CogSci Word Counts

Lauren Oey

11/24/2018

Cleaning up Data Frame

Adds vector of word counts per abstract. Glimpse at data frame “df” containing a row for each abstract.

```
df$length <- str_count(df$abstract, pattern=" ")+1
glimpse(df)
```

```
## Observations: 7,871
## Variables: 7
## $ year      <int> 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015, 2015...
## $ authors   <chr> "Anne Cleary, Hector Avila-Munoz, Evan Heit, Chris H...
## $ title     <chr> "Applying for National Science Foundation Funding in...
## $ abstract  <chr> "This half-day workshop will provide information and...
## $ html_link <chr> "https://mindmodeling.org/cogsci2015/papers/0001/ind...
## $ pdf_link  <chr> "https://mindmodeling.org/cogsci2015/papers/0001/pap...
## $ length    <dbl> 102, 208, 115, 178, 199, 160, 63, 209, 214, 92, 147,...
```

New Data Frame with a Word for Each Row

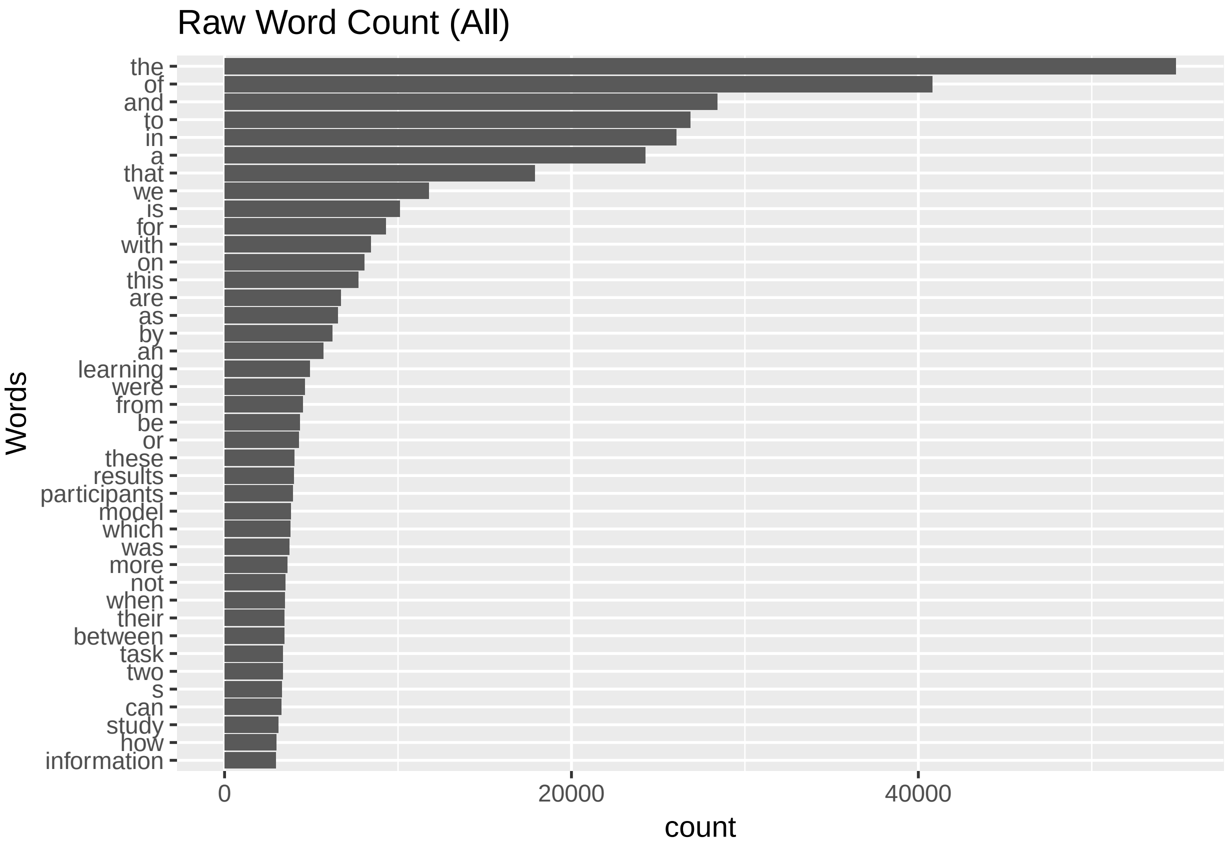
Removes punctuation and escape characters, “\n”, “\t”, “\f”. Creates exception for words containing punctuation, “e.g.” & “i.e.” Creates unique row for each word in abstract. Removes “words” that consist of an empty string. Creates vector of lower-case version of word. Writes to new data frame “words”.

```
words <- df %>%
  mutate(abstract_cleaned = str_replace_all(abstract, c("e\\.g\\."="e1g1", "i\\.e\\."="i1e1")),
         abstract_cleaned = str_replace_all(abstract_cleaned, c("[^a-zA-Z0-9\\&\\s]"=" ", "[\\n\\t\\f]"=
         abstract_cleaned = str_replace_all(abstract_cleaned, c("e1g1"="e.g.", "i1e1"="i.e.")),
         word = strsplit(abstract_cleaned, " ")) %>%
  unnest(word) %>%
  filter(word != "") %>%
  mutate(lowerword = tolower(word))
```

