The relation between the Lexicon and Grammar in child language development

lexicon-syntax

Eva Portelance

2019-03-19 10:37:42

Contents

The following study is a replication of Braginsky et al. (2015). The initial study has since been updated to include more languages in its cross-linguistic dataset. These more recent results can be found in chapters §13 and §14 of Frank, Braginsky, Marchman and Yurovsky's upcoming book (Frank et al., n.d.).

1 Introduction

Lexicon and grammar are intimately intertwined concepts. They both pertain to language: one being the collection of units of meaning - words or morphemes - and the other the set of rules which determine how we combine these units into meaningful utterances. Though the dependence between these two parts of language is clear from a theoretical standpoint, their separation historically into two separate conceptual parts in linguistic theory has led to a large body of work in language development which has assumed that word learning and syntactic development (learning to produce coherent sentences) are separate cognitive processes (Baker 2005). In more recent years, with the advent of instruments for the assessment of linguistic development in children and the collection of larger scale cross-linguistic data sets, researchers have been able to show that there is a correlation between the acquisition of words and the production of more complex syntactic structures. Bates et al. (1994) showed that vocabulary size was a much better predictor than age of syntactic development in English speaking children. Braginsky et al. (2015)[1] applied this same analysis to

[1] An updated version of this paper is available in the chapters §13 and §14 of Frank et al. (n.d.).

additional languages. They also measured how much additional variance could, in fact, be explained by age base factors, such as working memory and overall physical development. This strong correlation between lexicon growth and syntactic development lends weight to the hypothesis that the learning processes behind these language developments are related.

1.1 Research questions

This paper will address three research questions, all pertaining to the existence of a relation between word learning and grammar development. First, Do nouns, predicates, and functional words have different learning trajectories in early child language development? Second, is there a strong correlation between lexicon size and complexity score (a metric for syntactic development)? Third, if so, does a model with lexicon size as a predictor explain more of the variance than a model which age as a predictor?

1.2 Hypotheses

I attempt to reproduce the results found by Bates et al. (1994) and Braginsky et al. (2015). First, I expect nouns to be more easily acquired than function words in early language acquisition, but that as vocabulary size grows, the noun advantage disappears. Second, I expect that vocabulary size is a significant predictor of complexity score. Third, I expect a linear model with vocabulary size as a predictor to explain more variance in reported complexity scores than a model with age as a predictor.

1.3 Data

For my analyses, I am using cross-linguistic data available through the Wordbank Project repository (Frank et al. 2016). There is an API available through CRAN to access this repository, wordbankr (https://github.com/langcog/wordbankr).

This is a repository of CDI form administrations. CDI forms are self-assessed reports of a child's language development. They contain a word list section in which parents can report whether or not their child understands, produces, or has not yet acquired a given word. The word lists are language specific and around 300 words long for 'WG' forms and 600 words long for 'WS' forms. To determine the vocabulary size of a given child, we take the number of produced words over the total number of words on the given form. This returns a normalized vocabulary score representing the proportion of words acquired on the CDI form.

These forms can also contain a complexity section. This section is only present on forms administered to children who are 18 months or older ('WS' forms in the data). In this section, parents are asked to choose the form which best describes their child's production between a simple form and complex form of a sentence, eg. 'Sam happy' or 'Sam is happy'. The complexity score of a given form is the number of complex forms produced divided by the total number of complexity items on the form.

For this paper, I used the data available through wordbankr for the following languages from their respective sources.

- **English**: (American) Thal, Marchman, and Tomblin (2013), Fernald, Marchman, and Weisleder (2013), Fenson et al. (2014), Krista Byers-Heinlein (Concordia University), Linda Smith (Indiana University), Michael C. Frank (Stanford University), Virginia Marchman (Stanford University)
- **Danish**: Bleses et al. (2008)
- **French (French)**: Von Holzen, Nishibayashi, and Nazzi (2018), Sophie Kern (Centre national de la recherche scientifique (CNRS)), Christina Bergmann (Max Planck Institute for Psycholinguistics), Anne-Caroline Fievet (Laboratoire de Sciences Cognitives et Psycholinguistique (ENS, EHESS, CNRS)
- **French (Quebecois)**: Boudreault et al. (2007), Trudeau and Sutton (2011)
- **Hebrew**: Hila Gendler Shalev (Tel-Aviv University)
- **Kiswahili**: Alcock et al. (2015)

- **Spanish (Mexican)**: Jackson-Maldonado et al. (2003), Weisleder and Fernald (2013)
- **Slovak**: Svetlana Kapalková (Comenius University)
- **Norwegian**: Simonsen et al. (2014)
- **Kigiriama**: Alcock et al. (2015)

The following code block collects all the instrument data available through wordbankr for a set of selected languages for both 'WG' and 'WS' forms into single data frame where each observation is the answer (value) to an item (item_id) on a given completed instrument (data_id), in a given language (language), in addition to the linguistic information about this item (eg. type, definition, category, etc.). This information is then saved to a CSV for future use.

```
# Get information about forms and there availability by languages
df.administrations = get_administration_data()
df.instruments = get_instruments()

# The set of languages I found to have WG and WS forms with annotated item types
# including "word" and "complexity".
languages  = c("English (American)", "Danish", "French (French)",
               "French (Quebecois)","Hebrew", "Kiswahili", "Spanish (Mexican)",
               "Slovak", "Norwegian", "Kigiriama")

# A helper function to collect instrument data from multiple languages into a
# single data.frame
get_multiling_instrument_data <- function(languages, form){
  df.multiling_instrument_data = data.frame()
  for(lang in languages){
    df.lang_instrument_data = get_instrument_data(language = lang,
                                                  form = form,
                                                  iteminfo = TRUE ) %>%
      mutate(language = lang) %>%
      select(language, everything())
      df.multiling_instrument_data = rbind(df.multiling_instrument_data,
                                           df.lang_instrument_data)
  }
  return(df.multiling_instrument_data)
}

# Collect all instrument data for selected languages for both WG and WS forms
df.WG_multiling_instrument_data = get_multiling_instrument_data(languages, "WG")
df.WS_multiling_instrument_data = get_multiling_instrument_data(languages, "WS")

# I used the following to make sure selected languages where annotated for at least
# "word" and "complexity" in variable type
#df.WS_multiling_instrument_data %>% distinct(language, type)


# Write to csv for future use
write.csv(df.WG_multiling_instrument_data,
          file = "df_WG_multiling_instrument_data.csv",row.names=FALSE)
write.csv(df.WS_multiling_instrument_data,
          file = "df_WS_multiling_instrument_data.csv",row.names=FALSE)
```

```
# Load in data from csv (to avoid downloading from remote all the data)
df.WG_multiling_instrument_data <-
  read.csv(file="df_WG_multiling_instrument_data.csv", header=TRUE, sep=",")
```

```r
df.WS_multiling_instrument_data <-
  read.csv(file="df_WS_multiling_instrument_data.csv", header=TRUE, sep=",")

languages = df.WG_multiling_instrument_data %>% distinct(language) %>% .$language

# View head of data frame to see all variables
print(head(df.WS_multiling_instrument_data))
```

```
            language data_id    value num_item_id item_id definition type
1 English (American)  129242 produces           1  item_1    baa baa word
2 English (American)  129243                    1  item_1    baa baa word
3 English (American)  129244 produces           1  item_1    baa baa word
4 English (American)  129245 produces           1  item_1    baa baa word
5 English (American)  129246 produces           1  item_1    baa baa word
6 English (American)  129247 produces           1  item_1    baa baa word
  category lexical_category lexical_class uni_lemma complexity_category
1   sounds            other         other   baa baa
2   sounds            other         other   baa baa
3   sounds            other         other   baa baa
4   sounds            other         other   baa baa
5   sounds            other         other   baa baa
6   sounds            other         other   baa baa
```

