

# Analysis of Subject island results

Anastasia Kobzeva

## Data pre-processing and plotting

### Choose model type for analysis

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

filename_wh = sprintf("../data/results/%s/subject_wh_result.csv", model_type) # Norwegian wh
filename_rc = sprintf("../data/results/%s/subject_rc_result.csv", model_type) # Norwegian RC
filename_en = sprintf("../data/results/%s/subject_wh_en_result.csv", model_type) # English wh

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
```

### 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
- island = island condition
- control = control condition

### Loading in data and analysis functions, defining paths for plots

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

df = df %>%
  mutate(region = if_else(word == "." | word == "<eos>" & region == "end", "EOS", region))

control = df[endsWith(df$condition, "_subj"),] # subj control
island = df[endsWith(df$condition, "_pp-subj"),] # island
```

```

embed = df[endsWith(df$condition, "_subj-emb"),] # emb subj control

wh_control = control %>%
  filter(dependency == "Wh" & language == "Norwegian")
rc_control = control %>%
  filter(dependency == "RC" & language == "Norwegian")
wh_island = island %>%
  filter(dependency == "Wh" & language == "Norwegian")
rc_island = island %>%
  filter(dependency == "RC" & language == "Norwegian")
en_control = control %>%
  filter(language == "English")
en_island = island %>%
  filter(language == "English")
wh_embed = embed %>%
  filter(dependency == "Wh" & language == "Norwegian")
rc_embed = embed %>%
  filter(dependency == "RC" & language == "Norwegian")
en_embed = embed %>%
  filter(language == "English")

```

## Control, wh-dependency

Regions for subject position:

```

REGION_ORDER = c("prefix", "comp", "subj", "verb", "obj", "end", "EOS")
REGION_EXEMPLARS = c("The newspaper reported", "{th/wh}at", "the agreement/GAP", "will strengthen",
  "political interaction", "after the elections", ". <eos>")

# Changing the data according to the ROIs
wh_control = wh_control %>%
  mutate(region = if_else(region == "that" | region == "wh-subj" | region == "rp", "comp", region),
    region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))

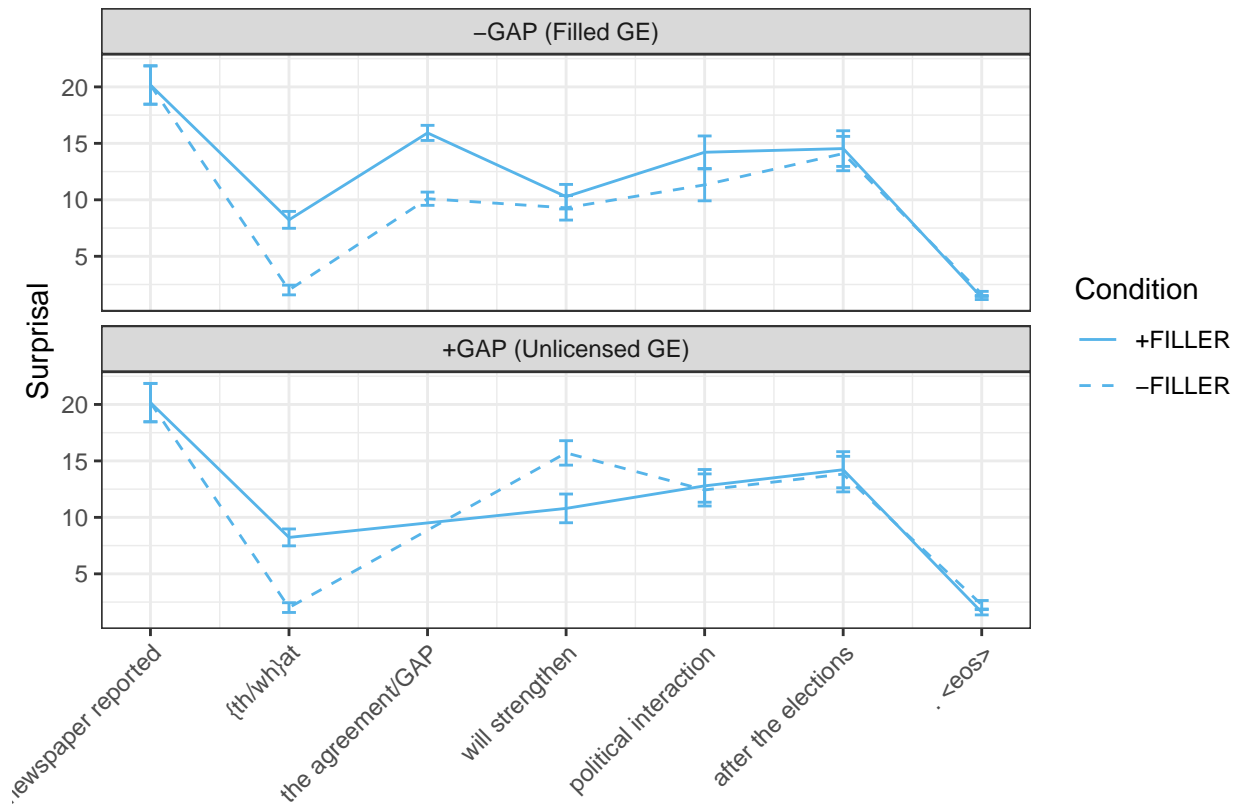
```

