slightly below corpus average, but Trump even more so. Trump also uses this phrase much more often in his corpus. 6904 hits are in the Trump corpus, while only 303 are in the Obama corpus.

Interesting is that "United States" has a very similar sentiment in both corpora. And for Trump "America" and "United States" have a very similar sentiment, while for Obama they do not.

4.5.5.6. Economy

Economy is also a very pressing domestic political topic in the United States. In both president's time in office, there was an economic crisis, the 2008 financial crisis, for Obama and the 2020 economic crisis due to the Corona pandemic for Trump. The word "crisis" is also included in this topic, but is definitely not exclusive to this topic. The results should thus be interpreted with care.

The words in this topic are: *econom*, growth, industry, crisis*.



Plot 62: Boxplot of sentiment in proximity to the words *econom*, growth, industry, crisis*. Additional horizontal lines to display the overall corpus average for the Trump and Obama corpora

For both corpora "econom*" is very close to corpus average. In the concordances "economic crisis" is only on rank 5 and 29, while most other words are positive, like growth.

Rank	Freq	Cluster
1	73	economic growth
5	43	economic crisis
10	18	economic development
15	14	economic recovery
18	11	economic security
20	10	economic opportunity
21	10	economic progress
24	9	economic future
29	8	economic crisis
30	8	economy grow

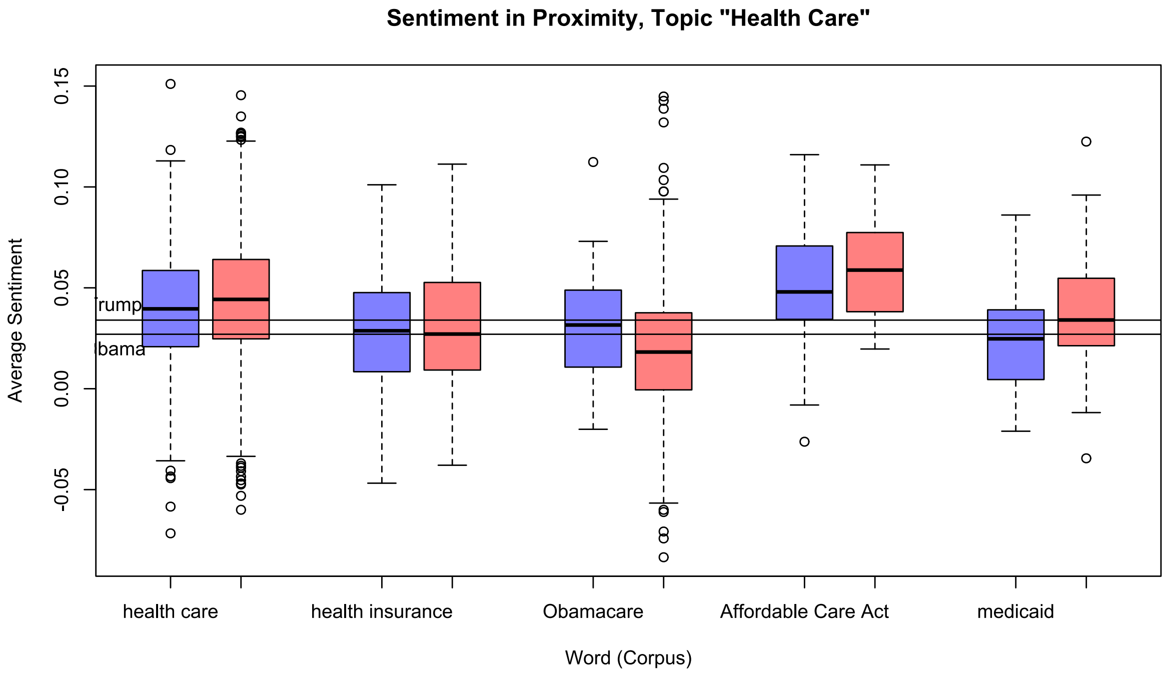
Table 21: 2-grams of “econom*” in the Obama corpus with the keyword on the left

Growth has a positive sentiment, while crisis has a negative sentiment. Both results fit expectations.

4.5.5.7. Healthcare

One of the central topics of Obama’s presidential campaign were his plans for better healthcare for the country, nicknamed "Obamacare". One of Trump’s campaign goals was to repeal Obamacare.

The words in this topic are: *health care, Obamacare, medicaid, Affordable Care Act, health insurance*.



Plot 63: Boxplot of sentiment in proximity to the words *health care, Obamacare, medicaid, Affordable Care Act, health insurance*. Additional horizontal lines to display the overall corpus average for the Trump and Obama corpora

"health care" has a positive sentiment in both corpora, while "health insurance" is lower. It is below corpus average for Trump and on corpus average for Obama.

The 2010 introduced Affordable Care Act (ACA) was one of Obama's central achievements and still today is an essential part of the United States health care system. It is colloquially known as "Obamacare". In his presidential campaign, Trump promised to "repeal and replace" Obamacare.[69] The sentiment results for these two terms is thus very interesting: Both terms are positive in the Obama corpus, while there is a big difference in the Trump corpus. Sentiment for "Obamacare" is much lower than "Affordable Care Act", even though they describe the same bill. The problem with nicknaming a bill after a president is that it makes it a partisan issue, writes John E. McDonough in 2012.[70] A sizable amount of Americans do not know that Obamacare and ACA refer to the same bill.[71] This explains why politicians like Trump, opponents of Obama, use this to frame their and their oponents bills to their advantage.

The sentiment of "medicaid", a social program, not health insurance, is about corpus average for both corpora.

[69] Trump, Donald J. (2016): *We will immediately repeal and replace ObamaCare - and nobody can do that like me. We will save $'s and have much better healthcare!*, Twitter, https://twitter.com/realDonaldTrump/status/697182075045179392 (retrieved 14.11.2016)

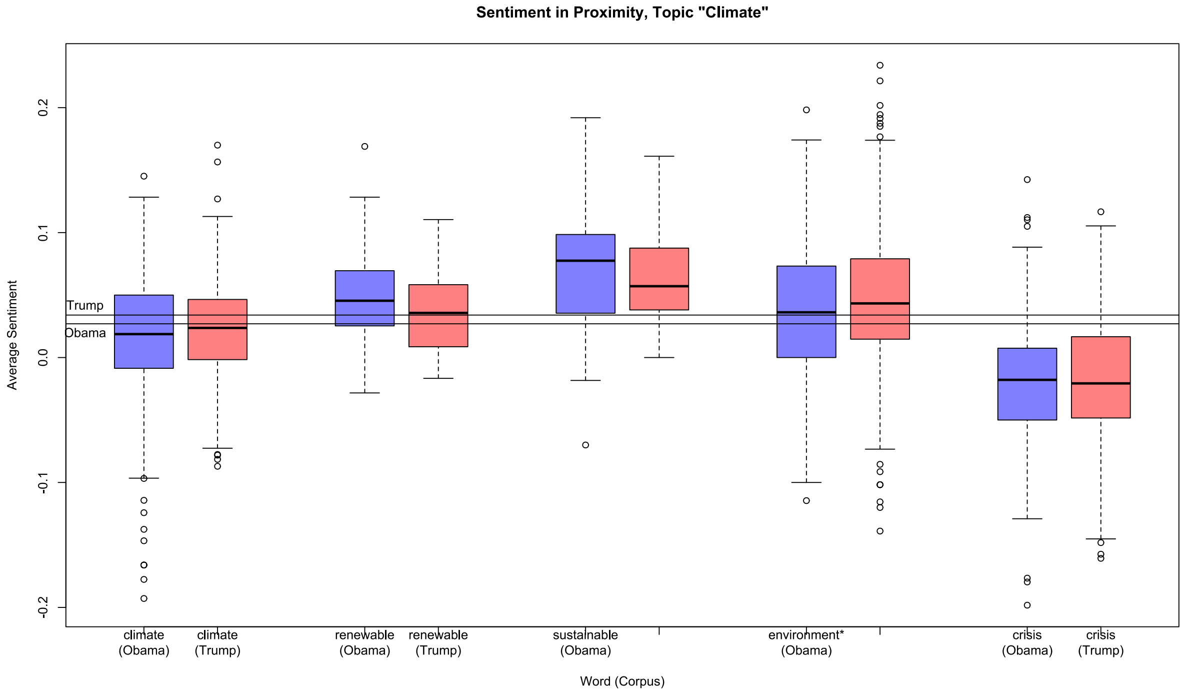
[70] McDonough, John E. (2012): *ACA vs. ObamaCare: What's In a Name?*, in: *boston.com* (14.01.2012), http://archive.boston.com/lifestyle/health/health_stew/2012/01/aca_vs_obamacare_whats_in_a_na.html (retrieved 09.082021)

[71] Dropp, Kyle; Nyhan, Brendan (2017): *One-Third Don't Know Obamacare and Affordable Care Act Are the Same*, in: *The New York Times* (07.02.2017), https://www.nytimes.com/2017/02/07/upshot/one-third-dont-know-obamacare-and-affordable-care-act-are-the-same.html (retrieved 09.08.2021)

4.5.5.8. Climate

Climate change has been one of the most important topic's in one way or another. As with the economy topic, the word crisis is in this topic, but the results need to be interpreted with care, since crisis appears in other contexts than the climate context aswell.

The words in this topic are: *climate, renewable, sustainable, environment*, crisis*.



Plot 64: Boxplot of sentiment in proximity to the words *climate, renewable, sustainable, environment*, crisis*. Additional horizontal lines to display the overall corpus average for the Trump and Obama corpora

"climate" has a negative sentiment for both the Obama and Trump corpus. This is not surprising as it is a complicated topic. Very surprising are the very low extremes for "climate" with Obama.

"renewable" has a positive sentiment for the Obama corpus, but only corpus average for the Trump corpus.

Both "sustainable" and "environment*" have a higher-than-average sentiment, with "sustainable" in the Obama Corpus having the highest average sentiment in this topic. The highest single sentiment is again held by the Trump corpus, because, like in the complete corpus, Trump has much higher variability in his sentiment.

Obama's corpus also has a higher deviation in the first three words. Only "environment" has a higher deviation with Trump's corpus.

"crisis" has a much lower sentiment than the rest of the corpus. It was also included in the topic economy (chapter 4.5.5.6), but had much more importance there, as the 2-gram "climate crisis" is only on rank 52 of "* crisis" 2-grams, with only 2 occurences, so it can be disregarded for this topic.

