

Analysis of results for Unboundedness

Anastasia Kobzeva

Data pre-processing and plotting

```
# Loading the required packages
library(tidyverse)
library(plotrix)
library(lme4)
library(gt)
library(modelsummary)
library(patchwork)

# To disable dplyr messages "summarise() has grouped output by..."
options(dplyr.summarise.inform = FALSE)
rm(list = ls()) # removing everything from the environment
```

Choose model type for analysis

```
# Change depending on the model
#model_type = "lstm"
model_type = "gpt2"

filename_wh = sprintf("../data/results/%s/unbound_wh_result.csv", model_type) # Norwegian wh
filename_rc = sprintf("../data/results/%s/unbound_rc_result.csv", model_type) # Norwegian RC
filename_en = sprintf("../data/results/%s/unbound_wh_en_result.csv", model_type) # English wh
```

File naming conventions

- raw = raw surprisal values
- fe = filler effects (surprisal difference between +filler, -filler conditions)
- no = Norwegian
- en = English
- wh = wh-dependencies
- rc = RC-dependencies
- regions = plots data from all sentence regions
- roi = plots data from regions of interest (where we look for FEs)
- uge = unlicensed gap effect
- fge = filled-gap effect
- all = all language-dependency combinations

Loading in data and analysis functions, defining paths for plots

```
# Loading analysis functions
source("analysis-functions.R")
```

```
df_wh = read.csv(filename_wh, fileEncoding = "UTF-8-BOM")
df_rc = read.csv(filename_rc, fileEncoding = "UTF-8-BOM")
df_en = read.csv(filename_en, fileEncoding = "UTF-8-BOM")
```

```
df_no = rbind(df_wh, df_rc)
df = rbind(df_no, df_en) # df with both languages
```

```
df_wh = df %>%
  filter(dependency == "Wh")
df_rc = df %>%
  filter(dependency == "RC")
```

Wh-dependencies

Wh-dependencies have the same structure in Norwegian and English, analyzing them together

```
df_wh = df_wh %>%
  mutate(region = if_else(word == "." | word == "<eos>" & region == "end", "EOS", region),
         region = if_else(region == "that" | region == "wh-obj", "prefix", region),
         region = if_else(region == "2-layers" | region == "3-layers" |
                           region == "4-layers" | region == "5-layers" |
                           region == "2-that" | region == "3-that" |
                           region == "4-that" | region == "5-that", "layers", region),
         region = factor(region, levels = REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "layers"))
```

introduce another column which states whether there's "that" in emb or not

```
df_wh = df_wh %>%
  add_column(that_comp = "")
```

populate the "that_comp" column using the layers column

```
df_wh = df_wh %>%
  mutate(that_comp = if_else(layers == "1" | layers == "2" | layers == "3" |
                             layers == "4" | layers == "5", "no", that_comp),
         that_comp = if_else(layers == "1-comp" | layers == "2-comp" | layers == "3-comp" |
                             layers == "4-comp" | layers == "5-comp", "yes", that_comp))
```

now only the numeric is left in layers

```
df_wh = df_wh %>%
  mutate(layers = gsub("-comp", "", layers))
```

Creating an aggregated data frame

```
d_agg = df_wh %>%
  group_by(region, sent_index, comp, gap, layers, language, that_comp) %>%
  summarise(surprisal=sum(surprisal)) %>%
  ungroup() %>%
  mutate(comp_numeric=if_else(comp == "what", 0.5, -0.5),
         comp=factor(comp, levels=c("what", "that")),
         gap=factor(gap, levels=c("no-gap", "gap")),
         layers=factor(layers, levels=c("1", "2", "3", "4", "5")))
```

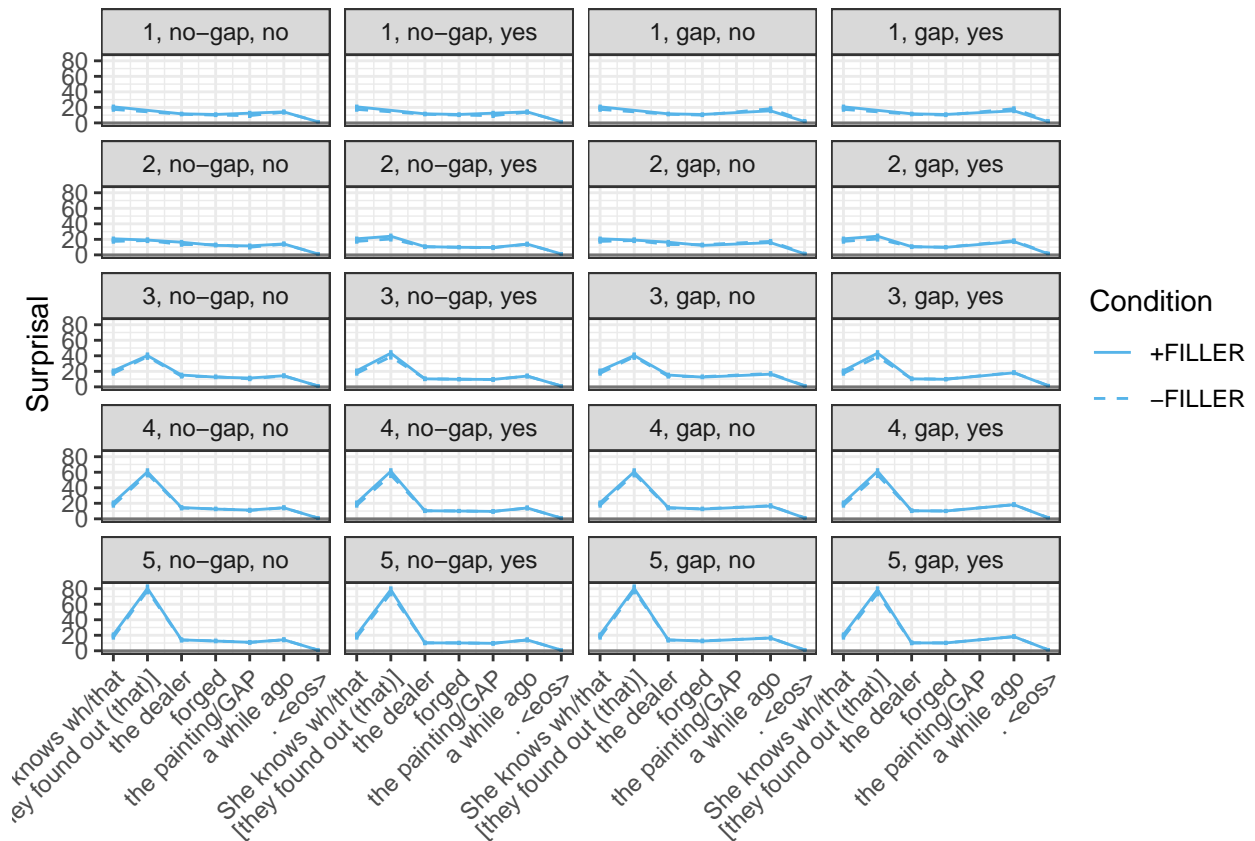
Plotting raw surprisal by region for Norwegian wh

```
d_agg %>%
  filter(language == "Norwegian") %>%
  group_by(region, gap, comp, layers, that_comp) %>%
  summarise(m=mean(surprisal),
```

```

s=std.error(surprisal),
upper=m + 1.96*s,
lower=m - 1.96*s) %>%
ungroup() %>%
mutate(region=as.numeric(region)) %>%
ggplot(aes(x=region, y=m, ymax=upper, ymin=lower, linetype=comp)) +
  facet_wrap(layers~gap~that_comp, ncol=4,
            labeller = label_wrap_gen(multi_line=FALSE)) +
  geom_line(color = "#56B4E9") + theme_bw() +
  geom_errorbar(linetype="solid", width=.1, color = "#56B4E9") +
  scale_x_continuous(breaks=seq(1, length(REGION_ORDER)), labels=REGION_EXEMPLARS) +
  geom_hline(yintercept=0, color="black", alpha=0.5) + ylab("Surprisal") +
  theme(axis.text.x = element_text(angle=45, hjust=1),
        axis.title.x = element_blank(), legend.margin=margin(c(0,0,0,0))) +
  scale_linetype_manual(name = "Condition", labels = c("+FILLER", "-FILLER"),
                        values = c('solid', 'dashed'))

