

Supplementary Materials: Stem Formation in French Verbs: Structure, Rules, and Allomorphy

Gustavo L. Estivalet and Fanny Meunier

Text S3: R script for statistical analyses

```
## Load packages
library(Matrix)
library(lme4)
library(lmerTest)
library(car)

## Clear all
rm(list=ls())
objects()

#####
#### Cross-modal priming ####
#####

## Read data
cross1 <- read.table(file = "c:/.../cross.csv", header = TRUE, sep = ";", dec = ",")
cross1$Subject <- as.factor(cross1$Subject)
cross1$ACC <- as.logical(cross1$ACC)
cross1$RTlog <- log(cross1$RT)
cross1$RTinv <- 1/cross1$RT * -1000
str(cross1)
head(cross1)

## Filler filter
cross2 <- subset(cross1, VerbType != "FILLER")
cross2 <- droplevels(cross2)
round(100 * (1 - nrow(cross2)/nrow(cross1)), 2)

## Timeout filter
cross3 <- subset(cross2, RT > 300 & RT < 1800)
cross3 <- droplevels(cross3)
round(100 * (1 - nrow(cross3)/nrow(cross2)), 2)

## Subject ACC
cross.subject.acc.prop <- 100 * prop.table(xtabs(~Subject+ACC, cross3), 1)
cross.subject.acc.prop[order(cross.subject.acc.prop[,1]),]

## Item ACC
cross.item.acc.prop <- 100 * prop.table(xtabs(~Target+ACC, cross3), 1)
cross.item.acc.prop[order(cross.item.acc.prop[,1]),]

## Reject items
cross4 <- subset(cross3, Target != "CEIGNONS" & Target != "DEPARTONS")
cross4 <- droplevels(cross4)
round(100 * (1 - nrow(cross4)/nrow(cross3)), 2)
```

```

## ACC filter
cross5 <- subset(cross4, ACC == TRUE)
cross5 <- droplevels(cross5)
round(100 * (1 - nrow(cross5)/nrow(cross4)), 2)
round(100 * (1 - nrow(cross5)/nrow(cross2)), 2)

## RT/SD verb type and priming condition
round(with(cross5, tapply(RT, list(VerbType, PrimeCond), mean)))
round(with(cross5, tapply(RT, list(VerbType, PrimeCond), sd)))

## Priming differences
cross.cat.prime.diff <- as.data.frame(with(cross5, tapply(RT, list(VerbType, PrimeCond), mean)))
cross.cat.prime.diff$ControlTest <- cross.cat.prime.diff$CONTROL - cross.cat.prime.diff$TEST
cross.cat.prime.diff$TestIdentity <- cross.cat.prime.diff$TEST - cross.cat.prime.diff$IDENTITY
round(cross.cat.prime.diff)

## Mixed-effects model - RT
cross5$PrimeCond <- relevel(cross5$PrimeCond, ref="TEST")
cross5$VerbType <- relevel(cross5$VerbType, ref="ER")

cross.rt.lmer <- lmer(RTinv ~ VerbType * PrimeCond + (1|Subject) + (1|Target), cross5)
summary(cross.rt.lmer)
anova(cross.rt.lmer)

## ACC verb type
round(100 * prop.table(xtabs(~VerbType+ACC, cross4)), 2)

## ACC priming condition
round(100 * prop.table(xtabs(~PrimeCond+ACC, cross4)), 2)

## Mixed-effects model - ACC
cross4$PrimeCond <- relevel(cross4$PrimeCond, ref="TEST")
cross4$VerbType <- relevel(cross4$VerbType, ref="ER")

cross.acc.glmer <- glmer(ACC ~ VerbType * PrimeCond + (1|Subject) + (1|Target), family =
binomial, cross4)
summary(cross.acc.glmer)
Anova(cross.acc.glmer)

#####
#### Masked priming ####
#####

## Read data
masked1 <- read.table(file = "c:/../masked.csv", header = TRUE, sep = ";", dec = ",")
masked1$Subject <- as.factor(masked1$Subject)
masked1$ACC <- as.logical(masked1$ACC)
masked1$RTlog <- log(masked1$RT)
masked1$RTinv <- 1/masked1$RT * -1000
levels(masked1$PrimeCond)[levels(masked1$PrimeCond) == "ORTHO"] <- "IDENTITY"
levels(masked1$PrimeCond)[levels(masked1$PrimeCond) == "SEMANTIC"] <- "TEST"
str(masked1)
head(masked1)

```

```

## Filler filter
masked2 <- subset(masked1, VerbType != "FILLER")
masked2 <- droplevels(masked2)
round(100 * (1 - nrow(masked2)/nrow(masked1)), 2)

## Timeout filter
masked3 <- subset(masked2, RT > 300 & RT < 1800)
masked3 <- droplevels(masked3)
round(100 * (1 - nrow(masked3)/nrow(masked2)), 2)

## Subject ACC
masked.subject.acc.prop <- 100 * prop.table(xtabs(~Subject+ACC, masked3), 1)
masked.subject.acc.prop[order(masked.subject.acc.prop[,1]),]

## Item ACC
masked.item.acc.prop <- 100 * prop.table(xtabs(~Target+ACC, masked3), 1)
masked.item.acc.prop[order(masked.item.acc.prop[,1]),]

## Reject items
masked4 <- subset(masked3, Target != "CEIGNONS")
masked4 <- droplevels(masked4)
round(100 * (1 - nrow(masked4)/nrow(masked3)), 2)

## ACC filter
masked5 <- subset(masked4, ACC == TRUE)
masked5 <- droplevels(masked5)
round(100 * (1 - nrow(masked5)/nrow(masked4)), 2)
round(100 * (1 - nrow(masked5)/nrow(masked2)), 2)

## RT/SD verb type and priming condition
round(with(masked5, tapply(RT, list(VerbType, PrimeCond), mean)))
round(with(masked5, tapply(RT, list(VerbType, PrimeCond), sd)))

## Priming differences
masked.cat.prime.diff <- as.data.frame(with(masked5, tapply(RT, list(VerbType, PrimeCond),
mean)))
masked.cat.prime.diff$ControlTest <- masked.cat.prime.diff$CONTROL -
masked.cat.prime.diff$TEST
masked.cat.prime.diff$TestIdentity <- masked.cat.prime.diff$TEST -
masked.cat.prime.diff$IDENTITY
round(masked.cat.prime.diff)

## Mixed-effects model - RT
masked5$PrimeCond <- relevel(masked5$PrimeCond, ref="TEST")
masked5$VerbType <- relevel(masked5$VerbType, ref="ER")

masked.rt.lmer <- lmer(RTinv ~ VerbType * PrimeCond + (1|Subject) + (1|Target), masked5)
summary(masked.rt.lmer)
anova(masked.rt.lmer)

## ACC verb type

```

```
round(100 * prop.table(xtabs(~VerbType+ACC, masked4)), 2)

## ACC priming condition
round(100 * prop.table(xtabs(~PrimeCond+ACC, masked4)), 2)

## Mixed-effects model - ACC
masked4$PrimeCond <- relevel(masked4$PrimeCond, ref="TEST")
masked4$VerbType <- relevel(masked4$VerbType, ref="ER")

masked.acc.glmer <- glmer(ACC ~ VerbType * PrimeCond + (1|Subject) + (1|Target), family =
binomial, masked4)
summary(masked.acc.glmer)
Anova(masked.acc.glmer)
```