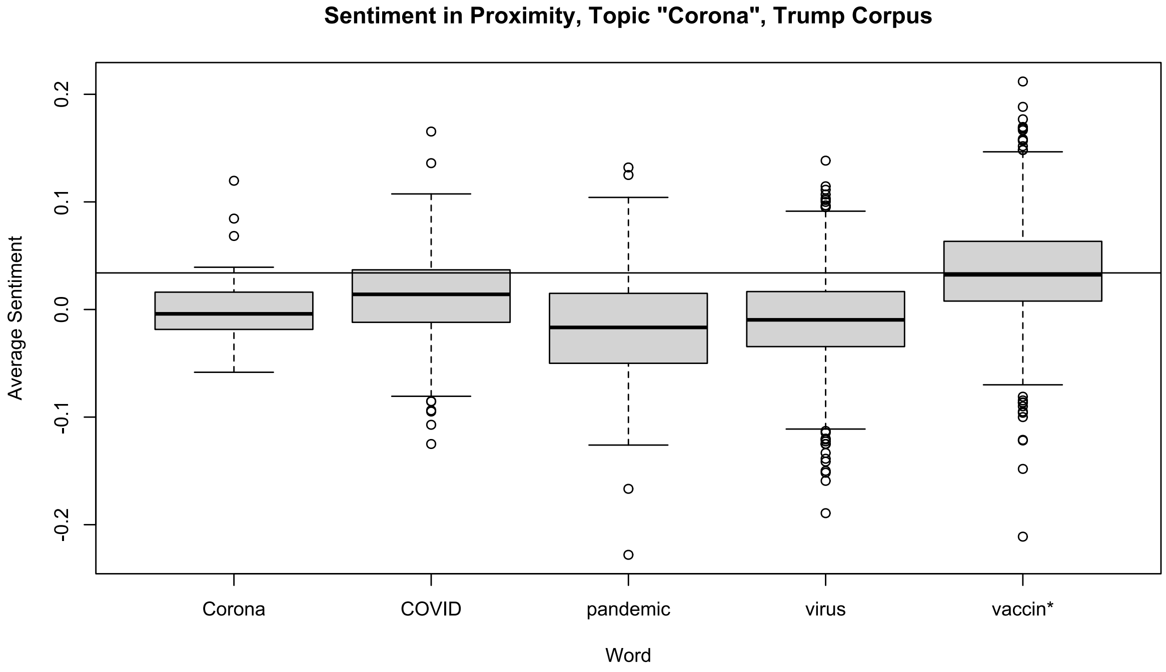
Rank	Freq	Cluster
1	181	this crisis
2	102	a crisis
3	101	the crisis
4	67	opioid crisis
5	56	financial crisis
6	54	economic crisis
7	34	humanitarian crisis
9	22	drug crisis
10	18	health crisis
11	16	security crisis
13	13	refugee crisis
14	12	border crisis
15	12	manufactured crisis
16	11	national crisis
17	9	covid crisis
52	2	climate crisis

Table 22: 2-grams of “crisis” in both corpora combined with the keyword on the right

4.5.5.9. COVID-19 (Trump specific)

The last topic is COVID-19. It cannot be used to compare the two presidents with each other, since in Obama's time in office there was no pandemic of this proportion. It was nevertheless a significant topic in the last year of Trump's presidency.

The words in this topic are: *Corona, COVID, pandemic, virus, vaccin**.



Plot 65: Boxplot of sentiment in proximity to the words *Corona, COVID, pandemic, virus, vaccin** in the Trump corpus. An additional horizontal line to display the overall corpus average for the Trump corpus.

Unsurprisingly, most words associated with the COVID-19 pandemic have a negative sentiment. "COVID" has a better sentiment than "Corona", probably because it's the less colloquial term, used in more neutral environments.

"vaccin*" is about corpus average and the most positive in this topic, probably because it is the beginning of solving the pandemic.

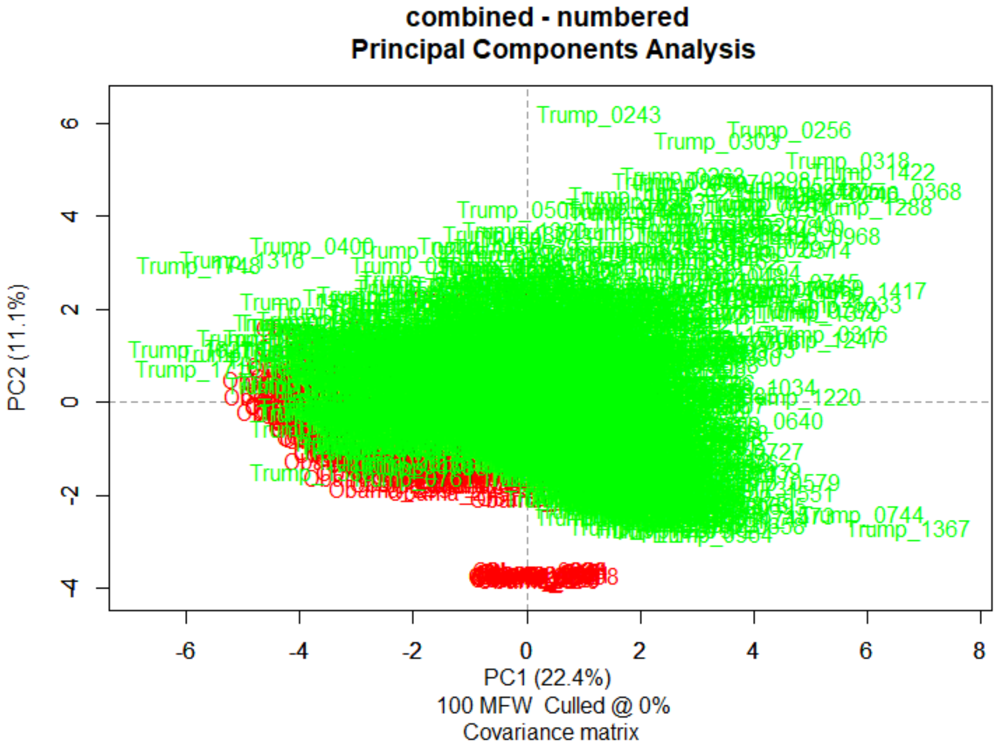
4.6. Authorship Attribution

In my first paper that led to the creation of the Obama corpus the original goal was authorship attribution to the different assisting speechwriters on Obama's staff. The conclusion was unsuccessful, probably because the team had very similar backgrounds, worked together for a decade and was trying to make it hard to distinguish, who wrote the speeches. Another issue was that I did not have any know attributed speeches to compare against.[72]

In Chapter 4.2 of the Trump speech corpus creation, I also tried a classic principal component analysis, to see if the results would be different from the Obama corpus but did not pursue it any further, when the results were a similar large cluster of texts.[73]

It would be interesting to see if the two corpora would separate into different clusters, when compared against each other. As established in the previous chapters, the two corpora are very different from each other, with Trump having a very distinct style from politicians in general.

4.6.1. Classic Burrow's Delta and PCA-Visualization



Plot 66: Principal component analysis visualization of Classic Burrow's Delta of both corpora. Texts from the Obama corpus are colored red, from the Trump corpus colored green.

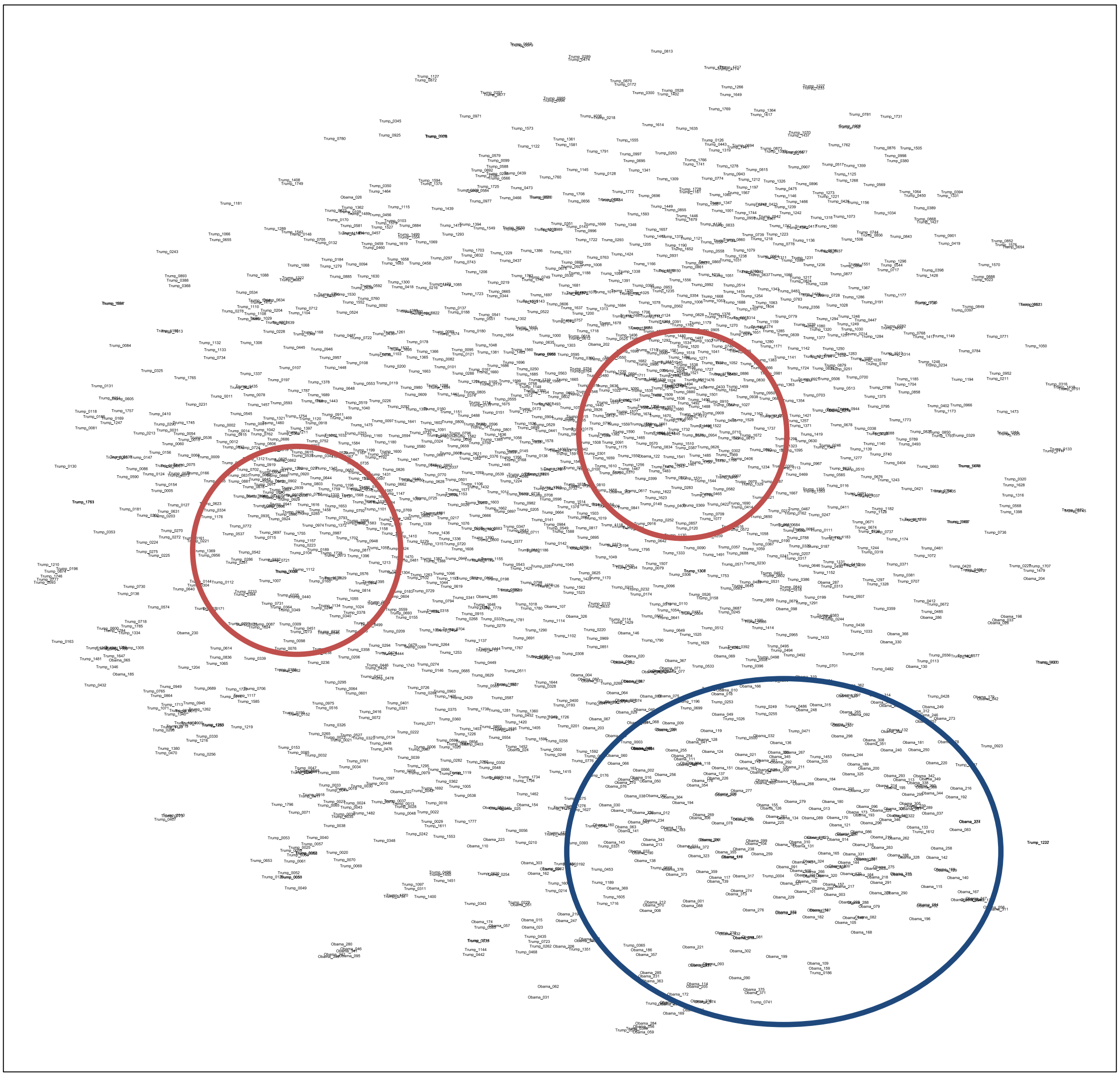
As already with the previous papers, the result with the Burrow's Delta and principle component analysis as Visualization is not very promising. Without the colorization, it would be impossible to differentiate between the Obama corpus texts and the Trump corpus texts.

[72] Lemmerich (2020)

[73] Lemmerich (2021a)

4.6.2. Classic Burrow’s Delta and tSNE-Visualization

The stylo package also includes tSNE visualization. tSNE was developed by Laurens van der Maaten and Geoffrey Hinton in 2008.[74]



Plot 67: tSNE visualization of the combined Obama and Trump corpora, calculated with *stylo*

The full resolution of this visualization can be seen as a SVG-file in the file-attachments 10.

Looking at this visualization, there are three groups that could be interpreted. There are two concentrations of Trump texts (highlighted in red here) and one main grouping for Obama texts (highlighted in blue). But there are still a lot of texts from the Trump corpus mixed in with the Obama texts. And without the naming of datapoints, it would be hard to differentiate them.