The following block creates hashmaps (data_id, age) from the administration data for the selected languages and forms. these hashmaps can be used via a helper function add_age(hashmap, data.frame) to map the age of the child to a given data_id since age is not included as a variable in the instrument data.

```r
# This is a helper function similar to the one used for collecting instrument
#data which merges the administration data for multiple languages of a given
#form into one central data frame.
get_multiling_administration_data <- function(languages, form){
    df.multiling_administration_data = data.frame()
  for(lang in languages){
    df.lang_administration_data = get_administration_data(language = lang,
                                                          form = form ) %>%
      mutate(language = lang) %>%
      select(language, everything())
      df.multiling_administration_data = rbind(df.multiling_administration_data,
                                               df.lang_administration_data)
  }
  return(df.multiling_administration_data)
}


# Collect the administration information for the chosen languages and forms
df.WG_multiling_administration_data = get_multiling_administration_data(languages, "WG")
df.WS_multiling_administration_data = get_multiling_administration_data(languages, "WS")

write.csv(df.WG_multiling_administration_data,
          file = "df_WG_multiling_administration_data.csv",row.names=FALSE)
write.csv(df.WS_multiling_administration_data,
          file = "df_WS_multiling_administration_data.csv",row.names=FALSE)
```

```r
df.WG_multiling_administration_data <-
  read.csv(file="df_WG_multiling_administration_data.csv", header=TRUE, sep=",")
```

```r
df.WS_multiling_administration_data <-
  read.csv(file="df_WS_multiling_administration_data.csv", header=TRUE, sep=",")

# The number of unique data_ids in the administration data is lower than in the
# instrument data ... this is weird.
#df.WG_multiling_administration_data %>% distinct(data_id) %>% count()
#df.WG_multiling_instrument_data %>% distinct(data_id) %>% count()
#df.WS_multiling_administration_data %>% distinct(data_id) %>% count()
#df.WS_multiling_instrument_data %>% distinct(data_id) %>% count()

# I will attempt to match as many ages as I can and for the missing data_ids i will
# mark the age as NA

# Create hashmap with data_ids as keys and age as values. Given the amount of
# observations, hashmap present a much more efficient alternative for searching for
# a given data_id's age than a data frame.
hm.WG_multiling_age <- hashmap(keys= df.WG_multiling_administration_data$data_id,
                               values =  df.WG_multiling_administration_data$age)

hm.WS_multiling_age <- hashmap(keys= df.WS_multiling_administration_data$data_id,
                               values = df.WS_multiling_administration_data$age)

# A helper function which adds the age information to a data frame given the right
# hashmap and data frame
add_age <- function(hm.age, df.data){
  df.result = df.data %>% mutate(age = hm.age[[data_id]])
  return (df.result)
}
```

In sections §2 and §3, I present two separate analyses of the relationship between word learning and grammar learning using two separate metrics for syntactic development: the acquisition of grammatical categories and syntactic complexity.

2 Analysis 1: grammatical categories

If word learning does not take into account any grammatical information about words, then we expect words from all lexical categories, i.e. nouns, predicates (verbs and adjectives), and functional words (conjunctions, particles, complementizers,...), to be acquired at the same rate. However, if grammatical categories matter, it is possible that instead, we observe that the rate of acquisition for different categories varies over time.

The following analysis will test whether or not the *noun bias* is attested cross-linguistically. The noun bias is a learning bias where children acquire nouns more easily than other categories at first (It is hypothesized that this is because most noun meanings are grounded in concrete referents). This analysis will also determine whether or not functional words are generally acquired later than other categories due to their more complex uses in language.

2.1 Data wrangling

In this analysis, I use data collected on 'WG' CDI forms for children whose ages ranged between 8 and 24 months in all the aforementioned languages.

The following code block calculates the vocabulary size as well as the total proportion of acquired nouns, predicates, and function words for every data_id (child/instrument). It produces two separate data frames based on our definition of "acquired words": the first requiring word production and the second requiring word comprehension (a superset of the first). These data frames are used to plot and model the relationship between vocabulary size and the proportion of different acquired lexical categories.

```
# There are 5 possible values for lexical_category in this data:
# [NA, other, nouns, predicates, function_words]
df.WG_multiling_instrument_data %>% distinct(lexical_category)

# Given that the Bates et al. 1994 study and Braginsky et al. 2015 replication
# only looked at three of these categories, I will only look at data points which
# match lexical_category in [nouns, predicates, function_words]
df.WG_multiling_lexcat_data <- df.WG_multiling_instrument_data %>%
  filter(lexical_category %in% c("nouns","predicates","function_words"))

# There are 10,272 unique data_ids
#df.WG_multiling_lexcat_data %>% distinct(data_id)

# We need to produce a data frame with 10,272 observations with their respective
# vocabulary size, and proportion of nouns, predicates and function_words. I will
# produce two data frames, one where counts are based on word production (value =
# produces) and the other where counts are based on word comprehension (value =
# understands | produces)

# This first data frame counts acquired words as 'produces'
df.WG_multiling_lexcat_produces <- df.WG_multiling_lexcat_data %>%
  group_by(data_id) %>%
# We will use this to normalize the vocab_size
  mutate(vocab_max = n()) %>%
  ungroup() %>%
  group_by(data_id, value) %>%
# These are temporary counts which will make sense once we spread the data
# according to lexical_category and value
  mutate(temp_vocab_size = n()) %>%
  ungroup() %>%
  group_by(data_id,lexical_category,value) %>%
  mutate(temp_score = n()) %>%
  ungroup() %>%
  group_by(data_id,lexical_category) %>%
# We will use this to normalize the the proportional counts for each
# lexical category
  mutate(max_score = n()) %>%
  ungroup() %>%
  select(language, data_id, lexical_category,
         value, vocab_max, temp_vocab_size, temp_score, max_score) %>%
  distinct(data_id, lexical_category, value, .keep_all = TRUE) %>%
# We combine lexical_category and value in order to properly spread the scores
# accross lexical_category and value
  mutate(lexical_category.value = paste(lexical_category, value, sep = ".")) %>%
# normalize scores
  mutate(norm_score = temp_score/max_score) %>%
# we only want to count produced words as part of the vocab, so other value
# types are set to 0 for later sum
  mutate(temp_vocab_size = ifelse(value == "produces", temp_vocab_size, 0)) %>%
```

```r
  select(language,data_id, vocab_max, temp_vocab_size, lexical_category.value, norm_score) %>%
  spread(lexical_category.value, norm_score, fill = 0) %>%
# get rid of NAs
  mutate(temp_vocab_size = ifelse(is.na(temp_vocab_size), 0, temp_vocab_size)) %>%
  group_by(data_id) %>%
# normalize vocabulary size
  mutate(vocab_size = sum(temp_vocab_size)/vocab_max) %>%
   ungroup() %>%
  mutate(nouns = ifelse(is.na(nouns.produces), 0, nouns.produces),
         predicates = ifelse(is.na(predicates.produces), 0, predicates.produces),
         function_words =
           ifelse(is.na(function_words.produces), 0, function_words.produces)) %>%
  select(language, data_id, vocab_size, nouns, predicates, function_words) %>%
# We still have null duplicates of observations (where all numeric variables = 0)
# and we need to get rid of them. We needed to keep them earlier to make sure not to
# filter out observations where no words are yet acquired (vocab_size = 0)
  arrange(data_id, desc(nouns), desc(predicates), desc(function_words)) %>%
  distinct(data_id, .keep_all = TRUE) %>%
# add the ages of each data_id
  add_age(hm.WG_multiling_age, .)

print(head(df.WG_multiling_lexcat_produces))

# This second data frame counts acquired words as 'understands' (understands + produces)
df.WG_multiling_lexcat_understands <- df.WG_multiling_lexcat_data %>%
# change all "produces" values to "understands", since production implies comprehension
  mutate(value_combined =
           ifelse((value %in% c("produces","understands")), "understands", NA)) %>%
  group_by(data_id) %>%
# We will use this to normalize the vocab_size
  mutate(vocab_max = n()) %>%
  ungroup() %>%
  group_by(data_id, value_combined) %>%
# These are temporary counts which will make sense once we spread the data
# according to lexical_category and value_combined
  mutate(temp_vocab_size = n()) %>%
  ungroup() %>%
  group_by(data_id,lexical_category,value_combined) %>%
  mutate(temp_score = n()) %>%
  ungroup() %>%
  group_by(data_id,lexical_category) %>%
# We will use this to normalize the the proportional counts for each
# lexical category
  mutate(max_score = n()) %>%
  ungroup() %>%
  select(language, data_id, lexical_category,
         value_combined, vocab_max, temp_vocab_size, temp_score, max_score) %>%
  distinct(data_id, lexical_category, value_combined, .keep_all = TRUE) %>%
# We combine lexical_category and value_combined in order to properly
# spread the scores accross lexical_category and value_combined
  mutate(lexical_category.value_combined =
           paste(lexical_category, value_combined, sep = ".")) %>%
# normalize scores
```

```r
  mutate(norm_score = temp_score/max_score) %>%
# get rid of NAs
  mutate(temp_vocab_size = ifelse(is.na(value_combined),0 , temp_vocab_size)) %>%
  select(language,data_id, vocab_max, temp_vocab_size,
         lexical_category.value_combined, norm_score) %>%
  spread(lexical_category.value_combined, norm_score, fill = 0) %>%
  mutate(temp_vocab_size = ifelse(is.na(temp_vocab_size), 0, temp_vocab_size)) %>%
  group_by(data_id) %>%
# normalize vocabulary size
  mutate(vocab_size = sum(temp_vocab_size)/vocab_max) %>%
   ungroup() %>%
  mutate(nouns =
           ifelse(is.na(nouns.understands), 0, nouns.understands),
         predicates =
           ifelse(is.na(predicates.understands), 0, predicates.understands),
         function_words =
           ifelse(is.na(function_words.understands), 0, function_words.understands)) %>%
  select(language, data_id, vocab_size, nouns, predicates, function_words) %>%
# We still have null duplicates of observations (where all numeric variables = 0)
# and we need to get rid of them. We needed to keep them earlier to make sure not to
# filter out observations where no words are yet acquired (vocab_size = 0)
  arrange(data_id, desc(nouns), desc(predicates), desc(function_words)) %>%
  distinct(data_id, .keep_all = TRUE) %>%
# add the ages of each data_id
  add_age(hm.WG_multiling_age, .)

print(head(df.WG_multiling_lexcat_understands))
```

```
  lexical_category
1             <NA>
2            other
3            nouns
4       predicates
5   function_words
# A tibble: 6 x 7
  language         data_id vocab_size nouns predicates function_words   age
  <fct>              <int>      <dbl> <dbl>      <dbl>          <dbl> <int>
1 French (Quebeco~   48708          0     0          0              0     8
2 French (Quebeco~   48709          0     0          0              0     8
3 French (Quebeco~   48710          0     0          0              0     8
4 French (Quebeco~   48711          0     0          0              0     8
5 French (Quebeco~   48712          0     0          0              0     8
6 French (Quebeco~   48713          0     0          0              0     8
# A tibble: 6 x 7
  language        data_id vocab_size  nouns predicates function_words   age
  <fct>             <int>      <dbl>  <dbl>      <dbl>          <dbl> <int>
1 French (Quebec~   48708     0      0          0                0        8
2 French (Quebec~   48709     0.0348 0.0406     0.0325           0        8
3 French (Quebec~   48710     0.133  0.127      0.163            0.04     8
4 French (Quebec~   48711     0.162  0.127      0.252            0        8
5 French (Quebec~   48712     0      0          0                0        8
6 French (Quebec~   48713     0.0493 0.0558     0.0325           0.08     8
```

