

Silent and spoken intervals in naming: Basic patterns

Athanasios Protopapas

7 February 2023

Contents

1	Introduction	1
2	Read in data	2
3	Analysis of discrete naming	2
3.1	Spoken intervals (articulation duration)	2
3.1.1	Graph of mean articulation duration per task and grade	2
3.1.2	Statistical comparisons of mean articulation duration by task	3
3.2	Silent intervals (onset latency)	5
3.2.1	Graph of mean onset latency per task and grade	5
3.2.2	Statistical comparisons of mean onset latency by task	6
4	Analysis of serial naming	7
4.1	Spoken intervals (articulation times)	8
4.1.1	Graph of mean articulation time per task and grade	8
4.1.2	Statistical comparisons of mean articulation time by task	9
4.2	Silent intervals (pause times)	10
4.2.1	Graph of mean pause time per task and grade	10
4.2.2	Statistical comparisons of mean pause time by task	12
5	Relationship between silent and spoken intervals in discrete tasks	13
6	Correlations between discrete and serial	14
7	Associations of naming response components with reading measures	15
7.1	Tables of correlations	15
7.2	Graphical display of correlations	16
8	Session information	19

```
library(psych)
library(mice)
library(lattice)
require(nlme)
require(multcomp)
```

1 Introduction

In this document we present a set of basic analyses of silent and spoken intervals from serial and discrete naming tasks, including comparisons of mean performance across grades and tasks as well as correlations among response components.

Data quality considerations and descriptive statistics are presented in separate documents: “data check” and “descriptives”.

2 Read in data

Data from reading tests and discrete and serial naming for 298 children in Grades 1, 3, and 5 have been combined in a single wide-format data frame.

```
read.table("read_discrete_serial_data.txt",stringsAsFactors=TRUE) -> rsdw
tasks <- c("digit","numword","word","dice","object")
tcols <- c("#D55E00","#E69F00","#009E73","#0180C7","#884584")
```

3 Analysis of discrete naming

A main premise of the present work is that we can use discrete naming as a baseline over which to explore the effects of the serial task format. In the discrete naming situation we expect to be able to directly apply knowledge from spoken word production in picture naming and single word reading studies.

The materials used in the five tasks have been carefully matched for phonological and orthographic length, phonological and orthographic complexity, and printed word frequency. We therefore expect them to be equally demanding in terms of producing the actual word. However, there may be differences arising from (a) whether access to the phonological form is mediated or unmediated, and (b) whether production can benefit from within-task repetition of the same items.

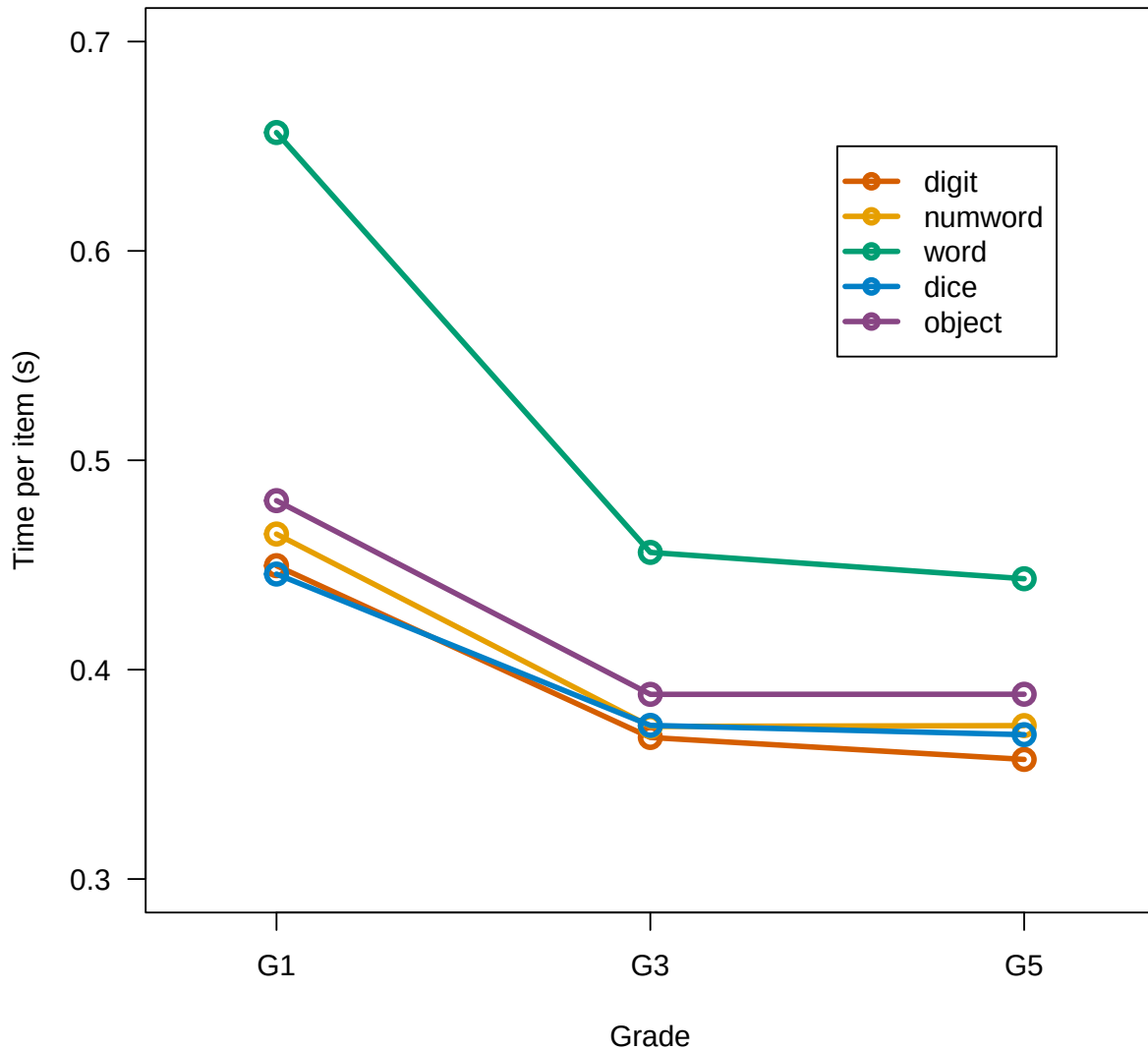
Note: In this and the following analyses we use log-transformed data as there were determined to produce acceptable distributions across variables, tasks, and grades. Rates (i.e., inverse-transformed data) lead to most normal-like distributions for total times, and have been the metric of choice in most past studies, but they lead to severely skewed distributions for serial pause times (i.e., silent intervals), perhaps because they shrink substantially toward zero with increasing skill. Thus, in keeping all comparable metrics on a common scale, we use log-transformed times per item for all metrics below.

3.1 Spoken intervals (articulation duration)

3.1.1 Graph of mean articulation duration per task and grade

The following graph displays the average times per item for each task and grade. In this summary graph we have back-transformed the data to times, to facilitate interpretability, after computing means on the log-transformed individual data. Thus we are displaying the geometric means of individual articulation times. (The distributions of individual data are presented in separate documents.)

```
discart <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logartic",tasks,sep="."),collapse=","),
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.3,0.7),xaxt="n",xlab="Grade",ylab="Time per item (s)",las=1,
axis(1,at=1:3,labels=paste("G",c(1,3,5),sep=""),cex=1.5)
for (itsk in 1:length(tasks)) {
  tsk <- paste("logartic",tasks[itsk],sep=".")
  lines(1:3,discart[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,discart[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}
legend(2.5,0.65,tasks,col=tcols,pch=21,lty=1,lwd=3)
```



Articulation times seem to differ somewhat among tasks despite the psycholinguistic matching. In the next session we test which of these points differ significantly within each grade.