[74] Van der Maaten, Laurens; Hinton, Geoffrey (2008): *Visualizing Data using t-SNE*, in: *Journal of Machine Learning Research* 9 (2008), https://jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf (retrieved 23.08.2021)

4.6.3. TF-IDF and tSNE-Visualization

Term Frequency – Inverse Data Frequency (TF-IDF) is the product of the Term Frequency and the Inverse Data Frequency. Term Frequency is the ratio of the number of times a type appears in a document compared to the total amount of tokens in a document. Inverse Data Frequency gives the weight of rare words across all texts in a corpus. There are different variations to calculate TF-IDF and I will be using the most common variant.[75]

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

$$idf\ (w) = \log\left(\frac{N}{df_t}\right)$$

Formula for calculating Term Frequency

Formula for calculating Inverse Data Frequency

i = term, j = document, $n_{i,j}$ = absolute frequency of a term in a document, $\sum_k n_{i,j}$ = sum of max. terms in all documents

N = number of documents, df_t = number of documents, that contain the term

TF-IDF produces a datamatrix that can be visualized with tSNE.

Scikit-learn offers a python package for converting an array of text into a matrix of TF-IDF features.[76] Yellowbrick offers a python package for tSNE visualization that works well with the Scikit TfidfVectorizer.[77]

Code

```
from os import listdir

from sklearn.feature_extraction.text import TfidfVectorizer
#this filter is needed as the FutureWarning will spam output otherwise
from warnings import simplefilter
#ignore all future warnings
simplefilter(action='ignore', category=FutureWarning)

from yellowbrick.text import tsne

path = "C:\\Users\\julian.lemmerich\\OneDrive\\User Data\\Uni\\Semester 8\\The
sis\\Corpora\\combined - named\\corpus"
filelist = listdir(path)
docs = []
for f in filelist:
    file = open(path + "\\" + f, encoding='utf-8')
    docs.append(file.read())
```

[75] Tripathi, Mayank (06.06.2018): *How to process textual data using TF-IDF in Python*, in: *FreeCodeCamp*, https://www.freecodecamp.org/news/how-to-process-textual-data-using-tf-idf-in-python-cd2bbc0a94a3/ retrieved 23.08.2021)

[76] scikit-learn developers: *sklearn.feature_extraction.text.TfidfVectorizer*, https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer.fit_transform (retrieved 23.08.2021)

[77] scikit-yb developers (13.02.2020): *t-SNE Corpus Visualization*. in: *Yellowbrick* https://www.scikit-yb.org/en/latest/api/text/tsne.html (retrieved 23.08.2021)

```
classlist = []
for f in filelist:
    classlist.append(f[0])
```

In the first half of the code, all the texts from the corpus are read into an array. The `TfidfVectorizer` supposedly also takes a list of files and iterates itself, but I did not obtain the expected results.

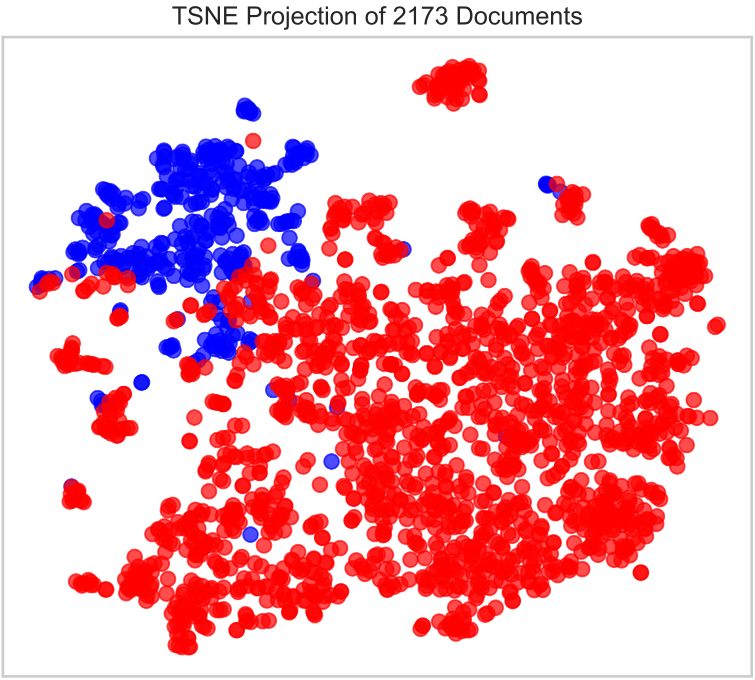
The `classlist` array is created for categorizing the text either to Obama ("O") or Trump ("T") for coloring and labeling in the visualization.

```
x = TfidfVectorizer().fit_transform(docs, y=None)
#encoding is utf-8 by default
```

`x` is not a matrix of TF-IDF features that can be visualizes by the `tsne` function. Parameter `y` is an array of equal length to `x` that categorizes the datapoints into a class for visualization.

```
tsne(x, y=classlist, colors=["blue", "red"])
```

Visualization



Plot 68: tSNE-visualization of TF-IDF matrix with text from the Obama corpus colored blue, and texts from the Trump corpus colored red

This visualization has similar issues to the tSNE-visualization of the Burrow's Delta. It is definitely better than the PCA-visualization in differentiating between the two presidents, but without the coloring it would be hard to identify two different corpora in this visualization.

4.6.4. Authorship Attribution: Conclusion

Even though these two corpora are very different from each other, as was demonstrated by the number of different methods in the previous chapters, none of the available methods for authorship attribution are able to clearly differentiate them from each other. Given this it is not surprising, that the authorship attribution to different writers trying for a uniform style in the two corpus creation papers was unsuccessful.

5. Limitations

The biggest limitation in this paper was the dataset, since the data was scraped, where a lot of errors were introduced in comparison to a 'cleaner' method of data-mining. The dataset was also in plaintext instead of XML, so exclusion of other speakers, for example in press events or the crowd in public events, was not easily possible. Paratext, like speaker denotation, was also included. These issues could lead to some datapoints being skewed, like the most frequent word in the Trump corpus in chapter 4.2.1. being "president", which was apparent, but there may be other similar phenomena that may not have been noticed in this paper.

The corpora for the two different presidents were also from two different sources (once *American Rhetoric* and one from the *American Presidency Project*). Ideally the datasets would be from the same source to assure uniformity in editing.

6. Discussion, Conclusion and Outlook

Not only the politics, but also the speeches of these two successive presidents were very different. Most of the findings in this thesis confirm previous results about presidential speeches. Trump especially is very different from previous presidents and other politicians. This may have made the gap between the two corpora even larger.

The differences between the two corpora really show in the part of speech distribution. Trump uses more verbs, names, and pronouns than Obama. The mean sentence length also really differentiates the two corpora: Obamas sentences are more than two times longer than Trumps sentences. The lexical diversity of the Trump corpus is lower, but not by a big margin. The number of big words is lower in the Obama corpus, but not by as big of a difference as was found by other papers.

The word frequencies show some distinct words for each president. "great" is one of Trump's often used words. It appears five times more often in the Trump corpus than in the Obama corpus. "health" and "care" on the other hand are very frequent words for Obama that do not appear in Trumps mfw-list. Both presidents, as well as the Corpus of Contemporary American English (COCA) have many variations of "America" in their corpus, which shows American patriotism.

The Flesch-Kincaid readability tests did not yield useful results. The highest scores, categorized as most complex by Flesch-Kincaid, would indicate a listener's age of 2700 years, which is an absurd score. The meaningfulness of this method is doubtful for political speeches.

Sentiment analysis brought forth a number of interesting results. The overall sentiment score for the Obama corpus is lower than for the Trump corpus. Trump shows a higher deviation though, claiming both the most positive and most negative speeches of this dataset. The sentiment in proximity to certain words also yielded some interesting results. The political position towards foreign nations aligned well with the sentiment in proximity to those nations. The same data can be observed for the corresponding head of states. I was not able to follow all interesting data points due to the quantity. Further research could concentrate on just one set of keywords and discuss the sentiment in close reading instead of distant reading, like this paper did. Different techniques of sentiment analysis will also yield different results. A better method for finding sentiment could definitely improve the results. Some of the problems with the method used in this paper have been known and already discussed (see chapter 4.5.2.).

Authorship attribution should also be further researched. There are a lot of different ways to calculate the distance between texts and then also visualize that. The visualization is a very important aspect. So far, I was not able to show the differences between the two corpora. They were pointed out in other methods though and I think it should be possible to differentiate them in authorship attribution as well.

There are many areas where further research can be conducted based on these corpora and the findings in this Paper: Comparing these two corpora to even more presidential and political corpora would be interesting to bring into perspective how usual or unusual these differences between successive presidents are.

The existing corpora can be improved to see if cleaner data yields different results. Obtaining the data in tagged form to differentiate speaker and paratext from the actual speech content would drastically improve the data. With such an improved dataset the analysis could be repeated.

This paper did not touch on bias and political ideology too much. The two very different corpora could be an interesting dataset to further research in this direction, for example by looking for similarities between differently biased news sources and attributing the corpora on this political spectrum. However, an already trained dataset for bias detection would be necessary for this method. Since it was not part of the scope of this paper, it was not included.

Political speeches and political rhetoric will always keep moving, so there will always be new material to analyse. This also means the same methods used in this paper can be used on new datasets for new and interesting results. After the end of Trump’s presidency, another Democratic politician became president: Joseph Biden. Comparing him to his predecessor could lead to similar results. Biden was also Obama’s Vice-President, so comparing his speeches as president to his own speeches as vice-president could be thought-provoking. Kamala Harris is the first female vice-president of the United States and thus brings an entirely new aspect, gender, to compare against.

7. References

Airoldi, E. M.; Anderson, A. G.; Fienberg, S. E.; Skinner, K. K. (2006): *Who Wrote Ronald Reagan's Radio Addresses?*, in: *Bayesian Analyst*, pp. 289-320, https://projecteuclid.org/download/pdf_1/euclid.ba/1340371064 (retrieved 30.06.2020)

The American Presidency Project, https://www.presidency.ucsb.edu/about

Bansal, Shivam; Aggarwal, Chaitanya: *Textstat*, Python package version 0.7.2, https://pypi.org/project/textstat/ or https://github.com/shivam5992/textstat

Benoit, K; Watanabe, K; Wang, H; Nulty, P; Obeng, A; Müller, S; Matsuo A (2018): *quanteda: An R package for the quantitative analysis of textual data.* in: *Journal of Open Source Software*, **3**(30), 774. doi: 10.21105/joss.00774, https://quanteda.io

Singh, Bhopinder: *When the Trumpets blare: Language of Statesmanship*, in: *DH News Service* (26.02.2020), https://www.deccanherald.com/opinion/main-article/when-the-trumpets-blare-language-of-statesmanship-808435.html (retrieved 22.03.2021)

Blei, David M. (2012): *Probabilistic topic models*, in: *Commun, ACM 55* (4), pp. 77–84. doi: 10.1145/2133806.2133826

Bouchet-Valat, Milan (2020): *SnowballC: Snowball Stemmers Based on the C 'libstemmer' UTF-8 Library*, R package version 0.7.0, https://CRAN.R-project.org/package=SnowballC

Brennan, Christopher (15.02.2017): *SEE IT: Trump adviser Stephen Miller booed off stage by classmates after high school speech*, in: *New York Daily News*, https://www.nydailynews.com/news/national/trump-advisor-stephen-miller-booed-stage-high-school-article-1.2973670 (retrieved 06.05.2019), now available at https://web.archive.org/web/20190506203314/https://www.nydailynews.com/news /national/trump-advisor-stephen-miller-booed-stage-high-school-article-1.2973670 (retrieved 22.03.2021)

Change.gov (26.11.2008): *President-Elect Barack Obama names two new White House staff members*, http://change.gov/newsroom/entry/president_elect_barack_obama_names_two_new_white_house_staff_members/ (retrieved 26.11.2008), now available at web-archive https://web.archive.org/web/20081126191037/http://change.gov/ newsroom/entry/president_elect_barack_obama_names_two_new_white_house_staff_members/

Collegiate School (27.10.2008): *Election 2008: Ben Rhodes '96, Speechwriter and Advisor to Barack Obama*, https://www.collegiateschool.org/news-detail?pk=453880 (retrieved 10.10.2020)

Conway, L. G.; Gornick, L. J.; Burfeind, C.; Mandella, P.; Kuenzli, A.; Houck, S. C.; Fullerton, D. T. (2012): *Does Complex or Simple Rhetoric Win Elections? An Integrative Complexity Analysis of U.S. Presidential Campaigns*, in: *Political Psychology* (33), pp. 599-618, doi:10.1111/j.1467-9221.2012.00910.x

Covington, Michael A; McFall, Joe D: *MATTR 2.0,* Institute for Artificial Intelligence. University of Gerorgia, http://ai1.ai.uga.edu/caspr/ (retrieved 20.03.2021)

Davies, M. (2010): *The Corpus of Contemporary American English as the first reliable monitor corpus of English*, in: *Literary and Linguistic Computing* 25 (4), pp. 447–464. doi: 10.1093/llc/fqq018.