Aggregating the data and plotting raw surprisal values:

```

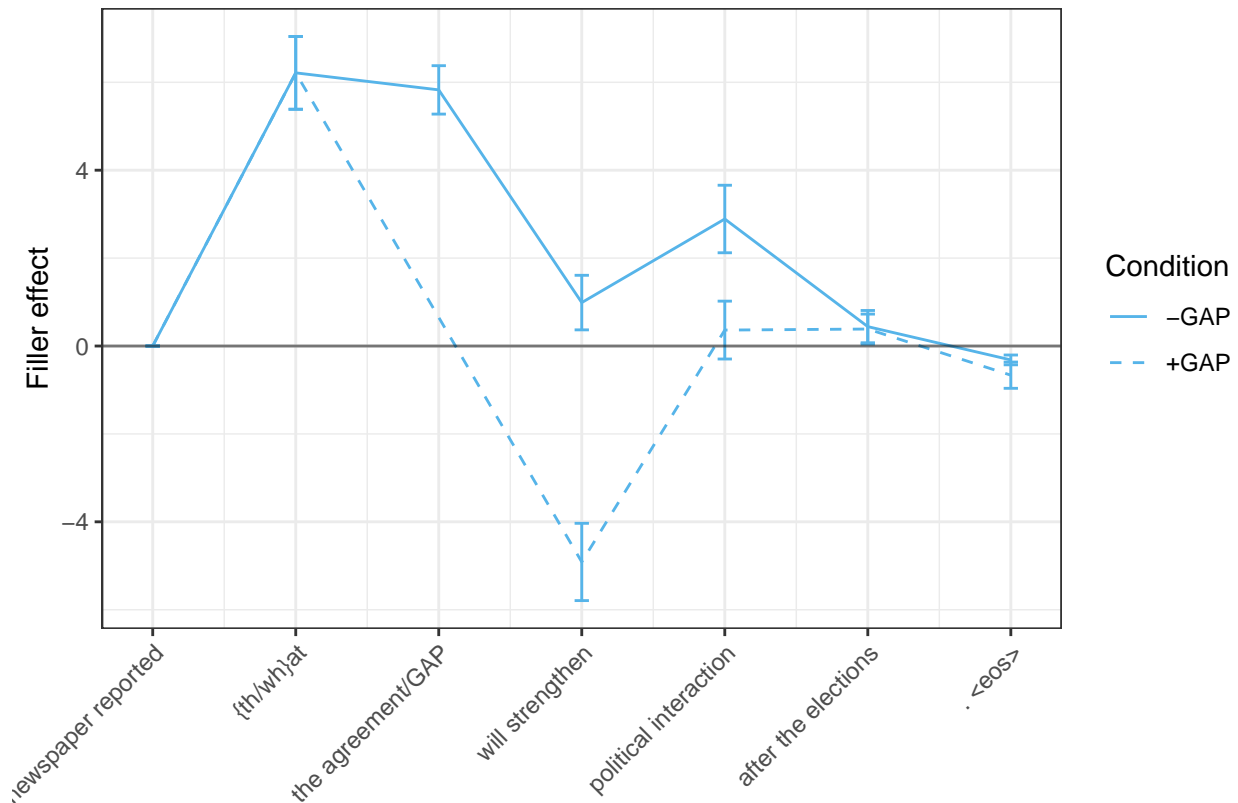
wh_control = region.surprisal(data = wh_control)
raw.surprisal.plot(data = wh_control, name = "no-wh-dep-control", path = regions_raw,
  regions = REGION_EXEMPLARS, color_choice = c("#56B4E9"))

```



Calculating filler effects and plotting them by region:

```
wh_control_fe = fe.calculation(data = wh_control)
fe.regions.plot(data = wh_control_fe, name = "no-wh-dep-control", path = regions_fe,
                regions = REGION_EXEMPLARS, color_choice = c("#56B4E9"))
```



```
wh_control_fe = wh_control_fe %>%
  filter(region == "subj" & gap == "no-gap" | region == "verb" & gap == "gap")

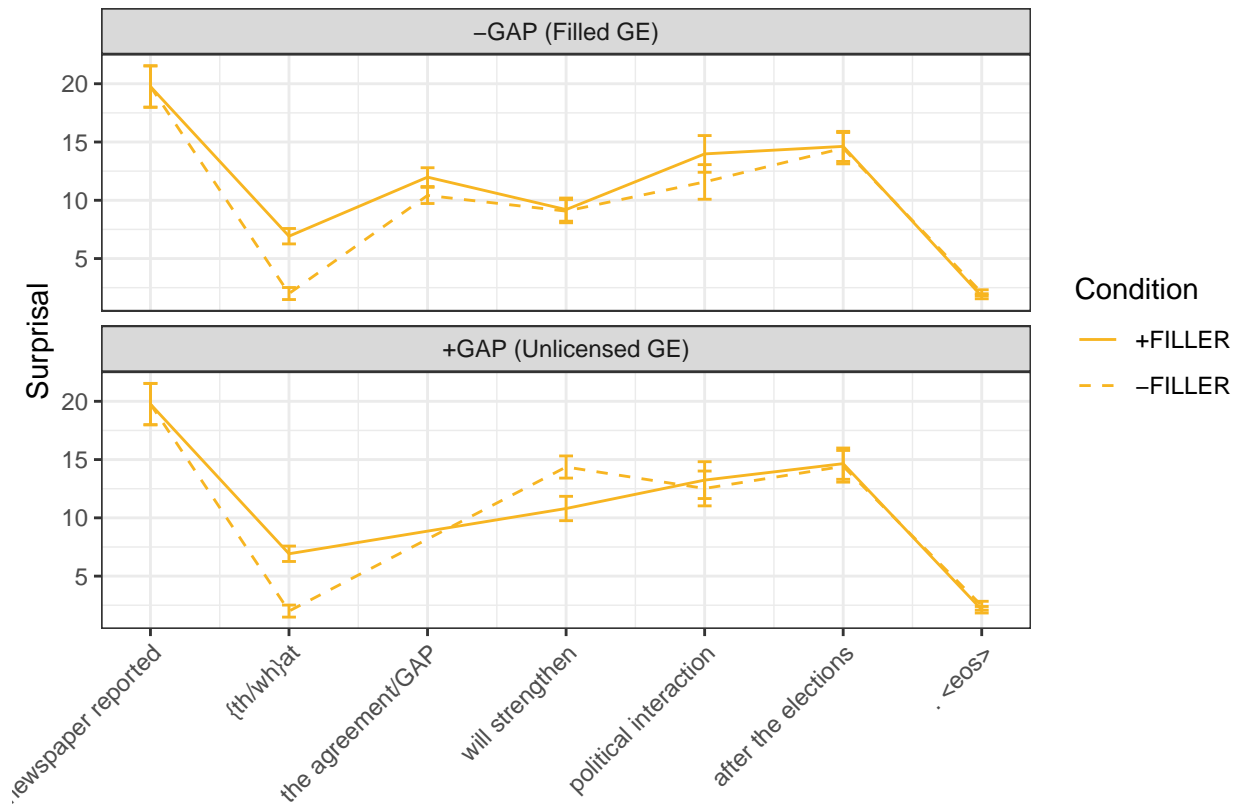
wh_control_fe_roi = fe.roi.stats(wh_control_fe)
wh_control_fe_roi$dependency = "Wh"
wh_control_fe_roi$language = "Norwegian"
```