3.1.2 Statistical comparisons of mean articulation duration by task

The following set of post-hoc contrasts considers within-grade successive points, that is, for each grade, task pairs that correspond to neighboring points in the graph above, traversed vertically from fastest, at the bottom, to slowest, toward the top. In addition, we test a few additional comparisons (of nonconsecutive points) of particular interest.

```
artnm <- paste("logartic",tasks,sep=".")
artlong <- reshape(rsdw[,c("sID","grade",artnm)],direction="long",
                   varying=artnm,timevar="task",times=tasks,idvar="sID",v.names="logartic")
artlong$task <- relevel(as.factor(artlong$task),ref="word")
artlong$grade <- relevel(artlong$grade,ref="G3")
```

```

artlong <- droplevels(artlong[!is.na(artlong$logartic),])
l0 <- lme(logartic~task*grade,random=~1|sID/task,data=artlong)
contrast.matrix.tasks <- rbind(
  #
  "G1 dice-digit" = c( 0, +1, -1, 0, 0, 0, 0, 0, +1, -1, 0, 0, 0, 0, 0, 0 ),
  "G1 digit-numword" = c( 0, 0, +1, -1, 0, 0, 0, 0, 0, +1, -1, 0, 0, 0, 0, 0 ),
  "G1 numword-object" = c( 0, 0, 0, +1, -1, 0, 0, 0, 0, 0, +1, -1, 0, 0, 0, 0 ),
  "G1 digit-object" = c( 0, 0, +1, 0, -1, 0, 0, 0, 0, +1, 0, -1, 0, 0, 0, 0 ),
  "G1 object-word" = c( 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0 ),

  "G3 digit-numword" = c( 0, 0, +1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 numword-dice" = c( 0, -1, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 dice-object" = c( 0, +1, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 object-word" = c( 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),

  "G5 digit-dice" = c( 0, -1, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, +1, 0, 0 ),
  "G5 dice-numword" = c( 0, +1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0, -1, 0 ),
  "G5 digit-object" = c( 0, 0, +1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0, -1 ),
  "G5 numword-object" = c( 0, 0, 0, +1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, -1 ),
  "G5 object-word" = c( 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1 )
)
summary(glht(l0,contrast.matrix.tasks))

```

```

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lme.formula(fixed = logartic ~ task * grade, data = artlong,
## random = ~1 | sID/task)
##
## Linear Hypotheses:
## Estimate Std. Error z value Pr(>|z|)
## G1 dice-digit == 0 -0.007374 0.015085 -0.489 1.0000
## G1 digit-numword == 0 -0.037929 0.015136 -2.506 0.1425
## G1 numword-object == 0 -0.030056 0.015136 -1.986 0.4359
## G1 digit-object == 0 -0.067985 0.015085 -4.507 <0.001 ***
## G1 object-word == 0 -0.312421 0.015233 -20.509 <0.001 ***
## G3 digit-numword == 0 -0.014305 0.014953 -0.957 0.9896
## G3 numword-dice == 0 -0.003035 0.014907 -0.204 1.0000
## G3 dice-object == 0 -0.043109 0.014862 -2.901 0.0474 *
## G3 object-word == 0 -0.154864 0.014862 -10.420 <0.001 ***
## G5 digit-dice == 0 -0.032673 0.015289 -2.137 0.3296
## G5 dice-numword == 0 -0.011503 0.015289 -0.752 0.9987
## G5 digit-object == 0 -0.084185 0.015240 -5.524 <0.001 ***
## G5 numword-object == 0 -0.040009 0.015240 -2.625 0.1046
## G5 object-word == 0 -0.131693 0.015240 -8.641 <0.001 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

```

The results confirm that words are articulated more slowly than objects in every grade; and objects are articulated more slowly than the other three types of items (in Grades 1 and 3; only than digits in Grade 5). Thus, neither psycholinguistic matching nor repetition is sufficient to fully control articulation times among items in naming tasks.

In contrast, digits, dice, and number words do not differ significantly in articulation duration. This was

expected because responses to these three tasks involve the exact same words uttered equally repetitively. Thus these three tasks are demonstrably mathed in articulation duration across grades.

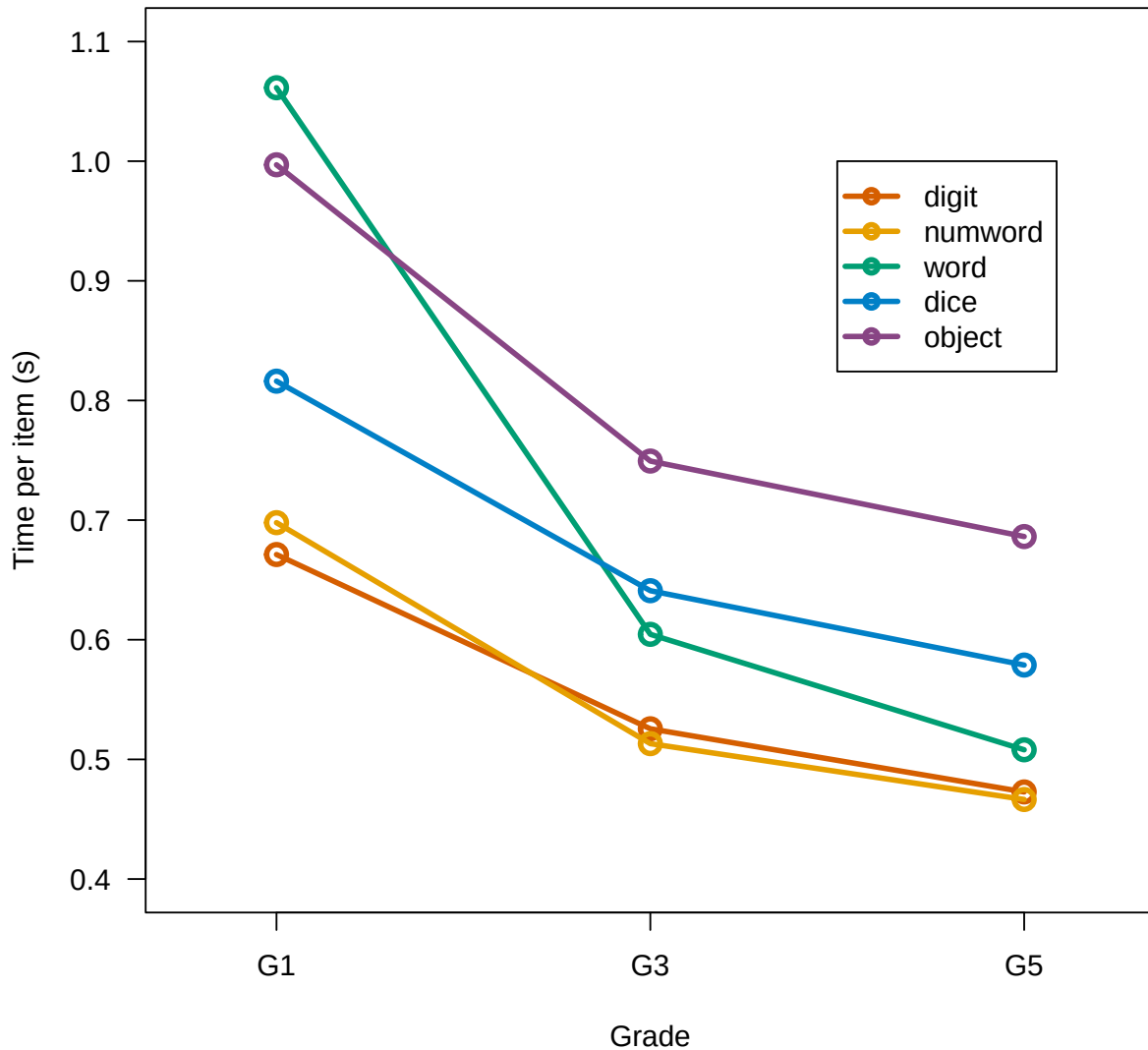
3.2 Silent intervals (onset latency)

We now turn to onset latency and apply the same approach.