```



```

fname = sprintf("plots/unbound-plots/%s-unbound-regions-row-no-wh.png", model_type)
ggsave(fname, width = 10, height = 10)

```

```

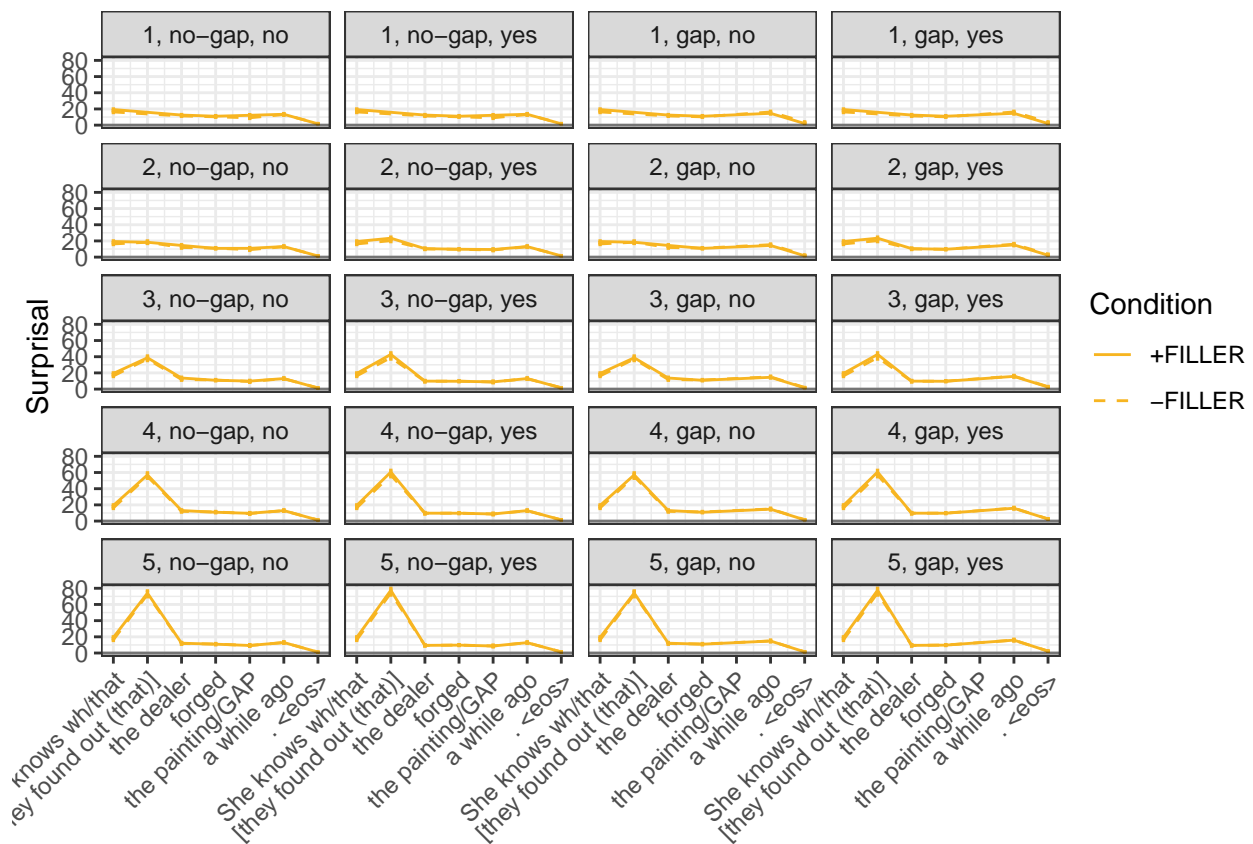
# Plotting raw surprisal by region for English wh
d_agg %>%
  filter(language == "English") %>%
  group_by(region, gap, comp, layers, that_comp) %>%
  summarise(m=mean(surprisal),
            s=std.error(surprisal),
            upper=m + 1.96*s,

```

```

    lower=m - 1.96*s) %>%
  ungroup() %>%
  mutate(region=as.numeric(region)) %>%
  ggplot(aes(x=region, y=m, ymax=upper, ymin=lower, linetype=comp)) +
    facet_wrap(layers~gap~that_comp, ncol=4,
      labeller = label_wrap_gen(multi_line=FALSE)) +
    geom_line(color = "#F7B521") + theme_bw() +
    geom_errorbar(linetype="solid", width=.1, color = "#F7B521") +
    scale_x_continuous(breaks=seq(1, length(REGION_ORDER)), labels=REGION_EXEMPLARS) +
    geom_hline(yintercept=0, color="black", alpha=0.5) + ylab("Surprisal") +
    theme(axis.text.x = element_text(angle=45, hjust=1),
      axis.title.x = element_blank(), legend.margin=margin(c(0,0,0,0))) +
    scale_linetype_manual(name = "Condition", labels = c("+FILLER", "-FILLER"),
      values = c('solid', 'dashed'))

```



```

fname = sprintf("plots/unbound-plots/%s-unbound-regions-raw-en-wh.png", model_type)
ggsave(fname, width = 10, height = 10)