Unfiltered Word Frequency Across All Abstracts

40 most frequent words in all abstracts. As predicted, has a Zipfian distribution.

```
words %>%
  group_by(lowerword) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  head(40) %>%
  ggplot(aes(x=reorder(lowerword,count), y=count)) +
  geom_bar(stat="identity") +
  ggtitle("Raw Word Count (All)") +
  scale_x_discrete("Words") +
  coord_flip()
```



```
ggsave("graphs/rawWordCount.png")
```

```
## Saving 6.5 x 4.5 in image
```

Including Lemma and Filter by Part-of-Speech (POS) Tagging

Gets lemma & POS info using CleanNLP. Writes to new data frame "obj_token". Filters function words (e.g. determiners, prepositions). Writes to new filtered data frame "obj_token_cleaned". Joins data about lemma ("obj_token_cleaned") to original words ("words") dataframe. New data frame "words_full" which also has the function words filtered out. New vector with lower-cased lemma.

```
## First time running, you need to run this to extract the lemma/POS info from CleanNLP
## This takes awhile, so it's better to save the CSV file and read in the file for future use

#cnlp_init_udpipe()
#obj <- cnlp_annotate(df$abstract, as_strings = TRUE, doc_ids=df$title)
#obj_token <- cnlp_get_token(obj)
#write.csv(obj_token, "token_info.csv")
obj_token <- read.csv("token_info.csv")

## Filters out most POS, keeping nouns, verbs, adjectives, proper nouns, adverbs, numbers,
## and INTJ, which is kind of a mix of multiple things
```

```
obj_token_cleaned <- obj_token %>%
  filter(upos %in% c("NOUN", "VERB", "ADJ", "PROPN", "ADV", "NUM", "INTJ")) %>%
  mutate(title = id) %>%
  select("title","word","lemma")

write.csv(obj_token_cleaned, "token_info_cleaned.csv")

words_full <- inner_join(words, unique(obj_token_cleaned), by=c("title","word")) %>%
  mutate(lowerlemma = tolower(lemma))
```

```
## Warning: Column `title` joining character vector and factor, coercing into
## character vector
```