Dennis (05.09.2017): *EXCEL VBA compare cell values to an Array*, https://stackoverflow.com/questions/46049323/excel-vba-compare-cell-values-to-an-array (retrieved 09.07.2021)

Dilai, Marianna; Onukevych, Yuliya; Dilay, Iryna (2018): *Sentiment analysis of the US and Ukrainian presidential speeches*, http://ena.lp.edu.ua:8080/handle/ntb/42572 (retrieved 03.03.2021)

dpa: *Trump wollte Nato angeblich mit Austritt der USA drohen*, in: *Süddeutsche Zeitung* (23.06.2020), https://www.sueddeutsche.de/politik/regierung-trump-wollte-nato-angeblich-mit-austritt-der-usa-drohen-dpa.urn-newsml-dpa-com-20090101-200623-99-527321 (retrieved 07.08.2021)

Dropp, Kyle; Nyhan, Brendan (2017): *One-Third Don't Know Obamacare and Affordable Care Act Are the Same*, in: *The New York Times*, 07.02.2017, https://www.nytimes.com/2017/02/07/upshot/one-third-dont-know-obamacare-and-affordable-care-act-are-the-same.html (retrieved 09.08.2021)

Eidenmuller, M. E.: *Obama Speeches*, https://www.americanrhetoric.com/barackobamaspeeches.htm (retrieved 27.03.2019)

Feinerer, Ingo; Hornik, Kurt; Meyer, David (2008): *Text Mining Infrastructure* in *R. Journal of Statistical Software* 25(5): 1-54. URL: https://www.jstatsoft.org/v25/i05/.

Feinerer, Ingo; Hornik, Kurt (2020): *tm: Text Mining Package*, R package version 0.7-8, https://CRAN.R-project.org/package=tm

Fellows, Ian (2018): *wordcloud: Word Clouds*, R package version 2.6, https://CRAN.R-project.org/package=wordcloud

Felsenthal, C. (19.02.2013): *Cody Keenan, Obama's Chief Speechwriter: Chicago-Born and (Mostly) Bred*, in: *Chicago Magazine*, https://www.chicagomag.com/Chicago-Magazine/Felsenthal-Files/February-2013/Cody-Keenan-Obamas-Chief-Speechwriter-Chicago-Born-and-Mostly-Bred/ (retrieved 10.10.2020)

Guerrero, Jean (10.08.2020): *The Man Who Made Stephen Miller*, in: *POLITICO Magazine* https://www.politico.com/news/magazine/2020/08/01/stephen-miller-david-horowitz-mentor-389933 (retrieved 22.03.2021)

Hains, Tim: *Trump: "If You're Running For President You Shouldn't Be Allowed To Use A Teleprompter*", in: RealClear Politics (25.08.2015), http://www.realclearpolitics.com/video/2015/08/25/trump_i_write_my_own_tweets_if_youre_running_for_president_you_should_be_allowed_to_have_teleprompters.html?jwsource=cl (retrieved 22.03.2021)

Helderman, Rosalind S. (11.02.2017): *Stephen Miller: A key engineer for Trump's 'America first' agenda*, in: *The Washington Post*, https://www.washingtonpost.com/politics/stephen-miller-a-key-engineer-for-trumps-america-first-agenda/2017/02/11/a70cb3f0-e809-11e6-bf6f-301b6b443624_story.html (retrieved 22.03.2021)

Herz, J.; Bellaachia, A. (2014): *The Authorship of Audacity: Data Mining and Stylometric Analysis of Barack Obama Speeches*, in: R. Stahlbock, G. M. Weiss, M. Abou-Nasr, & H.

R. Arabnia: *DMIN 2014 : proceedings of the 2014 International Conference on Data Mining*, http://worldcomp-proceedings.com/proc/p2014/DMI8024.pdf (retrieved 27.03.2019)

Horowitz, J. (02.09.2011): *Jon Lovett's written for the president, but will that get him to Hollywood?* in: *Washington Post*, https://www.washingtonpost.com/lifestyle/style/jon-lovetts-written-for-the-president-but-will-that-get-him-to-hollywood/2011/08/22/gIQAhZmIxJ_story.html (retrieved 10.10.2020)

Ioffe, Julia (27.06.2016): *The Believer. How Stephen Miller went from obscure Capitol Hill staffer to Donald Trump's warm-up act—and resident ideologue.*, in: *POLITICO*, https://www.politico.com/magazine/story/2016/06/stephen-miller-donald-trump-2016-policy-adviser-jeff-sessions-213992/ (retrieved 22.03.2021)

JensS (05.09.2017): Answer to *EXCEL VBA compare cell values to an Array*, https://stackoverflow.com/a/46050159/9397749 (09.07.2021)

Jockers, Matthew L. (05.06.2014): *A Novel Method for Detecting Plot*, https://www.matthewjockers.net/2014/06/05/a-novel-method-for-detecting-plot/ (retrieved 07.03.2021).

Jockers, Matthew L. (2015): *Syuzhet: Extract Sentiment and Plot Arcs from Text*, https://github.com/mjockers/syuzhet (retrieved 26.03.2021)

Johnson, Scott (29.03.2017): *How Trump Adviser Stephen Miller Divided a Santa Monica Synagogue,* https://www.hollywoodreporter.com/news/how-trump-adviser-stephen-miller-divided-a-santa-monica-synagogue-989250 (retrieved 22.03.2021)

Kranz, Michael; Cranley, Ellen (15.11.2019): *Meet Stephen Miller, the 34-year-old White House adviser who's being called to resign after leaked emails showed him sharing white supremacist links*, in: *Business Insider*, https://www.businessinsider.de/international/who-is-stephen-miller-trump-speechwriter-immigration-adviser-2018-1/?r=US&IR=T (retrieved 16.03.2021)

Van der Maaten, Laurens; Hinton, Geoffrey (2008): *Visualizing Data using t-SNE*, in: *Journal of Machine Learning Research* 9 (2008), https://jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf (retrieved 23.08.2021)

Kincaid, J.P.; Fishburne, R.P.; Rogers, R.L.; Chissom, B.S. (1975): *Derivation of new readability formulas (automated readability index, fog count, and flesch reading ease formula) for Navy enlisted personnel*, in: *Research Branch Report*, pp. 8–75, Chief of Naval Technical Training: Naval Air Station Memphis

Kreis, Ramona (2017): *Right-Wing Populism in Europe & USA*, in: *JLP 16*, https://www.jbe-platform.com/content/journals/10.1075/jlp.17032.kre (retrieved 22.03.2021)

Lemmerich, Julian (2020): *President Obama's Speeches. Corpus, Quantitative Analysis and Authorship Attribution*, Technische Universität Darmstadt, unpublished manuscript

Lemmerich, Julian (2021a): *Trump Speech Corpus*, Technische Universität Darmstadt, unpublished manuscript

Lemmerich, Julian (2021b): *Sentimentanalyse. Barack Obamas und Donald Trumps Reden im Vergleich*, Technische Universität Darmstadt, unpublished manuscript

Liu, Bing; Hu, Minqing; Cheng, Junsheng (2005): *Opinion observer*, in: *Proceedings of the 14th international conference on World Wide Web - WWW '05*, p. 342, doi: 10.1145/1060745.1060797 (retrieved 26.03.2021)

Liu, Dilin; Lei, Lei (2018): *The appeal to political sentiment: An analysis of Donald Trump's and Hillary Clinton's speech themes and discourse strategies in the 2016 US presidential election*, in: *Discourse, Context & Media 25*. pp. 143–152, doi: 10.1016/j.dcm.2018.05.001

Mango News (17.08.2015): *Donald Trump: Obama is Teleprompter Guy. We Dont Want Scripted President*, https://www.youtube.com/watch?v=hs5woj5Ae48 (retrieved 22.03.2021)

Parker, A.: *The New Team: Jonathan Favreau*, in: *The New York Times* (05.12.2008)

Peinado, Fernando: *How White House advisor Stephen Miller went from pestering Hispanic students to designing Trump's immigration policy*, in: *Univision News* (08.02.2017), https://www.univision.com/univision-news/politics/how-white-house-advisor-stephen-miller-went-from-pestering-hispanic-students-to-designing-trumps-immigration-policy (retrieved 22.03.2021)

Pilkington, E.: *Obama inauguration: Words of history … crafted by 27-year-old in Starbucks*, in: *The Guardian* (20.01.2009), https://www.theguardian.com/world/2009/jan/20/barack-obama-inauguration-us-speech (retrieved 10.10.2020)

Samuels, D.: *The Aspiring Novelist Who Became Obama's Foreign-Policy Guru*. in: *The New York Times Magazine* (05.05.2016), https://www.nytimes.com/2016/05/08/magazine/the-aspiring-novelist-who-became-obamas-foreign-policy-guru.html (retrieved 10.10.2021)

Schmidt, Thomas; Burghardt, Manuel; Dennerlein, Katrin (2018): *Kann man denn auch nicht lachend sehr ernsthaft sein? Zum Einsatz von Sentiment Analyse-Verfahren für die quantitative Untersuchung von Lessings Dramen*, in: *Book of Abstracts, DHd 2018*.

scikit-lean developers: *sklearn.feature_extraction.text.TfidfVectorizer*, https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer.fit_transform (retrieved 23.08.2021)

scikit-yb developers (13.02.2020): *t-SNE Corpus Visualization*. in: *Yellowbrick* https://www.scikit-yb.org/en/latest/api/text/tsne.html (retrieved 23.08.2021)

Scott, Mark (20.07.2015): *Remove All Rows Containing Certain Data*, http://excelzoom.com/remove-all-rows-containing-certain-data/ (retrieved 09.07.2021)

Simmler, Severin; Vitt, Thorsten; Pielström, Steffen (2019): *Topic Modeling with Interactive Visualizations in a GUI Tool*, in: *Proceedings of the Digital Humanities Conference*, https://dev.clariah.nl/files/dh2019/boa/0637.html

Singham, Luke (17.01.2021): *How to Make a Wordcloud Using R*, https://lukesingham.com/how-to-make-a-word-cloud-using-r/ (retrieved 18.07.2021)

Steinhauer, Jennifer; Thrush, Glenn: *Once Seen as Mere Gadfly, 'True Believer' Now Shapes Key Trump Policies*, in: *The New York Times* (12.02.2017), p. 20, https://www.nytimes.com/2017/02/11/us/politics/stephen-miller-donald-trump-adviser.html (retrieved 22.03.2021)

STHDA: *Text mining and word cloud fundamentals in R : 5 simple steps you should know*, http://www.sthda.com/english/wiki/text-mining-and-word-cloud-fundamentals-in-r-5-simple-steps-you-should-know (retrieved 18.07.2021)

Swafford, Annie (02.03.2015): *Problems with the Syuzhet Package*, https://annieswafford.wordpress.com/2015/03/02/syuzhet/ (retrieved 25.08.2021)

Tausczik, Yla R.; Pennebaker, James W. (2010): *The Psychological Meaning of Words: LIWC and Computerized Text Analysis Methods*, in: *Journal of Language and Social Psychology 29*, pp. 24–54, doi: 10.1177/0261927X09351676 (retrieved 26.03.2021).