Here are the respective number of observations per language.

```r
df.WG_multiling_lexcat_understands %>% group_by(language) %>% count()
```

```
# A tibble: 10 x 2
# Groups:   language [10]
   language                n
   <fct>               <int>
 1 Danish               2398
 2 English (American)   2454
 3 French (French)       222
 4 French (Quebecois)    537
 5 Hebrew                 62
 6 Kigiriama             132
 7 Kiswahili              51
 8 Norwegian            2926
 9 Slovak                657
10 Spanish (Mexican)     833
```

2.2 Model comparison

In the initial studies by Bates et al. (1994) and Braginsky et al. (2015), the relations between vocabulary size and the proportions of acquired nouns, predicates, and function words were non-linear. If these relations where linear, the number of acquired nouns, predicates and function words would all be proportional to the vocabulary size and proportional to each other, indicating that lexical category does not inform word learning. In what follows, I compare linear models encoding a linear relation, a quadratic relation and a non-linear (cubic) relation between the proportion of each lexical category and the vocabulary size for both the produced vocabulary and the understood vocabulary.

```r
# helper function to compare models of the linear, quadratic and non-linear
# relations between the proportion of acquired y and the proportion of acquired
# predictor x for a given language lang.
model_comparison <- function(df, lang, y, x){
  df$y <- eval(substitute(y), df)
  df$x <- eval(substitute(x), df)
  df.lang <- df %>%
    filter(language==lang)
  fit.linear= lm(formula = y ~ 0 + x, data= df.lang)

  fit.quadratic = lm(formula = y ~ 0 + x + I(x^2), data =df.lang)

  fit.nonlinear = lm(formula = y ~ 0 + x + I(x^2) + I(x^3), data = df.lang)
  print(c(lang, substitute(y)))
  print(anova(fit.linear, fit.quadratic, fit.nonlinear))
}

#Example
model_comparison(df.WG_multiling_lexcat_produces, "English (American)", nouns, vocab_size)
```

```
[[1]]
[1] "English (American)"

[[2]]
nouns

Analysis of Variance Table
```

```
Model 1: y ~ 0 + x
Model 2: y ~ 0 + x + I(x^2)
Model 3: y ~ 0 + x + I(x^2) + I(x^3)
  Res.Df     RSS Df Sum of Sq        F    Pr(>F)
1   2453 0.87591
2   2452 0.56364  1  0.312268 1363.688 < 2.2e-16 ***
3   2451 0.56125  1  0.002394   10.455  0.001239 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I do not print the model evaluation results because they are very long given that I am comparing models for every language for all three lexical categories.

```r
# This prints the model comparison for every language for all three lexical categories
# Comparisons for produced vocabulary
print("PRODUCED VOCABULARY")
for(lang in languages){
  model_comparison(df.WG_multiling_lexcat_produces, lang, nouns, vocab_size)
  model_comparison(df.WG_multiling_lexcat_produces, lang, predicates, vocab_size)
  model_comparison(df.WG_multiling_lexcat_produces, lang, function_words, vocab_size)
}
print("UNDERSTOOD VOCABULARY")
# Comparisons for understood vocabulary
for(lang in languages){
  model_comparison(df.WG_multiling_lexcat_understands, lang, nouns, vocab_size)
  model_comparison(df.WG_multiling_lexcat_understands, lang, predicates, vocab_size)
  model_comparison(df.WG_multiling_lexcat_understands, lang, function_words, vocab_size)
}
```

In the case of produced vocabulary, I find that the use of a model with a quadratic predictor over a single linear predictor is significant ($p < 0.05$) for the proportion of nouns as the dependent variable in all languages except French (Quebecois) (the cubic predictor is also significant ($p < 0.05$) in most languages, except English (American) and Hebrew). The model of a quadratic relation is also significant ($p < 0.05$) in the case of predicates in all languages except Slovak (the cubic predictor is significant in all languages, except Danish and Hebrew). In the case of function words, the use of a quadratic predictor is significant ($p < 0.05$) in all languages except French (French), Hebrew, and Kiswahili.

In the case of understood vocabulary, the picture is slightly different. For nouns, the inclusion of a quadratic predictor in the model is significant ($p < 0.05$) in all languages except Hebrew, Kiswahili, and Kigiriama. For predicates, the inclusion of quadratic predictor is not significant in most cases (except for French (Quebecois), Slovak, and Kigiriama), in other words, a linear relationship between the proportion of predicates and the proportion of words acquired (vocabulary size) best fits the data. As for functional words, a model with a quadratic predictor is significant ($p < 0.05$) in all languages except Kiswahili and Kigiriama (a cubic predictor is significant in all languages except Danish, Hebrew, Kiswahili, Spanish (Mexican), and Kigiriama).

In the result section, I plot the model configuration which was significant in the majority of languages for a given dependent variable.

2.3 Results

The following code calculates the model fit for produced nouns, predicates, and function words given the produced vocabulary size.