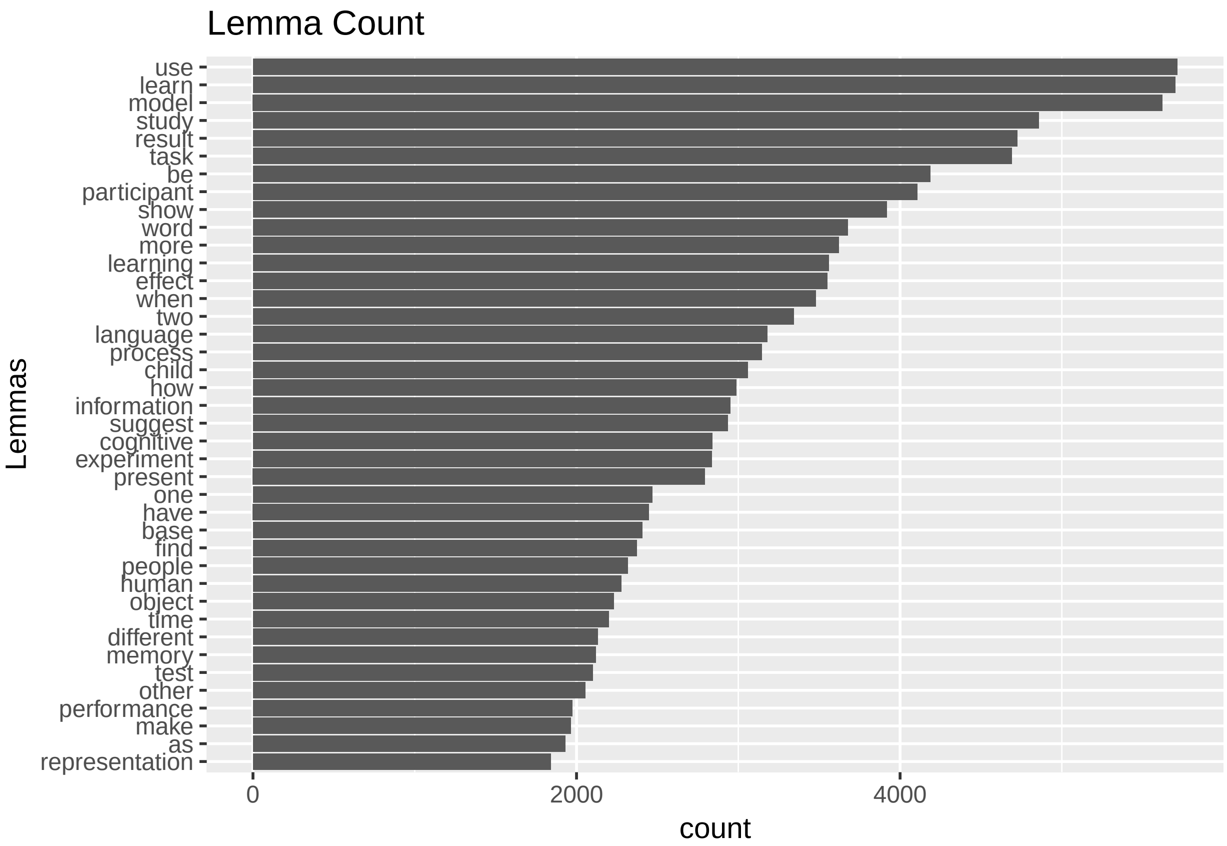
```
## Warning: Column `word` joining character vector and factor, coercing into
## character vector
```

Content Lemma Word Count

Count of content word lemmas. Visualization of 40 most frequent lemmas across all abstracts.

```
wc_overall <- words_full %>%
  group_by(lowerlemma) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
write.csv(wc_overall, "wc_overall.csv")

ggplot(head(wc_overall, 40), aes(x=reorder(lowerlemma,count), y=count)) +
  geom_bar(stat="identity") +
  ggtitle("Lemma Count") +
  scale_x_discrete("Lemmas") +
  coord_flip()
```



```
ggsave("graphs/overallPopWords.png")
```

