

### Supplemental Material S3. R code for data analysis.

**## Analysis 1:** the same code applies to all treatment steps, with exceptions for Steps 3 & 6##

#Defining variables

```
Step$acc<-as.factor(Step$Accuracy) #accuracy
Step$subject<-as.factor(Step$ID) #subjects
Step$language<-as.factor(Step$Lang) #language (Mandarin vs. English)
Step$session_c<-scale(Step$session,center=TRUE,scale=FALSE) #center session
Step$item<-as.factor(Step$Tx_Item) #treatment item
Step$aq_c<-scale(Step$aq,center=TRUE,scale=TRUE) #center & scale AQ
Step$exp<-as.numeric(Step$Exposure) #number of treated items over time
Step$TxType<-as.factor(Step$Tx_Type) #treated category (noun vs. verb)
Step$TxType_v<-relevel(Step$TxType,ref="V") #verb as the reference level
Step$order<-factor(Step$Tx_Order,order=T) #treatment order
```

#Plot the actual performance

```
Step$TxType<-factor(Step$TxType,levels = c("N","V"),
                    labels = c("Noun Tx","Verb Tx"))
ggplot(Step,aes(x=session,y=acc_num,color=language))+
  facet_wrap(~TxType)+
  geom_smooth(method='loess',se=T,fullrange=F,alpha=0.2)+
  scale_color_brewer(name="Tx Lang",palette="Dark2")+
  coord_cartesian(xlim=c(1,20), ylim=c(0,1)) +
  scale_y_continuous(breaks=seq(0,1,0.1))+
  labs(title="(a) Treatment Performance",x="Tx Session",y="Average
Accuracy",color="Treatment Lang")+
  theme_bw()+
  theme(panel.grid.minor=element_blank(),legend.position = "top",
        plot.title=element_text(size=14,hjust=0.5),text=element_text(size=12),plot.caption =
element_text(size=12,hjust=0))
```

#Performing mixed-effects models#

#Linear term for session

```
mSa<-
glmer(acc~session_c*TxType+aq_c+exp+order+(1|item)+(1+session_c|subject),data=Step,famil
y=binomial(link="logit"),control=glmerControl(optimizer =
"bobyqa",calc.derivs=FALSE,optCtrl = list(maxfun=2e5)))
summary(mSa)
mSb<-
glmer(acc~session_c*TxType_v+aq_c+exp+order+(1|item)+(1+session_c|subject),data=Step,fa
mily=binomial(link="logit"),control=glmerControl(optimizer =
"bobyqa",calc.derivs=FALSE,optCtrl = list(maxfun=2e5)))
summary(mSb) #change the reference level to "verb"
```

#For steps 3 & 6, ordered mixed-effects models were performed:

```
mSa<-
clmm(scale~session_c*TxType+aq_c+exp+order+(1+session_c|subject)+(1|item),data=Step,Hess=T)
summary(mSa)
mSb<-
clmm(scale~session_c*TxType_v+aq_c+exp+order+(1+session_c|subject)+(1|item),data=Step,Hess=T)
summary(mSb)

##Quadratic term for session
mSa2<-
glmer(acc~session_c*TxType+I(session_c^2)+aq_c+exp+order+(1|item)+(1+session_c|subject),
data=Step,family=binomial(link="logit"),control=glmerControl(optimizer =
"bobyqa",calc.derivs=FALSE,optCtrl = list(maxfun=2e5)))
summary(mSa2)
mSb2<-
glmer(acc~session_c*TxType_v+I(session_c^2)+aq_c+exp+order+(1|item)+(1+session_c|subject),
data=Step,family=binomial(link="logit"),control=glmerControl(optimizer =
"bobyqa",calc.derivs=FALSE,optCtrl = list(maxfun=2e5)))
summary(mSb2) #change the reference level to "verb"

#Quadratic term for Steps 3 & 6:
m3a2<-
clmm(scale~session_c*TxType+I(session_c^2)+aq_c+exp+order+(1+session_c|subject)+(1|item),
data=Step,Hess=T)
summary(m3a2)
m3b2<-
clmm(scale~session_c*TxType_v+I(session_c^2)+aq_c+exp+order+(1+session_c|subject)+(1|item),
data=Step,Hess=T)
summary(m3b2)

anova(mSa,mSa2) #Model comparison

#Follow-up analysis: Effect of treatment language
m_int<-
glmer(acc~session_c*TxType*language+aq_c+exp+order+(1|item)+(1|subject),data=Step,family=
binomial(link="logit"),control=glmerControl(optimizer = "bobyqa",calc.derivs=FALSE,optCtrl
= list(maxfun=2e5)))
summary(m_int)
m_int2<-
glmer(acc~session_c*TxType_v*language+aq_c+exp+order+(1|item)+(1|subject),data=Step,fam
ily=binomial(link="logit"),control=glmerControl(optimizer =
"bobyqa",calc.derivs=FALSE,optCtrl = list(maxfun=2e5)))
summary(summary)
```

## ##Analysis 2##

#Separate noun data and verb data

```
StepN<-Step[(Step$TxType=="N"),]
```

```
StepV<-Step[(Step$TxType=="V"),]
```

##Creating 10 time points, each was the average treatment performance across every two treatment sessions

