

A Quantitative Synthesis of Early Language Acquisition Using Meta-Analysis

Supplementary Information

Molly Lewis, Mika Braginsky, Sho Tsuji, Christina Bergmann, Page Piccinini, Alejandrina Cristia, and Michael C. Frank

2016-08-31

Contents

This document was created from an R Markdown file. Data from the paper can be interactively explore on the Metalab website. The manuscript itself was also produced from an R markdown file, and thus all analyses presented in the paper can be reproduced from this document.

```
source("../.../dashboard/global.R", chdir = TRUE)

# Remove incomplete/longitudinal MAs
all_data <- filter(all_data, dataset != "Statistical word segmentation") %>%
  filter(dataset != "Pointing and vocabulary (longitudinal)")

# need to recompute datasets summary data, based on filtered all_data
studies <- all_data %>%
  group_by(dataset) %>%
  summarise(num_experiments = n(),
            num_papers = length(unique(study_ID)))

subjects <- all_data %>%
  rowwise() %>%
  mutate(n_total = sum(c(n_1, n_2), na.rm = TRUE)) %>%
  distinct(dataset, study_ID, same_infant, .keep_all = TRUE) %>%
  group_by(dataset) %>%
  summarise(num_subjects = sum(n_total))

datasets <- datasets %>%
  rename(dataset = name) %>%
  select(-num_experiments, -num_papers, -num_subjects) %>%
  filter(dataset != "Statistical word segmentation") %>%
  filter(dataset != "Pointing and vocabulary (longitudinal)") %>%
  left_join(studies) %>%
  left_join(subjects) %>%
  rename(name = dataset)

# rename pointing and vocabulary
datasets$name = plyr::mapvalues(datasets$name, from = c("Pointing and vocabulary (concurrent)"),
                               to = c("Pointing and vocabulary"))

all_data$dataset = plyr::mapvalues(all_data$dataset , from = c("Pointing and vocabulary (concurrent)"),
                                  to = c("Pointing and vocabulary"))
```

Statistical approach

Effect sizes were computed by a script, `compute_es.R`. We calculated effect sizes from reported means and standard deviations where available, otherwise we rely on reported test-statistics (t and F). Several pre-existing MAs deal with special cases, and these are listed in the script. Except where noted, formulas are from Hedges & Olkin's textbook. All analyses are conducted with the `metafor` package, using random-effects models.

Search strategies

Meta-analyses were conducted by the authors for all but two phenomena (IDS preference and Pointing and Vocabulary). Data for these two phenomena were obtained by adapting effect size estimates from existing, published meta-analyses (Dunst, Gorman, & Hamby, 2012 and Colonna et al., 2010). Across phenomena, meta-analyses varied in their degree of systematicity in selecting papers. Below we describe the search strategy for each phenomenon. Descriptions are taken directly from the published source when available.

```
datasets %>%
  select(name, search_strategy) %>%
  mutate(name = as.factor(name)) %>%
  .[c(1,6, 4, 5, 7,8,10,12,2,9,3,11),] %>%
  kable(col.names = c("Phenomenon", "Search Strategy"), row.names = F)
```

Phenomenon	Search Strategy
Infant directed speech preference	Studies were located using motherese or parentese or fatherese or infant directed speech
Phonotactic learning	
Vowel discrimination (native)	A full search on scholar.google.com was conducted in September 2012 with the keywords
Vowel discrimination (non-native)	A full search on scholar.google.com was conducted in September 2012 with the keywords
Statistical sound category learning	
Word segmentation	We first generated a list of potentially relevant items to be included in our meta-analysis
Mutual exclusivity	We conducted a forward search based on citations of Markman and Wachtel (1988) in
Sound symbolism	We followed the PRISMA statement (Moher, Liberati, Tetzlaff, Altman, & The PRISMA
Label advantage in concept learning	We conducted a forward search based on Balaban and Waxman (1997) in Google Scholar
Online word recognition	We conducted a systematic literature review by using Google Scholar to identify peer
Gaze following	We identified papers using a Google Scholar search for "gaze following" and included
Pointing and vocabulary	The search method involved inspection of digital databases (Web of Knowledge, Psyc