```

```

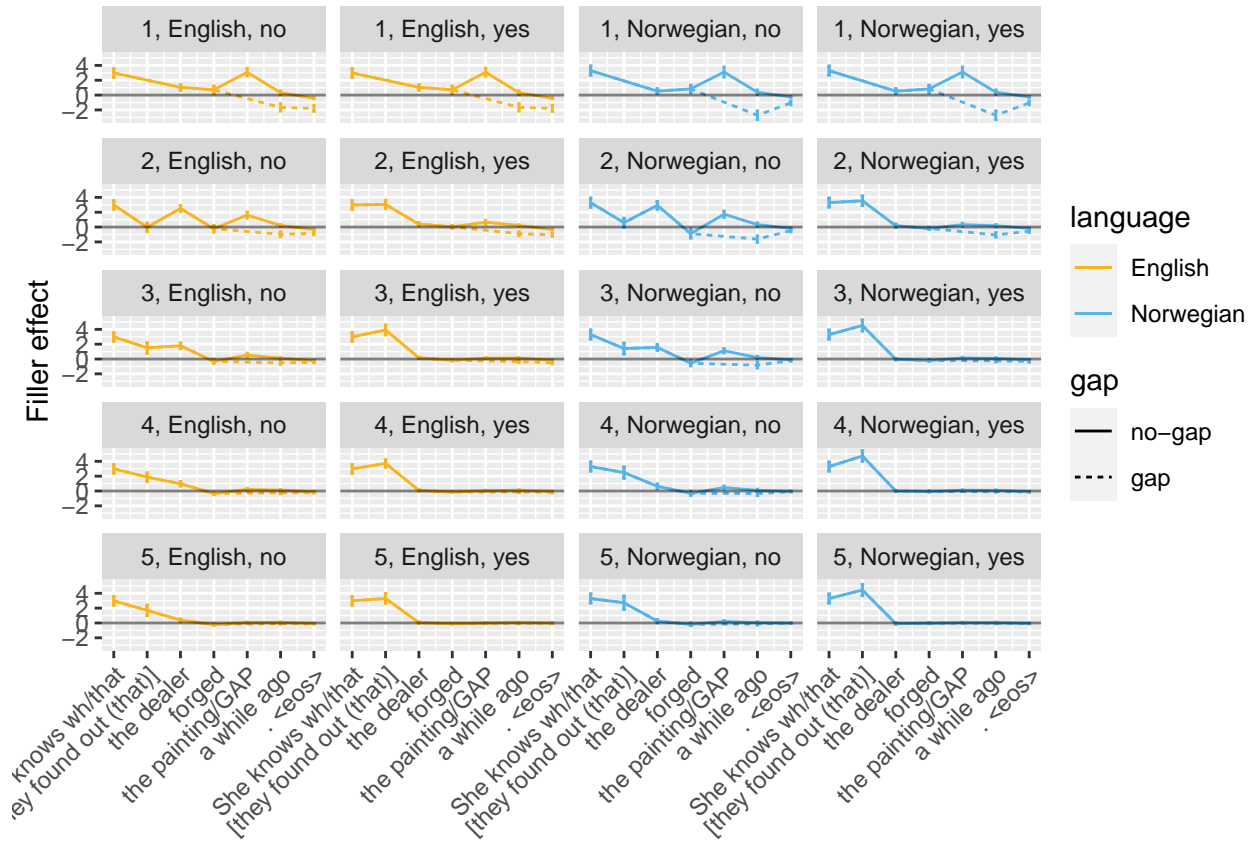
# Plotting filler effects (FGEs and UGEs) by region for wh-dep (both lang)
two_colors = c("#F7B521", "#56B4E9")
d_agg %>%
  select(-comp_numeric) %>%
  spread(comp, surprisal) %>%
  mutate(filler_effect=what~that) %>%
  group_by(language, region, gap, layers, that_comp) %>%
  summarise(m=mean(filler_effect),
    s=std.error(filler_effect),

```

```

    upper=m + 1.96*s,
    lower=m - 1.96*s) %>%
  ungroup() %>%
  mutate(region=as.numeric(region)) %>%
  ggplot(aes(x=region, y=m, ymax=upper, ymin=lower, linetype=gap, color=language)) +
    facet_wrap(layers~language~that_comp, ncol=4, labeller = label_wrap_gen(multi_line=FALSE)) +
    geom_line() +
    geom_errorbar(linetype="solid", width=.1) +
    scale_x_continuous(breaks=seq(1, length(REGION_ORDER)), labels=REGION_EXEMPLARS) +
    geom_hline(yintercept=0, color="black", alpha=0.5) +
    theme(axis.text.x = element_text(angle=45, hjust=1),
          axis.title.x = element_blank(), legend.margin=margin(c(0,0,0,0))) +
    ylab("Filler effect") + scale_colour_manual(values = two_colors)

```



```

fname = sprintf("plots/unbound-plots/%s-unbound-regions-fe-wh.png", model_type)
ggsave(fname, width = 10, height = 10)

```

Calculating filler effects in RoIs (region of the filled NP/post-gap region)

unlicensed gap effect

```

uge = d_agg %>%
  filter(region == "end" & gap == "gap") %>%
  select(-comp_numeric) %>%
  spread(comp, surprisal) %>%
  mutate(filler_effect=what~that)

```

filled gap effect

```

fge = d_agg %>%

```

```

filter(region == "obj" & gap == "no-gap") %>%
select(-comp_numeric) %>%
spread(comp, surprisal) %>%
mutate(filler_effect=what~`that`)

```

RC-dependencies

```

df_rc = df_rc %>%
  mutate(region = if_else(word == "." | word == "<eos>" & region == "end", "EOS", region),
         region = if_else(region == "that" | region == "pp-add" |
                           region == "rc-head-obj" | region == "rp", "prefix", region),
         region = if_else(region == "2-layers" | region == "3-layers" |
                           region == "4-layers" | region == "5-layers" |
                           region == "2-that" | region == "3-that" |
                           region == "4-that" | region == "5-that", "layers", region),
         region = factor(region, levels = REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "layers"))

```

introduce another column which states whether there's "that" in emb or not

```

df_rc = df_rc %>%
  add_column(that_comp = "")

```

populate the column same way as with wh-dep

```

df_rc = df_rc %>%
  mutate(that_comp = if_else(layers == "1" | layers == "2" | layers == "3" |
                             layers == "4" | layers == "5", "no", that_comp),
         that_comp = if_else(layers == "1-comp" | layers == "2-comp" | layers == "3-comp" |
                             layers == "4-comp" | layers == "5-comp", "yes", that_comp))

```

now only numeric is left in layers

```

df_rc = df_rc %>%
  mutate(layers = gsub("-comp", "", layers))

```

Creating an aggregated data frame

```

d_agg_rc = df_rc %>%
  group_by(region, sent_index, comp, gap, layers, language, that_comp) %>%
  summarise(surprisal=sum(surprisal)) %>%
  ungroup() %>%
  mutate(comp_numeric=if_else(comp == "what", 0.5, -0.5),
         comp=factor(comp, levels=c("what", "that")),
         gap=factor(gap, levels=c("no-gap", "gap")),
         layers=factor(layers, levels=c("1", "2", "3", "4", "5")))