3.2.1 Graph of mean onset latency per task and grade

We first plot the mean times per item for each task and grade (geometric means of individual mean onset latencies).

```
discons <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logonsRT",tasks,sep="."),collapse=","),
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.4,1.1),xaxt="n",xlab="Grade",ylab="Time per item (s)",la
axis(1,at=1:3,labels=paste("G",c(1,3,5),sep=""),cex=1.5)
for (itsk in 1:length(tasks)) {
  tsk <- paste("logonsRT",tasks[itsk],sep=".")
  lines(1:3,discons[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,discons[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}
legend(2.5,1,tasks,col=tcols,pch=21,lty=1,lwd=3)
```



As expected, onset latencies differ among tasks despite the psycholinguistic matching. The following set of post-hoc contrasts tests which successive points differ significantly.

3.2.2 Statistical comparisons of mean onset latency by task

As was done for articulation durations above, the following set of post-hoc contrasts of onset latencies considers within-grade successive points, that is, for each grade, task pairs that correspond to neighboring points in the graph above, traversed vertically from fastest, at the bottom, to slowest, toward the top.

```
onsnm <- paste("logonsRT",tasks,sep=".")
onslong <- reshape(rsdw[,c("sID","grade",onsnm)],direction="long",
                   varying=onsnm,timevar="task",times=tasks,idvar="sID",v.names="logonsRT")
onslong$task <- relevel(as.factor(onslong$task),ref="word")
onslong$grade <- relevel(onslong$grade,ref="G3")
onslong <- droplevels(onslong[!is.na(onslong$logonsRT),])
```

```

l1 <- lme(logonsRT~task*grade,random=~1|sID/task,data=onslong)
contrast.matrix.tasks <- rbind(
  #
  "G1 digit-numword" = c( 0, 0, +1, -1, 0, 0, 0, 0, 0, +1, -1, 0, 0, 0, 0, 0 ),
  "G1 numword-dice" = c( 0, -1, 0, +1, 0, 0, 0, 0, -1, 0, +1, 0, 0, 0, 0, 0 ),
  "G1 dice-object" = c( 0, +1, 0, 0, -1, 0, 0, 0, +1, 0, 0, -1, 0, 0, 0, 0 ),
  "G1 object-word" = c( 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0 ),

  "G3 numword-digit" = c( 0, 0, -1, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 digit-word" = c( 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 word-dice" = c( 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 dice-object" = c( 0, +1, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),

  "G5 numword-digit" = c( 0, 0, -1, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, +1, 0 ),
  "G5 digit-word" = c( 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0 ),
  "G5 word-dice" = c( 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0 ),
  "G5 dice-object" = c( 0, +1, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, +1, 0, -1 )
)
summary(glht(l1,contrast.matrix.tasks))

```

```

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lme.formula(fixed = logonsRT ~ task * grade, data = onslong,
## random = ~1 | sID/task)
##
## Linear Hypotheses:
##
## Estimate Std. Error z value Pr(>|z|)
## G1 digit-numword == 0 -0.05106 0.01667 -3.064 0.0250 *
## G1 numword-dice == 0 -0.15265 0.01667 -9.158 <0.001 ***
## G1 dice-object == 0 -0.19229 0.01661 -11.574 <0.001 ***
## G1 object-word == 0 -0.06626 0.01677 -3.950 <0.001 ***
## G3 numword-digit == 0 -0.02259 0.01647 -1.372 0.8546
## G3 digit-word == 0 -0.14102 0.01642 -8.590 <0.001 ***
## G3 word-dice == 0 -0.05839 0.01637 -3.567 0.0042 **
## G3 dice-object == 0 -0.15515 0.01637 -9.479 <0.001 ***
## G5 numword-digit == 0 -0.01298 0.01678 -0.773 0.9975
## G5 digit-word == 0 -0.07202 0.01678 -4.291 <0.001 ***
## G5 word-dice == 0 -0.13076 0.01684 -7.766 <0.001 ***
## G5 dice-object == 0 -0.16983 0.01684 -10.086 <0.001 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

```

The results are qualitatively different for Grade 1, where all points differ from each other, digits are faster than number words and objects, and words are slowest. In Grades 3 and 5, digits and number words are not significantly different from each other and are faster than words, which are faster than dice, which are faster than objects.

4 Analysis of serial naming

Having established a baseline in discrete naming, we now turn to the corresponding analysis of serial naming response components, namely spoken intervals (articulation time) and silent intervals (pause time). For direct comparability with discrete tasks, we calculate corresponding time variables using the adjusted times

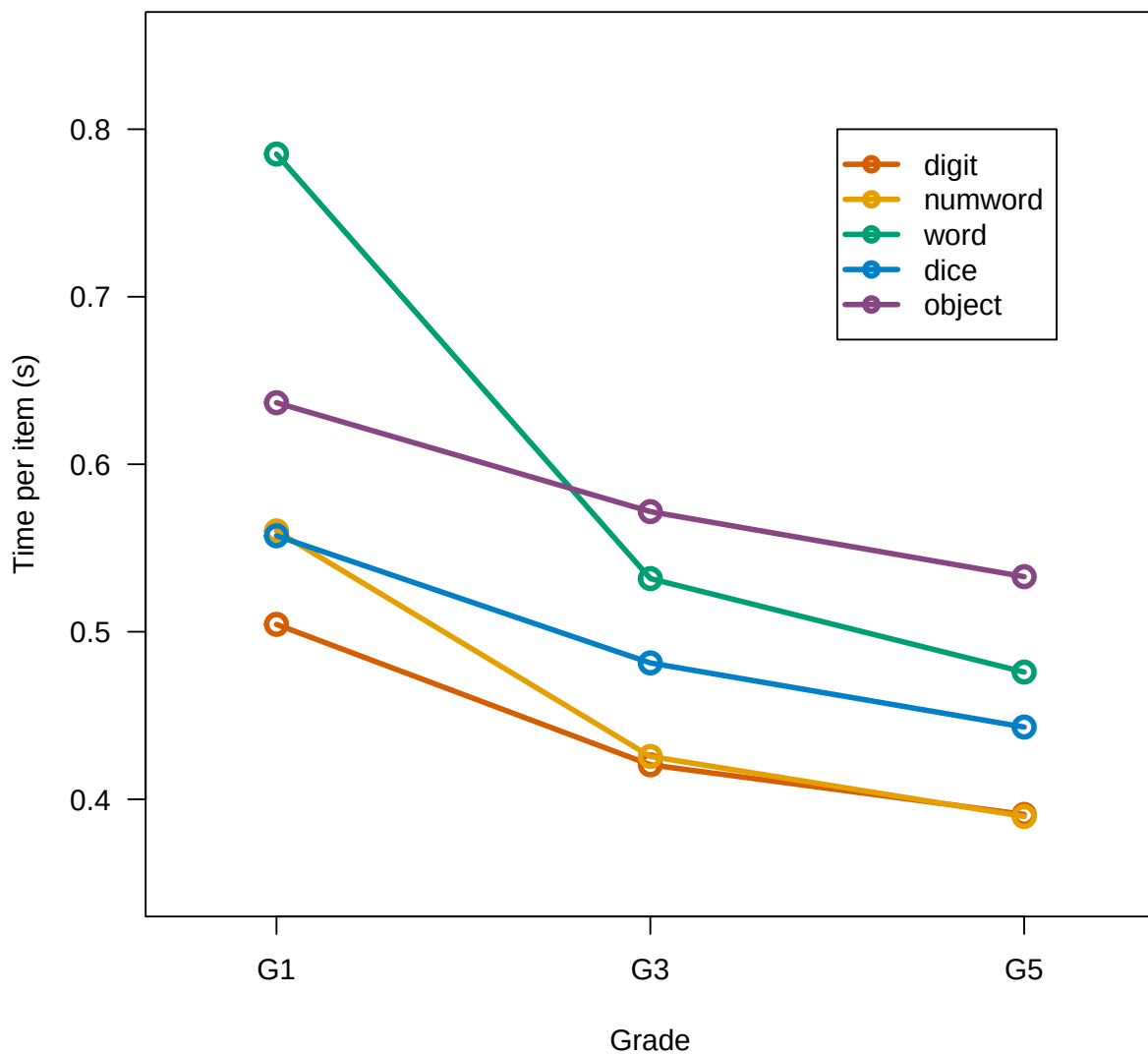
and error counts, effectively taking only correct responses into account, as for the discrete tasks.