Thelwall, M.; Buckley, K.; Paltoglou, G.; Cai, D.; Kappas, A. (2010): *Sentiment strength detection in short informal text*, in: *Journal of the American Society for Information Science and Technology* 61, pp. 2544–2558, http://www.scit.wlv.ac.uk/~cm1993/papers/SentiStrengthPreprint.doc (retrieved 25.03.2021)

Toutanova, Kristina; Klein, Dan; Manning, Christopher D.; Singer, Yoram: *Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network*, in: *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics* (2003), pp. 252–259, https://www.aclweb.org/anthology/N03-1033, doi: 10.3115/1117794.1117802.

Toutanova, Kristina; Manning, Christopher D: *Stanford Part-Of-Speech Tagger 2020*, https://nlp.stanford.edu/software/tagger.shtml (retrieved 20.03.2021)

Toutanova, Kristina; Manning, Christopher D.: *Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger, in: Proceedings of the 2000 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: Held in Conjunction with the 38th Annual Meeting of the Association for Computational Linguistics - Volume 13* (2000), pp. 63–70, doi: 10.3115/1117794.1117802.

Tripathi, Mayank (06.06.2018): *How to process textual data using TF-IDF in Python*. in: *FreeCodeCamp*, https://www.freecodecamp.org/news/how-to-process-textual-data-using-tf-idf-in-python-cd2bbc0a94a3/ retrieved 23.08.2021)

Trump, Donald J.(02.06.2016): *Bad performance by Crooked Hillary Clinton! Reading poorly from the telepromter! She doesn't even look presidential!,* https://twitter.com/realDonaldTrump/status/738449664752553984 (retrieved 07.01.2021), now available at https://web.archive.org/web/20210107005209/ https://twitter.com/realDonaldTrump/status/738449664752553984 (retrieved 22.03.2021).

Trump, Donald J. (2016): *We will immediately repeal and replace ObamaCare - and nobody can do that like me. We will save $'s and have much better healthcare! Twitter*. https://twitter.com/realDonaldTrump/status/697182075045179392 (retrieved 14.11.2016), now available at http://web.archive.org/web/20161114184728if_/https://twitter.com/realDonaldTrump/status/697182075045179392 (retrieved 09.08.2021)

Trump-Pence Presidential Transition Team (13.12.2016): *President-Elect Donald J. Trump Appoints Stephen Miller as Senior Advisor to the President For Policy*, New York https://www.presidency.ucsb.edu/node/320012 (retrieved 22.03.2021)

Trump-Pence Presidential Transition Team (05.01.2017): *President-Elect Donald J. Trump Transition Builds Out White House Policy Team,* New York, https://www.presidency.ucsb.edu/node/321136 (retrieved 22.03.2021)

Landwehr, Arthur (2020): *Trump schafft mit Truppenabzug Fakten*, in: *tagesschau.de* (17.11.2020), https://www.tagesschau.de/ausland/us-abzug-afghanistan-105.html (retrieved 07.08.2021)

LinkedIn: *Adam Frankel*, https://www.linkedin.com/in/adam-frankel-1721b715/ (retrieved 21.10.2020)

LinkedIn: *Kyle O'Connor*, https://www.linkedin.com/in/kyle-o-connor-2230b896/ (retrieved 10.10.2020)

LinkedIn: *Ryan Jarmula. Deputy Chief of Staff & District Director at Congressman Greg Pence*, https://www.linkedin.com/in/ryan-jarmula-a76a141b (retrieved 22.03.2021)

LinkedIn: *Vince Haley. Richmond, Virginia Area*, https://www.linkedin.com/in/vince-haley-220b842 (retrieved 22.03.2021)

Lippman, Daniel; Toosi, Nahal (2019): *Boris and Donald: A very special relationship*, in: *Politico* (12.12.2019), https://www.politico.com/news/2019/12/12/trump-boris-johnson-relationship-083732 (retrieved 07.08.2021)

Luther, Carsten: *Nato will Donald Trump mit höheren Verteidigungsausgaben besänftigen*, in: *Zeit Online* (29.11.2019), https://www.zeit.de/politik/ausland/2019-11/nato-gipfel-donald-trump-verteidigungsausgaben-zahlen (retrieved 07.08.20219)

McDonough, John E. (2012): *ACA vs. ObamaCare: What's In a Name?* In *boston.com* (14.01.2012), http://archive.boston.com/lifestyle/health/health_stew/2012/01/aca_vs_obamacare_whats_in_a_na.html (retrieved 09.082021)

Milbank, Dana: *Trump's fake-news presidency*, in: *The Washington Post* (18.11.2016), https://www.washingtonpost.com/opinions/trumps-fake-news-presidency/2016/11/18/72cc7b14-ad96-11e6-977a-1030f822fc35_story.html (retrieved 18.03.2021)

Neuwirth, Erich (2014): *RColorBrewer: ColorBrewer Palettes*, R package version 1.1-2, https://CRAN.R-project.org/package=RColorBrewer

Nussbaum, Matthew: *Trump and the teleprompter: A brief history*, in: *POLITICO* (06.07.2016), https://www.politico.com/story/2016/06/donald-trump-teleprompter-224039 (retrieved 22.03.2021)

ProPublica: *Gingrich Productions, Inc.*, https://projects.propublica.org/trump-town/organizations/gingrich-productions-inc (retrieved 22.03.2021)

ProPublica: *Ross P. Worthington* https://projects.propublica.org/trump-town/staffers/ross-p-worthington (retrieved 22.03.2021)

ProPublica: *Ross Worthington*, https://projects.propublica.org/trump-town/staffers/ross-worthington (retrieved 22.03.2021)

ProPublica: *Tyan Jarmula*, https://projects.propublica.org/trump-town/staffers/ryan-jarmula (retrieved 22.03.2021)

ProPublica: *Vincent M. Haley*, https://projects.propublica.org/trump-town/staffers/vincent-m-haley (retrieved 22.03.2021)

Reuters: *Trump says NATO is obsolete but still 'very important to me'*, in: *Reuters* (15.01.2017), https://www.reuters.com/article/us-usa-trump-nato-obsolete-idUSKBN14Z0YO (retrieved 07.08.2021)

Rogers, Katie: *The State of the Union Is Trump's Biggest Speech. Who Writes It?*, in: *The New York Times* (02.03.2020), https://www.nytimes.com/2020/02/03/us/politics/trump-state-of-the-union.html (retrieved 22.03.2021)

Santorini, Beatrice: *Part-of-Speech Tagging Guidelines for the Penn Treebank Project* (06.1990), https://catalog.ldc.upenn.edu/docs/LDC99T42/tagguid1.pdf (retrieved 20.03.2021)

Savoy, Jacques (2018a): *Analysis of the style and the rhetoric of the 2016 US presidential primaries*, in: *Digital Scholarship in the Humanities* 33, pp. 143–159, https://academic.oup.com/dsh/article/33/1/143/2993886, doi: 10.1093/llc/fqx007 (retrieved 06.03.2021)

Savoy, Jacques (2018b): *Trump's and Clinton's Style and Rhetoric during the 2016 Presidential Election*, in: *Journal of Quantitative Linguistics* 25, pp. 168–189, doi: 10.1080/09296174.2017.1349358 (retrieved 03.03.2021)

Scheuermann, Christoph; Hebel, Christina (2018): *Ziemlich neue Freunde*, in: *Spiegel Online*, 16.07.2018, https://www.spiegel.de/politik/ausland/donald-trump-und-wladimir-putin-anfang-einer-freundschaft-a-1218790.html (retrieved 07.08.2021)

Vote Smart: *Vincent Haley's Biography,* https://justfacts.votesmart.org/candidate /biography/156457/vincent-haley (retrieved 22.03.2021)

Vrana, Leo; Schneider, Gerold (2017): *Saying Whatever It Takes: Creating and Analyzing Corpora from US Presidential Debate Transcripts*, doi: 10.5167/uzh-145668, (retrieved 04.02.2021)

Walker, T. (06.02.2013). *Jon Favreau: From White House to silver screen*, in: *Independent*, https://www.independent.co.uk/news/world/americas/jon-favreau-white-house-silver-screen-8483994.html (retrieved 10.10.2020)

West Wing Writers. (06.2013). *West Wing Writers Welcomes New Staff, Congratulates Alumni Named to Obama Team*, http://www.westwingwriters.com/news/6-24-2013 (retrieved 10.10.2020)

White House Historical Association: *Donald Trump. THE 45TH PRESIDENT OF THE UNITED STATES*, https://www.whitehouse.gov/about-the-white-house/presidents/donald-j-trump/ (retrieved 25.03.2021)

Wickham, Hadley (2016): *ggplot2: Elegant Graphics for Data Analysis*

Wickham, Hadley (2019): *stringr: Simple, Consistent Wrappers for Common String Operations*, R package version 1.4.0, https://CRAN.R-project.org/package=stringr

Wickham, Hadley; François, Romain; Henry, Lionel; Müller, Kirill (2021): *dplyr: A Grammar of Data Manipulation*, R package version 1.0.7, https://CRAN.R-project.org/package=dplyr

Wickham, Hadley; Hester, Jim (2021): *readr: Read Rectangular Text Data*, R package version 2.0.0, https://CRAN.R-project.org/package=readr

Wikipedia (10.08.2021): *Barack Obama*, https://en.wikipedia.org/wiki/Barack_Obama (retrieved 10.08.2021)

Wolf, Zach Byron: *Trump breaks his own rule, uses teleprompter.*, in: *CNNPolitics* (22.03.2016), https://edition.cnn.com/2016/03/21/politics/trump-teleprompter-aipac-speech/index.html (retrieved 22.03.2021)

Xie, Yihui (2014): *knitr: A Comprehensive Tool for Reproducible Research in R*, in: Stodden, Victoria; Leisch, Friedrich; Peng, Roger D.: *Implementing Reproducible Computational Research*. Chapman and Hall/CRC. ISBN 978-1466561595

Xie, Yihui (2015): *Dynamic Documents with R and knitr*. 2nd edition. Chapman and Hall/CRC. ISBN 978-1498716963

Xie, Yihui (2021): *knitr: A General-Purpose Package for Dynamic Report Generation in R*, R package version 1.33.

Zhu, Hao (2021): *kableExtra: Construct Complex Table with 'kable' and Pipe Syntax*, R package version 1.3.4, https://CRAN.R-project.org/package=kableExtra

8. List of Figures

8.1. Diagrams

8.2. Tables

8.3. Plots

9. Appendix

9.1. Biographies

9.1.1. Speechwriters of Barack Obama[78]

Over the years of his presidency Obama employed a team of 7 speechwriters. These were Adam Frankel, Jon Favreau, Cody Keenan, Ben Rhodes, Jon Lovett, David Litt and Kyle O'Connor.

