Early blindness does not impair the detection of sound symbolic associations in natural language

Marco Barilari, Minah Chang and Roberto Bottini

Data Ananlysys for the paper "Early blindness does not impair the detection of sound symbolic associations in natural language" (under review)

Data preparation

- All Distractor trials were removed (4186 values) and not considered for further analyses.

Load merged data

1. subID - subject id
2. group - 'EB' either early or congenital blind, 'SC' sighted controls
3. session - it refers to the experiment version , either 'matlab' or 'gorilla'
4. category [integers 1, 2, 3, 4, 5, 6, 7]
   - category 1: size
   - category 2: brightness
   - category 3: loudness
   - category 4: velocity
   - category 5: movement
   - category 6: space
   - category 7: distance
5. wordcode - it refers to the identifier code of each stimulus, the code structure is speaker's gender + lamguage + xxx
6. language - language of origin of the stimulus
7. stimulus - sound transcription of the stimulus
8. meaning - meaning of the sitmulus refered the general synonim
9. value - 'good' or 'good_op', see note below.
10. response - participants response as generla synonim
11. grade - response coded as correct (1) or incorrect (0) compared to the 'meaning' column

*Note about 'value' header (native accuracy): given that some words in specific native languages groups can "sound" with the opposite meaning as presented in Tzeng et al., (2017), we recompute the accuracy for those stimuli that in our prestudy were mapped systematically to the opposite meaning. For these subset of stimuli, we consider the response correct if it does not match the original meaning and incorrect if it does. The 9th column "value" indicates whether the accurcy should be calculated as the word's meaning (good), or the

opposite meaning (good_op). Raw accuracy rate is calculated without taking into account the value of good or goodop.

```
allData<-read.csv("allData.csv", header = TRUE)

summary(allData$language)
```

```
##   albanian      dutch   gujarati indonesian     korean   mandarin   romanian
##       1104        782        598       1334       1610       2024       1242
##      tamil    turkish     yoruba
##       1380       1518        920
```

Descriptive analysis

Compute the over all raw and then native accuracy

```
allData$rawAccuracy <- as.integer(allData$response == allData$meaning)

allData$nativeAccuracy <- ifelse(allData$value == 'goodop',
                                 abs(allData$rawAccuracy - 1),
                                 allData$rawAccuracy + 0)
allData %>%
  group_by(group) %>%
  summarize(native_acc = mean(nativeAccuracy),
            native_sd = sd((nativeAccuracy)),
            .groups = 'keep')
```

```
## # A tibble: 2 x 3
## # Groups:   group [2]
##   group native_acc native_sd
##   <fct>      <dbl>     <dbl>
## 1 EB         0.624     0.484
## 2 SC         0.626     0.484
```

```
names(allData$language)
```

```
## NULL
```

```
allData$language <-  as.character(allData$language)

allData <- subset(allData, language != "albanian")

allData <- subset(allData, language != "dutch")

allData <- subset(allData, language != "romanian")
```

Accuracy per subjects and category

```
categoryAccuracySubj_tidy <- allData %>%
  group_by(subID, group, category) %>%
  summarize(nativeAccuracy_mean = mean(nativeAccuracy),
            nativeAccuracy_sd = sd(nativeAccuracy),
            n = n(),
            nativeAccuracy_se = nativeAccuracy_sd / sqrt(n),
            .groups = 'keep')
```

```r
print(categoryAccuracySubj_tidy)
```

```
## # A tibble: 322 x 7
## # Groups:   subID, group, category [322]
##    subID group category nativeAccuracy_~ nativeAccuracy_~     n nativeAccuracy_~
##    <fct> <fct>    <int>            <dbl>            <dbl> <int>            <dbl>
##  1 EB01  EB           1            0.902            0.300    41           0.0469
##  2 EB01  EB           2            0.686            0.471    35           0.0796
##  3 EB01  EB           3            0.643            0.497    14           0.133
##  4 EB01  EB           4            0.818            0.395    22           0.0842
##  5 EB01  EB           5            0.556            0.504    36           0.0840
##  6 EB01  EB           6            0.56             0.507    25           0.101
##  7 EB01  EB           7            0.677            0.475    31           0.0853
##  8 EB02  EB           1            0.732            0.449    41           0.0701
##  9 EB02  EB           2            0.514            0.507    35           0.0857
## 10 EB02  EB           3            0.357            0.497    14           0.133
## # ... with 312 more rows
```