English control:

```
# Changing the data according to the ROIs
en_control = en_control %>%
  mutate(region = if_else(region == "that" | region == "wh-subj", "comp", region),
         region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

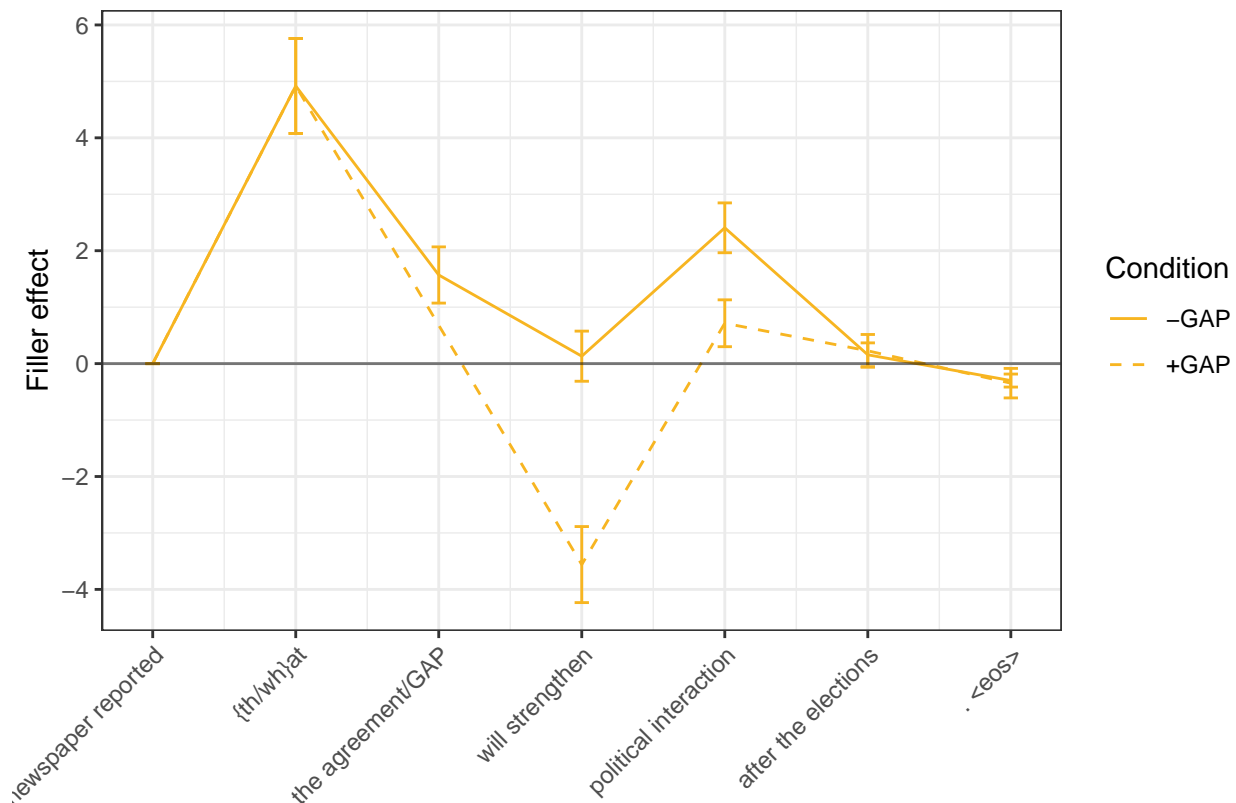
Aggregating the data and plotting raw surprisal values:

```
en_control = region.surprisal(data = en_control)
raw.surprisal.plot(data = en_control, name = "en-wh-dep-control", path = regions_raw,
                   regions = REGION_EXEMPLARS, color_choice = c("#F7B521"))
```



Calculating filler effects and plotting them by region:

```
en_control_fe = fe.calculation(data = en_control)
fe.regions.plot(data = en_control_fe, name = "en-wh-dep-control", path = regions_fe,
                regions = REGION_EXEMPLARS, color_choice = c("#F7B521"))
```



```
en_control_fe = en_control_fe %>%
  filter(region == "subj" & gap == "no-gap" | region == "verb" & gap == "gap")

en_control_fe_roi = fe.roi.stats(en_control_fe)
en_control_fe_roi$dependency = "Wh"
en_control_fe_roi$language = "English"
```

## Island, wh-dependency

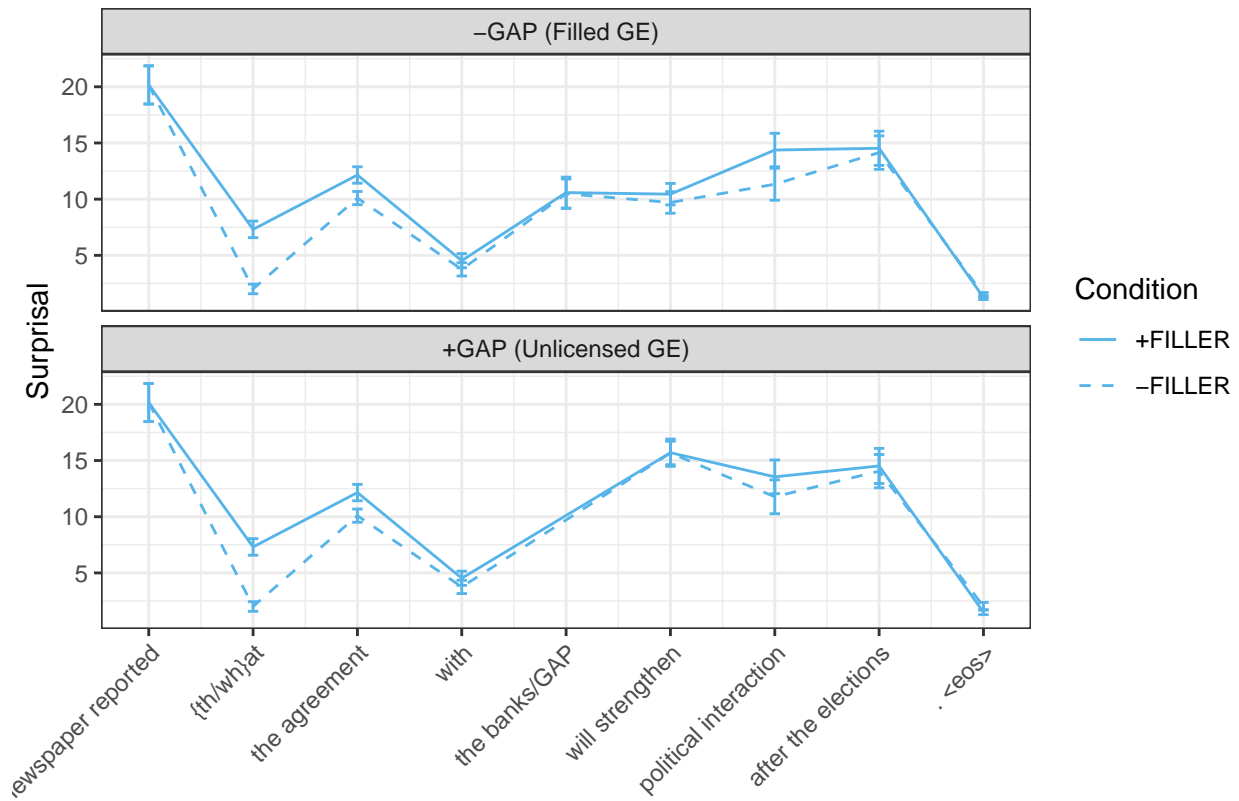
New regions with the PP:

```
REGION_ORDER = c("prefix", "comp", "subj", "prep", "obl", "verb", "obj", "end", "EOS")
REGION_EXEMPLARS = c("The newspaper reported", "{th/wh}at", "the agreement",
  "with", "the banks/GAP", "will strengthen",
  "political interaction", "after the elections", ". <eos>")
```

```
# Changing the data according to the ROIs
wh_island = wh_island %>%
  mutate(region = if_else(region == "that" | region == "wh-obl", "comp", region),
    region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

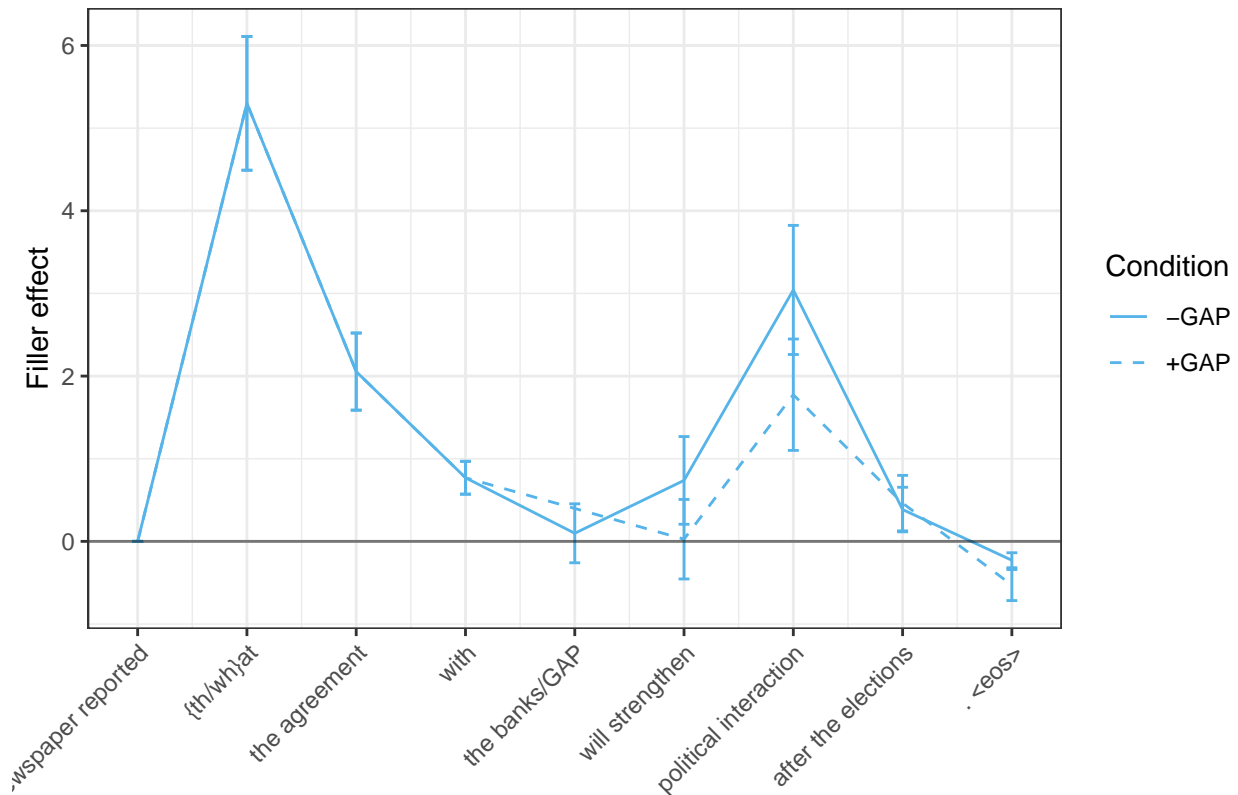
Aggregating the data and plotting raw surprisal values:

```
wh_island = region.surprisal(data = wh_island)
raw.surprisal.plot(data = wh_island, name = "no-wh-dep-island", path = regions_raw,
  regions = REGION_EXEMPLARS, color_choice = c("#56B4E9"))
```



Calculating filler effects and plotting them by region:

```
wh_island_fe = fe.calculation(data = wh_island)
fe.regions.plot(data = wh_island_fe, name = "no-wh-dep-island", path = regions_fe,
                regions = REGION_EXEMPLARS, color_choice = c("#56B4E9"))
```



```
wh_island_fe = wh_island_fe %>%
  filter(region == "obl" & gap == "no-gap" | region == "verb" & gap == "gap")

wh_island_fe_roi = fe.roi.stats(wh_island_fe)
wh_island_fe_roi$dependency = "Wh"
wh_island_fe_roi$language = "Norwegian"
```