P-curves

When available, we calculated p-values based on test statistics reported in the paper. However, when unavailable, we calculated p-values based on raw descriptive statistics (means and standard deviations) or reported effect sizes (the method used for IDS preference). The main text shows the results of the p-curve analysis based on p-values derived by both approaches. Here, we compare these results to same analysis on the subset of p-values derived only from reported test statistics. Below are p-curves based only on reported test-statistics.

```
ALPHA = .05
P_INCREMENT = .01

pc.data <- get_all_pc_data(all_data, ALPHA, P_INCREMENT, transform = FALSE)

p.source = pc.data %>%
  select(f.transform, f.value, dataset, study_ID, p_round) %>%
```

```

group_by(dataset) %>%
summarise(n.total = n(),
          n.transform = length(which(!is.na(f.transform))),
          sig.p = length(which(p_round < .05))) %>%
mutate(dataset = plyr::revalue(dataset,
                                c("Infant directed speech preference"="IDS preference",
                                   "Statistical sound category learning"="Statistical sound learning",
                                   "Label advantage in concept learning"="Concept-label advantage",
                                   "Vowel discrimination (native)"="Vowel discrimination\n(native)",
                                   "Vowel discrimination (non-native)"="Vowel discrimination\n(non-nati

dataset = as.factor(dataset),
dataset = gdata::reorder.factor(dataset,
                                new.order = c(2,6,10,11,9,12,4,8,3,5,1,7))) %>%
mutate(stat_only = ifelse(n.total > n.transform, 1, 0)) %>%
arrange(-stat_only) %>%
mutate(prop.ts = 1-n.transform/n.total,
       prop.ts.string = as.character(round(prop.ts, digits = 2))) %>%
as.data.frame()

prop.p <- function(d, i, p){
  sum(d[i] == p) / sum(d[i] < ALPHA)
}

get.CI <- function(my.p, v, ci) {
  boot.out = boot(data = v, prop.p, R = 1000, p = my.p)
  if (ci == "upper") {
    quantile(boot.out$t, 0.975, na.rm = T)
  } else {
    quantile(boot.out$t, 0.025, na.rm = T)
  }
}

get.all.CIS <- function(df) {
  ps <- seq(P_INCREMENT, ALPHA, P_INCREMENT)
  lower = ps %>%
    map(get.CI, df$p_round, "low") %>%
    unlist()
  upper = ps %>%
    map(get.CI, df$p_round, "upper") %>%
    unlist()
  data.frame(dataset = df$dataset[1],
             p = ps,
             ci.lower = lower,
             ci.upper = upper)
}

ci.data = pc.data %>%
  split(.$dataset) %>%
  map(function(data) get.all.CIS(data)) %>%
  bind_rows() %>%
  mutate(dataset = as.factor(dataset),
         dataset = plyr::revalue(dataset,
                                   c("Infant directed speech preference"="IDS preference",