```r
# no pooling between languages fit model to each lexical category
# For Production data ("produces")
```

```r
df.multiling_lexcat_produces_no_pooling =  df.WG_multiling_lexcat_produces %>%
  group_by(language) %>%
# fit function_words
  nest(vocab_size, function_words) %>%
  mutate(fit = map(data, ~ lm(function_words ~ 0 +vocab_size+I(vocab_size^2),
                              data = .)),
         augment = map(fit, augment)) %>%
  unnest(augment) %>%
  clean_names() %>%
  select(function_words_fitted = fitted) %>%
  cbind(df.WG_multiling_lexcat_produces, .)
# fit predicates
df.multiling_lexcat_produces_no_pooling =  df.WG_multiling_lexcat_produces %>%
  group_by(language) %>%
  nest(vocab_size, predicates) %>%
  mutate(fit = map(data, ~ lm(predicates ~ 0 +vocab_size+I(vocab_size^2)+I(vocab_size^3),
                              data = .)),
         augment = map(fit, augment)) %>%
  unnest(augment) %>%
  clean_names() %>%
  select(predicates_fitted = fitted) %>%
  cbind(df.multiling_lexcat_produces_no_pooling, .)
# fit nouns
df.multiling_lexcat_produces_no_pooling =  df.WG_multiling_lexcat_produces %>%
  group_by(language) %>%
  nest(vocab_size, nouns) %>%
  mutate(fit = map(data, ~ lm(nouns ~ 0 +vocab_size+I(vocab_size^2)+I(vocab_size^3),
                              data = .)),
         augment = map(fit, augment)) %>%
  unnest(augment) %>%
  clean_names() %>%
  select(nouns_fitted = fitted) %>%
  cbind(df.multiling_lexcat_produces_no_pooling, .)
```

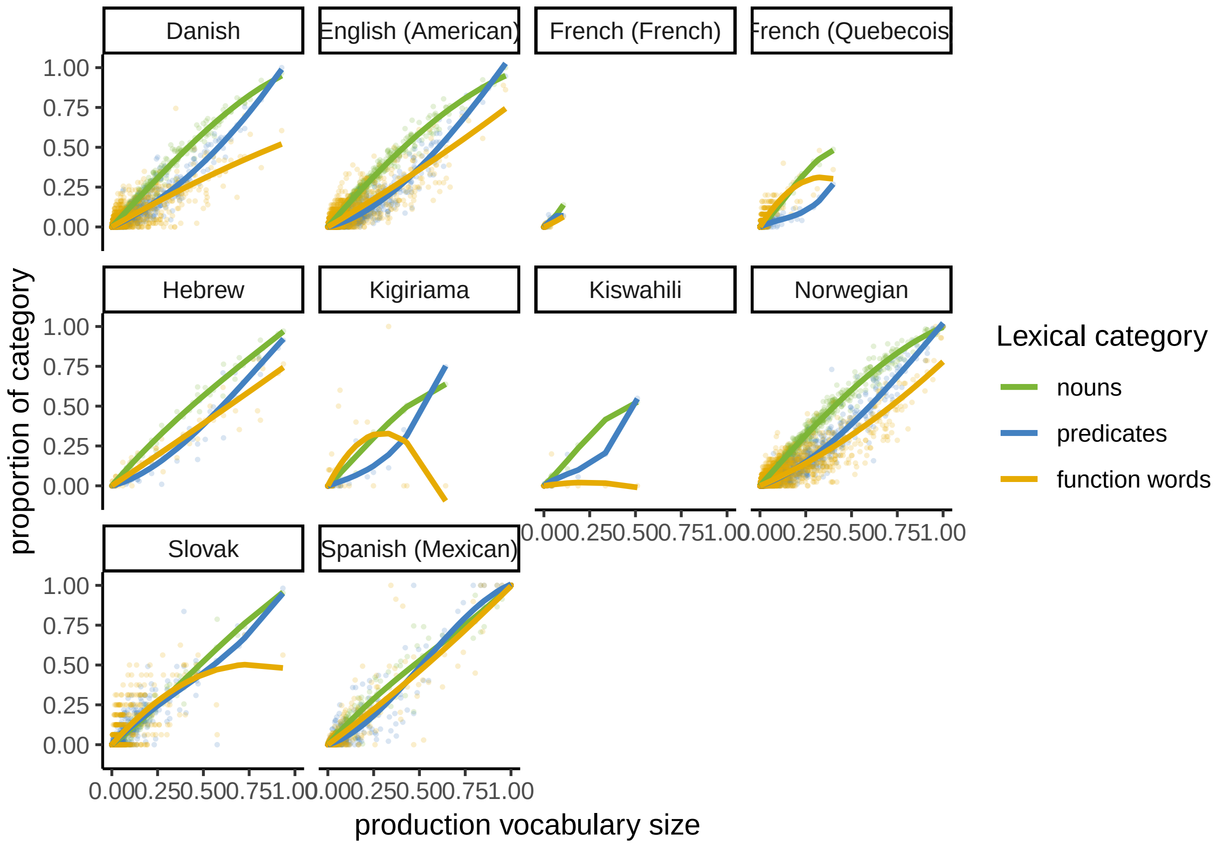
This block does the same as the previous one, fitting models for all three dependent variables, but for understood vocabulary.

```r
# For Comprehension data ("understands")
df.multiling_lexcat_understands_no_pooling =  df.WG_multiling_lexcat_understands %>%
  group_by(language) %>%
# fit function_words
  nest(vocab_size, function_words) %>%
  mutate(fit = map(data, ~ lm(function_words ~ 0 +vocab_size+I(vocab_size^2)+I(vocab_size^3),
                              data = .)),
         augment = map(fit, augment)) %>%
  unnest(augment) %>%
  clean_names() %>%
  select(function_words_fitted = fitted) %>%
  cbind(df.WG_multiling_lexcat_understands, .)
# fit predicates
df.multiling_lexcat_understands_no_pooling =  df.WG_multiling_lexcat_understands %>%
  group_by(language) %>%
  nest(vocab_size, predicates) %>%
  mutate(fit = map(data, ~ lm(predicates ~ 0 + vocab_size, data = .)),
```

```
         augment = map(fit, augment)) %>%
  unnest(augment) %>%
  clean_names() %>%
  select(predicates_fitted = fitted) %>%
  cbind(df.multiling_lexcat_understands_no_pooling, .)
# fit nouns
df.multiling_lexcat_understands_no_pooling =  df.WG_multiling_lexcat_understands %>%
  group_by(language) %>%
  nest(vocab_size, nouns) %>%
  mutate(fit = map(data, ~ lm(nouns ~ 0 + vocab_size+I(vocab_size^2), data = .)),
         augment = map(fit, augment)) %>%
  unnest(augment) %>%
  clean_names() %>%
  select(nouns_fitted = fitted) %>%
  cbind(df.multiling_lexcat_understands_no_pooling, .)
```

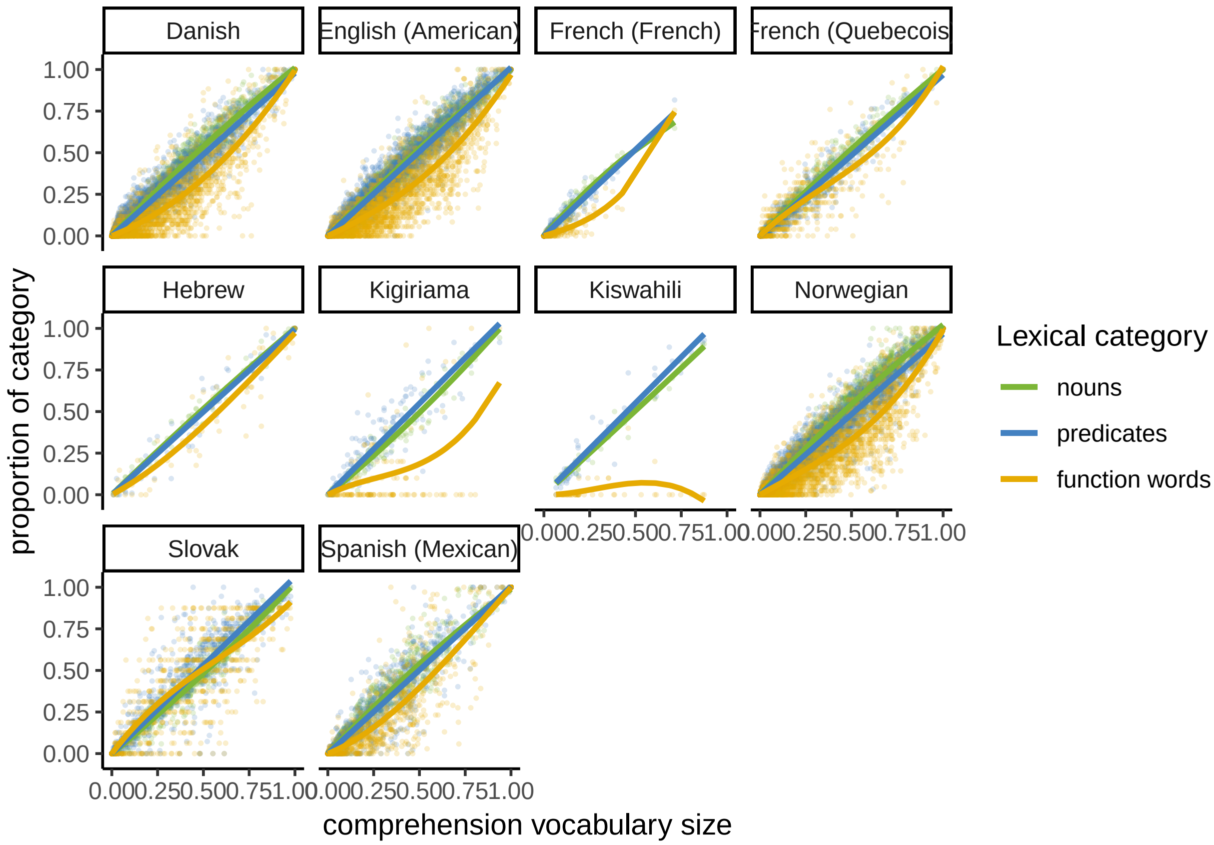
The following plots represent the fitted models for nouns, predicates, and function words (`proportion of lexical category  ~ vocabulary size`) in each language. This first set is for produced vocabulary.