Accuracy per group and category

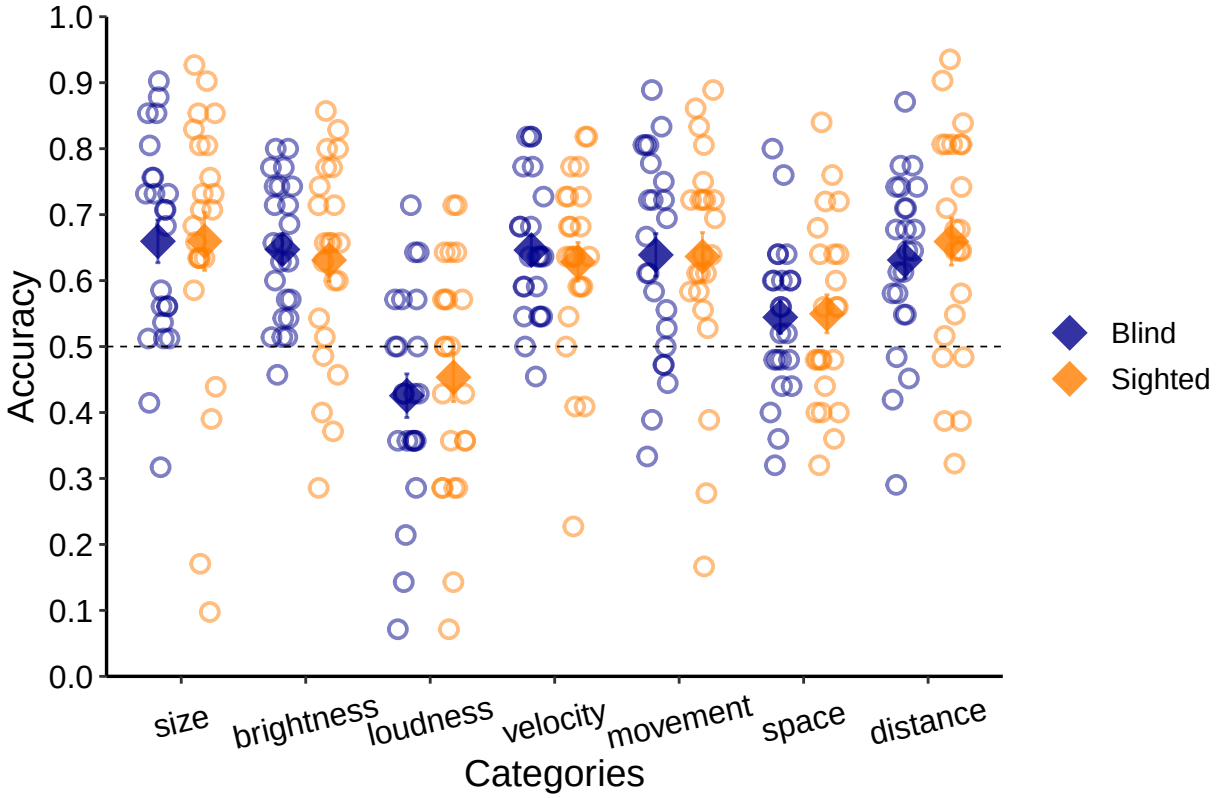
```r
categoryAccuracyGroup_tidy <- categoryAccuracySubj_tidy %>%
  group_by(group, category) %>%
  summarize(group_nativeAccuracy_mean = mean(nativeAccuracy_mean),
            group_accuracy_sd = sd(nativeAccuracy_mean),
            n = n(),
            group_nativeAccuracy_se = group_accuracy_sd / sqrt(n),
            .groups = 'keep')

print(categoryAccuracyGroup_tidy)
```

```
## # A tibble: 14 x 6
## # Groups:   group, category [14]
##    group category group_nativeAccuracy_~ group_accuracy_~     n group_nativeAcc~
##    <fct>    <int>                  <dbl>            <dbl> <int>            <dbl>
##  1 EB           1                  0.660            0.156    23           0.0325
##  2 EB           2                  0.647            0.105    23           0.0218
##  3 EB           3                  0.425            0.159    23           0.0331
##  4 EB           4                  0.646            0.104    23           0.0218
##  5 EB           5                  0.639            0.156    23           0.0324
##  6 EB           6                  0.544            0.117    23           0.0243
##  7 EB           7                  0.631            0.133    23           0.0277
##  8 SC           1                  0.660            0.212    23           0.0441
##  9 SC           2                  0.631            0.155    23           0.0324
## 10 SC           3                  0.453            0.177    23           0.0369
## 11 SC           4                  0.628            0.141    23           0.0294
## 12 SC           5                  0.636            0.175    23           0.0364
## 13 SC           6                  0.550            0.139    23           0.0289
## 14 SC           7                  0.659            0.172    23           0.0358
```