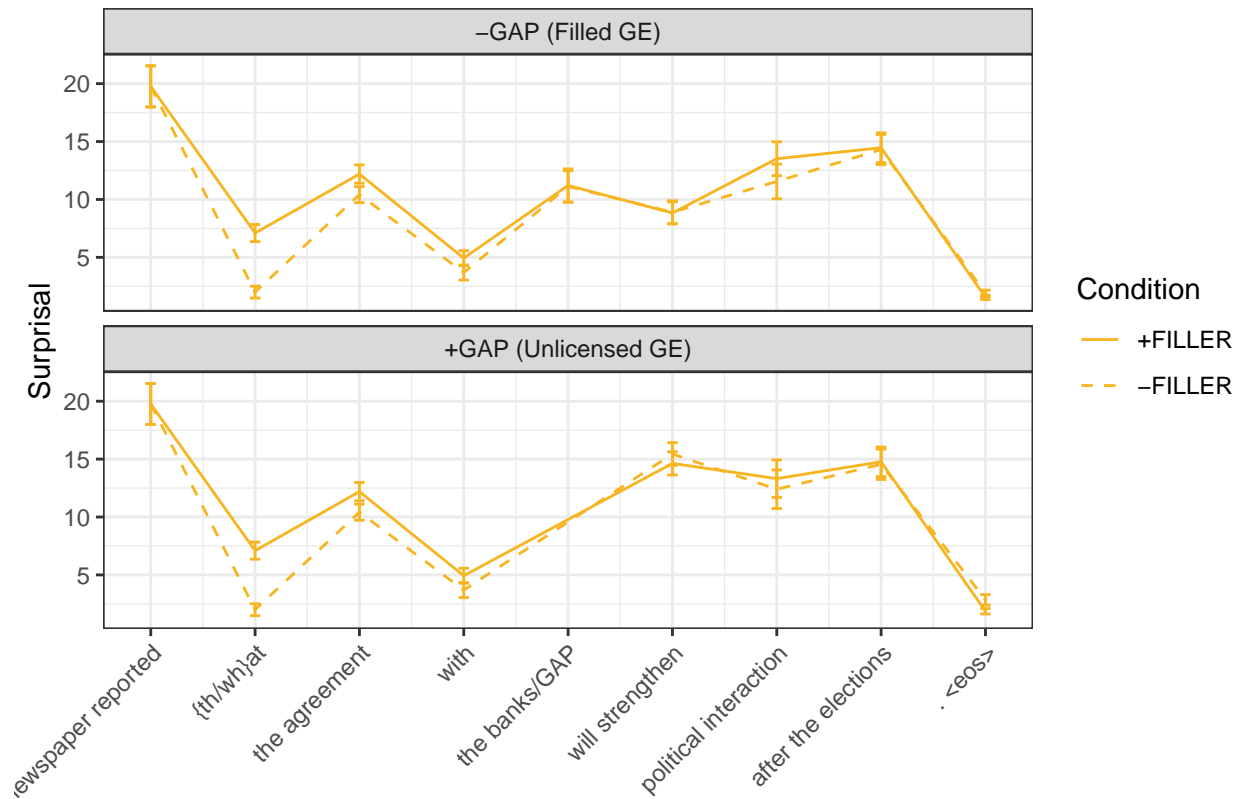
English island condition:

```
# Changing the data according to the ROIs
en_island = en_island %>%
  mutate(region = if_else(region == "that" | region == "wh-obl", "comp", region),
         region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

Aggregating the data and plotting raw surprisal values:

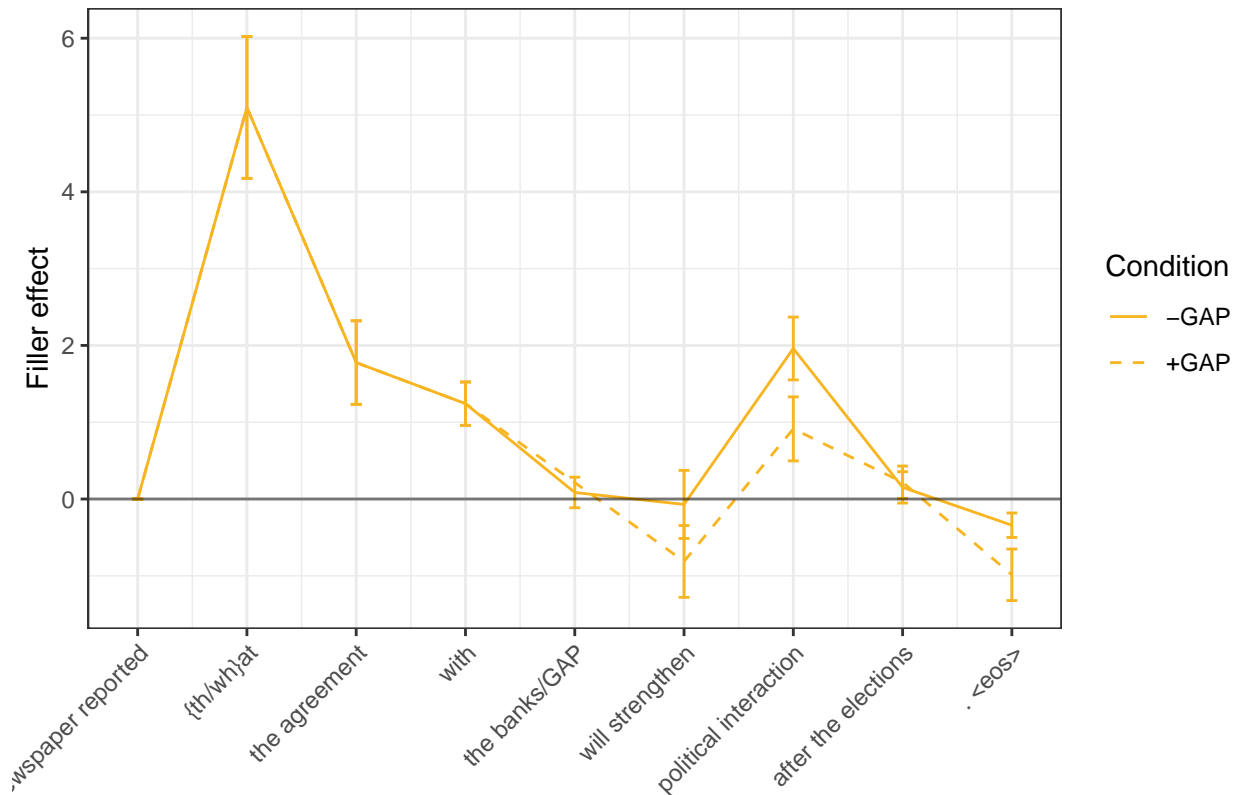
```
en_island = region.surprisal(data = en_island)
raw.surprisal.plot(data = en_island, name = "en-wh-dep-island", path = regions_raw,
                   regions = REGION_EXEMPLARS, color_choice = c("#F7B521"))
```





Calculating filler effects and plotting them by region:

```
en_island_fe = fe.calculation(data = en_island)
fe.regions.plot(data = en_island_fe, name = "en-wh-dep-island", path = regions_fe,
                regions = REGION_EXEMPLARS, color_choice = c("#F7B521"))
```



```
en_island_fe = en_island_fe %>%
  filter(region == "obl" & gap == "no-gap" | region == "verb" & gap == "gap")