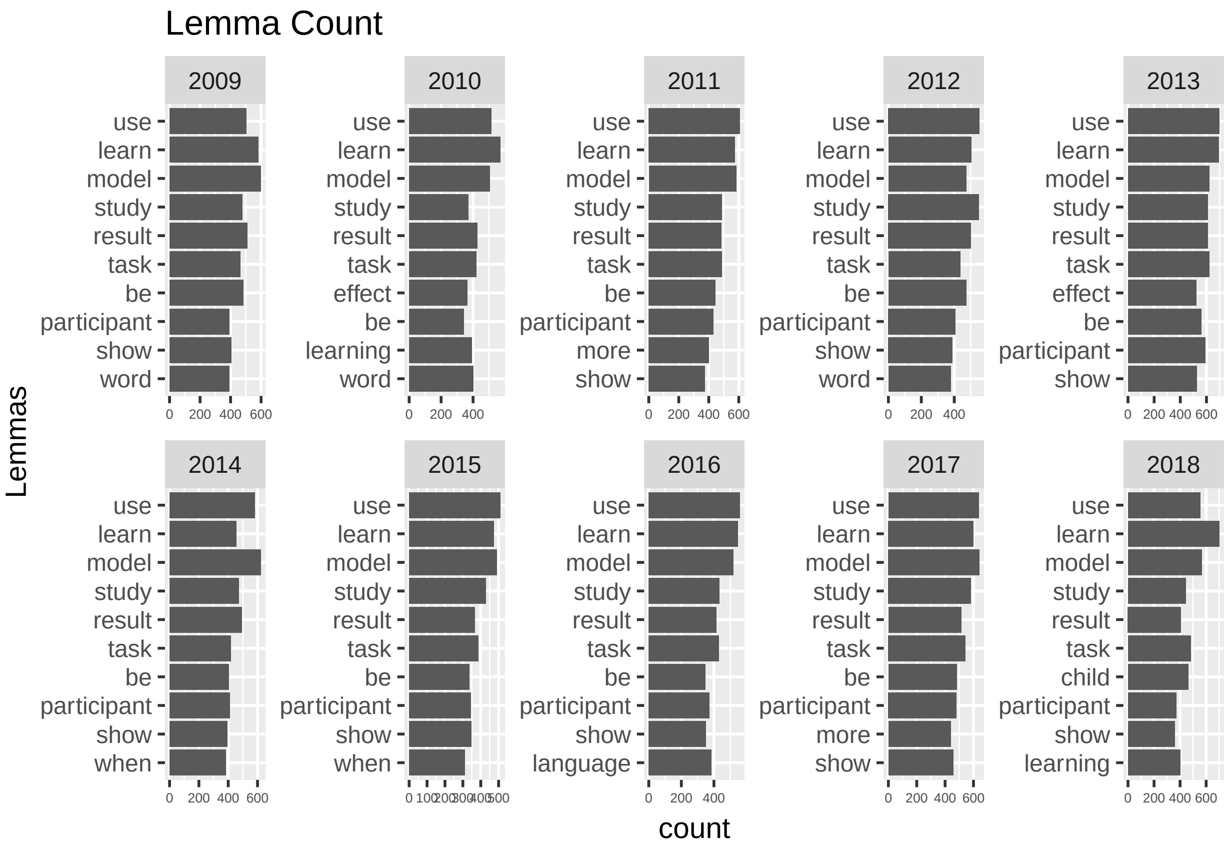
```
## Saving 6.5 x 4.5 in image
```

Content Lemma Word Count by Year

Count of content word lemmas by publication year. Visualization of 10 most frequent lemmas by year. Fairly consistent across years, mostly contains commonly used verbs (e.g. use, be, show), and words pertaining to the scientific procedure (e.g. participant, study, result, task, model).

```
wc_byYear <- words_full %>%
  group_by(year,lowerlemma) %>%
  summarise(count = n()) %>%
  arrange(year,desc(count))
write.csv(wc_byYear, "wc_byYear.csv")

wc_byYear %>%
  group_by(year) %>%
  top_n(10,count) %>%
  ggplot(aes(x=reorder(lowerlemma,count), y=count)) +
  geom_bar(stat="identity") +
  ggtitle("Lemma Count") +
  scale_x_discrete("Lemmas") +
  coord_flip() +
  facet_wrap(~year, nrow=2, scales="free") +
  theme(axis.text.x=element_text(size=5))
```



```
ggsave("graphs/wordsByYear.png")
```

```
## Saving 6.5 x 4.5 in image
```