Plot of the native accuracy per category.

```r
ggplot() +
  geom_jitter(data = categoryAccuracySubj_tidy,
```

```
            aes(x = as.factor(category),
                y = nativeAccuracy_mean,
                color = group),
            position = position_jitterdodge(),
            shape = 1,
            size = 2.5,
            stroke = 1,
            alpha = .5,
            show.legend = F) +
geom_errorbar(data = categoryAccuracyGroup_tidy,
              aes(y = group_nativeAccuracy_mean,
                  x = as.factor(category),
                  ymin = group_nativeAccuracy_mean - group_nativeAccuracy_se,
                  ymax = group_nativeAccuracy_mean + group_nativeAccuracy_se,
                  color = group),
              width = .08,
              alpha = .6,
              position = position_jitterdodge(0),
              show.legend = F) +
geom_point(data = categoryAccuracyGroup_tidy,
            aes(y = group_nativeAccuracy_mean,
                x = as.factor(category),
                color = group),
            position = position_jitterdodge(0),
            shape = 18,
            size = 6,
            alpha = .8) +
theme_classic() +
scale_color_manual(values=c('darkblue','darkorange1'), labels=c('Blind', 'Sighted')) +
scale_y_continuous(limits=c(0, 1), breaks=seq(0, 1, 0.10), expand = c(0,0)) +
scale_x_discrete(limits=c("1", "2", "3", "4", "5", "6", "7"),
                 labels = c("size", "brightness", "loudness", "velocity", "movement", "space", "distan
labs(x="Categories", y="Accuracy") +
ggtitle("") +
theme(
  text=element_text(size=14),
  axis.line = element_line(size = 0.6),
  axis.text.x = element_text(size=12,colour="black",
                             angle = 13,
                             vjust = .5,
                             hjust = 0.5),
  axis.text.y = element_text(size=12,
                             colour='black'),
  legend.title=element_blank())+
geom_hline(yintercept=c(0.5), linetype="dashed", colour="black", size=0.3)
```




```
# ggsave('categories_accuracy.tiff',
#        device="tiff",
#        units="in",
#        width=7.54,
#        height=4.54,
#        dpi=300)
```