Jon Favreau was Obama's first *White House Director of Speechwriting*. He was born in 1981 in Winchester, Massachusetts, US. He holds a degree in political science from the Jesuit *College of the Holy Cross* in Worchester, Massachusetts.[79] He was *Director of Speechwriting* from 2009 until 2013. But even before that, Favreau was part of Obama's Team. In 2005, at the age of 23, he began working for Obama, who was then a senator. In 2006 he started writing for Obama's presidential campaign[80] together with Cody Keenan and Ben Rhodes. Obama called Favreau his "mind reader".[81] Favreau himself refers to Robert F. Kennedy and Michael Gerson as influences to his writing style and a speech by Peggy Noonan, Ronald Reagan's speechwriter from 1984 to 1986, as his favorite speech.[82]

Adam Frankel was the second speechwriter hired for Obama's presidential campaign in 2007. He graduated from *Princeton University*, New Jersey, in 2003 and from *The London School of Economics and Political Science* with a master's degree in Theory and History of International Relations. In the White House his official title was *Special Assistant to the President and Senior Presidential Speechwriter* until 2011.[83]

After Favreau left the White House in March 2013, **Cody Keenan**, up until then the *Deputy Director of Speechwriting*, took over the position of *White House Director of Speechwriting*. Cody Keenan was born in 1980 in Chicago, Illinois, US. In 2008 he graduated from the *Kennedy School of Government* at *Harvard* with a master's degree in public policy and directly joined the White House at the age of 23, after interning there for Jon Favreau in the previous year.[84]

[78] Chapter 2 in Lemmerich, Julian (2020): *President Obama's Speeches. Corpus, Quantitative Analysis and Authorship Attribution*, Technische Universität Darmstadt, unpublished manuscript

[79] Parker, A. (05.12.2008): *The New Team: Jonathan Favreau*, in: *The New York Times*

[80] Change.gov (26.11.2008): *President-Elect Barack Obama names two new White House staff members*, in: *Change.gov - The Office of the President-Elect*, https://web.archive.org/web/20081126191037/http://change.gov/newsroom/entry/president_elect_barack_obama_names_two_new_white_house_staff_members/

[81] Pilkington, E. (20.01.2009): *Obama inauguration: Words of history ... crafted by 27-year-old in Starbucks*, in: *The Guardian*, https://www.theguardian.com/world/2009/jan/20/barack-obama-inauguration-us-speech (retrieved 10.10.2020)

[82] Walker, T. (06.02.2013): *Jon Favreau: From White House to silver screen*, in: *Independent*, https://www.independent.co.uk/news/world/americas/jon-favreau-white-house-silver-screen-8483994.html (retrieved 10.10.2020)

[83] In this case LinkedIn, which is usually not used in scientific context, provides a suitable source for prior employment and education: LinkedIn: *Adam Frankel*, https://www.linkedin.com/in/adam-frankel-1721b715/ (retrieved 21.10.2020)

[84] Felsenthal, C. (19.02.2013): *Cody Keenan, Obama's Chief Speechwriter: Chicago-Born and (Mostly) Bred*, in: *Chicago Magazine*, https://www.chicagomag.com/Chicago-Magazine/Felsenthal-Files/February-2013/Cody-Keenan-Obamas-Chief-Speechwriter-Chicago-Born-and-Mostly-Bred/ (retrieved 10.10.2020)

Ben Rhodes's official position in the White House was *Deputy National Security Advisor for Strategic Communications*. He was born in 1977 in New York City, US. He graduated from *Rice University* in 2000 with a major in English and Political Science followed by a Master of Fine Arts in creative writing from New York University in 2002.[85] He joined Obama's Team on his campaign trail in 2007 at the age of 30. As *Deputy National Security Advisor* he often travelled for negotiations with foreign nations. He wrote and co-wrote a lot of the speeches pertaining to foreign policy.[86]

Jon Lovett was born in 1982 in Woodbury, New York, US. He graduated from *Williams College* in Massachusetts in 2004 with a degree in mathematics. He was hired as assistant to Hillary Clintons speechwriter in 2005 and joined the presidential speechwriters at the White House at the age of 26 after Obama's successful campaign for president. His most notable speeches were about the financial reform.[87]

David Litt was born in New York City in 1987 and attended Yale University, where he was also editor-in-chief of the Yale Record. He first got in contact with the White House in an internship program and was later hired as writer on Obama's reelection campaign in 2012 at the age of 25 and then joined the team of presidential speechwriters at the White House in 2013.[88]

Kyle O'Connor joined the White House as *Assistant Speechwriter* in 2009 and was promoted to presidential speechwriter in 2011. After Favreau left in 2013 and Cody Keenan took over the position of *White House Director of Speechwriting*, Kyle O'Connor became *Deputy Director of Speechwriting*. He studied political philosophy, policy and law at the *University of Virginia*.[89]

There are a few similarities in the team of Obama's speechwriters. They started writing speeches for him at a young age – most of them shortly after graduating from University in their 20s and being born in the span of one decade – 1977, '80, '81, '82 and '87. Their linguistically formative years were spent in similar regions to Obama's – New York and Harvard most prominently. All of Barack Obama's writers were young white men, while the rest of Obama's team was a lot more diverse. Notable is that for example Ronald Reagan had women and Michelle Obama had men in her speechwriting team. All these factors in Obama's team may increase similarities in their writing, which is a positive trait for a team of ghostwriters trying to create a unified writing style.

[85] Collegiate School (27.10.2008): *Election 2008: Ben Rhodes '96, Speechwriter and Advisor to Barack Obama*, https://www.collegiateschool.org/news-detail?pk=453880 (retrieved 10.10.2020)

[86] Samuels, D. (05.05.2016): *The Aspiring Novelist Who Became Obama's Foreign-Policy Guru*, in: *The New York Times Magazine*, https://www.nytimes.com/2016/05/08/magazine/the-aspiring-novelist-who-became-obamas-foreign-policy-guru.html (retrieved 10.10.2020)

[87] Horowitz, J. (02.09.2011): *Jon Lovett's written for the president, but will that get him to Hollywood?*, in: *Washington Post*, https://www.washingtonpost.com/lifestyle/style/jon-lovetts-written-for-the-president-but-will-that-get-him-to-hollywood/2011/08/22/gIQAhZmIxJ_story.html (retrieved 10.10.2020)

[88] West Wing Writers. (24.06.2013): *West Wing Writers Welcomes New Staff, Congratulates Alumni Named to Obama Team*, http://www.westwingwriters.com/news/6-24-2013 (retrieved 10.10.2020)

[89] In this case too LinkedIn, which is usually not used in scientific context, provides a suitable source for prior employment and education: LinkedIn: *Kyle O'Connor*, https://www.linkedin.com/in/kyle-o-connor-2230b896/ (retrieved 10.10.2020)

9.1.2. Speechwriters of Donald Trump[90]

Trump doesn’t want his speechwriters to be known to the public, because he presents himself as an independent and eloquent speaker.[91] He heavily criticized his predecessor Barack Obama and his opponent Hillary Clinton for reading from a teleprompter and having assistance in writing the speeches: “She’s just reading it off a teleprompter. Believe me, they write that for her,” and “She doesn't even look presidential [reading off a teleprompter]!”, Trump said of Clinton.[92] He even went as far as repeatedly saying that teleprompter should not be allowed if you are running for president.[93] When asked in a press conference, if he was writing his speeches himself, he answered, that he thinks about them himself: »I think about my speeches a lot. Essentially, I don’t use notes and I definitely don’t read the speeches. [...] I do a lot of things by myself. [...] People are shocked at how smart I am.«[94]

All of this shows that Trump is very proud of his unscripted speaking style. He wants to present an image of coming up with his speeches himself. Thus, Trump tries to hide any assistance he gets in the writing process of his speeches. This makes research into his speechwriters rather complicated. Not a lot of information is public about them. Administration officials have declined requests by journalists to talk about the speechwriting process.[95]

Stephen Miller was *White House Director of Speechwriting* for the full term of Donald Trump’s presidency. His role as *Senior Advisor to the President* was however more prominent.[96]

He was born in 1985 in California as the son of a liberal-leaning Jewish family.[97] Between middle school and high school he underwent a political radicalization.[98] In high school he started appearing on conservative radio talk shows, where he also met David Horowitz,[99] an

[90] Chapter 2.3.2 and 2.3.3 of Lemmerich, Julian (2021a): *Trump Speech Corpus*, Technische Universität Darmstadt, unpublished manuscript

[91] Lt. Gen. Singh, Bhopinder: *When the Trumpets blare: Language of Statesmanship*, in: *DH News Service* (26.02.2020)

[92] Nussbaum, Matthew: *Trump and the teleprompter: A brief history*, in: *POLITICO* (06.07.2016)

Donald J. Trump (02.06.2016): *Bad performance by Crooked Hillary Clinton! Reading poorly from the telepromter! She doesn't even look presidential!*, cited by Nussbaum (2016).

[93] Tim Hains: *Trump: "If You're Running For President You Shouldn't Be Allowed To Use A Teleprompter*", in: *RealClear Politics* (25.08.2015)

Nussbaum (2016)

Wolf, Zach Byron: *Trump breaks his own rule, uses teleprompter.*, in: *CNNPolitics* (22.03.2016).

[94] Hains (2015).

Mango News (17.08.2015): *Donald Trump: Obama is Teleprompter Guy. We Dont Want Scripted President*.

[95] Katie Rogers: *The State of the Union Is Trump’s Biggest Speech. Who Writes It?*, in: *The New York Times* (02.03.2020)

[96] Trump-Pence Presidential Transition Team (13.12.2016): *President-Elect Donald J. Trump Appoints Stephen Miller as Senior Advisor to the President For Policy*

[97] Kranz, Michael; Cranley, Ellen: *Meet Stephen Miller, the 34-year-old White House adviser who’s being called to resign after leaked emails showed him sharing white supremacist links*, in: *Business Insider* (15.11.2019)

[98] Peinado, Fernando: *How White House advisor Stephen Miller went from pestering Hispanic students to designing Trump's immigration policy*, in: *Univision News* (08.02.2017)

[99] Johnson, Scott: *How Trump Adviser Stephen Miller Divided a Santa Monica Synagogue* (29.03.2017)

anti-muslim and anti-immigrant extremist, who from then on was a mentor and influential figure in his early life.[100] Miller was known for “riling up his fellow classmates with controversial statements”.[101] In 2007, at the age of 22, he graduated with a Bachelor’s degree in Political Science from *Duke University*, where he also wrote for the school’s newspaper. His column “Miller Time” got national awareness for its controversial positions.[102]

After college Miller was recommended to Tea-Party Republican Michele Bachmann by Horowitz and he began working as press secretary for her. In 2009, again recommended by Horowitz, he began working for later *US Attorney General* Jeff Sessions, climbing up to be his communication’s director.[103] In this role, he wrote many of Sessions’ speeches in Congress against a proposed immigration reform bill.[104]

In 2016 Miller joined Donald Trump’s presidential campaign as policy advisor by recommendation of Jeff Sessions. He was *White House Director of Speechwriting*, but his main role was *Senior Advisor* in multiple roles, shaping first domestic policy then immigration policy.[105]

Besides Stephen Miller, three more advisors helped in speechwriting for Donald Trump.[106] They remained rather unknown from the public throughout Trump’s whole presidency.