Content Lemma Word Count by Paper

Count of content lemmas by paper.

```
wc_byTitle <- words_full %>%
  group_by(year,title,lowerlemma) %>%
  summarise(count = n()) %>%
  arrange(year,title,desc(count))
kable(head(wc_byTitle, 20))
```

year	title	lowerlemma	count
2009	'If only' counterfactuals and the exceptionality effect	exceptional	6
2009	'If only' counterfactuals and the exceptionality effect	outcome	6
2009	'If only' counterfactuals and the exceptionality effect	alternative	5
2009	'If only' counterfactuals and the exceptionality effect	lead	5
2009	'If only' counterfactuals and the exceptionality effect	when	4
2009	'If only' counterfactuals and the exceptionality effect	action	3
2009	'If only' counterfactuals and the exceptionality effect	better	3
2009	'If only' counterfactuals and the exceptionality effect	change	3
2009	'If only' counterfactuals and the exceptionality effect	counterfactual	3
2009	'If only' counterfactuals and the exceptionality effect	effect	3
2009	'If only' counterfactuals and the exceptionality effect	only	3

year	title	lowerlemma	count
2009	‘If only’ counterfactuals and the exceptionality effect	also	2
2009	‘If only’ counterfactuals and the exceptionality effect	experiment	2
2009	‘If only’ counterfactuals and the exceptionality effect	experiments	2
2009	‘If only’ counterfactuals and the exceptionality effect	not	2
2009	‘If only’ counterfactuals and the exceptionality effect	reverse	2
2009	‘If only’ counterfactuals and the exceptionality effect	show	2
2009	‘If only’ counterfactuals and the exceptionality effect	usual	2
2009	‘If only’ counterfactuals and the exceptionality effect	1	1
2009	‘If only’ counterfactuals and the exceptionality effect	2	1

```
write.csv(wc_byTitle, "wc_byTitle.csv")
```

New Data Frame with an Author for Each Row

Creates new row for each author in $authors vector. Replaces non-ASCII characters (e.g ü to u); some authors had duplicate names with or without these characters. Removes final comma in authors list, created during data extraction. Fixes weird bug where some authors had additional institution information, due to weird formatting in HTML from which data was extracted. To do this, it removes any text appearing before “\n” (fix gathered from glancing at the data and seeing this recurring issue). Fixes weird bug where there are some double white spaces. Fixes weird bug where numbers appear next to some names, probably indicative of a sub- or superscript in the print. Replaces “&” with “,”. Splits authors list by “,” into new rows, so a new row appears for each word corresponding to each author. Filters out remaining institutions still appearing among authors by removing authors with numbers in their name (indicative of an address) and using a few recurring key words that are unlikely to also be a persons name (e.g. University). Removes extra white space appearing before and after names. Creates new factor $authorAbbr tackling issue of authors with multiple names (e.g. names with or without middle initials, nicknames) by extracting last name and first character of first name (e.g. E Vul). This allows Ed Vul to publish as Edward Vul, Ed Vul, Eddy Vul, Eduardo Vul, E. Vul, Edward Scissorhands Vul, etc. and it will all be categorized as E Vul. Potential issue 1: Edgar Vul would also be categorized as E Vul, potentially leading to issues if Edward and Edgar Vul are indeed different humans. Potential issue 2: first name nicknames that differ in the first letter from the full first name appear as different humans when in fact they should be the same author, e.g. Elizabeth Bonawitz = E Bonawitz; Liz Bonawitz = L Bonawitz. Writes to new data frame “byAuthor”.