Statistical analyses

Prepare the data

```
summary(allData)
```

```
##      subID       group         session         category         wordcode
##  EB01   : 204   EB:4692   gorilla:5100   Min.   :1.000   gm1nu22:  46
##  EB02   : 204   SC:4692   matlab :4284   1st Qu.:2.000   gm1nu26:  46
##  EB03   : 204                            Median :4.000   gm1nu28:  46
##  EB04   : 204                            Mean   :3.863   gm1nu3 :  46
##  EB05   : 204                            3rd Qu.:6.000   gm1nu36:  46
##  EB06   : 204                            Max.   :7.000   gm1nu4 :  46
##  (Other):8160                                            (Other):9108
##    language            stimulus       meaning        value         response
##  Length:9384        xia     :  92   small  : 966   good  :5704   small  : 986
##  Class :character   a le    :  46   big    : 920   goodop:3680   big    : 900
##  Mode  :character   aasu    :  46   still  : 874                 moving : 854
##                     acaru   :  46   bright : 828                 bright : 837
```

```
##                    acele  :  46   near   : 828                 still  : 802
##                    adu    :  46   dark   : 782                 dark   : 773
##                    (Other):9062   (Other):4186                 (Other):4232
##      grade         rawAccuracy     nativeAccuracy
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :1.0000   Median :1.0000   Median :1.0000
##  Mean   :0.6187   Mean   :0.5573   Mean   :0.6187
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##
```

```
allData$subID <- factor(allData$subID)
allData$category <- factor(allData$category)
allData$group <- factor(allData$group)
allData$session <- factor(allData$session)
allData$language <- factor(allData$language)
allData$stimulus <- factor(allData$stimulus)
```

Groups accuracy agaisnt chance

```
native_EB <- subset(allData, group == 'EB')
binom.test(sum(native_EB$grade), nrow(native_EB), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(native_EB$grade) and nrow(native_EB)
## number of successes = 2899, number of trials = 4692, p-value < 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6037802 0.6317933
## sample estimates:
## probability of success
##              0.6178602
```

```
native_SC <- subset(allData, group == 'SC')
binom.test(sum(native_SC$grade), nrow(native_SC), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(native_SC$grade) and nrow(native_SC)
## number of successes = 2907, number of trials = 4692, p-value < 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6054961 0.6334854
## sample estimates:
## probability of success
##              0.6195652
```

Accuracy per category against chance

Category 1: size

```r
catOne_EB <- subset(allData, category == 1 & group == 'EB')
binom.test(sum(catOne_EB$grade), nrow(catOne_EB), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catOne_EB$grade) and nrow(catOne_EB)
## number of successes = 622, number of trials = 943, p-value < 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6283613 0.6898321
## sample estimates:
## probability of success
##               0.659597
```

```r
catOne_SC <- subset(allData, category == 1 & group == 'SC')
binom.test(sum(catOne_SC$grade), nrow(catOne_SC), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catOne_SC$grade) and nrow(catOne_SC)
## number of successes = 622, number of trials = 943, p-value < 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6283613 0.6898321
## sample estimates:
## probability of success
##               0.659597
```

Category 2: brightness

```r
catThree_EB <- subset(allData, category == 2 & group == 'EB')
binom.test(sum(catThree_EB$grade), nrow(catThree_EB), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catThree_EB$grade) and nrow(catThree_EB)
## number of successes = 521, number of trials = 805, p-value < 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6130828 0.6802444
## sample estimates:
## probability of success
##               0.647205
```

```r
catThree_SC <- subset(allData, category == 2 & group == 'SC')
binom.test(sum(catThree_SC$grade), nrow(catThree_SC), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catThree_SC$grade) and nrow(catThree_SC)
## number of successes = 508, number of trials = 805, p-value = 9.784e-14
## alternative hypothesis: true probability of success is not equal to 0.5
```

```
## 95 percent confidence interval:
##  0.5966710 0.6644771
## sample estimates:
## probability of success
##              0.6310559
```

