* [Google n-gram data](https://jtimm.net/2019/04/14/lexical-change-procrustes/" \l "google-n-gram-data" \t "_blank)
* [Nearest-neighbors](https://jtimm.net/2019/04/14/lexical-change-procrustes/#nearest-neighbors)
* [Procrustes, PCA & visualizing semantic change](https://jtimm.net/2019/04/14/lexical-change-procrustes/#procrustes-pca-visualizing-semantic-change)
* [Detecting semantic change](https://jtimm.net/2019/04/14/lexical-change-procrustes/#detecting-semantic-change)
* [Summary](https://jtimm.net/2019/04/14/lexical-change-procrustes/#summary)
* [References](https://jtimm.net/2019/04/14/lexical-change-procrustes/#references)

I have developed a [Git Hub guide](https://github.com/jaytimm/google_ngrams_and_R) that demonstrates a simple workflow for sampling Google n-gram data and building historical word embeddings with the aim of investigating lexical semantic change. Here, we build on this workflow, and unpack some methods presented in Hamilton, Leskovec, and Jurafsky ([2016](https://jtimm.net/2019/04/14/lexical-change-procrustes/#ref-hamilton2016diachronic)) & Li et al. ([2019](https://jtimm.net/2019/04/14/lexical-change-procrustes/#ref-li2019macroscope)) for aligning historical matrices/embeddings and visualizing semantic change.

library(tidyverse)

**Google n-gram data**

As detailed in the Git Hub post, the historical embeddings utilized here span eight quarter-centuries from 1808 to 2008, and are derived from a ~1% sample of Google’s **English One Million** 5-gram corpus. Long story short of workflow:

* Build term-feature matrices based on simple co-occurrence;
* Weight/transform matrices via positive-pointwise mutual information;
* Homogenize feature set across all time periods;
* Compress feature set to 250 latent dimensions via SVD.

The count of terms included in each quarter-century varies from ~16k to ~18k. Matrices are available on Git Hub [here](https://github.com/jaytimm/google_ngrams_and_R/tree/master/google_one_percent_embeddings).

setwd(local\_dir)

tfms\_mats <- readRDS('tfms\_mats.rds')

For subsequent analyses, we reduce matrices to only terms which occur in every quarter-century, which amounts to roughly 11k forms.

all\_terms <-Reduce(intersect,

lapply(tfms\_mats, rownames))

tfms\_trimmed <- lapply(1:8, function (x)

tfms\_mats[[x]][rownames(tfms\_mats[[x]]) %in% all\_terms,])

names(tfms\_trimmed) <- names(tfms\_mats)

**Frequency data** for all terms included in the historical matrices are available [here](https://github.com/jaytimm/google_ngrams_and_R/blob/master/freqs_by_gen.rds). These frequencies are based on the sampled 5-gram corpus, and will obviously differ some from values obtained from the full corpus.

setwd(local\_freq)

freqs\_by\_gen <- readRDS('freqs\_by\_gen.rds')

While we are here, we examine historical frequencies for a small set of forms.

sample <- c('GRASP', 'SIGNIFICANT', 'COMMITMENT', 'HOPE')

freqs\_by\_gen %>%

filter(form %in% sample) %>%

ggplot(aes(x = quarter, y = ppm,

color = form, group = form)) +

geom\_line(size = 1.5) +

ggthemes::scale\_color\_stata()+

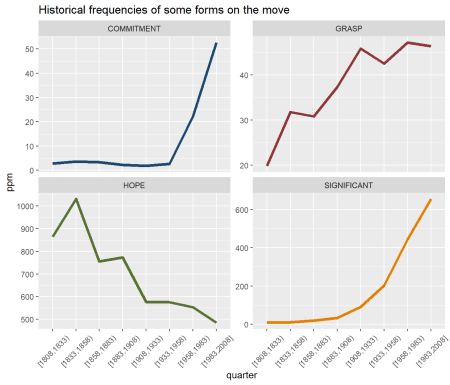
facet\_wrap(~form, scales = 'free\_y', ncol=2) +

labs(title="Historical frequencies of some forms on the move") +

theme(legend.position="none",

axis.text.x=element\_text(angle=45,

hjust=0.5, vjust=0.5))



**Nearest-neighbors**

Based on these historical matrices, we first investigate historical nearest-neighbors/synonyms for a given term. Using the neighbors function from the LSAfun package, we extract the top 10 nearest-neighbors for each quarter-century for the form *SIGNIFICANT*.

search <- toupper('significant')

nns <- lapply(tfms\_trimmed, LSAfun::neighbors, x = search , n = 10)

#disability, disorder, CELL, quit, significant,

List output from above is a bit messy; the function below cleans this output some for subsequent visualization/analyses.