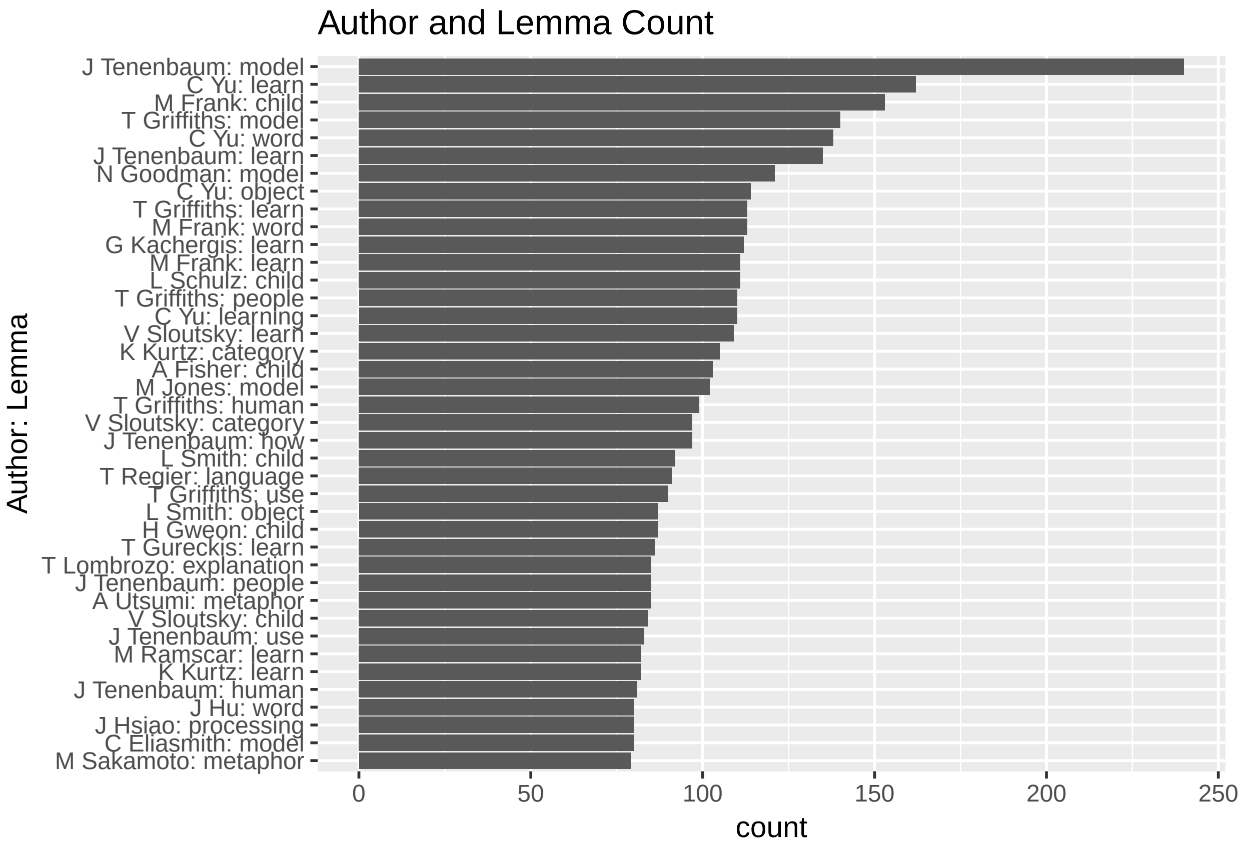
```
byAuthor <- words_full %>%
  mutate(authors=stri_trans_general(authors, "latin-ascii"),
         author=str_replace_all(authors, ",$", ""),
         author=str_replace_all(author, c(".*\n"="", " *\\(.*?\\)"="", "  "=" ")),
         author=str_replace_all(author, " & ", ", "),
         author=strsplit(author, ", ")) %>%
  unnest(author) %>%
  filter(!grepl("University|Institute|Center|Centre|Centro|School|Department|Dept|Unit|Hospital|\\d",
                author)) %>%
  mutate(author=trimws(author),
         authorAbbr = paste(substring(author,1,1), word(author, -1)))
```