Category 3: loudness

```
catFour_EB <- subset(allData, category == 3 & group == 'EB')
binom.test(sum(catFour_EB$grade), nrow(catFour_EB), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catFour_EB$grade) and nrow(catFour_EB)
## number of successes = 137, number of trials = 322, p-value = 0.008711
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.3708169 0.4814987
## sample estimates:
## probability of success
##              0.4254658
```

```
catFour_SC <- subset(allData, category == 3 & group == 'SC')
binom.test(sum(catFour_SC$grade), nrow(catFour_SC), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catFour_SC$grade) and nrow(catFour_SC)
## number of successes = 146, number of trials = 322, p-value = 0.1059
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.3981378 0.5095592
## sample estimates:
## probability of success
##              0.4534161
```

Category 4: velocity

```
catSix_EB <- subset(allData, category == 4 & group == 'EB')
binom.test(sum(catSix_EB$grade), nrow(catSix_EB), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catSix_EB$grade) and nrow(catSix_EB)
## number of successes = 327, number of trials = 506, p-value = 4.665e-11
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6028367 0.6879333
## sample estimates:
## probability of success
##              0.6462451
```

```r
catSix_SC <- subset(allData, category == 4 & group == 'SC')
binom.test(sum(catSix_SC$grade), nrow(catSix_SC), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catSix_SC$grade) and nrow(catSix_SC)
## number of successes = 318, number of trials = 506, p-value = 8.137e-09
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.5847144 0.6706922
## sample estimates:
## probability of success
##              0.6284585
```

Category 5: movement

```r
catSeven_EB <- subset(allData, category == 5 & group == 'EB')
binom.test(sum(catSeven_EB$grade), nrow(catSeven_EB), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catSeven_EB$grade) and nrow(catSeven_EB)
## number of successes = 529, number of trials = 828, p-value = 1.15e-15
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6051199 0.6716651
## sample estimates:
## probability of success
##              0.6388889
```

```r
catSeven_SC <- subset(allData, category == 5 & group == 'SC')
binom.test(sum(catSeven_SC$grade), nrow(catSeven_SC), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catSeven_SC$grade) and nrow(catSeven_SC)
## number of successes = 527, number of trials = 828, p-value = 3.606e-15
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6026662 0.6693052
## sample estimates:
## probability of success
##              0.6364734
```

Category 6: space

```r
catEight_EB <- subset(allData, category == 6 & group == 'EB')
binom.test(sum(catEight_EB$grade), nrow(catEight_EB), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catEight_EB$grade) and nrow(catEight_EB)
```

```
## number of successes = 313, number of trials = 575, p-value = 0.03696
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.5026319 0.5856059
## sample estimates:
## probability of success
##             0.5443478
```

```
catEight_SC <- subset(allData, category == 6 & group == 'SC')
binom.test(sum(catEight_SC$grade), nrow(catEight_SC), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catEight_SC$grade) and nrow(catEight_SC)
## number of successes = 316, number of trials = 575, p-value = 0.01945
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.5078624 0.5907563
## sample estimates:
## probability of success
##             0.5495652
```

Category 7: distance

```
catNine_EB <- subset(allData, category == 7 & group == 'EB')
binom.test(sum(catNine_EB$grade), nrow(catNine_EB), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catNine_EB$grade) and nrow(catNine_EB)
## number of successes = 450, number of trials = 713, p-value = 2.462e-12
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.5945344 0.6666476
## sample estimates:
## probability of success
##              0.631136
```

```
catNine_SC <- subset(allData, category == 7 & group == 'SC')
binom.test(sum(catNine_SC$grade), nrow(catNine_SC), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(catNine_SC$grade) and nrow(catNine_SC)
## number of successes = 470, number of trials = 713, p-value < 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6230907 0.6939585
## sample estimates:
## probability of success
##             0.6591865
```