Vincent M. Haley[107]. *Advisor for Policy, Strategy and Speechwriting*: Haley holds an undergraduate degree from the *College of William & Mary*, a Law and Master's degree from the *University of Virginia*, and a Master of Laws in Foreign Affairs and European Union Law from the *College of Europe*. During the president-elect's successful campaign, Haley developed ethics

[100] Helderman, Rosalind S.: *Stephen Miller: A key engineer for Trump’s ‘America first’ agenda*, in: *The Washington Post* (11.02.2017)

[101] Brennan, Christopher: *SEE IT: Trump adviser Stephen Miller booed off stage by classmates after high school speech*, in: *New York Daily News* (15.02.2017)

[102] Ioffe, Julia: *The Believer. How Stephen Miller went from obscure Capitol Hill staffer to Donald Trump’s warm-up act—and resident ideologue.*, in: *POLITICO* (27.06.2016)

[103] Guerrero, Jean: *The Man Who Made Stephen Miller*, in: *POLITICO Magazine* (01.08.2020)

[104] Steinhauer, Jennifer; Thrush, Glenn: *Once Seen as Mere Gadfly, ‘True Believer’ Now Shapes Key Trump Policies*, in: *The New York Times* (12.02.2017)

[105] Ibid.

[106] Trump-Pence Presidential Transition Team (05.01.2017): *President-Elect Donald J. Trump Transition Builds Out White House Policy Team*

[107] Called *“Vince Haley”* instead of *“Vincent Haley”* by the Trump-Administration Press-releases.

ProPublica (2017): *Vincent M. Haley*

reform policies.[108] Amongst other things he worked on the State of the Union Speech in 2020 together with Ross Worthington.[109]

Ross P. Worthington. *Advisor for Policy, Strategy and Speechwriting*: Before his position in the White House, Worthington served alongside Haley[110] as *Research Director*, *Deputy Communications Director* and *Primary Writer* for Republican Newt Gingrich. He is a graduate of Brown University, where he concentrated in Political Theory.[111] In 2018 he became *Deputy Assistant to the President and Advisor for Policy, Strategy and Speechwriting*.[112] He worked under Stephen Miller and together with Harley since the early days of the Trump campaign.[113]

Ryan Jarmula. *Advisor for Policy Development and Speechwriting*: Ryan Jarmula graduated from the *Indiana University Bloomington* in 2007 with a Bachelor's degree in Political Science. Before joining Trump's campaign, he worked on then-congressman Mike Pence's staff, first as assistant, then as speechwriter. He served as *Special Assistant* to President Donald Trump for Speechwriting and Policy Development until January of 2019.[114]

[108] Finding Sources was not very easy. There are a few "politician transparency" websites, that have information on Haley, but they do not cite any sources for these. The information of these three sources overlapped mostly:

LinkedIn: *Vince Haley. Richmond, Virginia Area*

Vote Smart: *Vincent Haley's Biography*

Trump-Pence Presidential Transition Team (05.01.2017): *President-Elect Donald J. Trump Transition Builds Out White House Policy Team*

[109] Rogers, Katie: *The State of the Union Is Trump's Biggest Speech. Who Writes It?*, in: *The New York Times* (02.03.2020)

[110] ProPublica (2017): *Gingrich Productions, Inc.*

[111] Trump-Pence Presidential Transition Team (05.01.2017): *President-Elect Donald J. Trump Transition Builds Out White House Policy Team*

ProPublica (2017): *Ross Worthington*

[112] The White House (06.09.2018): *President Donald J. Trump Announces Appointments for the Executive Office of the President*

ProPublica (2017): *Ross P. Worthington*

[113] Rogers (2020)

[114] LinkedIn: *Ryan Jarmula. Deputy Chief of Staff & District Director at Congressman Greg Pence*

Trump-Pence Presidential Transition Team (2017)

ProPublica, *Tyan Jarmula.* 2017

9.2. Differences to COCA (to 4.2.)

This Code is no longer needed, since the whole calculation of differences in word frequencies was cut from the paper. The Code might still be interesting to some.

The chapter was cut, because differences of relative frequency can often lead quite nieche and not very important words to get very high onto these difference lists. This is amplified by only having the top 5000 most frequent words from the COCA corpus and a different tokenization in the COCA corpus, which makes matching the words up to each other quite difficult. (This code does not rectify the issue.) Instead this comparison was then done in 4.2.1 manually and not by the computer. The interesting and characteristic words were extracted by close reading of the lists.

Code: calculating differences to COCA

```
obamafreq <- read.csv(file.choose())
trumpfreq <- read.csv(file.choose())
cocafreq <- read.csv(file.choose())

#since the percent from csv is read as string not as number, it needs to be co
nverted first
obamafreq$percent <- as.numeric(sub("%","",obamafreq$percent))/100
trumpfreq$percent <- as.numeric(sub("%","",trumpfreq$percent))/100
cocafreq$percent <- as.numeric(sub("%","",cocafreq$percent))/100

## Obama

diffwords <- c()
diffnr <- c()
absolutewords <- c()

for (i in 1:length(obamafreq$words)) {
  diffwords <- c(diffwords, obamafreq$words[i])
  absolutewords <- c(absolutewords, obamafreq$word.freq[i])
  difference = cocafreq$percent[match(obamafreq$words[i], cocafreq$word)] - ob
amafreq$percent[i]
  diffnr <- c(diffnr, difference)
}

obama.diffdata <- data.frame(diffwords, diffnr, absolutewords)
write.csv(obama.diffdata, file.choose())

## Trump

#reset
diffwords <- c()
diffnr <- c()
absolutewords <- c()

for (i in 1:length(trumpfreq$word)) {
  diffwords <- c(diffwords, trumpfreq$word[i])
  absolutewords <- c(absolutewords, trumpfreq$word.freq[i])
  difference = cocafreq$percent[match(trumpfreq$word[i], cocafreq$word)] - tru
mpfreq$percent[i]
  diffnr <- c(diffnr, difference)
}

trump.diffdata <- data.frame(diffwords, diffnr, absolutewords)
write.csv(trump.diffdata, file.choose())

# positive difference means used more often, negative means used less often
```

9.3. Stopwords

There are different versions of the stopword list.

9.3.1. Stopwords v1

The original one (“v1”) is unchanged from Katharina Herget.[115]

List

```
a
able
about
above
abst
accordance
according
accordingly
across
act
actually
added
adj
adopted
affected
affecting
affects
after
afterwards
again
against
ah
all
almost
alone
along
already
also
although
always
am
among
amongst
an
and
announce
another
any
anybody
anyhow
anymore
anyone
anything
anyway
anyways
anywhere
apparently
approximately
are
aren
arent
arise
around
as
aside
ask
asking
at
auth
available
away
awfully
b
back
be
became
because
become
becomes
becoming
been
before
beforehand
begin
beginning
beginnings
begins
behind
being
believe
below
beside
besides
between
beyond
biol
both
brief
briefly
but
by
c
ca
came
can
cannot
can't
cause
causes
certain
certainly
co
com
come
comes
contain
containing
contains
could
couldnt
d
date
did
didn't
different
do
does
doesn't
doing
done
don't
down
downwards
due
during
e
each
ed
edu
effect
eg
eight
eighty
either
else
elsewhere
end
ending
enough
especially
et
et-al
etc
even
ever
every
everybody
everyone
everything
everywhere
ex
except
f
far
few
ff
fifth
first
five
fix
followed
following
follows
for
former
formerly
forth
found
four
from
further
furthermore
g
gave
get
gets
getting
give
given
gives
giving
go
goes
gone
got
gotten
h
had
happens
hardly
has
hasn't
have
haven't
having
he
hed
hence
her
here
hereafter
hereby
herein
heres
hereupon
hers
herself
hes
hi
hid
him
himself
his
hither
home
how
howbeit
however
hundred
i
id
ie
if
i'll
im
immediate
immediately
importance
important
in
inc
indeed
index
information
instead
into
invention
inward
is
isn't
it
itd
it'll
its
itself
i've
j
just
k
keep
keeps
kept
keys
kg
km
know
known
knows
l
largely
last
lately
later
latter
latterly
least
less
lest
let
lets
like
liked
likely
line
little
'll
look
looking
looks
ltd
m
made
mainly
make
makes
```

[115] Part of course materials for 02-15-1053-gk *Grundkurs Literaturwissenschaft mit Profil Digital Philology* at Technische Universität Darmstadt WS 2017/2018 by Katharina Herget.

```
many
may
maybe
me
mean
means
meantime
meanwhile
merely
mg
might
million
miss
ml
more
moreover
most
mostly
mr
mrs
much
mug
must
my
myself
n
na
name
namely
nay
nd
near
nearly
necessarily
necessary
need
needs
neither
never
nevertheless
new
next
nine
ninety
no
nobody
non
none
nonetheless
noone
nor
normally
nos
not
noted
nothing
now
nowhere
o
obtain
obtained
obviously
of
off
often
oh
ok
okay
old
omitted
on
once
one
ones
only
onto
or
ord
other
others
otherwise
ought
our
ours
ourselves
out
outside
over
overall
owing
own
p
page
pages
part
particular
particularly
past
per
perhaps
placed
please
plus
poorly
possible
possibly
potentially
pp
predominantly
present
previously
primarily
probably
promptly
proud
provides
put
q
que
quickly
quite
qv
r
ran
rather
rd
re
readily
really
recent
recently
ref
refs
regarding
regardless
regards
related
relatively
research
respectively
resulted
resulting
results
right
run
s
said
same
saw
say
saying
says
sec
section
see
seeing
seem
seemed
seeming
seems
seen
self
selves
sent
seven
several
shall
she
shed
she'll
shes
should
shouldn't
show
showed
shown
showns
shows
significant
significantly
similar
similarly
since
six
slightly
so
some
somebody
somehow
someone
somethan
something
sometime
sometimes
somewhat
somewhere
soon
sorry
specifically
specified
specify
specifying
state
states
still
stop
strongly
sub
substantially
successfully
such
sufficiently
suggest
sup
sure
t
take
taken
taking
tell
tends
th
than
thank
thanks
thanx
that
that'll
thats
that've
the
their
theirs
them
themselves
then
thence
there
thereafter
thereby
thered
therefore
therein
there'll
thereof
therere
theres
thereto
thereupon
there've
these
they
theyd
they'll
theyre
they've
think
this
those
thou
though
thoughh
thousand
throug
through
throughout
thru
thus
til
tip
to
together
too
took
toward
towards
tried
tries
truly
try
trying
ts
twice
two
u
un
under
unfortunately
unless
unlike
unlikely
until
unto
up
upon
ups
us
use
used
useful
usefully
usefulness
uses
using
usually
v
value
various
've
very
via
viz
vol
vols
vs
w
want
wants
was
wasn't
way
we
wed
welcome
we'll
went
were
weren't
we've
what
whatever
what'll
whats
when
whence
whenever
where
whereafter
whereas
whereby
wherein
wheres
whereupon
wherever
whether
which
while
whim
whither
who
whod
whoever
whole
who'll
whom
whomever
whos
whose
why
widely
willing
wish
with
within
without
won't
words
world
would
wouldn't
www
x
y
yes
yet
```

```
you
youd
you'll
your
youre
yours
yourself
yourselves
you've
z
zero
```

9.3.2. Stopwords v2

This iteration added only 4 entries in addition to the existing v1 list. Depending on the tokenization, they were sometimes listed as their own words instead of as part of a longer word.

```
's
'd
're
'm
```