Content Lemma Word Count by Author

Count of content lemmas by author. Visualization of 40 most words used by a single author, visualized as “Author: Lemma”.

```
wc_byAuthor <- byAuthor %>%
  group_by(authorAbbr, lowerlemma) %>%
  summarise(count = n()) %>%
  arrange(desc(count),authorAbbr)
write.csv(wc_byAuthor, "wc_byAuthor.csv")

wc_byAuthor %>%
  mutate(authorLemma = paste(authorAbbr, lowerlemma, sep=": ")) %>%
  head(40) %>%
  ggplot(aes(x=reorder(authorLemma,count), y=count)) +
  geom_bar(stat="identity") +
  ggtitle("Author and Lemma Count") +
  scale_x_discrete("Author: Lemma") +
  coord_flip()
```



```
ggsave("graphs/AuthorWords.png")
```

```
## Saving 6.5 x 4.5 in image
```

Content Lemma Word Count by Author and Year

Count of content lemmas by author and year. Allows for looking at trends in author word usage over time.

```
wc_byAuthorYear <- byAuthor %>%
  group_by(year, authorAbbr, lowerlemma) %>%
  summarise(count = n()) %>%
  arrange(year, authorAbbr,desc(count))
write.csv(wc_byAuthorYear, "wc_byAuthorYear.csv")
```

Leaders in the Field by Most Authored Papers by Year

Counts number of papers published by each author each year. Selects the top 3 most publishing author for each year. Creates "leaders" data frame.

```
leaders <- byAuthor %>%
  select("year","authorAbbr","title") %>%
  unique() %>%
  group_by(year, authorAbbr) %>%
  summarise(totalPapers = n()) %>%
  top_n(2, totalPapers)
leaders %>% arrange(year,desc(totalPapers)) %>%
  kable(caption="Top 2 Authors w/ Most Papers by Year")
```

Table 2: Top 2 Authors w/ Most Papers by Year

year	authorAbbr	totalPapers
2009	T Griffiths	11
2009	J Tenenbaum	10
2010	L Smith	10
2010	J Tenenbaum	8
2010	T Griffiths	8
2011	T Griffiths	9
2011	W Fu	8
2012	T Griffiths	12
2012	J Tenenbaum	11
2013	R Dale	12
2013	C Eliasmith	8
2013	M Frank	8
2014	J Hu	11
2014	J Tenenbaum	11
2015	J Tenenbaum	15
2015	N Goodman	11
2016	M Frank	12
2016	J Tenenbaum	10
2016	N Goodman	10
2017	J Tenenbaum	15
2017	T Griffiths	15
2018	J Tenenbaum	18
2018	T Griffiths	11

Authors with Most Words in Abstract by Year

Counts total number of words in abstracts by each author each year. Selects the top 3 most verbose author for each year. Approximately corresponds to top 3 authors by number of papers published (i.e. authors who have more papers correspondingly have more words).

```
byAuthor %>%
  group_by(year, authorAbbr) %>%
  summarise(total = n()) %>%
  top_n(2, total) %>%
  arrange(year, desc(total)) %>%
  kable(caption="Top 3 Authors w/ Most Abstract Words by Year")
```