4.1 Spoken intervals (articulation times)

4.1.1 Graph of mean articulation time per task and grade

Here are the back-transformed mean times per item for each task and grade (i.e., geometric means of individual articulation times).

```
serart <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logarttE",tasks,sep="."),collapse=","),
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.35,0.85),xaxt="n",xlab="Grade",ylab="Time per item (s)",
axis(1,at=1:3,labels=paste("G",c(1,3,5),sep=""),cex=1.5)
for (itsk in 1:length(tasks)) {
  tsk <- paste("logarttE",tasks[itsk],sep=".")
  lines(1:3,serart[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,serart[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}
legend(2.5,0.8,tasks,col=tcols,pch=21,lty=1,lwd=3)
```



It seems that serial articulation times pattern more like discrete onset latencies than like discrete articulation times.

The following set of post-hoc contrasts tests which of these points differ significantly for each grade.

4.1.2 Statistical comparisons of mean articulation time by task

```
serartnm <- paste("logarttE", tasks, sep=".")
serartlong <- reshape(rsdw[,c("sID", "grade", serartnm)], direction="long",
                      varying=serartnm, timevar="task", times=tasks, idvar="sID", v.names="logarttE")
serartlong$task <- relevel(as.factor(serartlong$task), ref="word")
serartlong$grade <- relevel(serartlong$grade, ref="G3")
serartlong <- droplevels(serartlong[!is.na(serartlong$logarttE),])
l2 <- lme(logarttE~task*grade, random=~1|sID/task, data=serartlong)
```

```

contrast.matrix.tasks <- rbind(
#           I    3c    3d    3n    3o    G1    G5    1c    1d    1n    1o    5c    5d    5n    5o
"G1 digit-numword" = c( 0,  0, +1, -1,  0,  0,  0,  0, +1, -1,  0,  0,  0,  0,  0 ),
"G1 numword-dice"  = c( 0, -1,  0, +1,  0,  0,  0, -1,  0, +1,  0,  0,  0,  0,  0 ),
"G1 dice-object"   = c( 0, +1,  0,  0, -1,  0,  0, +1,  0,  0, -1,  0,  0,  0,  0 ),
"G1 object-word"   = c( 0,  0,  0,  0, +1,  0,  0,  0,  0,  0, +1,  0,  0,  0,  0 ),

"G3 digit-numword" = c( 0,  0, +1, -1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0 ),
"G3 numword-dice"  = c( 0, -1,  0, +1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0 ),
"G3 dice-word"     = c( 0, +1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0 ),
"G3 word-object"   = c( 0,  0,  0,  0, -1,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0 ),

"G5 digit-numword" = c( 0,  0, +1, -1,  0,  0,  0,  0,  0,  0,  0,  0, +1, -1,  0 ),
"G5 numword-dice"  = c( 0, -1,  0, +1,  0,  0,  0,  0,  0,  0,  0, -1,  0, +1,  0 ),
"G5 dice-word"     = c( 0, +1,  0,  0,  0,  0,  0,  0,  0,  0,  0, +1,  0,  0,  0 ),
"G5 word-object"   = c( 0,  0,  0,  0, -1,  0,  0,  0,  0,  0,  0,  0,  0,  0, -1 )
)
summary(glht(l2,contrast.matrix.tasks))

```

```

##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: lme.formula(fixed = logarttE ~ task * grade, data = serartlong,
##       random = ~1 | sID/task)
##
## Linear Hypotheses:
##               Estimate Std. Error z value Pr(>|z|)
## G1 digit-numword == 0 -0.1017177  0.0138909  -7.323  <1e-04 ***
## G1 numword-dice == 0  0.0035150  0.0138272   0.254    1.000
## G1 dice-object == 0  -0.1347041  0.0138468  -9.728  <1e-04 ***
## G1 object-word == 0  -0.2090768  0.0137588 -15.196  <1e-04 ***
## G3 digit-numword == 0 -0.0097313  0.0135517  -0.718    0.999
## G3 numword-dice == 0 -0.1247522  0.0136350  -9.149  <1e-04 ***
## G3 dice-word == 0   -0.1040112  0.0136757  -7.606  <1e-04 ***
## G3 word-object == 0  -0.0711783  0.0136337  -5.221  <1e-04 ***
## G5 digit-numword == 0  0.0008077  0.0138550   0.058    1.000
## G5 numword-dice == 0 -0.1278412  0.0138550  -9.227  <1e-04 ***
## G5 dice-word == 0   -0.0706516  0.0139433  -5.067  <1e-04 ***
## G5 word-object == 0  -0.1144809  0.0141709  -8.079  <1e-04 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

```

The results confirm that the pattern for serial articulation duration is nothing like the pattern for discrete articulation duration. All points are significantly different except for three nearly identical pairs shown on the graph, namely dice and number words in Grade 1 and digits and number words in Grades 3 and 5.