```

Plotting raw surprisal by region for Norwegian RC

```

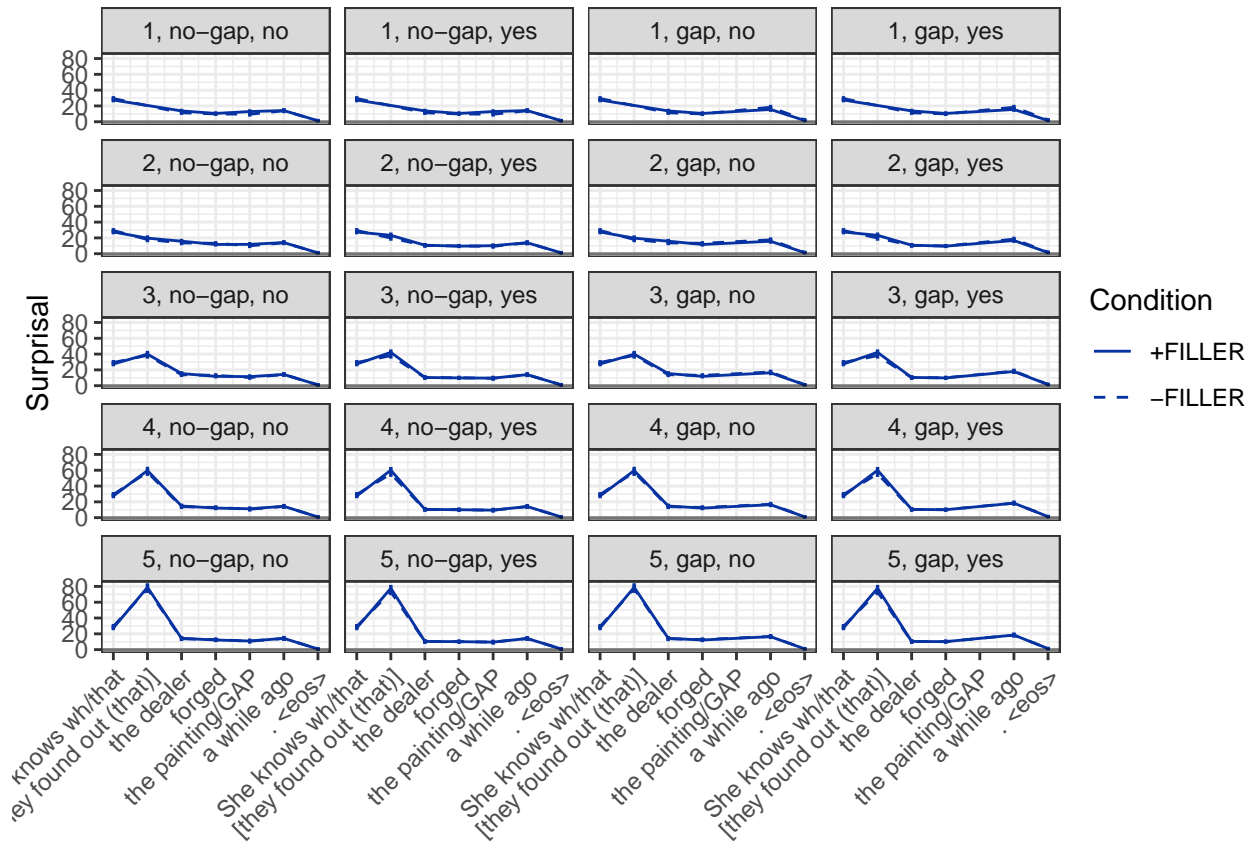
d_agg_rc %>%
  group_by(region, gap, comp, layers, that_comp) %>%
  summarise(m=mean(surprisal),
            s=std.error(surprisal),
            upper=m + 1.96*s,
            lower=m - 1.96*s) %>%
  ungroup() %>%
  mutate(region=as.numeric(region)) %>%
  ggplot(aes(x=region, y=m, ymax=upper, ymin=lower, linetype=comp)) +
  facet_wrap(layers~gap~that_comp, ncol=4,

```

```

    labeller = label_wrap_gen(multi_line=FALSE)) +
  geom_line(color = "#0732A2") + theme_bw() +
  geom_errorbar(linetype="solid", width=.1, color = "#0732A2") +
  scale_x_continuous(breaks=seq(1, length(REGION_ORDER)), labels=REGION_EXEMPLARS) +
  geom_hline(yintercept=0, color="black", alpha=0.5) + ylab("Surprisal") +
  theme(axis.text.x = element_text(angle=45, hjust=1),
        axis.title.x = element_blank(), legend.margin=margin(c(0,0,0,0))) +
  scale_linetype_manual(name = "Condition", labels = c("+FILLER", "-FILLER"),
                        values = c('solid', 'dashed'))

```



```

fname = sprintf("plots/unbound-plots/%s-unbound-regions-raw-no-rc.png", model_type)
ggsave(fname, width = 10, height = 10)