Session by Category by Group interaction maximal model

```
nativelModel <- glmer(nativeAccuracy ~ session + category + group + session*category*group +
        (1 + category | subID) + (1 + category | language:stimulus),
      data = allData,
      family = binomial,
      control = glmerControl(optimizer="bobyqa"))
```

```
## Warning in commonArgs(par, fn, control, environment()): maxfun < 10 *
## length(par)^2 is not recommended.

## Warning in commonArgs(par, fn, control, environment()): maxfun < 10 *
## length(par)^2 is not recommended.

## Warning in optwrap(optimizer, devfun, start, rho$lower, control = control, :
## convergence code 1 from bobyqa: bobyqa -- maximum number of function evaluations
## exceeded

## boundary (singular) fit: see help('isSingular')
```

```
summary(nativelModel)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: nativeAccuracy ~ session + category + group + session * category *
##     group + (1 + category | subID) + (1 + category | language:stimulus)
##    Data: allData
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##  11972.6  12573.0  -5902.3  11804.6     9300
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1101 -1.0208  0.5535  0.7385  2.9113
##
## Random effects:
##  Groups            Name        Variance Std.Dev. Corr
##  language:stimulus (Intercept) 0.23089  0.4805
##                    category2   0.05104  0.2259   -0.79
##                    category3   0.19379  0.4402   -0.55  0.53
##                    category4   0.13741  0.3707   -0.55  0.46  0.47
##                    category5   0.13183  0.3631   -0.77  0.55  0.55  0.71
##                    category6   0.08058  0.2839   -0.59  0.83  0.53  0.46  0.50
##                    category7   0.11413  0.3378   -0.93  0.90  0.61  0.57  0.74
##  subID             (Intercept) 0.72979  0.8543
##                    category2   0.26534  0.5151   -0.93
##                    category3   1.54610  1.2434   -0.95  0.94
##                    category4   0.42699  0.6534   -0.94  0.93  0.97
##                    category5   0.48970  0.6998   -0.68  0.76  0.63  0.58
##                    category6   0.37396  0.6115   -0.95  0.95  0.95  0.97  0.63
##                    category7   0.41383  0.6433   -0.77  0.77  0.63  0.76  0.60
##
##
##
```

```
##
##
##
##
##   0.84
##
##
##
##
##
##
##   0.77
## Number of obs: 9384, groups:  language:stimulus, 203; subID, 46
##
## Fixed effects:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      0.677791   0.275705   2.458   0.0140 *
## sessionmatlab                    0.178615   0.383901   0.465   0.6417
## category2                        0.003792   0.227737   0.017   0.9867
## category3                       -0.964444   0.428419  -2.251   0.0244 *
## category4                       -0.018093   0.276398  -0.065   0.9478
## category5                        0.090837   0.265461   0.342   0.7322
## category6                       -0.427643   0.258385  -1.655   0.0979 .
## category7                       -0.284150   0.251490  -1.130   0.2585
## groupSC                          0.196036   0.368564   0.532   0.5948
## sessionmatlab:category2         -0.267248   0.300394  -0.890   0.3737
## sessionmatlab:category3         -0.260512   0.585588  -0.445   0.6564
## sessionmatlab:category4         -0.225142   0.362726  -0.621   0.5348
## sessionmatlab:category5         -0.465439   0.360102  -1.293   0.1962
## sessionmatlab:category6         -0.317402   0.340148  -0.933   0.3508
## sessionmatlab:category7          0.191267   0.345422   0.554   0.5798
## sessionmatlab:groupSC           -0.415560   0.547190  -0.759   0.4476
## category2:groupSC               -0.050134   0.289113  -0.173   0.8623
## category3:groupSC               -0.197718   0.562428  -0.352   0.7252
## category4:groupSC               -0.197934   0.348302  -0.568   0.5698
## category5:groupSC               -0.152981   0.346216  -0.442   0.6586
## category6:groupSC               -0.261218   0.326366  -0.800   0.4235
## category7:groupSC                0.279834   0.331963   0.843   0.3992
## sessionmatlab:category2:groupSC -0.080617   0.429864  -0.188   0.8512
## sessionmatlab:category3:groupSC  0.690357   0.832738   0.829   0.4071
## sessionmatlab:category4:groupSC  0.239757   0.516890   0.464   0.6428
## sessionmatlab:category5:groupSC  0.244286   0.514507   0.475   0.6349
## sessionmatlab:category6:groupSC  0.604757   0.485900   1.245   0.2133
## sessionmatlab:category7:groupSC -0.306706   0.494742  -0.620   0.5353
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 28 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
## optimizer (bobyqa) convergence code: 1 (bobyqa -- maximum number of function evaluations exceeded)
## boundary (singular) fit: see help('isSingular')
## maxfun < 10 * length(par)^2 is not recommended.
```

```r
Anova(nativelModel)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: nativeAccuracy
##                          Chisq Df Pr(>Chisq)
## session                 1.1616  1     0.2811
## category               30.6818  6  2.915e-05 ***
## group                   0.1748  1     0.6759
## session:category        7.9449  6     0.2422
## session:group           0.4445  1     0.5050
## category:group          2.5267  6     0.8655
## session:category:group  6.5372  6     0.3658
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Analyses not considering those words that rquired inversion mapping