```

```

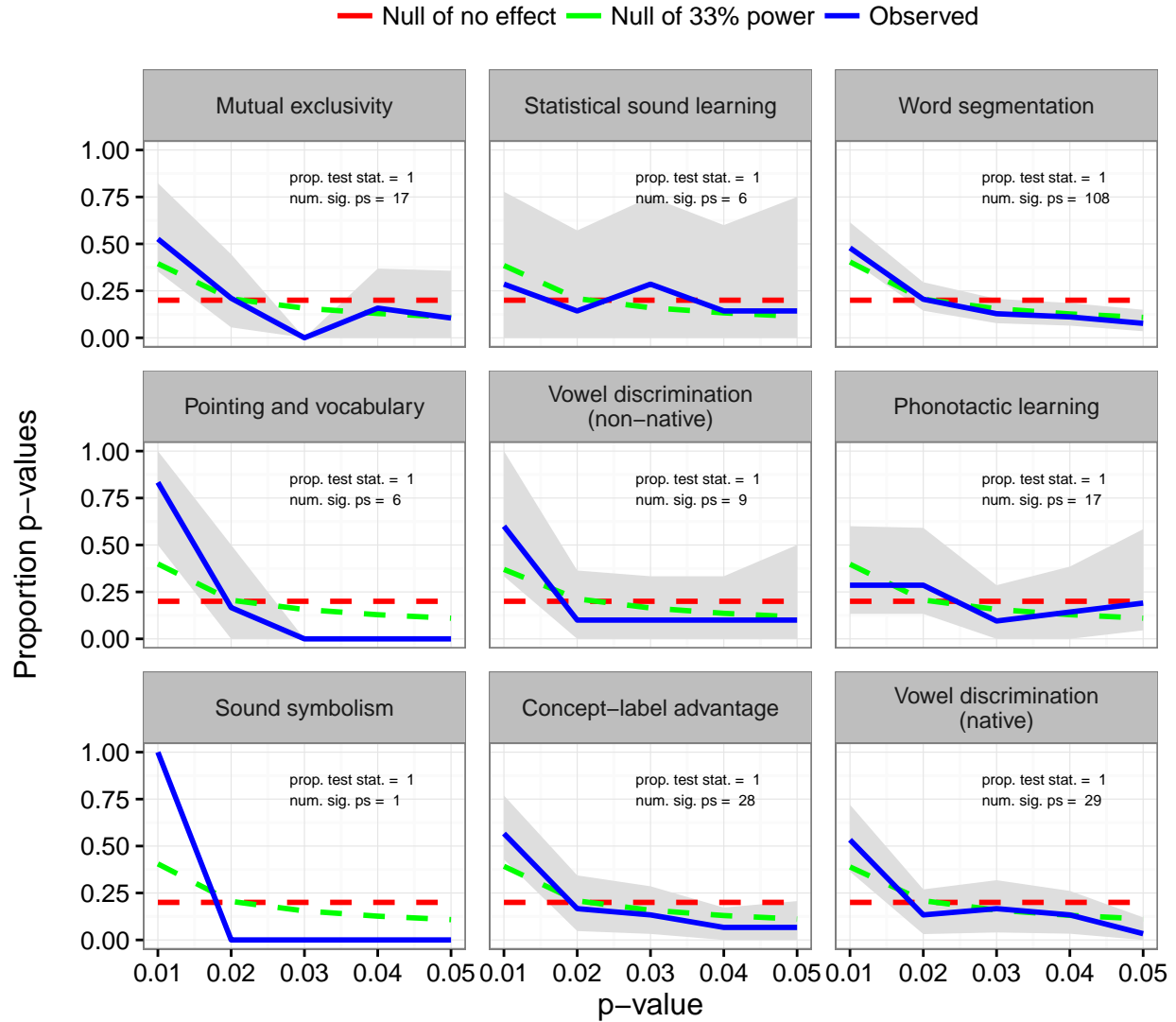
                                "Statistical sound category learning"="Statistical sound learning",
                                "Label advantage in concept learning"="Concept-label advantage",
                                "Vowel discrimination (native)"="Vowel discrimination\n(native)",
                                "Vowel discrimination (non-native)"="Vowel discrimination\n(non-nati

pc.data %>%
  group_by(dataset) %>%
  do(get_p_curve_df(., ALPHA, P_INCREMENT)) %>%
  ungroup() %>%
  mutate(dataset = as.factor(dataset),
         dataset = plyr::revalue(dataset,
                                c("Infant directed speech preference"="IDS preference",
                                  "Statistical sound category learning"="Statistical sound learning",
                                  "Label advantage in concept learning"="Concept-label advantage",
                                  "Vowel discrimination (native)"="Vowel discrimination\n(native)",
                                  "Vowel discrimination (non-native)"="Vowel discrimination\n(non-nati

         dataset = gdata::reorder.factor(dataset,
         new.order = c(2,6,10,11,9,12,4,8,3,5,1,7))) %>%

ggplot() +
  facet_wrap(~ dataset, nrow = 3) +
  geom_ribbon(aes(ymin = ci.lower, ymax = ci.upper, x = p), fill = "grey87", data = ci.data) +
  geom_line(size = 1, aes(x = p, y = value, linetype = measure, color = measure)) +
  scale_colour_manual(name = "", values = c("red", "green", "blue"),
                     labels=c("Null of no effect", "Null of 33% power", "Observed")) +
  scale_linetype_manual(values = c("dashed", "dashed", "solid"), guide = FALSE) +
  ylab("Proportion p-values\n") +
  xlab("p-value") +
  geom_text(aes(label = paste("prop. test stat. = ", prop.ts.string,
                             "\nnum. sig. ps = ", sig.p),
                x = .028, y = .8), data = p.source,
            colour = "black", size = 2, hjust = 0) +
  theme_bw() +
  theme(legend.position = "top",
        legend.key = element_blank(),
        legend.background = element_rect(fill = "transparent"),
        strip.text.x = element_text(size = 9),
        axis.title = element_text(colour = "black", size = 12),
        panel.margin = unit(.65, "lines"),
        strip.background = element_rect(fill="grey"))