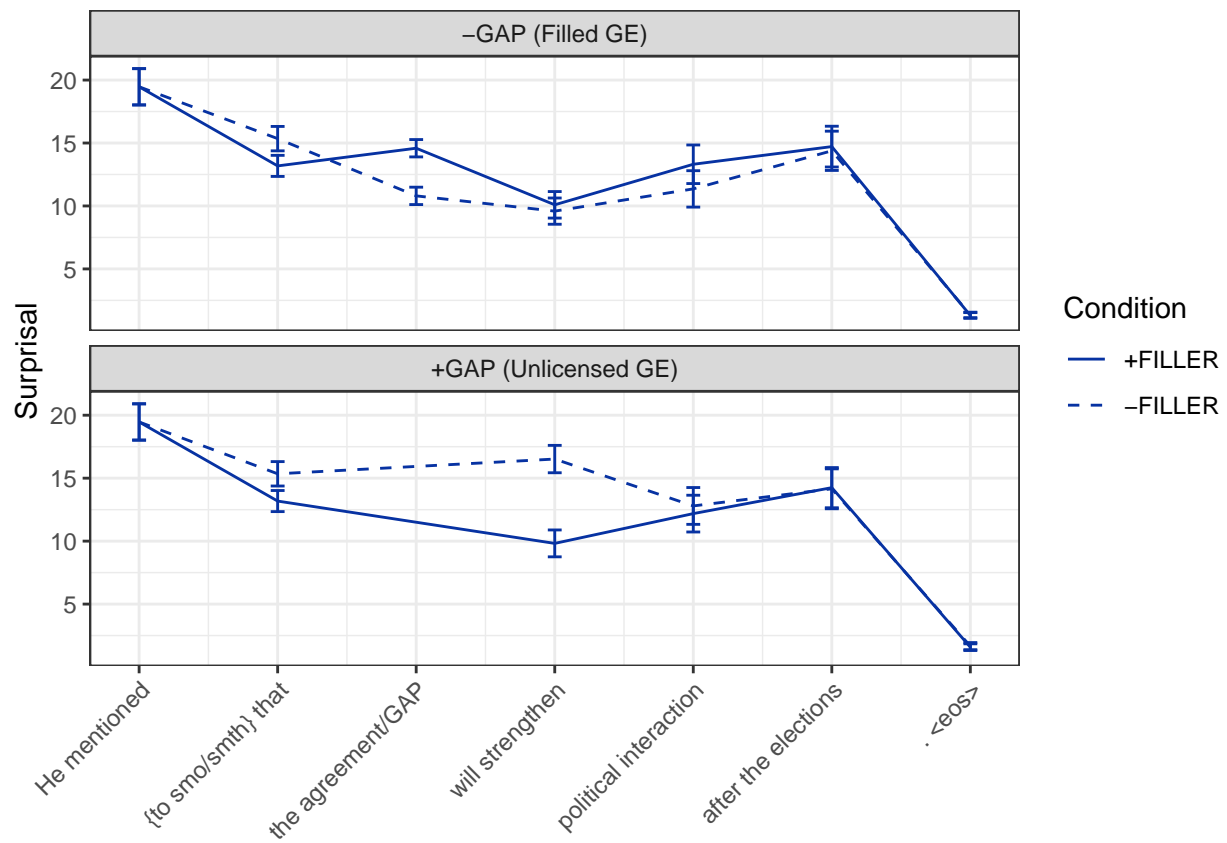
en_island_fe_roi = fe.roi.stats(en_island_fe)
en_island_fe_roi$dependency = "Wh"
en_island_fe_roi$language = "English"
```

### Control, RC-dependency

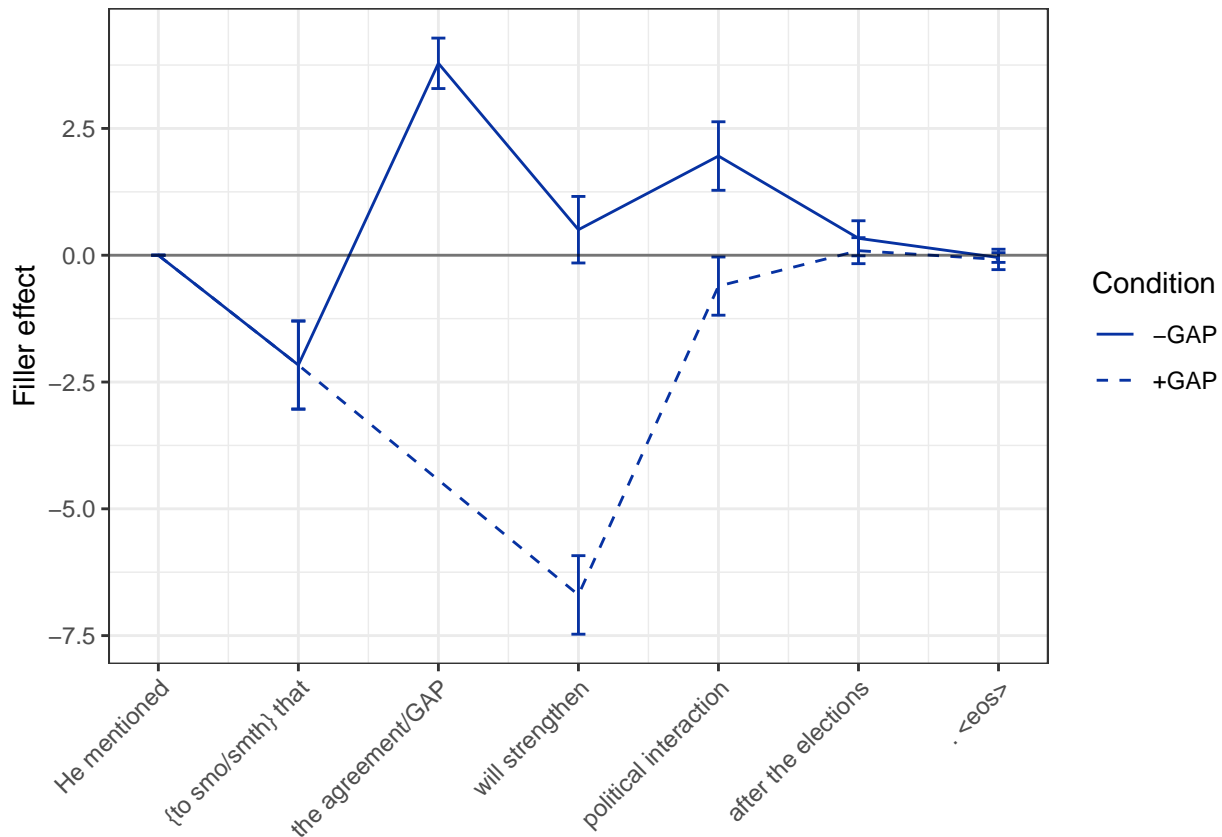
```
REGION_ORDER = c("prefix", "comp", "subj", "verb", "obj", "end", "EOS")
REGION_EXEMPLARS = c("He mentioned", "{to smth/somth} that", "the agreement/GAP", "will strengthen",
  "political interaction", "after the elections", ". <eos>")

rc_control = rc_control %>%
  mutate(region = if_else(region == "pp-add" | region == "that" | region == "rp"
    | region == "pr-subj" | region == "about", "comp", region),
    region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))

rc_control = region.surprisal(data = rc_control)
raw.surprisal.plot(data = rc_control, name = "no-rc-dep-control", path = regions_raw,
  regions = REGION_EXEMPLARS, color_choice = c("#0732A2"))
```