```

```

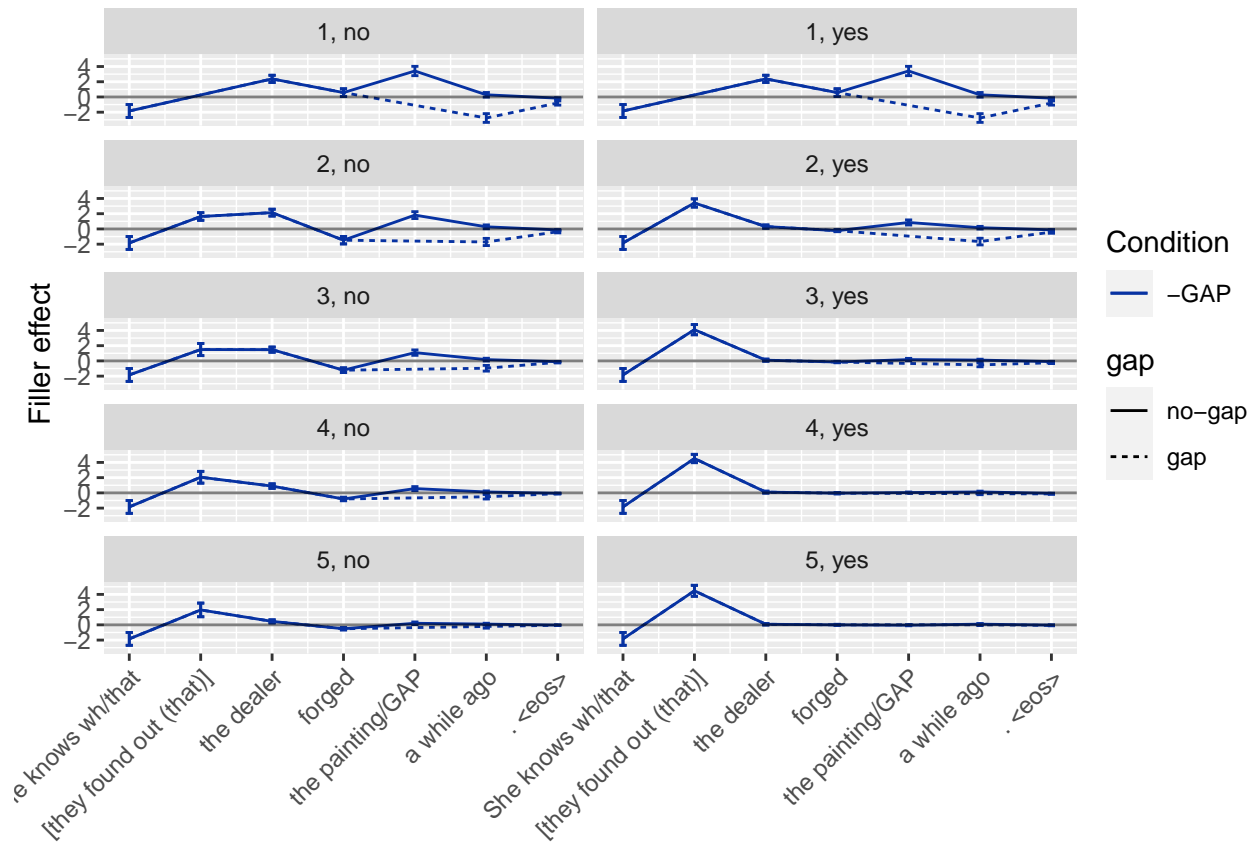
# Plotting filler effects (FGEs and UGEs) by region for RC-dep
d_agg_rc %>%
  select(-comp_numeric) %>%
  spread(comp, surprisal) %>%
  mutate(filler_effect=what~that) %>%
  group_by(language, region, gap, layers, that_comp) %>%
  summarise(m=mean(filler_effect),
            s=std.error(filler_effect),
            upper=m + 1.96*s,
            lower=m - 1.96*s) %>%
  ungroup() %>%
  mutate(region=as.numeric(region)) %>%
  ggplot(aes(x=region, y=m, ymax=upper, ymin=lower, linetype=gap, color=language)) +
  facet_wrap(layers~that_comp, ncol=2, labeller = label_wrap_gen(multi_line=FALSE)) +

```

```

geom_line() +
geom_errorbar(linetype="solid", width=.1) +
scale_x_continuous(breaks=seq(1, length(REGION_ORDER)), labels=REGION_EXEMPLARS) +
geom_hline(yintercept=0, color="black", alpha=0.5) +
theme(axis.text.x = element_text(angle=45, hjust=1),
      axis.title.x = element_blank(), legend.margin=margin(c(0,0,0,0))) +
ylab("Filler effect") +
scale_colour_discrete(name="Condition", labels=c("-GAP", "+GAP"))

```



```

fname = sprintf("plots/unbound-plots/%s-unbound-regions-fe-no-rc.png", model_type)
ggsave(fname, width = 10, height = 10)

```

Calculating filler effects in RoIs (region of the filled NP/post-gap region)

unlicensed gap effect

```

uge_rc = d_agg_rc %>%
  filter(region == "end" & gap == "gap") %>%
  select(-comp_numeric) %>%
  spread(comp, surprisal) %>%
  mutate(filler_effect=what-`that`)

```

filled gap effect

```

fge_rc = d_agg_rc %>%
  filter(region == "obj" & gap == "no-gap") %>%
  select(-comp_numeric) %>%
  spread(comp, surprisal) %>%
  mutate(filler_effect=what-`that`)

```



```

uge_rc$language[uge_rc$language == "Norwegian"] <- "Norwegian - RC"
fge_rc$language[fge_rc$language == "Norwegian"] <- "Norwegian - RC"

# Creating a data frame with both effects for all lang-dep combinations
d_filler_effect = Reduce(function(x, y) merge(x, y, all=TRUE),
                          list(uge, fge, uge_rc, fge_rc))
d_filler_effect$language[d_filler_effect$language == "English"] <- "English - Wh"
d_filler_effect$language[d_filler_effect$language == "Norwegian"] <- "Norwegian - Wh"

d_filler_effect = d_filler_effect %>%
  select(-region) %>%
  #Error Calculation
  #Across condition mean response
  group_by(language, sent_index, that_comp) %>%
  mutate(across_condition_mean = mean(filler_effect)) %>%
  ungroup() %>%
  #Item mean-extracted-response measure
  mutate(item_mean = filler_effect - across_condition_mean) %>%
  #Across item item-mean error
  group_by(language, layers, gap, that_comp) %>%
  mutate(err = std.error(item_mean, na.rm=T)) %>%
  ungroup() %>%
  select(-item_mean, -across_condition_mean)

# Add model column and save the aggregated data
d_filler_effect['model'] = toupper(model_type)
dfname = sprintf("../data/results/%s/unbound_%s_agg.csv", model_type, model_type)
write.csv(d_filler_effect, dfname, row.names=FALSE)

# Plotting the effects in the regions of interest
customs_two <- c("#FF5B97", "#096FCA")

gap.labs <- c("-GAP (Filled gap effect)", "+GAP (Unlicensed gap effect)")
names(gap.labs) <- c("no-gap", "gap")

that_comp.labs <- c("Without 'that'", "With 'that'")
names(that_comp.labs) <- c("no", "yes")

d_filler_effect$language <- factor(d_filler_effect$language,
  levels = c('Norwegian - RC', 'Norwegian - Wh', 'English - Wh'), ordered = TRUE)

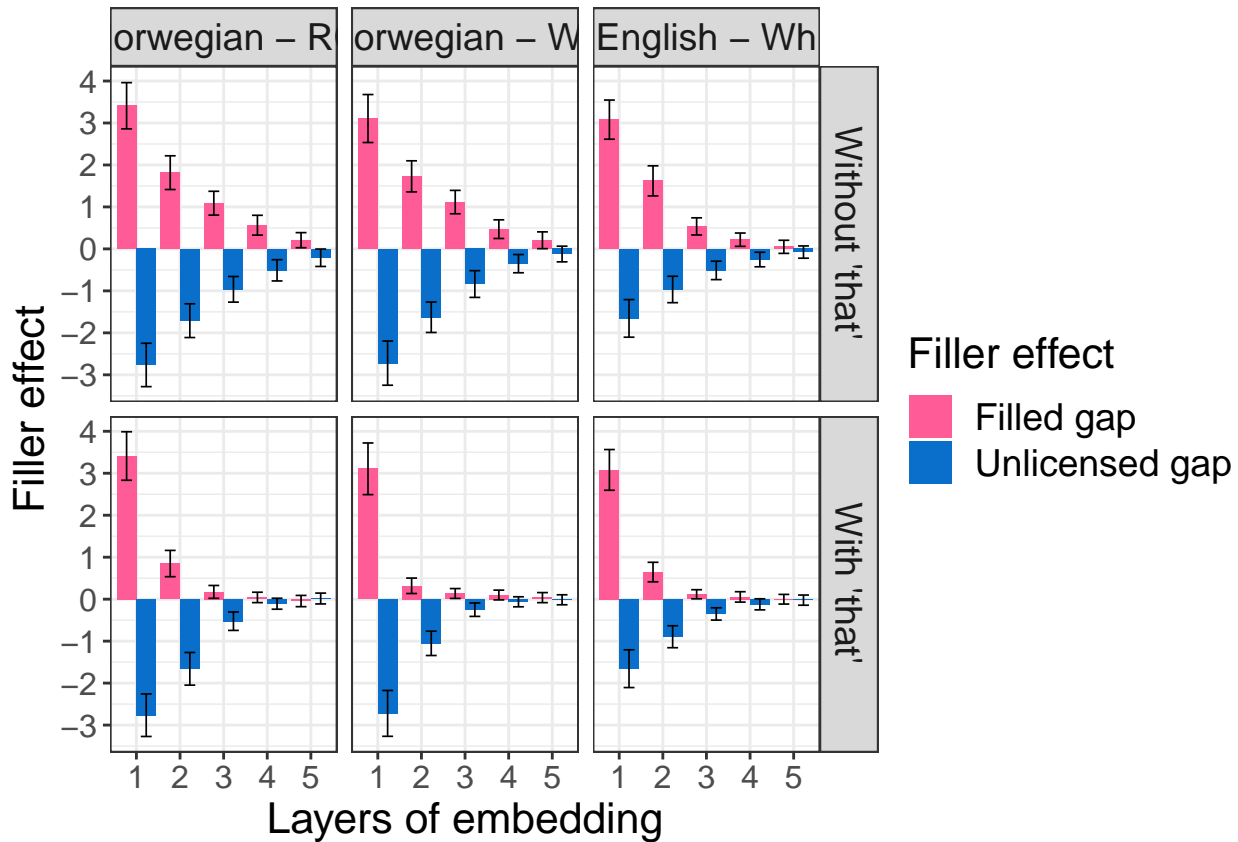
d_filler_effect %>%
  group_by(language, layers, gap, that_comp) %>%
  summarise(m=mean(filler_effect),
            s=mean(err),
            upper=m+1.96*s,
            lower=m-1.96*s) %>%
  ungroup() %>%
  ggplot(aes(x=layers, y=m, ymin=lower, ymax=upper, fill=gap)) +
  geom_bar(stat="identity", position="dodge") +
  geom_errorbar(color="black", width=.5, position=position_dodge(width=.9), linewidth=0.3) +
  theme_bw() +
  ylab("Filler effect") +
  facet_grid(that_comp~language, labeller = labeller(gap = gap.labs, that_comp = that_comp.labs)) +
  xlab("Layers of embedding") +