4.2 Silent intervals (pause times)

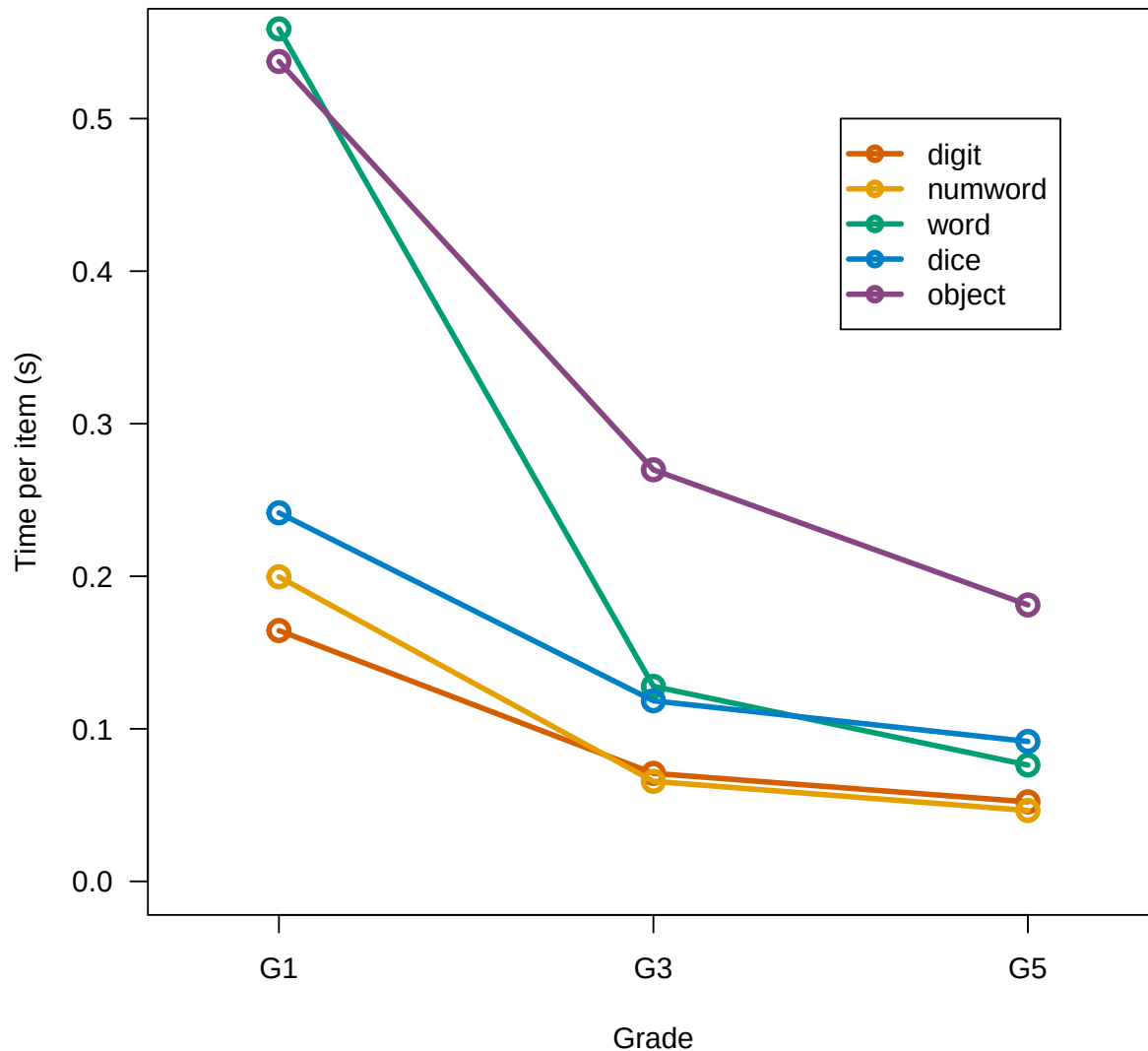
4.2.1 Graph of mean pause time per task and grade

We now turn to the (adjusted) silent intervals in the serial responses and apply the same approach. We first plot the (geometric) average times per task and grade.

```

serpau <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logpautE",tasks,sep="."),collapse=","),
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0,0.55),xaxt="n",xlab="Grade",ylab="Time per item (s)",las=
axis(1,at=1:3,labels=paste("G",c(1,3,5),sep=""),cex=1.5)
for (itsk in 1:length(tasks)) {
  tsk <- paste("logpautE",tasks[itsk],sep=".")
  lines(1:3,serpau[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,serpau[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}
legend(2.5,0.5,tasks,col=tcols,pch=21,lty=1,lwd=3)

```



Interestingly, serial pause times pattern very similarly to the serial articulation times. The following set of post-hoc contrasts tests which successive points differ significantly.

4.2.2 Statistical comparisons of mean pause time by task

```

paunm <- paste("logpautE",tasks,sep=".")
paulong <- reshape(rsdw[,c("sID", "grade", paunm)],direction="long",
                  varying=paunm,timevar="task",times=tasks,idvar="sID",v.names="logpautE")
paulong$task <- relevel(as.factor(paulong$task),ref="word")
paulong$grade <- relevel(paulong$grade,ref="G3")
paulong <- droplevels(paulong[!is.na(paulong$logpautE),])
l3 <- lme(logpautE~task*grade,random=~1|sID/task,data=paulong)
contrast.matrix.tasks <- rbind(
  #
  "G1 digit-numword" = c( 0, 0, +1, -1, 0, 0, 0, 0, 0, +1, -1, 0, 0, 0, 0, 0 ),
  "G1 numword-dice" = c( 0, -1, 0, +1, 0, 0, 0, 0, -1, 0, +1, 0, 0, 0, 0, 0 ),
  "G1 dice-word" = c( 0, +1, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0 ),
  "G1 word-object" = c( 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0, 0 ),

  "G3 numword-digit" = c( 0, 0, -1, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 digit-word" = c( 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 word-dice" = c( 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 dice-object" = c( 0, +1, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),

  "G5 numword-digit" = c( 0, 0, -1, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, +1, 0 ),
  "G5 digit-word" = c( 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0, 0 ),
  "G5 word-dice" = c( 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, 0, 0 ),
  "G5 dice-object" = c( 0, +1, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, +1, 0, 0, -1 )
)
summary(glht(l3,contrast.matrix.tasks))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lme.formula(fixed = logpautE ~ task * grade, data = paulong,
## random = ~1 | sID/task)
##
## Linear Hypotheses:
##
## Estimate Std. Error z value Pr(>|z|)
## G1 digit-numword == 0 -0.17849 0.06066 -2.942 0.0367 *
## G1 numword-dice == 0 -0.19824 0.06039 -3.283 0.0118 *
## G1 dice-word == 0 -0.85343 0.06027 -14.159 <0.001 ***
## G1 word-object == 0 0.05719 0.06009 0.952 0.9865
## G3 numword-digit == 0 -0.07909 0.05918 -1.336 0.8739
## G3 digit-word == 0 -0.57546 0.05936 -9.695 <0.001 ***
## G3 word-dice == 0 0.06296 0.05972 1.054 0.9714
## G3 dice-object == 0 -0.82142 0.05990 -13.713 <0.001 ***
## G5 numword-digit == 0 -0.11901 0.06051 -1.967 0.4176
## G5 digit-word == 0 -0.38099 0.06089 -6.257 <0.001 ***
## G5 word-dice == 0 -0.19034 0.06089 -3.126 0.0203 *
## G5 dice-object == 0 -0.66812 0.06149 -10.865 <0.001 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

```

The results confirm that objects are consistently slowest (longest silent intervals). Number words do not differ from digits. Words differ from both slower and faster tasks in Grades 3 and 5.