```
ggplot(data = df.multiling_lexcat_produces_no_pooling,
       mapping = aes(x = vocab_size,
                     group= language)) +
  geom_point(aes(y=nouns), size= 0.3, color = "#7CB637", alpha= 0.2) +
  geom_point(aes(y=predicates),size= 0.3, color = "#4381C1", alpha= 0.2) +
  geom_point(aes(y=function_words),size= 0.3, color = "#E6AB02", alpha= 0.2) +
  geom_line(aes(y=nouns_fitted, color = "nouns"), size= 1) +
  geom_line(aes(y=predicates_fitted, color = "predicates"), size=1) +
  geom_line(aes(y=function_words_fitted, color = "function words"),size=1) +
  scale_color_manual(name = "Lexical category",
                      breaks = c("nouns", "predicates", "function words"),
                      values = c("nouns" ="#7CB637",
                                 "predicates" = "#4381C1",
                                 "function words" = "#E6AB02")) +
  facet_wrap(vars(language)) +
  ylab("proportion of category") +
  xlab("production vocabulary size") +
  theme(legend.position = "right")
```



The second set of plots the results for understood vocabulary. (`proportion of lexical category ~ vocabulary size`)

```
ggplot(data = df.multiling_lexcat_understands_no_pooling,
       mapping = aes(x = vocab_size,
                     group= language)) +
  geom_point(aes(y=nouns), size= 0.3, color =  "#7CB637", alpha= 0.2) +
  geom_point(aes(y=predicates),size= 0.3, color = "#4381C1", alpha= 0.2) +
  geom_point(aes(y=function_words),size= 0.3, color = "#E6AB02", alpha= 0.2) +
  geom_line(aes(y=nouns_fitted, color = "nouns"), size= 1) +
  geom_line(aes(y=predicates_fitted, color = "predicates"), size=1) +
  geom_line(aes(y=function_words_fitted, color = "function words"),size=1) +
  scale_color_manual(name = "Lexical category",
                      breaks = c("nouns", "predicates", "function words"),
                      values = c("nouns" ="#7CB637",
                                 "predicates" = "#4381C1",
                                 "function words" = "#E6AB02")) +
  facet_wrap(vars(language)) +
  ylab("proportion of category") +
  xlab("comprehension vocabulary size") +
  theme(legend.position = "right")
```



2.4 Discussion

In the case of produced vocabulary, we do observe a noun bias in most of the languages, except for French (Quebecois), Kigiriama, Slovak, and Spanish (Mexican). In the case of French (French), there is not enough diversity in vocabulary sizes among participants to be able to conclude anything. Function words do seem to lag behind in there acquisition at first in many languages, though this does not seem to be the case in French (Quebecois), Slovak, or Spanish (Mexican).

In the case of understood vocabulary, we do not observe a noun bias in most of the languages, except Danish and Norwegian. There is a bias against function words in all of the languages, but Slovak.

Thus, unlike Bates et al. (1994), I do not find a strong noun bias. Like Braginsky et al. (2015), I find that the noun bias varies cross-linguistically. I did, however, find that a bias against function words was present in most languages suggesting that even though function words are more prevalent in language, their syntactic complexity does impact their acquisition trajectory. This was also found by Braginsky et al. (2015).

3 Analysis 2: syntactic complexity

The complexity score is a simple metric of syntactic development which can be calculated from CDI forms. It represents a raw count of the number of acquired morphosyntactic phenomena in a given language's version of the form (e.g. In English, the use of copulas, conjuncts, pronouns, tense, verbal agreement, etc.). These raw counts are then normalized by dividing them by the theoretical maximum score for each language. As such, they are a proxy for measuring syntactic development.

If word learning and grammar learning are overlapping learning processes then we expect there to be a strong correlation between vocabulary scores and complexity scores cross-linguistically. Furthermore, we should expect that this is not solely an effect of age. The following analysis will determine whether or not we observe this correlation between vocabulary size and complexity score, as well as whether it is stronger than the correlation with age.

3.1 Data wrangling

In this analysis, I use data collected on 'WS' CDI forms for children whose ages ranged between 16 and 36 months in all the languages but Hebrew and Slovak which had to be excluded due to data coding issues.

The following code block calculates the vocabulary score and complexity score for each data_id (child/instrument) and collects them in a data frame (either the raw score or the normalized score). This data is to be used to plot and model the correlation between vocabulary size ('word' score) and complexity ('complexity' score).

```
# There are many distinct types anotated for in this data, but I will assume that:
# only items of type = "word" go into calculating the vocabulary score;
# only items of type = "complexity" go into calculating the complexity score;
df.WS_multiling_instrument_data %>% distinct(type)

# Filter to keep only items which are part of either the complexity
df.WS_multiling_complexity_data <- df.WS_multiling_instrument_data %>%
  filter(type=="word" | type == "complexity")



# I have to exclude the data from Slovak and Hebrew because they use a 1-4 choice
# system as there value for some complexity items and I have no way of knowing which
# of the 4 variants corresponds to an acquired complexity item.

#df.WS_multiling_complexity_data %>% distinct(value)
#test <- df.WS_multiling_complexity_data %>% filter(value == 1) %>% distinct(language)

df.WS_multiling_complexity_data <- df.WS_multiling_complexity_data %>%
  filter(language !="Hebrew" & language != "Slovak")

# value can have any of the following values = [produces, NA, "", complex, simple]
# I consider something acquired for the purpose of calculating a vocabulary or
# complexity score if value is in [produces,complex]
#df.WS_multiling_complexity_data %>% ungroup() %>% distinct(type,value)


# There are 21,640 distinct data_ids in this data frame which means I want to end
# up with a data frame containing 21,640 observations with both a complexity score
# and a vocabulary score

# The following chain computes these scores
df.WS_multiling_complexity_data <- df.WS_multiling_complexity_data %>%
  group_by(data_id,type,value) %>%
  mutate(temp_score = n()) %>%
  ungroup() %>%
  group_by(data_id,type) %>%
# max score is the theoretical max score on a given form for either vocabulary or
# complexity. Given that these values vary across languages, we can use this to
# normalize scores
```

```r
  mutate(max_score = n()) %>%
  ungroup() %>%
  select(language, data_id, type, value, temp_score, max_score) %>%
# remove duplicate information
  distinct(data_id, type, value, .keep_all = TRUE) %>%
# keep scores for produced/complex values and scores which are zero (temp_score ==
# max_score if value==simple/NA/"" for all complexity or word items)
  filter(value=="produces" | value == "complex" | temp_score == max_score) %>%
# set score to zero if value is neither produces or complex
  mutate(score = ifelse((is.na(value) | !(value=="produces" | value == "complex")),
                       0,
                       temp_score)) %>%
# calculate normalized scores
  mutate(norm_score = score/max_score) %>%
  select(language,data_id,type,score, max_score, norm_score) %>%
  arrange(data_id, type)

print(head(df.WS_multiling_complexity_data))