strip\_syns <- function (x) {

lapply(1:length(x), function(y)

cbind(form = as.character(names(x[[y]])),

quarter = names(x[y]),

data.frame(sim = x[[y]], row.names=NULL)) ) %>%

bind\_rows()}

We apply function, and add historical frequency data to get a better sense of how changes in nearest-neighbors align (or do not align) with changes in term frequency historically.

syns <- nns %>%

strip\_syns() %>%

inner\_join(freqs\_by\_gen) %>%

mutate(ppm = round(ppm, 1), sim = round(sim,3)) %>%

select(-freq) %>%

group\_by(quarter) %>%

arrange(desc(sim))%>%

ungroup()

Via gridExtra, we plot nearest-neighbors of *SIGNIFICANT* by quarter-century.

g <- list(length(tfms\_mats))

tt <- gridExtra::ttheme\_default(base\_size = 6.75)

for (i in 1:length(tfms\_mats)) {

g[[i]] <- syns %>%

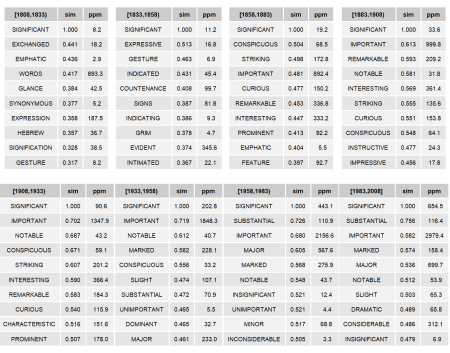
filter (quarter == names(tfms\_mats[i])) %>%

rename(!!names(tfms\_mats[i]) := form) %>%

select(-quarter)%>%

gridExtra::tableGrob(rows=NULL, theme = tt) }

gridExtra::grid.arrange(grobs = g, nrow = 2)



So, a form **on the semantic move** historically:

* Roughly: *CONSPICUOUS/ REMARKABLE* → *SUBSTANTIAL/ MARKED*;
* Also a form that has become decidedly more frequent;
* Collectively, changes (very roughly) reflect (1) the (novel) use of *SIGNIFICANT* in the (burgeoning, ca. late 19th century) field of Statistics, and (2) the continued development of the field of Statistics over the past century or so.

**Procrustes, PCA & visualizing semantic change**

While each of our historical term-feature matrices is comprised of 250 latent dimensions, per the nature of SVD this space is defined differently for each matrix. And while this is non-problematic when computing pair-wise cosine similarities within a given time period (as per above), it is problematic if the goal is to compare word embeddings for a single form across multiple time periods. Which is what we want to do now.

In order to perform this type of comparison, we need to map each historical matrix to a singularly defined coordinate space. Here, this means aligning all non-modern matrices to the modern matrix. We do so using **orthogonal Procrustes**, and implement the algorithm using the procrustes function from the vegan package.

procrustes\_clean <- lapply(1:7, function (x) {

pc <- vegan::procrustes (X = tfms\_trimmed[[8]],

Y = tfms\_trimmed[[x]])

pc <- pc$Yrot

dimnames(pc) <-list(rownames(tfms\_trimmed[[8]]),

c(1:250))

pc } )

procrustes\_clean[[8]] <- tfms\_trimmed[[8]]

names(procrustes\_clean) <- names(tfms\_mats)

With compare-able matrices in tow, our first goal is to **visualize semantic change** for a given term in two-dimensional space. General workflow:

1. Find (some subset of) historical/modern nearest-neighbors for a given term;
2. Extract embeddings for **(1)** from modern matrix;
3. Extract embeddings for given term from all historical matrices;
4. Append **(2)** & **(3)** as single matrix;
5. Perform Principal Component Analysis (PCA) on **(4)**;
6. Plot **(5)** in 2D space

This workflow (roughly) mirrors methods presented in Li et al. ([2019](https://jtimm.net/2019/04/14/lexical-change-procrustes/#ref-li2019macroscope)), which is a slightly simplified/modified version of methods presented in Hamilton, Leskovec, and Jurafsky ([2016](https://jtimm.net/2019/04/14/lexical-change-procrustes/#ref-hamilton2016diachronic)).

**What are we actually doing?** Well, we are basically defining a 2d space via modern word embeddings of nearest-neighbors, and allowing a given term (via historical word embeddings) to move within this (defined) space historically. Procrustes-enabled: all forms in all matrices are situated in the same coordinate space.

Obviously this is a bit weird, as nearest-neighbors are in motion within this space historically as well. However, we have to hold something constant to visualize change – modern embeddings for nearest-neighbors would seem most intuitive/insightful.