```r
allDataNoInversion <- subset(allData, value == 'good')

allDataNoInversion$rawAccuracy <- as.integer(allDataNoInversion$response == allDataNoInversion$meaning)

allDataNoInversion$nativeAccuracy <- ifelse(allDataNoInversion$value == 'goodop',
                                abs(allDataNoInversion$rawAccuracy - 1),
                                allDataNoInversion$rawAccuracy + 0)
allDataNoInversion %>%
  group_by(group) %>%
  summarize(native_acc = mean(nativeAccuracy),
            .groups = 'keep')
```

```
## # A tibble: 2 x 2
## # Groups:   group [2]
##   group native_acc
##   <fct>      <dbl>
## 1 EB         0.647
## 2 SC         0.643
```

Groups accuracy against chance

```r
native_EB <- subset(allDataNoInversion, group == 'EB')
binom.test(sum(native_EB$grade), nrow(native_EB), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(native_EB$grade) and nrow(native_EB)
## number of successes = 1845, number of trials = 2852, p-value < 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6290551 0.6644718
## sample estimates:
## probability of success
##              0.6469144
```

```r
native_SC <- subset(allDataNoInversion, group == 'SC')
binom.test(sum(native_SC$grade), nrow(native_SC), p = .5)
```

```
##
##  Exact binomial test
##
## data:  sum(native_SC$grade) and nrow(native_SC)
## number of successes = 1833, number of trials = 2852, p-value < 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.6248052 0.6603153
## sample estimates:
## probability of success
##              0.6427069
```

Session by Category by Group interaction maximal model on the subset of words that did not require inversion mapping

```r
nativelNoInversionModel <- glmer(nativeAccuracy ~ session + category + group + session*category*group +
        (1 + category | subID) + (1 + category | language:stimulus),
      data = allDataNoInversion,
      family = binomial,
      control = glmerControl(optimizer="bobyqa"))
```

```
## Warning in commonArgs(par, fn, control, environment()): maxfun < 10 *
## length(par)^2 is not recommended.

## Warning in commonArgs(par, fn, control, environment()): maxfun < 10 *
## length(par)^2 is not recommended.

## boundary (singular) fit: see help('isSingular')
```

```r
summary(nativelNoInversionModel)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: nativeAccuracy ~ session + category + group + session * category *
##     group + (1 + category | subID) + (1 + category | language:stimulus)
##    Data: allDataNoInversion
## Control: glmerControl(optimizer = "bobyqa")
##
##      AIC      BIC   logLik deviance df.resid
##   7120.9   7679.4  -3476.5   6952.9     5620
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5049 -1.0041  0.5249  0.6934  3.4004
##
## Random effects:
##  Groups            Name        Variance Std.Dev. Corr
##  language:stimulus (Intercept) 0.2337   0.4835
##                    category2   0.1335   0.3654   -0.44
##                    category3   0.1871   0.4325   -0.97  0.48
##                    category4   0.1395   0.3735   -0.90  0.46  0.89
```