# The following data frame contains exactly one observation for each data_id with
# normalized scores for both vocabulary and complexity. It will be
# used for plotting and models.
df.WS_multiling_complexity_normalized_score <- df.WS_multiling_complexity_data %>%
  select(language, data_id, type, norm_score) %>%
  spread(type, norm_score, fill = 0) %>%
# add the ages of each data_id
  add_age(hm.WS_multiling_age, .)

print(head(df.WS_multiling_complexity_normalized_score))
```

```
                   type
1                  word
2         how_use_words
3          word_endings
4      word_forms_nouns
5      word_forms_verbs
6    word_endings_nouns
7    word_endings_verbs
8               combine
9            complexity
10           word_forms
11         verb_endings
12       pretend_parent
13      pretend_objects
14 small_parts_of_words
15      word_complexity
16            new_words
17            use_items
# A tibble: 6 x 6
  language  data_id type       score max_score norm_score
  <fct>       <int> <fct>      <dbl>     <int>      <dbl>
1 Norwegian   60401 complexity     0        42     0
2 Norwegian   60401 word          44       731     0.0602
```

```
3 Norwegian   60402 complexity     0       42     0
4 Norwegian   60402 word           9      731     0.0123
5 Norwegian   60403 word          59      731     0.0807
6 Norwegian   60404 word         119      731     0.163
# A tibble: 6 x 5
  language data_id complexity  word   age
  <fct>      <int>      <dbl> <dbl> <int>
1 Danish    110697      0.364 0.712    29
2 Danish    110698      0.515 0.634    29
3 Danish    110699      0.364 0.579    29
4 Danish    110700      0.909 0.699    29
5 Danish    110701      0.303 0.572    29
6 Danish    110702      0.242 0.599    29
```

Here are number of observations for each language.

```r
df.WS_multiling_complexity_normalized_score %>% group_by(language) %>% count()
```

```
# A tibble: 8 x 2
# Groups:   language [8]
  language               n
  <fct>              <int>
1 Danish              3714
2 English (American)  5846
3 French (French)      665
4 French (Quebecois)   827
5 Kigiriama            100
6 Kiswahili             90
7 Norwegian           9304
8 Spanish (Mexican)   1094
```

3.2 Model comparison

The following tests determine whether the relation between vocabulary size and complexity score as well as the relation between age and complexity score are best represented by a linear, quadratic, or cubic relation.

```r
languages2 = df.WS_multiling_complexity_normalized_score %>%
  distinct(language) %>% .$language

#EXAMPLE
print("VOCABULARY SCORE AS PREDICTOR")
model_comparison(df.WS_multiling_complexity_normalized_score, "English (American)", complexity, word)
print("AGE AS PREDICTOR")
model_comparison(df.WS_multiling_complexity_normalized_score, "English (American)", complexity, age)
```

```
[1] "VOCABULARY SCORE AS PREDICTOR"
[[1]]
[1] "English (American)"

[[2]]
complexity

Analysis of Variance Table

Model 1: y ~ 0 + x
```

```
Model 2: y ~ 0 + x + I(x^2)
Model 3: y ~ 0 + x + I(x^2) + I(x^3)
  Res.Df    RSS Df Sum of Sq        F    Pr(>F)
1   5845 290.92
2   5844 257.44  1    33.481 760.9354 < 2.2e-16 ***
3   5843 257.09  1     0.342   7.7779  0.005306 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
[1] "AGE AS PREDICTOR"
[[1]]
[1] "English (American)"

[[2]]
complexity

Analysis of Variance Table

Model 1: y ~ 0 + x
Model 2: y ~ 0 + x + I(x^2)
Model 3: y ~ 0 + x + I(x^2) + I(x^3)
  Res.Df    RSS Df Sum of Sq        F    Pr(>F)
1   5519 432.05
2   5518 318.85  1   113.199 1971.961 < 2.2e-16 ***
3   5517 316.70  1     2.151   37.463 9.961e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

I do not print the model evaluation results for all languages for both predictors because they are quite long.

```r
print("VOCABULARY SCORE AS PREDICTOR")
for(lang in languages2){
  model_comparison(df.WS_multiling_complexity_normalized_score, lang, complexity, word)
}

print("AGE AS PREDICTOR")
for(lang in languages2){
  model_comparison(df.WS_multiling_complexity_normalized_score, lang, complexity, age)
}
```

In the case of vocabulary size as a predictor, the model which contains a quadratic predictor is a significantly better fit ($p < 0.05$) for all the languages except French (French) and Kiswahili. A model of a quadratic relation is also the best fit when age is the predictor, in all but French (Quebecois), French (French), and Kiswahili.

For these reasons, I fit a model of a quadratic relation to the data for both the model with vocabulary size as a predictor and the model with age as a predictor. I calculate the R-squared coefficient for each one of these models in each language to determine how much of the variance these models account for and whether vocabulary size is a better predictor than age.

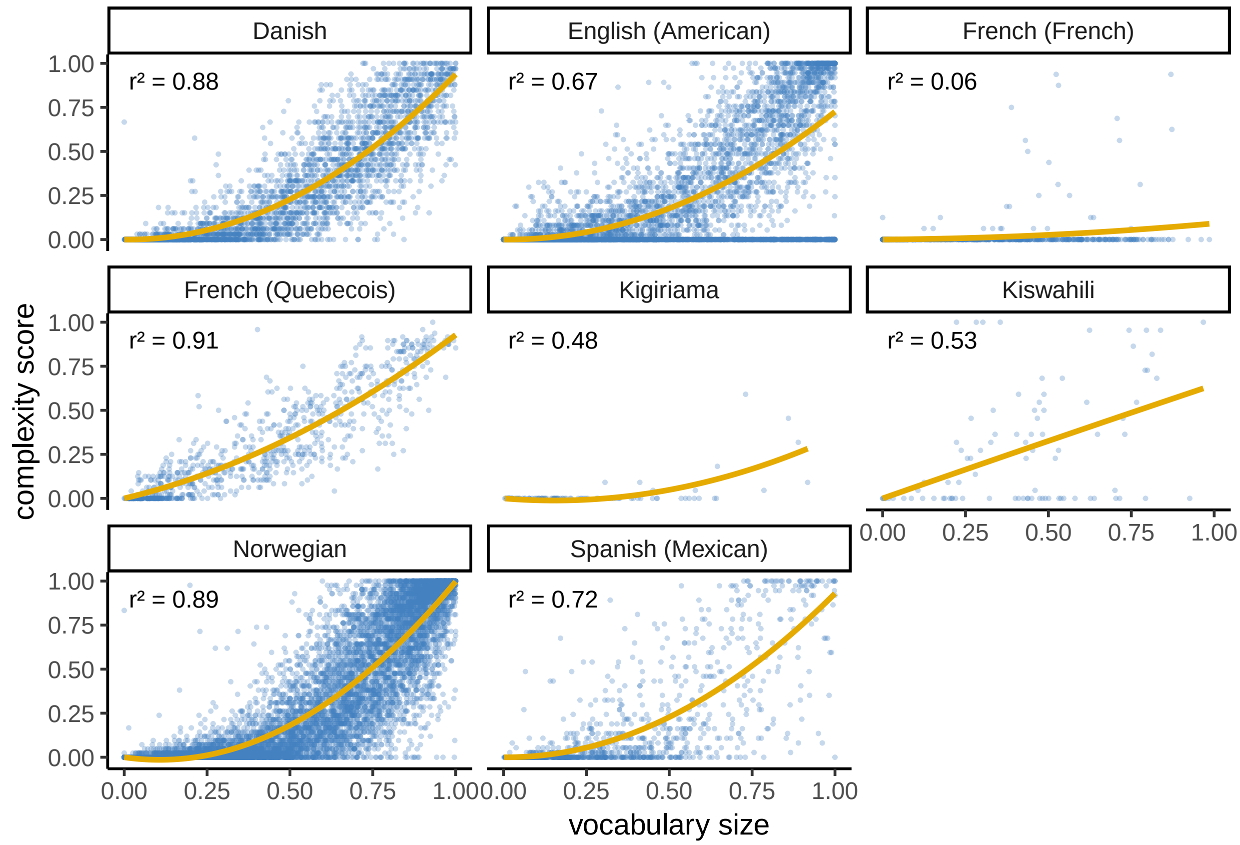
```r
# The best model cross linguisistically is the model with a quadratic predictor
# I will retrieve the r_squared scores for each language .

vocab_model <- function(data) {
  lm(complexity ~ 0 + word + I(word^2), data = data)
}
```

```
df.WS_multiling_complexity_vocab_models = df.WS_multiling_complexity_normalized_score %>%
  group_by(language) %>%
  nest() %>%
  mutate(
  fit.quadratic = map(data,vocab_model),
  rsq = map_dbl(fit.quadratic, ~summary(.x)$r.squared),
  rsq_print = sprintf("r² = %.2f", rsq)
  )

age_model <- function(data) {
  lm(complexity ~ 0 + age + I(age^2), data = data)
}

df.WS_multiling_complexity_age_models = df.WS_multiling_complexity_normalized_score %>%
  group_by(language) %>%
  nest() %>%
  mutate(
  fit.quadratic = map(data,age_model),
  rsq = map_dbl(fit.quadratic, ~summary(.x)$r.squared),
  rsq_print = sprintf("r² = %.2f", rsq)
  )
```