```
rc_control_fe = fe.calculation(data = rc_control)
fe.regions.plot(data = rc_control_fe, name = "no-rc-dep-control", path = regions_fe,
                regions = REGION_EXEMPLARS, color_choice = c("#0732A2"))
```



```
rc_control_fe = rc_control_fe %>%
  filter(region == "subj" & gap == "no-gap" | region == "verb" & gap == "gap")

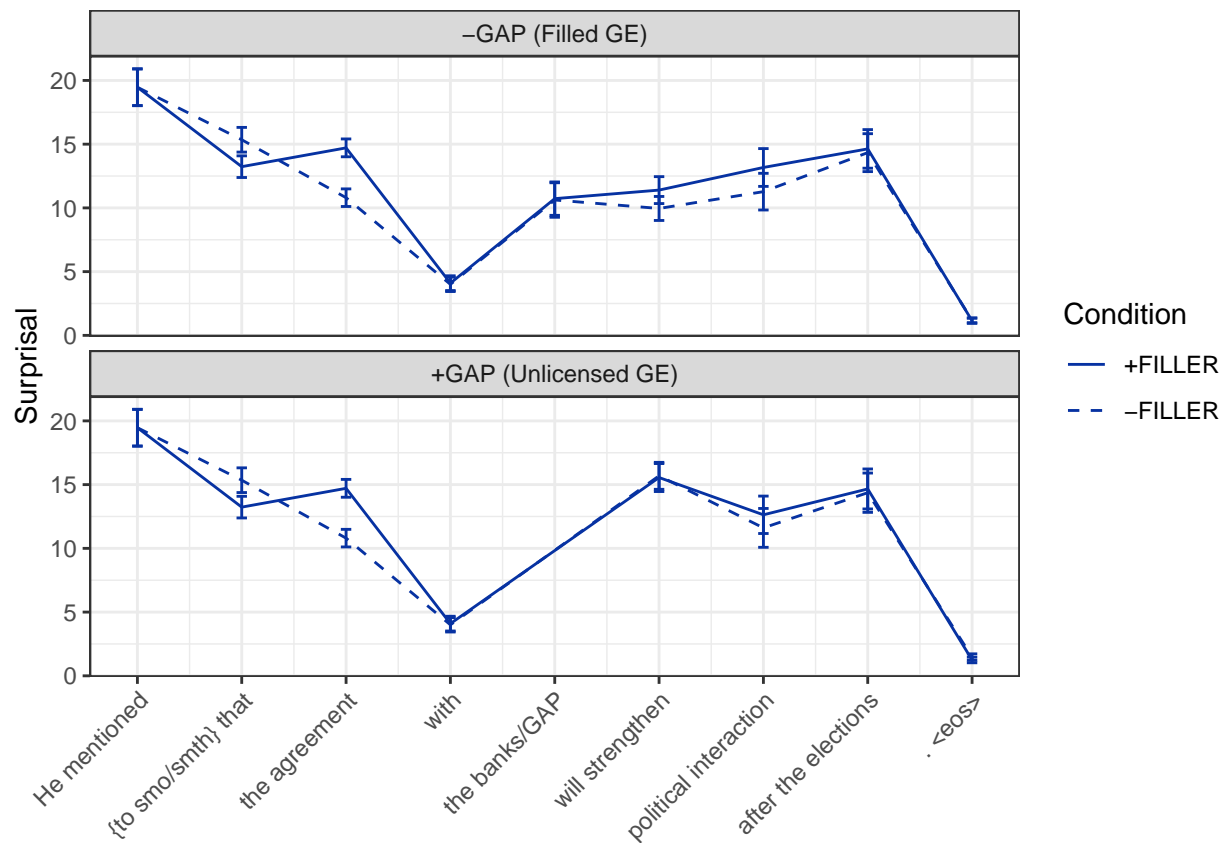
rc_control_fe_roi = fe.roi.stats(rc_control_fe)
rc_control_fe_roi$dependency = "RC"
rc_control_fe_roi$language = "Norwegian"
```

## Island, RC-dependency