To de-clutter our (forthcoming) visualizations some, we limit our analysis to even-numbered quarter-centuries.

gens <- c("[1833,1858)", "[1883,1908)",

"[1933,1958)", "[1983,2008]")

**1. Find (some subset of) historical/modern nearest-neighbors for a given term;**

plot\_syns <- syns %>%

filter(!form %in% search & quarter %in% gens) %>%

group\_by(quarter) %>%

slice(1:15) %>%

ungroup() %>%

select(form) %>%

distinct()

**2. Extract embeddings for (1) from modern matrix;**

syn\_modern <- tfms\_trimmed[[8]][rownames(tfms\_trimmed[[8]]) %in%

plot\_syns$form,]

**3. Extract embeddings for given term from all historical matrices;**

term\_proc <- lapply(procrustes\_clean[c(2,4,6,8)],

function (x) x[search,]) %>%

bind\_rows() %>% t()

**4. Append (2) & (3) as single matrix;**

full <- rbind(term\_proc, syn\_modern)

**5. Perform Principal Component Analysis (PCA) on (4);**

pca <- prcomp(full)$x[,1:2] %>%

data.frame() %>%

rownames\_to\_column() %>%

mutate(cs = ifelse(grepl('\\[', rowname), 'steelblue', 'black')) %>%

mutate(sz = ifelse(cs == 'steelblue', 3.5, 2.5))

**6. Plot (5) in 2D space.**

set.seed(99)

#Hack to build arrows in viz.

to\_end <- pca %>% slice(1:4)

for (i in 1:(nrow(to\_end)-1)) {

to\_end$PC3[i] <- to\_end$PC1[i+1]

to\_end$PC4[i] <- to\_end$PC2[i+1] }

ggplot(pca, aes(x=PC1, y=PC2))+

geom\_point(color = 'darkgray', size = 1)+

ggrepel::geom\_text\_repel(

data = pca, label = pca$rowname,

color = pca$cs, segment.size = .25,

direction = "y", hjust = 0, size = pca$sz,

segment.alpha = .25) +

geom\_segment(data = to\_end %>% slice(1:3),

aes(x=PC1, y=PC2, xend =PC3, yend = PC4),

color = 'steelblue',

size = .6, linetype = 2,

arrow = arrow(length=unit(0.30,"cm"),

type = "closed",

angle = 25))+

ggthemes::theme\_fivethirtyeight() +

ggtitle(paste0(search, ' in nearest-neighbor space historically')) +

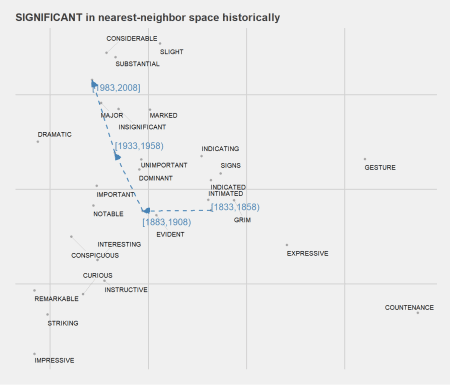
theme(axis.title.x=element\_blank(),

axis.text.x=element\_blank(),

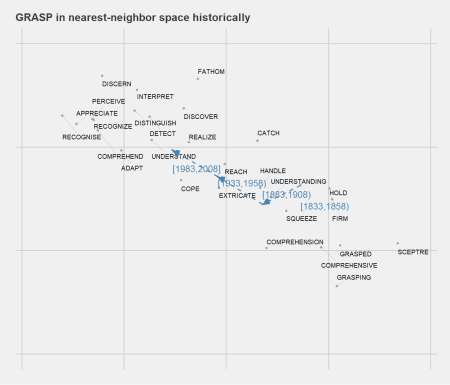
axis.title.y=element\_blank(),

axis.text.y=element\_blank(),

plot.title = element\_text(size=12))



**For good measure**, we consider another example: *GRASP*. Historical nearest-neighbor space is visualized below. Very generally, *physical(ly) GRASP* → *mental(ly) GRASP*. By virtue of having a non-lemmatized cache of word embeddings, lots of fun details.