3.3 Results

The following plot presents the relation between vocabulary size and complexity score (complexity ~ vocabulary size). Each point is an observation and the line represents the fitted model of a quadratic relation. R-squared coefficients are reported for each language.

```
#vocabulary size as a predictor
ggplot(data = df.WS_multiling_complexity_normalized_score,
       aes(x=word, y=complexity, group=language)) +
  facet_wrap(vars(language)) +
  geom_point(size= 0.3, color = "#4381C1", alpha= 0.3) +
  geom_smooth(method = "lm",
              formula = y ~ 0 + x + I(x^2), size = 1,
              color = "#E6AB02",
              se = FALSE) +
  geom_text(aes(label = rsq_print), x = 0.15, y = 0.9, size = 3,
              data = df.WS_multiling_complexity_vocab_models) +
  ylab("complexity score") +
  xlab("vocabulary size")
```

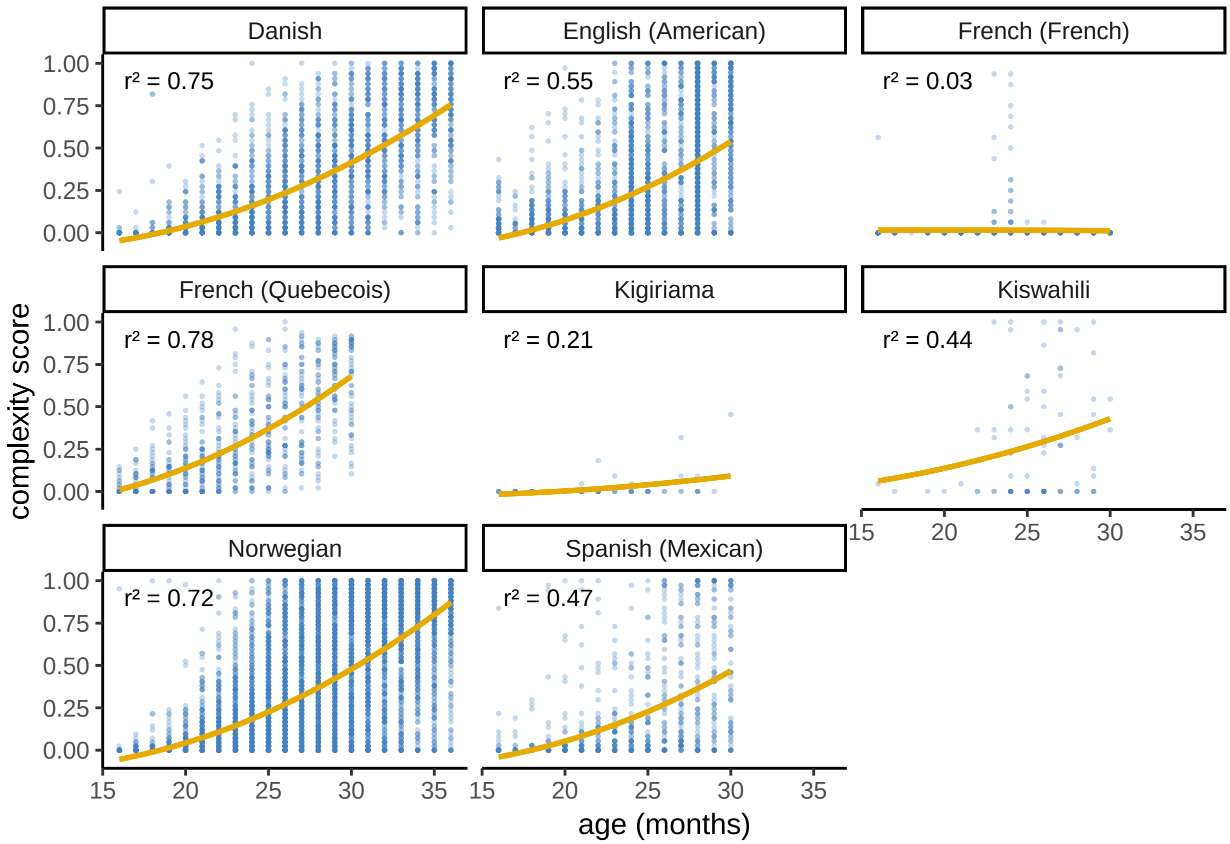


This plot presents the relation between age and complexity score (complexity ~ age). Each point is an observation and the line represents the fitted model of a quadratic relation. R-squared coefficients are reported for each language.

```r
#age as a predictor
ggplot(data = df.WS_multiling_complexity_normalized_score,
       aes(x=age, y=complexity, group=language)) +
  facet_wrap(vars(language)) +
  geom_point(size= 0.3, color = "#4381C1", alpha= 0.3) +
  geom_smooth(method = "lm",
              formula = y ~ 0 + x + I(x^2), size = 1,
              color = "#E6AB02",
              se = FALSE) +
  geom_text(aes(label = rsq_print), x = 19, y = 0.9, size = 3,
              data = df.WS_multiling_complexity_age_models) +
  ylab("complexity score") +
  xlab("age (months)")
```

```
Warning: Removed 368 rows containing non-finite values (stat_smooth).
```

```
Warning: Removed 368 rows containing missing values (geom_point).
```



3.4 Discussion

First, a model with vocabulary size as a predictor explains much more of the variance in complexity scores than a model with age as a predictor. Furthermore, vocabulary size and complexity score are highly correlated in most languages, with R-squared coefficients above 0.66 in all languages except French (French), Kigiriama, and Kiswahili.

The French (French) data only had 16 complexity items on their CDI forms, which might explain why most of the complexity scores are around zero. Kigiriama (n= 184) and Kiswahili (n=178) have much lower n than the other languages. They also have a maximum theoretical score of only 22 items, while other languages had between 33 and 48 complexity items as part of their CDI forms. These reasons may explain why the model fits are not as tight for these languages.

Overall, like Bates et al. (1994) and Braginsky et al. (2015), I find that vocabulary size strongly correlates with complexity score cross-linguistically.

4 Conclusion

The results of both analysis 1, the relation between vocabulary size and the proportion of acquired words in each lexical category, and analysis 2, the relation between vocabulary size and complexity score, suggest that the processes involved in syntactic development and word learning are interlinked.

In future work, I hope to explore the nature of the interactions between word learning and grammar learning. To do so, I will explicitly model possible dependencies between lexicon growth and grammar learning in generative models representing language production/comprehension. These models will then be used in tandem with Bayesian statistical inference as a proxy for the human learning process. I hope to then compare different models' outputs to actual children's speech production to determined which hypothesized model best explains children's production trajectory.

References

Alcock, KJ, K Rimba, P Holding, P Kitsao-Wekulo, Amina Abubakar, and CRJC Newton. 2015. "Developmental Inventories Using Illiterate Parents as Informants: Communicative Development Inventory (Cdi) Adaptation for Two Kenyan Languages." *Journal of Child Language* 42 (4): 763–85.

Baker, Mark C. 2005. "Mapping the Terrain of Language Learning." *Language Learning and Development* 1 (1): 93–129.

Bates, Douglas, and Martin Maechler. 2018. *Matrix: Sparse and Dense Matrix Classes and Methods*. https://CRAN.R-project.org/package=Matrix.