```

```

theme(axis.text = element_text(size = 12),
      strip.text = element_text(size = 14),
      legend.text = element_text(size = 14),
      legend.title = element_text(size = 16),
      axis.title = element_text(size = 16)) +
scale_y_continuous(breaks = scales::pretty_breaks(n = 8)) +
theme(legend.position = "right", legend.margin=margin(c(0,0,0,0))) +
scale_fill_manual(values = customs_two, name = "Filler effect",
                  labels = c("Filled gap", "Unlicensed gap"))

```



```

fname = sprintf("plots/unbound-plots/%s-unbound-fe-all.png", model_type)
ggsave(fname, width = 10, height = 6)

```

Run all of the code above for both models (lstm, gpt2) before running the code below.

```

# Read in the aggregated data for both models
lstm = read.csv("../data/results/lstm/unbound_lstm_agg.csv", fileEncoding = "UTF-8-BOM")
gpt2 = read.csv("../data/results/gpt2/unbound_gpt2_agg.csv", fileEncoding = "UTF-8-BOM")
d = rbind(lstm, gpt2)

# Plotting the effects in the regions of interest for both models
d$model = factor(d$model, levels = c("LSTM", "GPT2"), ordered = TRUE)
d$language = factor(d$language, levels = c('Norwegian - RC', 'Norwegian - Wh',
                                           'English - Wh'), ordered = TRUE)
d$gap = factor(d$gap, levels = c("no-gap", "gap"), ordered = TRUE)

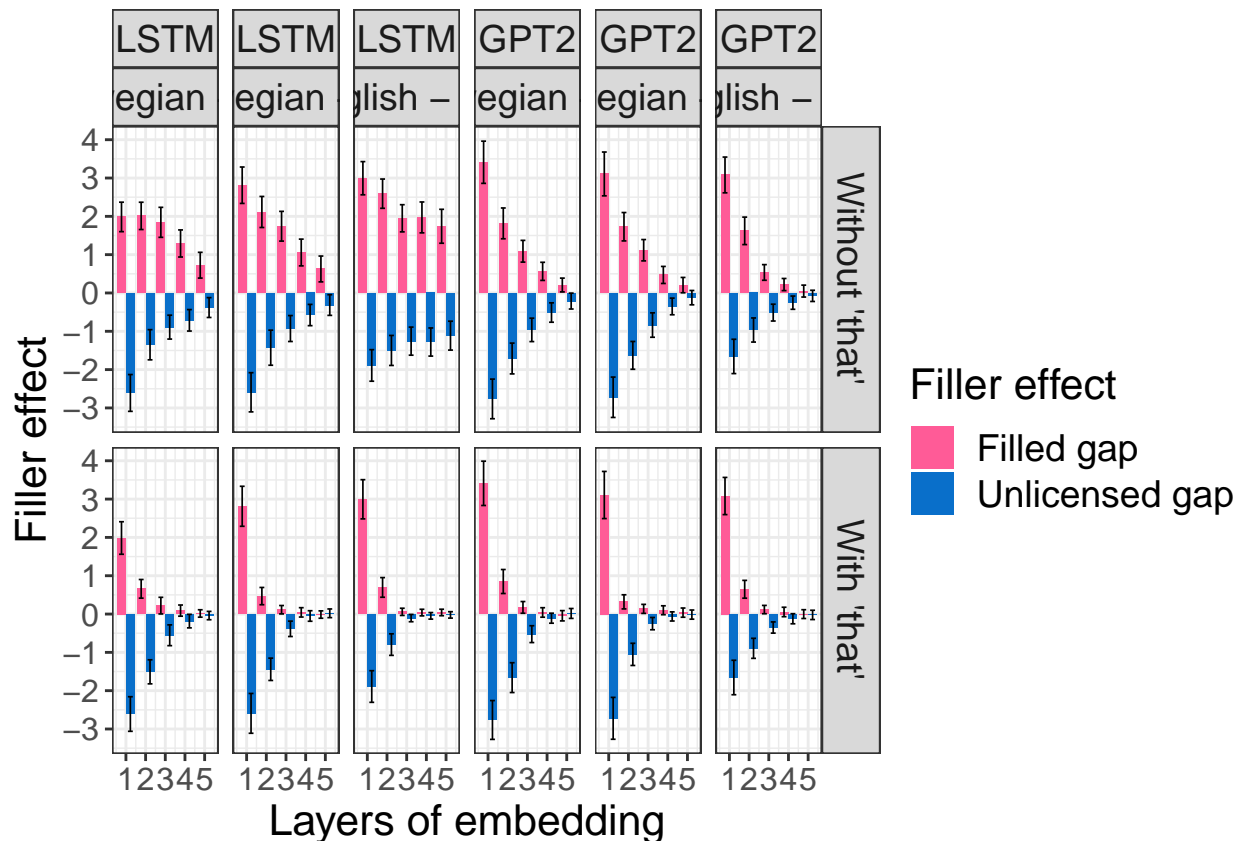
d %>%
  group_by(model, language, layers, gap, that_comp) %>%