#Noun

```
num_session<-length(unique(StepN$session))
```

```
avg_accuracy<-rep(NA,num_session-1)
```

```
df<-data.frame(c(1:10))
```

```
for(j in 1:10){
```

```
  for(i in 1:num_session){
```

```
    avg_accuracy[i]<-mean(StepN$acc_num[(StepN$session==i | StepN$session==i+1) &  
StepN$subject==unique(StepN$subject)[j]])
```

```
  }
```

```
  df[,ncol(df)+1]=avg_accuracy[c(1,3,5,7,9,11,13,15,17,19)]
```

```
  names(df)[ncol(df)]<-paste0("avg_accuracy_final_",j)
```

```
}
```

##Verb

```
num_session<-length(unique(StepV$session))
```

```
avg_accuracy<-rep(NA,num_session-1)
```

```
df<-data.frame(c(1:10))
```

```
for(j in 1:10){
```

```
  for(i in 1:num_session){
```

```
    avg_accuracy[i]<-mean(StepV$acc_num[(StepV$session==i | StepV$session==i+1) &  
StepV$subject==unique(StepV$subject)[j]])
```

```
  }
```

```
  df[,ncol(df)+1]=avg_accuracy[c(1,3,5,7,9,11,13,15,17,19)]
```

```
  names(df)[ncol(df)]<-paste0("avg_accuracy_final_",j)
```

```
}
```

#Defining variables

```
all$subj<-as.factor(all$ID) #subjects
```

```
all$aq_c<-scale(all$aq,center=T,scale=T) #centered and scaled AQ
```

```
all$lang<-as.factor(all$Tx_Lang) #treatment language
```

```
all$session_c<-scale(all$session,center=T,scale=F) #centered session
```

```
all$cond<-as.factor(all$Condition) #probe stimuli condition
```

```
all$probe<-as.numeric(all$Probe_score) #probe score
```

```
all$tx<-as.numeric(all$Tx_score) #treatment performance
```

```
all$type<-as.factor(all$Tx_type) #treatment target (noun vs. verb)
```

```
all$type_v<-relevel(all$type,ref="V") #change the reference level to "verb"
```

```
all$step<-as.factor(all$Tx_step) #treatment step
```

```
#separate by the probe stimuli condition in each treatment
all_V<-all[(all$type=="V"),] #verb treatment
all_trainedV<-all_V[(all_V$cond=="Trained"),]
all_untrainedV<-all_V[(all_V$cond=="Untrained"),]
all_trainedTrV<-all_V[(all_V$cond=="TR_Trained"),]
all_untrainedTrV<-all_V[(all_V$cond=="TR_Untrained"),]

all_N<-all[(all$type=="N"),] #noun treatment
all_trainedN<-all_N[(all_N$cond=="Trained"),]
all_untrainedN<-all_N[(all_N$cond=="Untrained"),]
all_trainedTrN<-all_N[(all_N$cond=="TR_Trained"),]
all_untrainedTrN<-all_N[(all_N$cond=="TR_Untrained"),]

#Mixed-effect models#
#Further separate by the treatment step
trainedN<-all_trainedN[(all_trainedN$Tx_step=="S"),] #replace "S" with the specific treatment
step number
untrainedN<-all_untrainedN[(all_untrainedN$Tx_step=="S"),]
trainedTrN<-all_trainedTrN[(all_trainedTrN$Tx_step=="S"),]
untrainedTrN<-all_untrainedTrN[(all_untrainedTrN$Tx_step=="S"),]

MtrainedN<-lmer(probe~tx*session_c+aq_c+(1|subj),data=trainedN) #trained set
summary(MtrainedN)
eta_squared(MtrainedN,alternative="two.sided") #Partial eta-squared

MuntrainedN<-lmer(probe~tx*session_c+aq_c+(1|subj),data=untrainedN) #untrained set
summary(MuntrainedN)
eta_squared(MuntrainedN,alternative="two.sided")

MtrainedTrN<-lmer(probe~tx*session_c+aq_c+(1|subj),data=trainedTrN) #translations of the
trained set
summary(MtrainedTrN)
eta_squared(MtrainedTrN,alternative="two.sided")

MuntrainedTrN<-lmer(probe~tx*session_c+aq_c+(1|subj),data=untrainedTrN) #translations of
the untrained set
summary(MuntrainedTrN)
eta_squared(MuntrainedTrN,alternative="two.sided")

##Verb
trainedV<-all_trainedV[(all_trainedV$Tx_step=="S"),] #replace "S" with the specific treatment
step number
untrainedV<-all_untrainedV[(all_untrainedV$Tx_step=="S"),]
trainedTrV<-all_trainedTrV[(all_trainedTrV$Tx_step=="S"),]
untrainedTrV<-all_untrainedTrV[(all_untrainedTrV$Tx_step=="S"),]
```

```
MtrainedV<-lmer(probe~tx*session_c+aq_c+(1|subj),data=trainedV) #trained set
summary(MtrainedV)
eta_squared(MtrainedV,alternative="two.sided") #Partial eta-squared
```

```
MuntrainedV<-lmer(probe~tx*session_c+aq_c+(1|subj),data=untrainedV) #untrained set
summary(MuntrainedV)
eta_squared(MuntrainedV,alternative="two.sided")
```

```
MtrainedTrV<-lmer(probe~tx*session_c+aq_c+(1|subj),data=trainedTrV) #translations of the
trained set
summary(MtrainedTrV)
eta_squared(MtrainedTrV,alternative="two.sided")
```

```
MuntrainedTrV<-lmer(probe~tx*session_c+aq_c+(1|subj),data=untrainedTrV) #translations of
the untrained set
summary(MuntrainedTrV)
eta_squared(MuntrainedTrV,alternative="two.sided")
```