```



Below we compare the test of right-skew presented in the main text for both the full set of p-values and those only derived from test statistics. In no case does the significance of the test differ between the two analyses.

```

stouffer.data = pc.data %>%
  group_by(dataset) %>%
  do(data.frame(stouffer = stouffer_test(., ALPHA))) %>%
  filter(stouffer.pp.measure == "ppr.full") %>%
  full_join(datasets %>% select(name, short_name), by= c("dataset" = "name")) %>%
  select(short_name, stouffer.Z.pp, stouffer.p.Z.pp) %>%
  mutate_each_(funs(round(., digits = 2)), vars = c("stouffer.p.Z.pp",
                                                    "stouffer.Z.pp")) %>%
  mutate(stouff_string = ifelse(is.na(as.character(stouffer.Z.pp)), "",
                                paste0(stouffer.Z.pp, " (", stouffer.p.Z.pp, ")"))) %>%
  mutate(stouff_string = sub("(0)", "< .01)", stouff_string, fixed = T)) %>%
  select(dataset, stouff_string)

stouffer.data_all = get_all_pc_data(all_data, ALPHA, P_INCREMENT, transform = TRUE) %>%
  group_by(dataset) %>%
  do(data.frame(stouffer = stouffer_test(., ALPHA))) %>%

```

```

filter(stouffer.pp.measure == "ppr.full") %>%
full_join(datasets %>% select(name, short_name), by= c("dataset" = "name")) %>%
select(short_name, stouffer.Z.pp, stouffer.p.Z.pp) %>%
mutate_each_(funs(round(., digits = 2)), vars = c("stouffer.p.Z.pp",
                                                "stouffer.Z.pp")) %>%
mutate(stouff_string = ifelse(is.na(as.character(stouffer.Z.pp)), "",
                             paste0(stouffer.Z.pp, " (", stouffer.p.Z.pp, ")"))) %>%
mutate(stouff_string = sub("(0)", "< .01)", stouff_string, fixed = T)) %>%
select(dataset, stouff_string)

# p-curve data using form all conditions (same as reported in paper)
pc.data.all <- get_all_pc_data(all_data, ALPHA, P_INCREMENT, transform = TRUE)

left_join(stouffer.data_all, stouffer.data, by = "dataset") %>%
kable(col.names = c("Phenomenon", "p-curve skew (all conditions)",
                    "p-curve skew (p-values only from test-statistics)"))

```

Phenomenon	p-curve skew (all conditions)	p-curve skew (p-values only from test-statistics)
Gaze following	-17.77 (< .01)	
Infant directed speech preference	-10.7 (< .01)	
Label advantage in concept learning	-7.3 (< .01)	-6.15 (< .01)
Mutual exclusivity	-12.87 (< .01)	-5 (< .01)
Online word recognition	-14.8 (< .01)	
Phonotactic learning	-1.52 (0.06)	-1.52 (0.06)
Pointing and vocabulary	-6.33 (< .01)	-6.33 (< .01)
Sound symbolism	-2.33 (0.01)	-2.16 (0.02)
Statistical sound category learning	-1.16 (0.12)	-0.82 (0.21)
Vowel discrimination (native)	-9.69 (< .01)	-5.42 (< .01)
Vowel discrimination (non-native)	-8.89 (< .01)	-3.24 (< .01)
Word segmentation	-9.26 (< .01)	-9.67 (< .01)

Method heterogeneity

Below we present model coefficients for method for datasets with more than one method. Coefficients are estimated from random effect meta-analytic models.

```

single_method_datasets = c("Gaze following",
                           "Label advantage in concept learning",
                           "Mutual exclusivity",
                           "Pointing and vocabulary",
                           "Online word recognition")

method.betas = data.frame()
for (i in 1:length(datasets$name)) {

  if (!(datasets$name[i] %in% single_method_datasets)) {
    d = filter(all_data, dataset == datasets$name[i])
    model = rma(d_calc ~ method - 1, vi = d_var_calc, data = d, method = "REML")

    d = data.frame(dataset = datasets$name[i],
                  method = row.names(model$b),

```

```

    betas = model$b,
    ci.lb = model$ci.lb,
    ci.ub = model$ci.ub,
    row.names = NULL)

    method.betas = rbind(method.betas,d)
  }
}

method.betas$method = gsub("method", "", method.betas$method)

ggplot(method.betas, aes(x = dataset, y = betas, ymin = ci.lb, ymax = ci.ub, color = method)) +
  geom_hline(aes(yintercept = 0), linetype = "dashed") +
  geom_pointrange() +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))

```