5 Relationship between silent and spoken intervals in discrete tasks

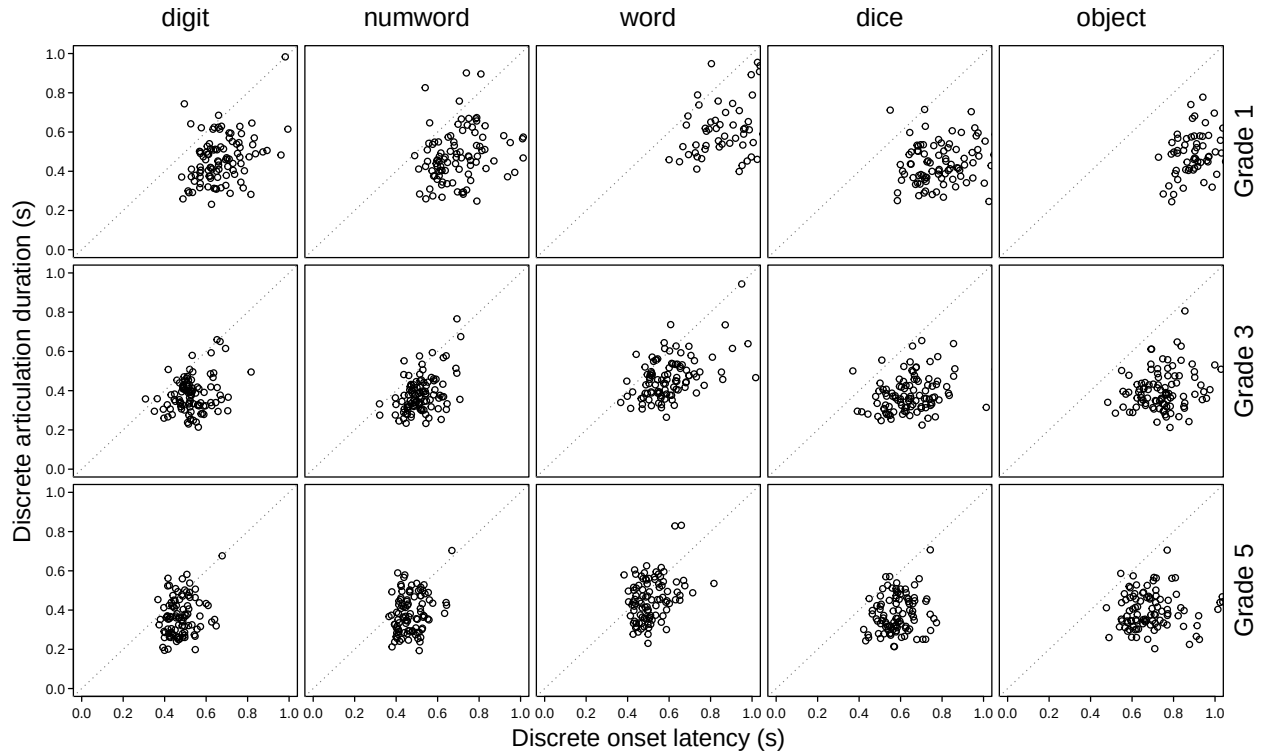
One way to visually examine how silent and spoken intervals compare to each other across children for each grade and task is to consider the location of points in a scatterplot against the diagonal.

```
par(mfrow=c(3,5))
par(oma=c(3.5,3.5,2,2))
par(mai=c(0.03,0.03,0.03,0.03))
par(mgp=c(2,0.5,0))
par(tcl= -0.3)

v1s <- "articsec"
v2s <- "onsRTsec"

for (grd in levels(rsdw$grade)) {
  for (tsk in tasks) {
    rsdw[,paste(v1s,tsk,sep=".")] <- rsdw[,paste("artic",tsk,sep=".")] / 1000
    rsdw[,paste(v2s,tsk,sep=".")] <- exp(rsdw[,paste("logonsRT",tsk,sep=".")]) / 1000
    plot(as.formula(paste(paste(v1s,tsk,sep="."),paste(v2s,tsk,sep="."),sep="~")),
         rsdw,subset= grade==grd,
         xlim=c(0,1),ylim=c(0,1),xaxt="n",yaxt="n")
    if (grd=="G5") { axis(1) }
    if (tsk==tasks[1]) { axis(2,las=1) }
    abline(0,1,lty=3,col="grey50")
    if (grd=="G1") { mtext(tsk,3,line=1,cex=1.25)}
    if (tsk==tasks[length(tasks)]) { mtext(paste("Grade",substr(grd,2,2)),4,line=1,cex=1.25) }
  }
}

mtext("Discrete onset latency (s)", 1, outer=T, cex=1.2, line=2)
mtext("Discrete articulation duration (s)", 2, outer=T, cex=1.2, line=2.2)
```



The vast majority of points lie below the diagonal, consistent with a longer time needed for pre-articulation processing of the stimuli than for articulation. However, nonnegligible proportions of points lie around or above the diagonal for digits, number words, and words in Grade 5. These represent children who may have reached high levels of processing efficiency for these – most easily automatized – stimuli that they can process them faster than they can articulate them.

6 Correlations between discrete and serial

Individual differences across formats pattern consistent with mean performance (Spearman's ρ):

```
cor_serdis <- round(rbind(
  diag(cor( rsdw[,paste(c("logonsRT"),tasks,sep=".")], rsdw[,paste(c("logpautE"),tasks,sep=".")], use="pa"),
  diag(cor( rsdw[,paste(c("logonsRT"),tasks,sep=".")], rsdw[,paste(c("logarttE"),tasks,sep=".")], use="pa"),
  diag(cor( rsdw[,paste(c("logartic"),tasks,sep=".")], rsdw[,paste(c("logpautE"),tasks,sep=".")], use="pa"),
  diag(cor( rsdw[,paste(c("logartic"),tasks,sep=".")], rsdw[,paste(c("logarttE"),tasks,sep=".")], use="pa"),
),2)
colnames(cor_serdis) <- tasks
rownames(cor_serdis) <- c("Discrete onset - Serial pause", "Discrete onset - Serial articulation",
  "Discrete articulation - Serial pause", "Discrete articulation - Serial articulation")
cor_serdis
```

	digit	numword	word	dice	object
Discrete onset - Serial pause	0.64	0.70	0.85	0.72	0.72
Discrete onset - Serial articulation	0.61	0.69	0.80	0.58	0.45
Discrete articulation - Serial pause	0.23	0.33	0.59	0.20	0.25
Discrete articulation - Serial articulation	0.50	0.53	0.76	0.49	0.54

Specifically, silent intervals across formats (discrete and serial) are most strongly correlated, for all tasks (coefficients between 0.64 and 0.85). They are followed by the correlations of discrete onset latency with serial articulation time (0.45–0.80). As expected from the patterns of mean performance, serial articulation

time is more closely related to discrete onset latency than to discrete articulation duration.