Table 3: Top 3 Authors w/ Most Abstract Words by Year

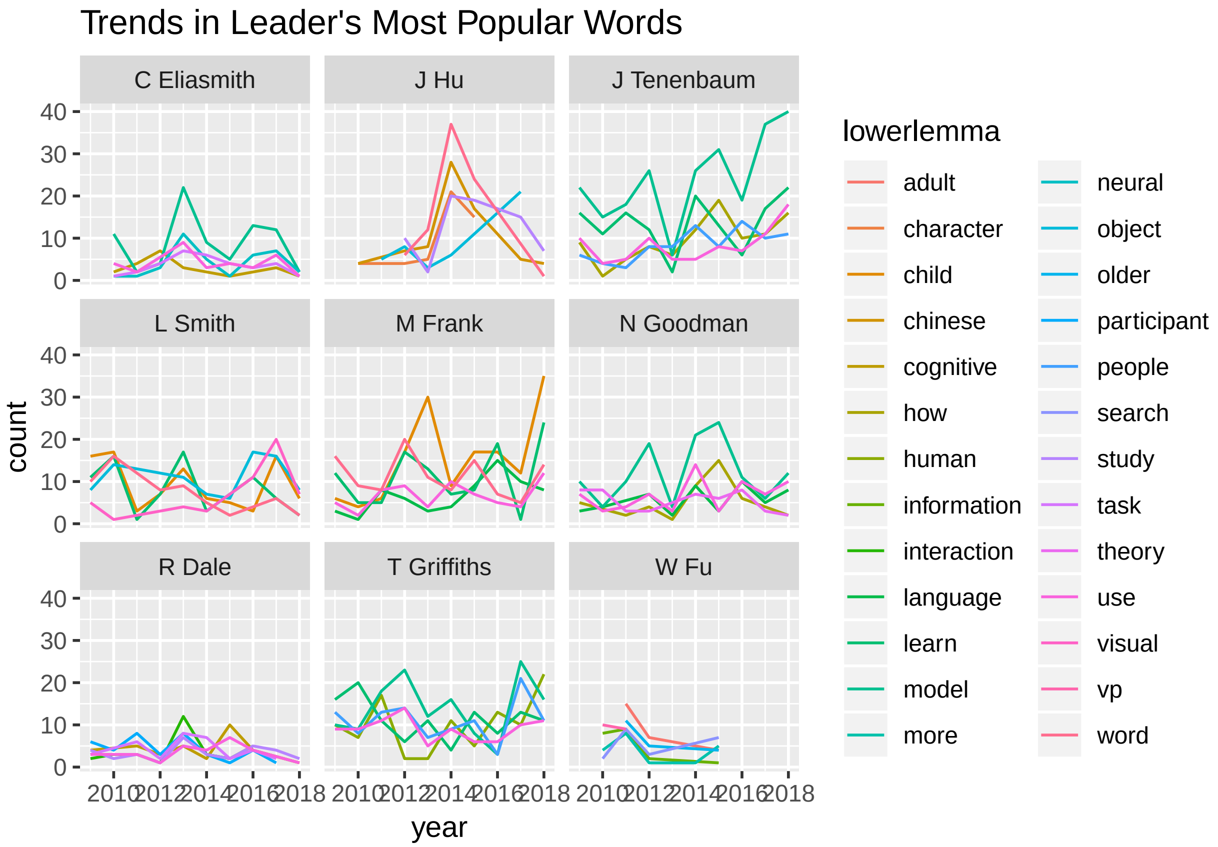
year	authorAbbr	total
2009	T Griffiths	920
2009	J Tenenbaum	849
2010	L Smith	874
2010	M Lee	754
2011	W Fu	736
2011	T Griffiths	730
2012	J Tenenbaum	1082
2012	T Griffiths	965
2013	R Dale	1075
2013	I McLaren	914
2014	J Hu	1256
2014	H Chen	1059
2015	J Tenenbaum	1277
2015	N Goodman	925
2016	M Frank	968
2016	J Tenenbaum	874
2017	T Griffiths	1292
2017	J Tenenbaum	1280
2018	J Tenenbaum	1651
2018	T Griffiths	1067

Favorite Words of Field Leaders and Trends

Extracts the 5 most frequent words used by the leaders determined in the “leaders” data frame. Visualizes the frequency of word usages for each author over time.

```
leaderFaveWords <- byAuthor %>%
  filter(authorAbbr %in% leaders$authorAbbr) %>%
  group_by(authorAbbr, lowerlemma) %>%
  summarise(count = n()) %>%
  top_n(5, count) %>%
  mutate(authorWord = paste(authorAbbr, lowerlemma, sep="_")) %>%
  arrange(authorAbbr, desc(count))

wc_byAuthorYear %>%
  filter(authorAbbr %in% unique(leaderFaveWords$authorAbbr)) %>%
  mutate(authorWord=paste(authorAbbr, lowerlemma, sep="_")) %>%
  filter(authorWord %in% unique(leaderFaveWords$authorWord)) %>%
  ggplot(aes(x=year, y=count, colour=lowerlemma)) +
  geom_line(stat="identity") +
  ggtitle("Trends in Leader's Most Popular Words") +
  facet_wrap(~authorAbbr)
```



```
ggsave("graphs/leaderPopWords.png")
```

```
## Saving 6.5 x 4.5 in image
```