9.3.3. Stopwords v3

This iteration was made specifically for topic modelling with the DARIAH topic explorer. It adds the following entries (in addition to v2):

```
it's
we're
that's
they're
i'm
he's
they're
you're
```

9.4. Word Frequency Tables by Year

9.4.1. Obama

2004, 2005, 2006

1	52	will
2	36	america
3	34	people
4	32	rights
5	29	american
6	24	parks
7	23	country
8	23	life
9	23	work
10	22	time
11	21	john
12	20	nation
13	18	hope
14	18	president
15	18	rosa
16	18	voting
17	17	civil
18	17	today
19	17	years
20	16	place
21	16	voters
22	15	day
23	15	vote
24	14	united
25	14	war
26	13	americans
27	13	children
28	13	election
29	13	history
30	13	kerry

2007

1	63	will
2	51	president
3	44	war
4	39	people
5	34	america
6	28	country
7	27	iraq
8	25	american
9	24	work
10	22	system
11	22	time
12	20	change
13	20	today
14	20	year
15	20	years
16	16	future
17	15	working
18	14	immigrants
19	14	race
20	13	americans
21	13	care
22	13	nation
23	12	face
24	12	family
25	11	decision
26	11	point
27	11	politics
28	10	best
29	10	bill
30	10	day

2008

1	164	will
2	107	people
3	93	time
4	90	america
5	78	country
6	73	american
7	69	change
8	59	hope
9	55	americans
10	52	work
11	51	children
12	44	moment
13	44	nation
14	44	years
15	42	economic
16	41	black
17	41	campaign
18	40	white
19	39	jobs
20	38	care
21	38	president
22	38	united
23	37	economy
24	36	better
25	36	great
26	35	tonight
27	34	health
28	34	help
29	33	stand
30	32	future

2009

1	1057	will
2	525	people
3	382	america
4	291	time
5	257	health
6	241	american
7	237	care
8	213	work
9	204	security
10	204	united
11	201	years
12	198	today
13	194	future
14	190	nation
15	189	president
16	187	country
17	182	war
18	170	going
19	167	americans
20	164	nations
21	154	system
22	151	economy
23	151	government
24	137	help
25	136	day
26	135	children
27	130	jobs
28	130	peace
29	130	year
30	127	good

2010

1	739	will
2	615	people
3	384	going
4	302	time
5	297	america
6	287	health
7	247	country
8	245	president
9	244	care
10	230	insurance
11	223	american
12	216	work
13	211	year
14	206	today
15	196	government
16	189	years
17	185	well
18	179	nation
19	178	americans
20	167	system
21	167	united
22	160	security
23	158	reform
24	157	families
25	146	economy
26	145	day
27	144	help
28	139	energy
29	139	good
30	136	jobs

2011

1	776	will
2	570	people
3	359	america
4	271	united
5	255	country
6	226	work
7	220	american
8	212	today
9	208	time
10	205	going
11	205	years
12	187	nation
13	175	security
14	173	americans
15	170	jobs
16	159	economy
17	158	government
18	154	future
19	142	nations
20	133	president
21	131	tax
22	128	war
23	124	help
24	120	day
25	118	change
26	116	rights
27	111	energy
28	110	well
29	104	middle
30	104	region

2012

1	531	president
2	519	will
3	508	people
4	448	governor
5	418	going
6	354	romney
7	322	america
8	248	years
9	225	time
10	221	jobs
11	215	country
12	202	work
13	194	obama
14	186	united
15	185	american
16	179	well
17	155	tax
18	150	government
19	150	today
20	144	nuclear
21	140	nation
22	138	year
23	136	future
24	133	israel
25	131	help
26	129	economy
27	127	energy
28	123	americans
29	119	care
30	119	military

2013

1	646	people
2	631	will
3	298	america
4	279	going
5	257	time
6	251	work
7	247	today
8	230	years
9	217	american
10	217	health
11	216	country
12	204	care
13	202	united
14	192	americans
15	192	president
16	186	good
17	180	insurance
18	170	congress
19	170	families
20	165	well
21	154	security
22	147	war
23	142	law
24	138	young
25	130	peace
26	128	day
27	126	help
28	122	government
29	122	nation
30	118	jobs

2014

1	494	people
2	461	will
3	252	america
4	224	today
5	218	going
6	212	united
7	207	president
8	184	american
9	175	country
10	173	work
11	162	time
12	151	government
13	144	years
14	142	young
15	135	nations
16	134	americans
17	129	help
18	128	security
19	122	countries
20	102	well
21	101	question
22	97	iraq
23	97	support
24	95	leaders
25	95	war
26	94	good
27	93	military
28	90	working
29	90	year
30	88	russia

2015

1	854	people
2	719	will
3	488	going
4	426	america
5	417	work
6	381	president
7	346	today
8	339	united
9	337	country
10	307	american
11	303	years
12	302	time
13	295	good
14	275	iran
15	261	young
16	248	government
17	239	americans
18	235	countries
19	229	well
20	226	help
21	203	deal
22	202	working
23	199	security
24	197	families
25	178	better
26	169	women
27	167	change
28	163	isil
29	162	children
30	161	great

2016

1	1170	people
2	595	will
3	574	president
4	524	going
5	398	work
6	388	time
7	379	america
8	376	united
9	374	years
10	351	country
11	333	good
12	305	young
13	302	today
14	283	americans
15	280	question
16	280	well
17	279	american
18	274	obama
19	258	lot
20	246	things
21	243	better
22	235	government
23	219	great
24	214	change
25	203	countries
26	194	help
27	173	vietnam
28	170	gun
29	166	democracy
30	164	law

9.4.2. Trump

2015

1	161	trump
2	124	people
3	101	going
4	79	bartiromo
5	78	great
6	54	will
7	52	well
8	50	country
9	48	good
10	47	china
11	44	money
12	37	lot
13	33	donald
14	32	big
15	29	things
16	28	jobs
17	26	nice
18	24	audience
19	24	charlie
20	24	love
21	22	deal
22	22	time
23	21	bring
24	21	company
25	21	gasparino
26	21	member
27	21	thing
28	20	trade
29	19	bad
30	19	billion

2016

1	2517	will
2	1976	going
3	1400	people
4	1172	country
5	1097	clinton
6	1076	hillary
7	962	american
8	799	jobs
9	710	america
10	647	great
11	417	president
12	411	time
13	404	trump
14	356	trade
15	354	americans
16	338	united
17	333	government
18	332	years
19	288	percent
20	287	win
21	282	obama
22	276	money
23	272	plan
24	261	nation
25	256	vote
26	252	change
27	240	year
28	238	good
29	237	work
30	235	deal

2017

1	7515	president
2	4834	going
3	4655	will
4	4329	people
5	3619	great
6	2288	country
7	1950	american
8	1833	good
9	1662	united
10	1638	well
11	1624	time
12	1558	america
13	1521	lot
14	1376	today
15	1289	trump
16	1210	tax
17	1203	years
18	1179	job
19	1062	jobs
20	973	work
21	965	big
22	915	things
23	905	long
24	895	care
25	837	love
26	810	working
27	774	day
28	767	nation
29	715	better
30	701	countries

2018

1	13264	president
2	8607	going
3	7545	people
4	6890	great
5	5484	will
6	3871	good
7	3823	country
8	3741	lot
9	3407	well
10	2931	years
11	2727	time
12	2222	job
13	2071	american
14	2003	trump
15	1988	united
16	1856	things
17	1842	big
18	1783	america
19	1705	today
20	1641	thing
21	1596	incredible
22	1510	deal
23	1461	work
24	1453	year
25	1415	trade
26	1376	jobs
27	1372	love
28	1345	coming
29	1331	billion
30	1329	long

2019

1	15674	president
2	6656	people
3	6566	going
4	5615	great
5	4465	will
6	3748	lot
7	3514	well
8	3489	trump
9	3436	good
10	3257	country
11	2892	years
12	2594	time
13	2144	china
14	2085	job
15	2024	american
16	1969	things
17	1961	deal
18	1896	united
19	1806	today
20	1656	big
21	1508	trade
22	1492	border
23	1485	thing
24	1483	incredible
25	1357	year
26	1345	america
27	1321	work
28	1295	sir
29	1283	tremendous
30	1179	long

2020

1	21251	president
2	10031	going
3	9870	people
4	6251	great
5	6067	will
6	4958	well
7	4452	lot
8	4252	good
9	3898	country
10	3242	time
11	3134	job
12	2633	things
13	2570	years
14	2474	american
15	2294	china
16	2279	big
17	2264	thing
18	2205	today
19	1841	trump
20	1826	work
21	1822	working
22	1773	incredible
23	1761	coronavirus
24	1614	day
25	1555	testing
26	1538	year
27	1510	sir
28	1483	united
29	1478	countries
30	1455	america

2021

1	193	people
2	137	president
3	134	going
4	99	country
5	97	ballots
6	93	will
7	88	votes
8	74	election
9	74	great
10	64	numbers
11	62	years
12	53	lot
13	51	number
14	50	georgia
15	49	vote
16	47	love
17	46	thousands
18	46	well
19	42	america
20	39	time
21	39	trump
22	38	county
23	37	law
24	37	things
25	36	good
26	35	big
27	34	brad
28	32	american
29	32	find
30	32	thing

10. File-Attachments

	Filename	Size	Description
📁	**Code**		
>	concat_files.ps1	538 B	Code from chapter 4.2.
VBA	excel_deleterows.bas	1,14 KB	Code from chapter 4.2.
R	freq-comparison.r	1,38 KB	Code from chapter 9.2.
	readability.py	2,45 KB	Code from chapter 4.3.
R	sentiment-graphs.r	2,99 KB	Code from chapter 4.5.4.
R	sentiment-proximity.r	4,94 KB	Code from chapter 4.5.5.
R	textlength-graphs.r	1,92 KB	Code from chapter 4.1.6.
	tfidf-tsne.py	897 B	Code from chapter 4.6.3.
R	wordcloud.r	1,46 KB	Code from chapter 4.2.
R	wordcloud-v2.r	721 B	Code from chapter 4.2.
R	wordfreq-to-text.r	212 B	Code from chapter 4.2.
📁	**Corpora**		
📁	obama_corpus	6,7 MB	Corpus from Lemmerich 2020
📄	obama_corpus.txt	6,7 MB	Obama corpus combined into one file
📄	obama-tagged.txt	21,8 MB	Obama corpus, POS tagged
📁	trump_corpus	30,4 MB	Filtered corpus from Lemmerich 2021a
📄	trump_corpus.txt	30,4 MB	Trump corpus combined into one file
📄	trump-tagged.txt	106,0 MB	Trump corpus, POS tagged
📁	**Data**		
📄	Stopwordlist_en.txt	4,10 KB	Original stopwordlist (see chapter 9.3.1.)
📄	Stopwordlist_en_v2.txt	4,11 KB	Extended stopwordlist
📄	Stopwordlist_en_v3.txt	4,16 KB	Stopwordlist version 3
📁	mfw_yearly	38 Files	Full tables from chapter 4.2.2.
X	MFW_tables.xlsx	4,17 MB	Full tables from chapter 4.2.1.
📄	readability_kincaid_obama.csv	24,7 KB	Full Table of Obama from chapter 4.3.2.
📄	readability_kincaid_trump.csv	177 KB	Full Table of Trump from chapter 4.3.2.
SVG	Classicdelta_tsne.svg	1,41 MB	Full resolution 4.6.2.