Bates, Douglas, Martin Maechler, Ben Bolker, and Steven Walker. 2018. *Lme4: Linear Mixed-Effects Models Using 'Eigen' and S4*. https://CRAN.R-project.org/package=lme4.

Bates, Elizabeth, Virginia Marchman, Donna Thal, Larry Fenson, Philip Dale, J Steven Reznick, Judy Reilly, and Jeff Hartung. 1994. "Developmental and Stylistic Variation in the Composition of Early Vocabulary." *Journal of Child Language* 21 (1). Cambridge University Press: 85–123.

Bleses, Dorthe, Werner Vach, Malene Slott, Sonja Wehberg, Pia Thomsen, Thomas O Madsen, and Hans Basbøll. 2008. "The Danish Communicative Developmental Inventories: Validity and Main Developmental Trends." *Journal of Child Language* 35 (3): 651–69.

Boudreault, MC, EA Cabirol, D Poulin-Dubois, A Sutton, and N Trudeau. 2007. "MacArthur Communicative Development Inventories: Validity and Preliminary Normative Data." *La Revue d'orthophonie et d'audiologie* 31 (1): 27–37.

Bowerman, Melissa. 1973a. *Early Syntactic Development: A Cross-Linguistic Study with Special Reference to Finnish*. Cambridge University Press.

———. 1973b. "Structural Relationships in Children's Utterances: Syntactic or Semantic?" In *Cognitive Development and Acquisition of Language*, 197–213.

Braginsky, Mika. 2018. *Wordbankr: Accessing the Wordbank Database*. https://CRAN.R-project.org/package=wordbankr.

Braginsky, Mika, Daniel Yurovsky, Virginia A Marchman, and Michael C Frank. 2015. "Developmental Changes in the Relationship Between Grammar and the Lexicon." In *CogSci*, 256–61.

Braine, Martin DS. 1963. "The Ontogeny of English Phrase Structure: The First Phase." *Language*, 1–13.

Brown, R, and U Bellugi. 1963. "The Acquisition of Syntax." In *Verbal Behavior and Learning: Problems and Processes*, edited by C. N. Cofer and B. S. Musgrave, 158–97. McGraw-Hill.

Brown, Roger. 1973. *A First Language: The Early Stages*. Harvard U. Press.

Brown, Roger, and Ursula Bellugi. 1964. "Three Processes in the Child's Acquisition of Syntax." *Harvard Educational Review* 34 (2). Harvard Education Publishing Group: 133–51.

C, Bates Judith, and Elizabeth Goodman. 1997. "On the Inseparability of Grammar and the Lexicon: Evidence from Acquisition, Aphasia and Real-Time Processing." *Language and Cognitive Processes* 12 (5-6). Taylor & Francis: 507–84.

Champely, Stephane. 2018. *Pwr: Basic Functions for Power Analysis*. https://CRAN.R-project.org/

package=pwr.

Chomsky, Noam. 2001. *The Minimalist Program*. MIT press.

Fenson, Larry, Virginia A Marchman, Donna J Thal, Philip S Dale, J Steven Reznick, and others. 2014. *MacArthur-Bates Communicative Development Inventories: User's Guide and Technical Manual*. PB Brookes.

Fernald, Anne, Virginia A Marchman, and Adriana Weisleder. 2013. "SES Differences in Language Processing Skill and Vocabulary Are Evident at 18 Months." *Developmental Science* 16 (2): 234–48.

Firke, Sam. 2018. *Janitor: Simple Tools for Examining and Cleaning Dirty Data*. https://CRAN.R-project.org/package=janitor.

Frank, Michael C, Mika Braginsky, Daniel Yurovsky, and Virginia A Marchman. 2016. "Wordbank: An Open Repository for Developmental Vocabulary Data." *Journal of Child Language* 44 (3). Cambridge University Press: 677–94.

Frank, Michael, Mika Braginsky, Virginia Marchman, and Daniel Yurovsky. n.d. "Variability and Consistency in Early Language Learning: The Wordbank Project."

Gentner, Dedre. 1978. "On Relational Meaning: The Acquisition of Verb Meaning." *Child Development*, 988–98.

Henry, Lionel, and Hadley Wickham. 2018. *Purrr: Functional Programming Tools*. https://CRAN.R-project.org/package=purrr.

Jackson-Maldonado, Donna, Donna Thal, Larry Fenson, VA Marchman, T Newton, and BT Conboy. 2003. "MacArthur Inventarios Del Desarrollo de Habilidades Comunicativas (Inventarios): User's Guide and Technical Manual." *Baltimore, MD: Brookes*.

MacWhinney, Brian. 1982. "Basic Syntactic Processes." *Language Acquisition* 1: 73–136.

Miller, Wick, and Susan Ervin. 1964. "The Development of Grammar in Child Language." *Monographs of the Society for Research in Child Development*, 9–34.

Müller, Kirill, and Hadley Wickham. 2019. *Tibble: Simple Data Frames*. https://CRAN.R-project.org/package=tibble.

Ninio, Anat. 1988. "On Formal Grammatical Categories in Early Child Language." Lawrence Erlbaum Associates, Inc.

———. 2006. *Language and the Learning Curve: A New Theory of Syntactic Development*. Oxford University Press.

Perfors, Amy, and Joshua Tenenbaum. 2009. "Learning to Learn Categories." In. Cognitive Science Society.

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Robinson, David, and Alex Hayes. 2018. *Broom: Convert Statistical Analysis Objects into Tidy Tibbles*. https://CRAN.R-project.org/package=broom.

Russell, Nathan. 2017. *Hashmap: The Faster Hash Map*. https://CRAN.R-project.org/package=hashmap.

Sanchez, Alessandro, Stephan Meylan, Mika Braginsky, Kyle MacDonald, Daniel Yurovsky, and Michael C Frank. 2018. "Childes-Db: A Flexible and Reproducible Interface to the Child Language Data Exchange System." PsyArXiv.

Simonsen, Hanne Gram, Kristian E Kristoffersen, Dorthe Bleses, Sonja Wehberg, and Rune N Jørgensen. 2014. "The Norwegian Communicative Development Inventories: Reliability, Main Developmental Trends and Gender Differences." *First Language* 34 (1): 3–23.

Smith, Linda B, Susan S Jones, Barbara Landau, Lisa Gershkoff-Stowe, and Larissa Samuelson. 2002. "Object

Name Learning Provides on-the-Job Training for Attention." *Psychological Science* 13 (1): 13–19.

Thal, D. J., V. A. Marchman, and J. B. Tomblin. 2013. *Late Talking Toddlers: Characterization and Prediction of Continued Delay.* Edited by L. Rescorla and P. Dale. Baltimore, MD.: Brookes Publishing.

Trudeau, Natacha, and Ann Sutton. 2011. "Expressive Vocabulary and Early Grammar of 16-to 30-Month-Old Children Acquiring Quebec French." *First Language* 31 (4): 480–507.

Vihman, Marilyn May. 1999. "The Transition to Grammar in a Bilingual Child: Positional Patterns, Model Learning, and Relational Words." *International Journal of Bilingualism* 3: 267–99.

Von Holzen, Katie, Leo-Lyuki Nishibayashi, and Thierry Nazzi. 2018. "Consonant and Vowel Processing in Word Form Segmentation: An Infant Erp Study." *Brain Sciences* 8 (2): 24.

Weisleder, Adriana, and Anne Fernald. 2013. "Talking to Children Matters: Early Language Experience Strengthens Processing and Builds Vocabulary." *Psychological Science* 24 (11): 2143–52.

Wickham, Hadley. 2017. *Tidyverse: Easily Install and Load the 'Tidyverse'*. https://CRAN.R-project.org/package=tidyverse.

———. 2018a. *Forcats: Tools for Working with Categorical Variables (Factors)*. https://CRAN.R-project.org/package=forcats.

———. 2018b. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

———. 2019a. *Feather: R Bindings to the Feather 'Api'*. https://CRAN.R-project.org/package=feather.

———. 2019b. *Modelr: Modelling Functions That Work with the Pipe*. https://CRAN.R-project.org/package=modelr.

Wickham, Hadley, and Lionel Henry. 2018. *Tidyr: Easily Tidy Data with 'Spread()' and 'Gather()' Functions*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, and Kara Woo. 2018. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2018. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, Jim Hester, and Romain Francois. 2018. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Xie, Yihui. 2018. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://CRAN.R-project.org/package=knitr.