```
##                     category5   0.1386   0.3723   -0.80  0.39  0.80  0.79
##                     category6   0.1262   0.3552   -0.82  0.58  0.76  0.67  0.67
##                     category7   0.1252   0.3538   -0.93  0.50  0.91  0.87  0.66
##  subID              (Intercept) 0.8030   0.8961
##                     category2   0.2392   0.4891   -0.86
##                     category3   1.6443   1.2823   -0.94  0.95
##                     category4   0.3264   0.5714   -0.82  0.93  0.95
##                     category5   0.4154   0.6445   -0.57  0.73  0.55  0.44
##                     category6   0.4271   0.6536   -0.96  0.86  0.93  0.91  0.45
##                     category7   0.3094   0.5563   -0.79  0.66  0.63  0.63  0.56
##
##
##
##
##
##
##
##
##   0.84
##
##
##
##
##
##
##   0.83
## Number of obs: 5704, groups:  language:stimulus, 124; subID, 46
##
## Fixed effects:
##                                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      0.76620    0.29358   2.610  0.00906 **
## sessionmatlab                    0.20734    0.40862   0.507  0.61187
## category2                        0.03393    0.26006   0.130  0.89620
## category3                       -1.20254    0.48315  -2.489  0.01281 *
## category4                       -0.13191    0.28207  -0.468  0.64004
## category5                       -0.05289    0.28869  -0.183  0.85463
## category6                       -0.16690    0.29966  -0.557  0.57755
## category7                       -0.22060    0.25780  -0.856  0.39216
## groupSC                          0.25193    0.39141   0.644  0.51981
## sessionmatlab:category2         -0.21100    0.32867  -0.642  0.52089
## sessionmatlab:category3          0.11497    0.68135   0.169  0.86600
## sessionmatlab:category4         -0.17962    0.38213  -0.470  0.63832
## sessionmatlab:category5         -0.36992    0.38670  -0.957  0.33876
## sessionmatlab:category6         -0.70512    0.39937  -1.766  0.07747 .
## sessionmatlab:category7          0.14585    0.35121   0.415  0.67793
## sessionmatlab:groupSC           -0.56770    0.58269  -0.974  0.32992
## category2:groupSC               -0.08340    0.31586  -0.264  0.79174
## category3:groupSC               -0.11296    0.65427  -0.173  0.86293
## category4:groupSC                0.03174    0.36891   0.086  0.93143
## category5:groupSC               -0.02329    0.37340  -0.062  0.95026
## category6:groupSC               -0.61736    0.38291  -1.612  0.10690
## category7:groupSC                0.08042    0.33588   0.239  0.81078
## sessionmatlab:category2:groupSC -0.23643    0.47122  -0.502  0.61584
## sessionmatlab:category3:groupSC  0.09899    0.97003   0.102  0.91872
## sessionmatlab:category4:groupSC  0.28674    0.55077   0.521  0.60264
```

```
## sessionmatlab:category5:groupSC  0.40731    0.55691   0.731  0.46455
## sessionmatlab:category6:groupSC  1.15699    0.56941   2.032  0.04216 *
## sessionmatlab:category7:groupSC -0.24035    0.50246  -0.478  0.63241
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation matrix not shown by default, as p = 28 > 12.
## Use print(x, correlation=TRUE)  or
##     vcov(x)        if you need it
```

```
## optimizer (bobyqa) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
## maxfun < 10 * length(par)^2 is not recommended.
```

```
Anova(nativelNoInversionModel)
```

```
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: nativeAccuracy
##                           Chisq Df Pr(>Chisq)
## session                  1.2706  1    0.25965
## category                29.5663  6  4.752e-05 ***
## group                    0.1847  1    0.66736
## session:category         4.7542  6    0.57570
## session:group            0.8674  1    0.35167
## category:group           3.7135  6    0.71538
## session:category:group 12.6542  6    0.04887 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```