```

```

summarise(m=mean(filler_effect),
          s=mean(err),
          upper=m+1.96*s,
          lower=m-1.96*s) %>%
  ungroup() %>%
  ggplot(aes(x=layers, y=m, ymin=lower, ymax=upper, fill=gap)) +
  geom_bar(stat="identity", position="dodge") +
  geom_errorbar(color="black", width=.5, position=position_dodge(width=.9), linewidth=0.3) +
  theme_bw() +
  ylab("Filler effect") +
  facet_grid(that_comp~model + language, labeller = labeller(gap = gap.labs, that_comp = that_comp.labs)) +
  xlab("Layers of embedding") +
  theme(axis.text = element_text(size = 12),
        strip.text = element_text(size = 14),
        legend.text = element_text(size = 14),
        legend.title = element_text(size = 16),
        axis.title = element_text(size = 16)) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 8)) +
  theme(legend.position = "right", legend.margin=margin(c(0,0,0,0))) +
  scale_fill_manual(values = customs_two, name = "Filler effect",
                   labels = c("Filled gap", "Unlicensed gap"))

```



```

ggsave("plots/unbound-plots/unbound-fe-two-models.png", width = 12, height = 8)

```

```

p1 = d %>%
  filter(model == "LSTM") %>%
  group_by(language, layers, gap, that_comp) %>%

```

```

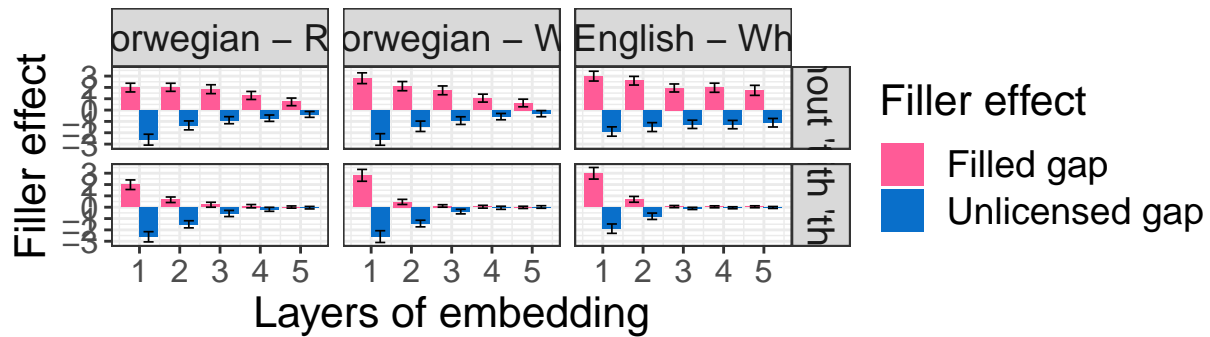
    summarise(m=mean(filler_effect),
              s=mean(err),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
ggplot(aes(x=layers, y=m, ymin=lower, ymax=upper, fill=gap)) +
  geom_bar(stat="identity", position="dodge") +
  geom_errorbar(color="black", width=.5, position=position_dodge(width=.9), linewidth=0.3) +
  theme_bw() +
  ylab("Filler effect") +
  facet_grid(that_comp~language, labeller = labeller(gap = gap.labs,that_comp = that_comp.labs))+
  xlab("Layers of embedding") +
  theme(axis.text = element_text(size = 12),
        strip.text = element_text(size = 14),
        legend.text = element_text(size = 14),
        legend.title = element_text(size = 16),
        axis.title = element_text(size = 16),
        plot.title = element_text(size = 18)) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 8)) +
  theme(legend.position = "right", legend.margin=margin(c(0,0,0,0))) +
  scale_fill_manual(values = customs_two, name = "Filler effect",
                    labels = c("Filled gap", "Unlicensed gap")) +
  ggtitle("LSTM models")

p2 = d %>%
  filter(model == "GPT2") %>%
  group_by(language, layers, gap, that_comp) %>%
    summarise(m=mean(filler_effect),
              s=mean(err),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
ggplot(aes(x=layers, y=m, ymin=lower, ymax=upper, fill=gap)) +
  geom_bar(stat="identity", position="dodge") +
  geom_errorbar(color="black", width=.5, position=position_dodge(width=.9), linewidth=0.3) +
  theme_bw() +
  ylab("Filler effect") +
  facet_grid(that_comp~language, labeller = labeller(gap = gap.labs,that_comp = that_comp.labs))+
  xlab("Layers of embedding") +
  theme(axis.text = element_text(size = 12),
        strip.text = element_text(size = 14),
        legend.text = element_text(size = 14),
        legend.title = element_text(size = 16),
        axis.title = element_text(size = 16),
        plot.title = element_text(size = 18)) +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 8)) +
  theme(legend.position = "right", legend.margin=margin(c(0,0,0,0))) +
  scale_fill_manual(values = customs_two, name = "Filler effect",
                    labels = c("Filled gap", "Unlicensed gap")) +
  ggtitle("GPT2 models")

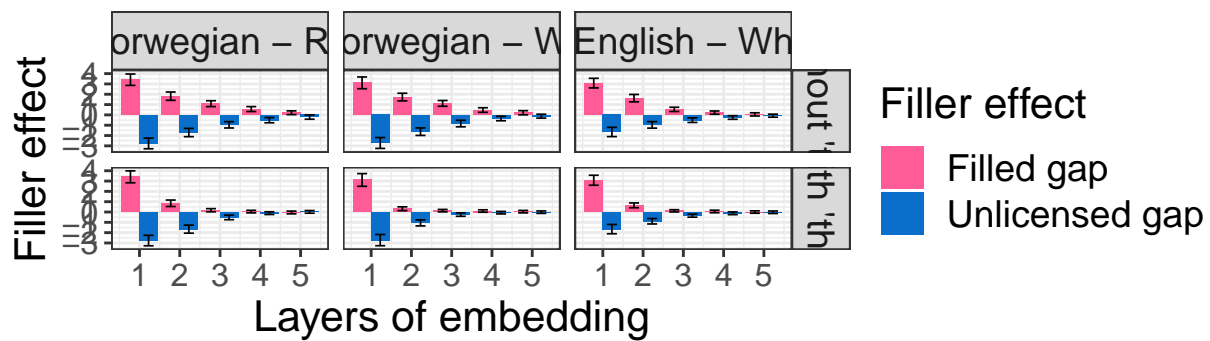
```

p1 / p2

LSTM models



GPT2 models



```
ggsave("plots/unbound-plots/unbound-two-models.png", width = 10, height = 10)
ggsave("plots/unbound-plots/unbound-two-models.pdf", width = 10, height = 10)
```