7 Associations of naming response components with reading measures

7.1 Tables of correlations

To connect with previous research on “RAN components”, here are the nonparametric correlations (Spearman’s ρ) between reading measures and silent and spoken intervals in discrete and serial naming tasks.

```
rmeas <- c("logwacc", "sqpwacc", "wflut", "pwflu")
nmeas <- c("logonsRT", "logartic", "logpautE", "logarttE")
by(rsdw, rsdw$grade, function(x){-round(cor(x[,c(rmeas,
                                                apply(expand.grid(nmeas, tasks), 1, function(x){paste(x[1], x[2]),
                                                method="spearman", use="pairwise") [1:4, -c(1:4)]
, 2])) -> rncor
for (grd in levels(rsdw$grade)) {
  cat(paste("\n***", grd, "\n"))
  print(t(rncor[[grd]]))
}
```

```
##
## *** G1
##          logwacc sqpwacc wflut pwflu
## logonsRT.digit      0.27   -0.10  0.35  0.47
## logartic.digit      0.10   -0.07  0.15  0.18
## logpautE.digit      0.21   -0.02  0.35  0.46
## logarttE.digit      0.02    0.07  0.36  0.47
## logonsRT.numword    0.46   -0.02  0.58  0.54
## logartic.numword    0.18    0.04  0.25  0.30
## logpautE.numword    0.32   -0.01  0.64  0.46
## logarttE.numword    0.24    0.05  0.57  0.46
## logonsRT.word       0.33    0.01  0.74  0.68
## logartic.word       0.20   -0.03  0.51  0.44
## logpautE.word       0.32   -0.06  0.78  0.70
## logarttE.word       0.12    0.02  0.62  0.53
## logonsRT.dice       0.28    0.04  0.44  0.46
## logartic.dice       0.12   -0.02  0.21  0.26
## logpautE.dice       0.27   -0.02  0.35  0.44
## logarttE.dice      -0.07    0.08  0.26  0.29
## logonsRT.object     0.23    0.18  0.27  0.27
## logartic.object     0.10    0.03  0.24  0.26
## logpautE.object     0.16    0.15  0.32  0.21
## logarttE.object    -0.12    0.14  0.14  0.15
##
## *** G3
##          logwacc sqpwacc wflut pwflu
## logonsRT.digit      0.26    0.43  0.45  0.47
## logartic.digit     -0.02    0.07  0.26  0.12
## logpautE.digit      0.19    0.23  0.45  0.47
## logarttE.digit      0.14    0.04  0.52  0.49
## logonsRT.numword    0.28    0.32  0.50  0.49
## logartic.numword    0.00    0.05  0.30  0.18
```

```
## logpautE.numword    0.32    0.33  0.59  0.56
## logarttE.numword    0.25    0.22  0.65  0.60
## logonsRT.word       0.38    0.44  0.74  0.67
## logartic.word       0.05    0.03  0.39  0.26
## logpautE.word       0.44    0.37  0.77  0.66
## logarttE.word       0.27    0.21  0.72  0.59
## logonsRT.dice       0.28    0.34  0.46  0.51
## logartic.dice      -0.05    0.00  0.23  0.13
## logpautE.dice       0.22    0.26  0.47  0.48
## logarttE.dice       0.16    0.19  0.45  0.44
## logonsRT.object     0.11    0.24  0.27  0.30
## logartic.object     -0.04    0.01  0.20  0.06
## logpautE.object     0.14    0.15  0.33  0.33
## logarttE.object     0.19    0.16  0.32  0.25
##
## *** G5
##               logwacc sqpwacc wflut pwflu
## logonsRT.digit    0.10    0.02  0.31  0.24
## logartic.digit    0.02    0.16  0.12  0.06
## logpautE.digit    0.19    0.11  0.43  0.43
## logarttE.digit    0.09    0.12  0.43  0.40
## logonsRT.numword  0.11    0.06  0.38  0.32
## logartic.numword  0.07    0.20  0.20  0.13
## logpautE.numword  0.16    0.12  0.60  0.51
## logarttE.numword  0.13    0.16  0.50  0.44
## logonsRT.word     0.17    0.05  0.54  0.44
## logartic.word     0.07    0.15  0.27  0.16
## logpautE.word     0.22    0.20  0.57  0.55
## logarttE.word     0.15    0.18  0.53  0.46
## logonsRT.dice     0.10    0.07  0.28  0.24
## logartic.dice     0.04    0.19  0.16  0.12
## logpautE.dice     0.23    0.09  0.31  0.36
## logarttE.dice     0.07    0.16  0.34  0.24
## logonsRT.object   0.16    0.05  0.34  0.33
## logartic.object   0.04    0.19  0.21  0.13
## logpautE.object   0.23    0.11  0.25  0.29
## logarttE.object   -0.01    0.06  0.14  0.11
```

7.2 Graphical display of correlations

To facilitate comparisons, the above correlations are displayed graphically below. In the following graph, each row of panels concerns one reading measure (word reading accuracy, pseudoword reading accuracy, word list reading fluency, and pseudoword list reading fluency). Each column of panels concerns one type of naming stimulus (digits, number words, words, dice, and objects). Within each panel, there are four bar triplets: The two triplets on the left (lighter colors) concern discrete naming tasks and the two triplets on the right (darker colors) concern serial naming tasks. For each pair of triplets, silent intervals are on the left and spoken intervals on the right. Each triplet displays correlation coefficients between the corresponding reading and naming measures for three groups, namely Grades 1, 3, and 5, distinguished by increasing shading density.

```
#pdf("corread.pdf",width=10,height=7)
par(mfcol=c(4,5))
par(oma=c(7,4,1,2))
par(mai=c(0.05,0.05,0.05,0.05))
```

```

par(mgp=c(2,0.6,0))
par(tcl= -0.2)

rmnice <- c("Word accuracy","Pseud. accuracy","Word fluency","Pseud. fluency")

# Run a dummy plot to obtain the horizontal coordinates for placement
barplot(rbind(rncor[["G1"]][1,paste(nmeas,c(tasks[1]),sep=".")],
             rncor[["G1"]][1,paste(nmeas,c(tasks[1]),sep=".")],
             rncor[["G1"]][1,paste(nmeas,c(tasks[1]),sep=".")]),
        ylim=c(0,1),width=0.2,beside=TRUE,plot=FALSE) -> mp

for (it in 1:length(tasks)) {

  tsk <- tasks[it]

  for (ir in 1:length(rmeas)) {

    rms <- rmeas[ir]

    # First plot the bars
    barplot(
      rbind(rncor[["G1"]][rms,paste(nmeas,c(tsk),sep=".")],
            rncor[["G3"]][rms,paste(nmeas,c(tsk),sep=".")],
            rncor[["G5"]][rms,paste(nmeas,c(tsk),sep=".")]),
      ylim=c(0,1),axes=FALSE,axisnames=FALSE,xpd=NA,
      col=rep(c("#FFD0B0","#B0D0FF","#C08000","#0080C0"),each=3),
      width=0.2,beside=TRUE)

    # Then draw the shading lines on top
    barplot(
      rbind(rncor[["G1"]][rms,paste(nmeas,c(tsk),sep=".")],
            rncor[["G3"]][rms,paste(nmeas,c(tsk),sep=".")],
            rncor[["G5"]][rms,paste(nmeas,c(tsk),sep=".")]),
      ylim=c(0,1),axes=FALSE,axisnames=FALSE,xpd=NA,
      col=rep(c("#FF9070","#7090FF","#604000","#004060"),each=3),density=rep(c(0,15,30),4),
      width=0.2,beside=TRUE,add=TRUE)

    # Axes, legends, etc.
    if (ir==4) {
      axis(1,at=c(mean(mp[2,c(1,2)]),mean(mp[2,c(3,4)])),labels=c("Discrete","Serial"),las=2,tick=FALSE)
      axis(1,at=mp[2,],labels=c("onset latency","articulation","pause","articulation"),
           las=2,tick=FALSE,cex.axis=1.2,xpd=NA)
    } else {
      axis(1,labels=NA,tick=FALSE)
    }

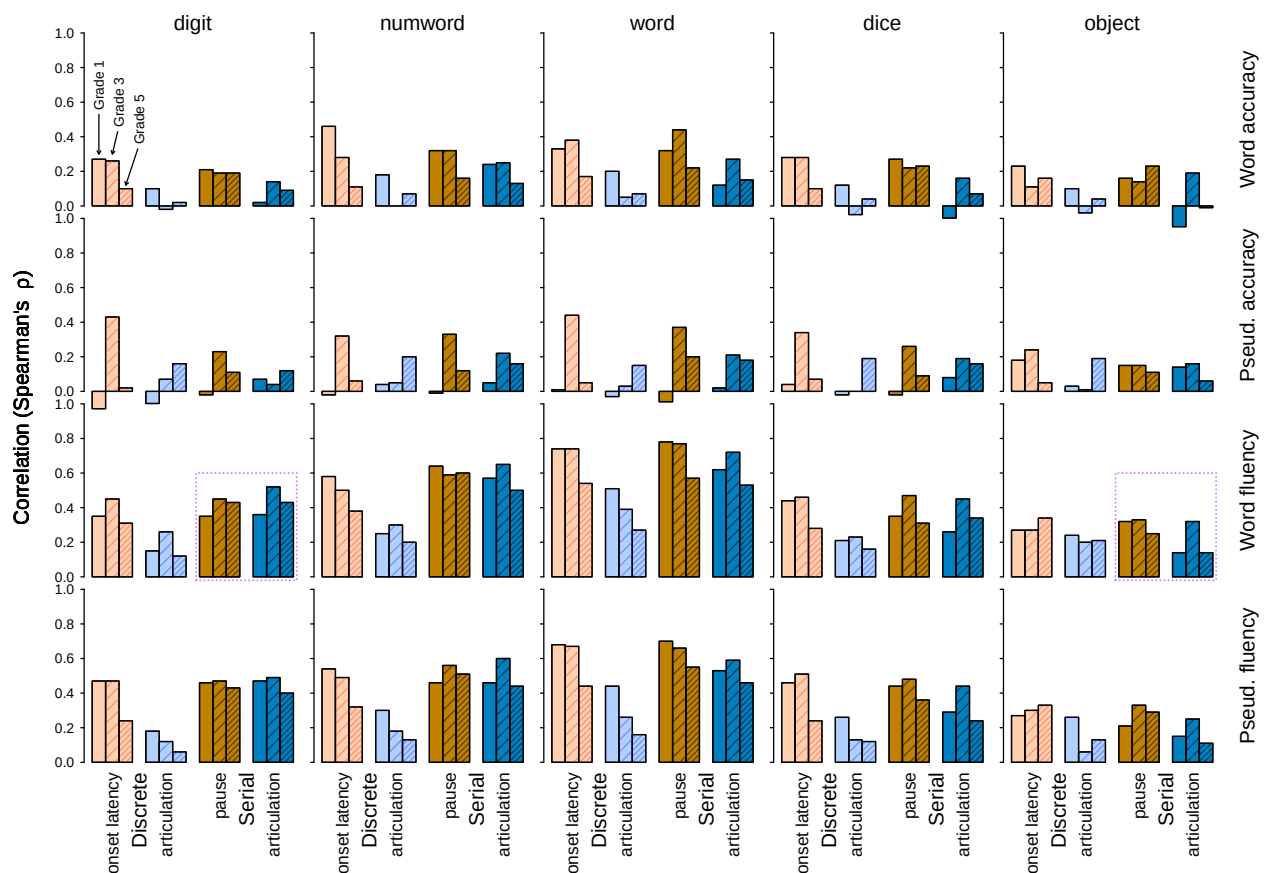
    if (ir==1) {
      mtext(tsk,3)
      if (it==1) {
        text(mp[1,1],0.7,"Grade 1",adj=c(0.5,0.5),srt=90,cex=1)
        text(mp[2,1]+0.1,0.6,"Grade 3",adj=c(0.5,0.5),srt=90,cex=1)
        text(mp[3,1]+0.2,0.5,"Grade 5",adj=c(0.5,0.5),srt=90,cex=1)
        arrows(mp[1,1],0.53,mp[1,1],0.30,length=0.03)
        arrows(mp[2,1]+0.08,0.43,mp[2,1],0.30,length=0.03)
        arrows(mp[3,1]+0.18,0.33,mp[3,1],0.12,length=0.03)
      }
    }
  }
}