**Detecting semantic change**

Per Hamilton, Leskovec, and Jurafsky ([2016](https://jtimm.net/2019/04/14/lexical-change-procrustes/#ref-hamilton2016diachronic)), we can also compare a single form’s embeddings at two (or more) points in time to detect changes in distribution and, in theory, lexical semantics. Hamilton, Leskovec, and Jurafsky ([2016](https://jtimm.net/2019/04/14/lexical-change-procrustes/#ref-hamilton2016diachronic)) dub such changes **semantic displacement** – below we compute the degree of semantic displacement (as cosine distances) for all forms in our historical lexicon between quarter-centuries [1883,1908) and [1983,2008].

sim\_deltas <- sapply(rownames(procrustes\_clean[[8]]), function (x)

lsa::cosine(procrustes\_clean[["[1883,1908)"]][x,],

procrustes\_clean[["[1983,2008]"]][x,]) ) %>%

#Clean --

as.tibble() %>%

rownames\_to\_column(var = 'form') %>%

mutate(value = round(value, 3)) %>%

arrange(desc(value)) %>%

rename(sim\_t1\_t2 = value)

The table below highlights forms with the highest degree of semantic displacement from [1883,1908) to [1983,2008]. Again, we add some frequency data to fill out the story some.

freq\_deltas <- freqs\_by\_gen %>%

filter(quarter %in% c("[1883,1908)", "[1983,2008]") &

form %in% rownames(procrustes\_clean[[8]])) %>%

select(-freq) %>%

spread(quarter, ppm) %>%

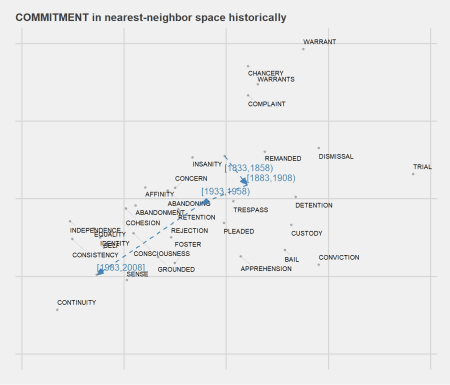
mutate(log\_freq\_t1\_t2 = round(log(`[1983,2008]`/`[1883,1908)`),3)) %>%

inner\_join(sim\_deltas)

freq\_deltas %>% arrange(sim\_t1\_t2) %>% slice(1:6) %>% knitr::kable()

| **form** | **[1883,1908)** | **[1983,2008]** | **log\_freq\_t1\_t2** | **sim\_t1\_t2** |
| --- | --- | --- | --- | --- |
| MID | 2.50 | 25.54 | 2.324 | 0.081 |
| EXPLOIT | 3.20 | 12.13 | 1.333 | 0.086 |
| PITT | 7.61 | 5.09 | -0.402 | 0.113 |
| LURE | 2.00 | 4.36 | 0.779 | 0.114 |
| SICKLE | 1.98 | 4.91 | 0.908 | 0.120 |
| COMMITMENT | 2.32 | 52.74 | 3.124 | 0.123 |

**Based on these findings**, we take a quick look at *COMMITMENT* in nearest-neighbor space historically. So, curious.



The plot below illustrates the relationship between **delta frequency and semantic displacement** from t1 to t2 for all forms included in our lexicon. So, increases in frequency from t1 to t2 are generally accompanied by higher degrees of semantic displacement (ie, lower cosine similarity). Does novel use cause increases in frequency, or vice versa?

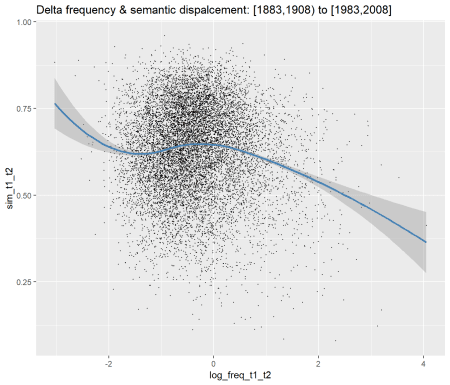
freq\_deltas %>%

ggplot(aes(x = log\_freq\_t1\_t2, y =sim\_t1\_t2)) +

geom\_point(size = .25)+

geom\_smooth(method="loess", se=T, color = 'steelblue')+

ggtitle('Delta frequency & semantic dispalcement: [1883,1908) to [1983,2008]')



**Summary**

Grains of salt abound. These methods are imperfect, and I think best suited for exploration. Examples of actual usage always tell a more complete story when investigating lexical change. Also, the word embeddings used here could be based on a larger corpus. That said, lots of fun to be had with the 1% embeddings, as well as some exploratory utility. Available [here](https://github.com/jaytimm/google_ngrams_and_R/tree/master/google_one_percent_embeddings).

**References**

Hamilton, William L, Jure Leskovec, and Dan Jurafsky. 2016. “Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change.” *arXiv Preprint arXiv:1605.09096*.

Li, Ying, Tomas Engelthaler, Cynthia SQ Siew, and Thomas T Hills. 2019. “The Macroscope: A Tool for Examining the Historical Structure of Language.” *Behavior Research Methods*. Springer, 1–14.