Statistical analysis with difference coding

Defining contrasts

```
# layers - factor with 5 levels
# creating a matrix for backward difference coding
# filling it in by row
cells = c(-(4/5), -(3/5), -(2/5), -(1/5),
          (1/5), -(3/5), -(2/5), -(1/5),
          (1/5), (2/5), -(2/5), -(1/5),
          (1/5), (2/5), (3/5), -(1/5),
          (1/5), (2/5), (3/5), (4/5))
bdc_contrasts = matrix(cells, nrow=5, ncol=4)
print(bdc_contrasts)
```

```
##      [,1] [,2] [,3] [,4]
## [1,] -0.8 -0.6 -0.4 -0.2
## [2,] -0.6 -0.4 -0.2  0.2
## [3,] -0.4 -0.2  0.2  0.4
## [4,] -0.2  0.2  0.4  0.6
## [5,]  0.2  0.4  0.6  0.8
```

```
contrasts(d_filler_effect$layers) = bdc_contrasts
```

Without complementizer that in the embedding

```
regressions = list()

models = c("Norwegian - RC", "Norwegian - Wh", "English - Wh")

for (i in models) {

  fge_model = d_filler_effect %>%
    filter(gap == "no-gap" & language == i & that_comp == "no") %>%
    lmer(filler_effect ~ layers + (1 | sent_index), data=.)
  uge_model = d_filler_effect %>%
    filter(gap == "gap" & language == i & that_comp == "no") %>%
    lmer(filler_effect ~ layers + (1 | sent_index), data=.)
  # first char and last two chars: NRC, NWh, EWh
  model_name = paste0(substr(i, 1, 1), substr(i, nchar(i)-1, nchar(i)))
  regressions[[paste0(model_name, "_fge")]] <- fge_model
  regressions[[paste0(model_name, "_uge")]] <- uge_model
}

# Minimal pretty table to be saved in Latex
latex_table_without_that_contr = modelsummary(regressions, output = "gt",
  stars = TRUE, gof_omit = ".*",
  estimate = "{estimate}{stars}", statistic = NULL, fmt = 1) %>%
  cols_label(
    NRC_fge = "FGE",
    NRC_uge = "UGE",
    NWh_fge = "FGE",
    NWh_uge = "UGE",
    EWh_fge = "FGE",
    EWh_uge = "UGE",
  ) %>%
  # column labels
  tab_spanner(label = 'Norwegian - RC', columns = 2:3) %>%
  tab_spanner(label = 'Norwegian - Wh', columns = 4:5) %>%
  tab_spanner(label = 'English - Wh', columns = 6:7)

# Table with more info to be saved in html
html_table_without_that_contr = modelsummary(regressions, output = "gt",
  stars = TRUE, gof_omit = ".*",
  estimate = "{estimate}{stars} ({std.error})",
  statistic = "t = {statistic}", fmt = 1) %>%
  cols_label(
    NRC_fge = "FGE",
    NRC_uge = "UGE",
    NWh_fge = "FGE",
    NWh_uge = "UGE",
    EWh_fge = "FGE",
    EWh_uge = "UGE",
  ) %>%
  # column labels
  tab_spanner(label = 'Norwegian - RC', columns = 2:3) %>%
```

```

tab_spanner(label = 'Norwegian - Wh', columns = 4:5) %>%
tab_spanner(label = 'English - Wh', columns = 6:7)

```

With complementizer

```

regressions = list()

models = c("Norwegian - RC", "Norwegian - Wh", "English - Wh")

for (i in models) {

  fge_model = d_filler_effect %>%
    filter(gap == "no-gap" & language == i & that_comp == "yes") %>%
    lmer(filler_effect ~ layers + (1 |sent_index), data=.)
  uge_model = d_filler_effect %>%
    filter(gap == "gap" & language == i & that_comp == "yes") %>%
    lmer(filler_effect ~ layers + (1 |sent_index), data=.)
  # first char and last two chars: NRC, NWh, EWh
  model_name = paste0(substr(i, 1, 1), substr(i, nchar(i)-1, nchar(i)))
  regressions[[paste0(model_name, "_fge")]] <- fge_model
  regressions[[paste0(model_name, "_uge")]] <- uge_model
}

# Minimal pretty table to be saved in Latex
latex_table_with_that_contr = modelsummary(regressions, output = "gt", stars = TRUE, gof_omit = ".*",
  estimate = "{estimate}{stars}", statistic = NULL, fmt = 1) %>%
  cols_label(
    NRC_fge = "FGE",
    NRC_uge = "UGE",
    NWh_fge = "FGE",
    NWh_uge = "UGE",
    EWh_fge = "FGE",
    EWh_uge = "UGE",
  ) %>%
  # column labels
  tab_spanner(label = 'Norwegian - RC', columns = 2:3) %>%
  tab_spanner(label = 'Norwegian - Wh', columns = 4:5) %>%
  tab_spanner(label = 'English - Wh', columns = 6:7)

# Table with more info to be saved in html
html_table_with_that_contr = modelsummary(regressions, output = "gt", stars = TRUE, gof_omit = ".*",
  estimate = "{estimate}{stars} ({std.error})",
  statistic = "t = {statistic}", fmt = 1) %>%
  cols_label(
    NRC_fge = "FGE",
    NRC_uge = "UGE",
    NWh_fge = "FGE",
    NWh_uge = "UGE",
    EWh_fge = "FGE",
    EWh_uge = "UGE",
  ) %>%
  # column labels
  tab_spanner(label = 'Norwegian - RC', columns = 2:3) %>%
  tab_spanner(label = 'Norwegian - Wh', columns = 4:5) %>%

```

```
tab_spanner(label = 'English - Wh', columns = 6:7)
```

Saving it all

```
stats_fname = sprintf("stats/unbound-stats/unbound-%s-", model_type)
html_table_without_that_contr |> gtsave(paste(stats_fname, "without-that.html", sep = ""))
latex_table_without_that_contr |> gtsave(paste(stats_fname, "without-that.tex", sep = ""))
html_table_with_that_contr |> gtsave(paste(stats_fname, "with-that.html", sep = ""))
latex_table_with_that_contr |> gtsave(paste(stats_fname, "with-that.tex", sep = ""))
```