```

```

    }
  }
  if (ir==3) {
    if (it %in% c(1,5)) { rect(mp[1,3]-0.15,-0.02,mp[3,4]+0.15,0.6,col=NA,border="#D080FF",lwd=1.5,lt)
    }
    if (it==5) { mtext(rmnice[ir],4,line=1.25) }
    if (it==1) {
      axis(2,las=1)
    } else {
      axis(2,labels=NA)
    }
  }
}
mtext(expression(paste("Correlation (Spearman's ",rho,")"),sep=""),2,outer=TRUE,line=2.5)
}

```



#dev.off()

In general, and as expected from the literature, we see that (a) reading fluency measures are predicted much more strongly than reading accuracy, and (b) a bit more strongly from digit naming (leftmost column) than from object naming (rightmost column). (c) The duration of spoken intervals from discrete naming (light blue bars) is the weakest predictor of reading measures across grades and tasks; and (d) the duration of silent intervals from discrete naming (onset latency; pink bars) is almost as good a predictor of reading measures as the duration of silent intervals from serial tasks (pause time; dark goldenrod bars). (e) Naming tasks involving print (that is, words and number words) are, naturally, the stronger predictors of reading measures.

The correlations most nearly (though not exactly) comparable with the literature are enclosed in lilac dashed boxes on the third row: they concern correlations between word list reading fluency and the “RAN components” of digit and object naming. We can see that silent and spoken intervals from digit naming (leftmost column) are associated with fluency to about the same extent, around 0.4. Associations with intervals from object naming (rightmost column) are lower, and less strong for spoken than for silent intervals.

8 Session information

```
date()

## [1] "Fri Feb  9 12:27:11 2024"

sessionInfo()

## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=Norwegian Bokmål_Norway.utf8  LC_CTYPE=Norwegian Bokmål_Norway.utf8
## [3] LC_MONETARY=Norwegian Bokmål_Norway.utf8 LC_NUMERIC=C
## [5] LC_TIME=Norwegian Bokmål_Norway.utf8
##
## time zone: Europe/Berlin
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] multcomp_1.4-24 TH.data_1.1-2  MASS_7.3-60      survival_3.5-5  mvtnorm_1.2-2  nlme_3.1-162
## [7] lattice_0.21-8  mice_3.16.0      psych_2.3.3
##
## loaded via a namespace (and not attached):
## [1] sandwich_3.0-2  utf8_1.2.3      generics_0.1.3  tidyr_1.3.0     shape_1.4.6
## [6] lme4_1.1-33     digest_0.6.31   magrittr_2.0.3  mitml_0.4-5     evaluate_0.21
## [11] grid_4.3.1      iterators_1.0.14 fastmap_1.1.1   foreach_1.5.2   jomo_2.7-6
## [16] glmnet_4.1-7    Matrix_1.5-4.1  nnet_7.3-19     backports_1.4.1 purrr_1.0.1
## [21] fansi_1.0.4     codetools_0.2-19 mnormt_2.1.1    cli_3.6.1       rlang_1.1.1
## [26] splines_4.3.1   yaml_2.3.7      pan_1.6         tools_4.3.1     parallel_4.3.1
## [31] nloptr_2.0.3    minqa_1.2.5     dplyr_1.1.2     boot_1.3-28.1   broom_1.0.5
## [36] rpart_4.1.19    vctrs_0.6.3     R6_2.5.1        zoo_1.8-12      lifecycle_1.0.3
## [41] pkgconfig_2.0.3 pillar_1.9.0     glue_1.6.2      Rcpp_1.0.10     highr_0.10
## [46] xfun_0.39       tibble_3.2.1    tidyselect_1.2.0 rstudioapi_0.14 knitr_1.43
## [51] htmltools_0.5.5 rmarkdown_2.22  compiler_4.3.1
```