```
REGION_ORDER = c("prefix", "comp", "subj", "prep", "obl", "verb", "obj", "end", "EOS")
REGION_EXEMPLARS = c("He mentioned", "{to smo/smith} that", "the agreement",
  "with", "the banks/GAP", "will strengthen",
  "political interaction", "after the elections", ". <eos>")
```

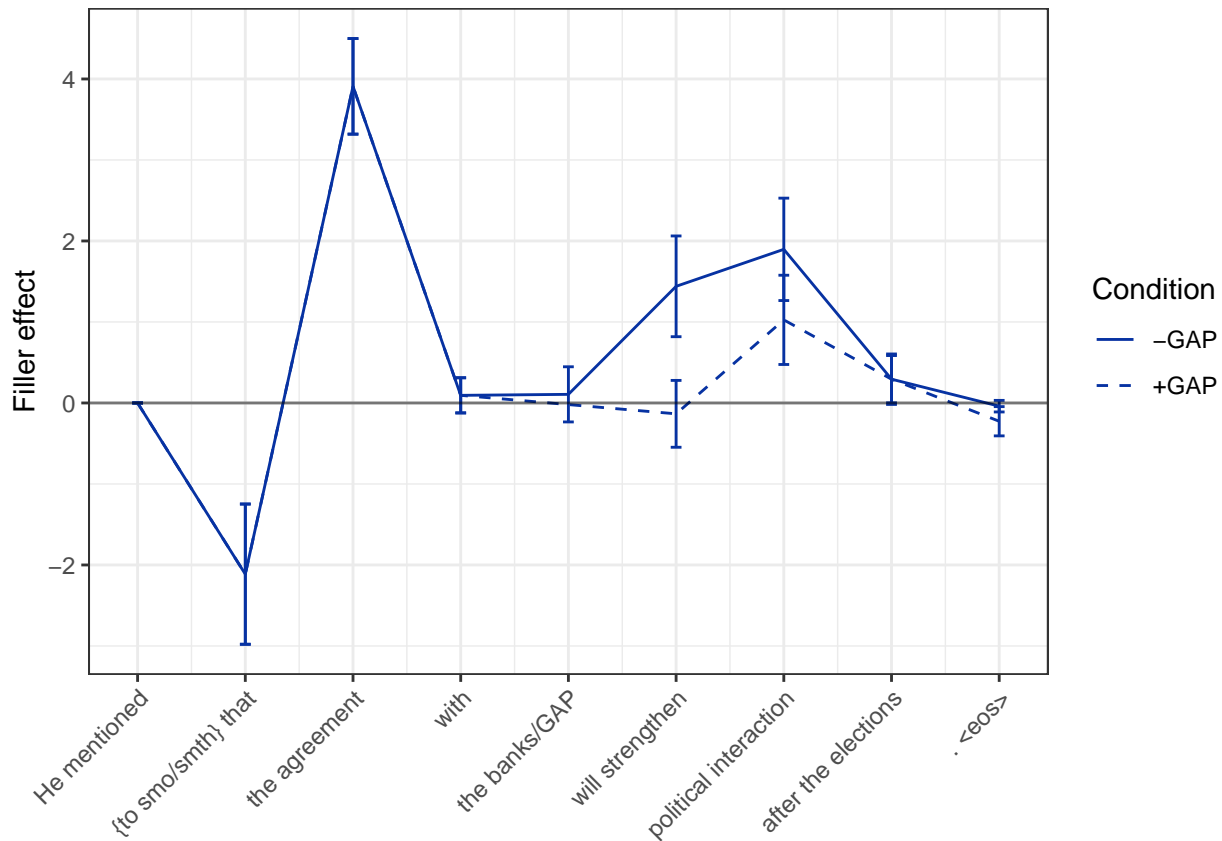
```
# Changing the data according to the ROIs
rc_island = rc_island %>%
  mutate(region = if_else(region == "pp-add" | region == "that" | region == "about"
    | region == "pr-obl" | region == "rp", "comp", region),
    region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

```
rc_island = region.surprisal(data = rc_island)
raw.surprisal.plot(data = rc_island, name = "no-rc-dep-island", path = regions_raw,
  regions = REGION_EXEMPLARS, color_choice = c("#0732A2"))
```



Calculating filler effects and plotting them by region:

```
rc_island_fe = fe.calculation(data = rc_island)
fe.regions.plot(data = rc_island_fe, name = "no-rc-dep-island", path = regions_fe,
                regions = REGION_EXEMPLARS,color_choice = c("#0732A2"))
```



```
rc_island_fe = rc_island_fe %>%
  filter(region == "obl" & gap == "no-gap" | region == "verb" & gap == "gap")

rc_island_fe_roi = fe.roi.stats(rc_island_fe)
rc_island_fe_roi$dependency = "RC"
rc_island_fe_roi$language = "Norwegian"
```

## Embed, wh-dependency

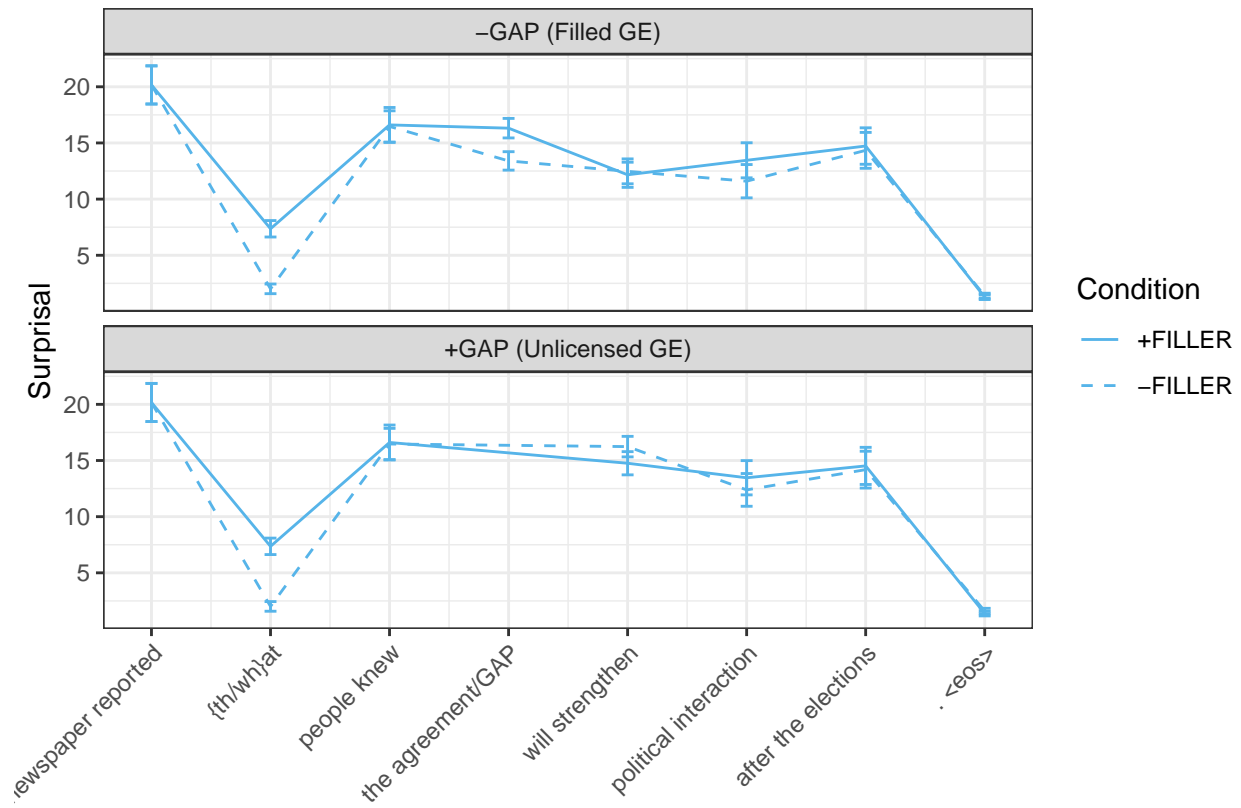
Regions for embed subject position:

```
REGION_ORDER = c("prefix", "comp", "embed", "subj", "verb", "obj", "end", "EOS")
REGION_EXEMPLARS = c("The newspaper reported", "{th/wh}at", "people knew", "the agreement/GAP", "will s",
  "political interaction", "after the elections", ". <eos>")

# Changing the data according to the ROIs
wh_embed = wh_embed %>%
  mutate(region = if_else(region == "that" | region == "wh-subj" | region == "rp", "comp", region),
    region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

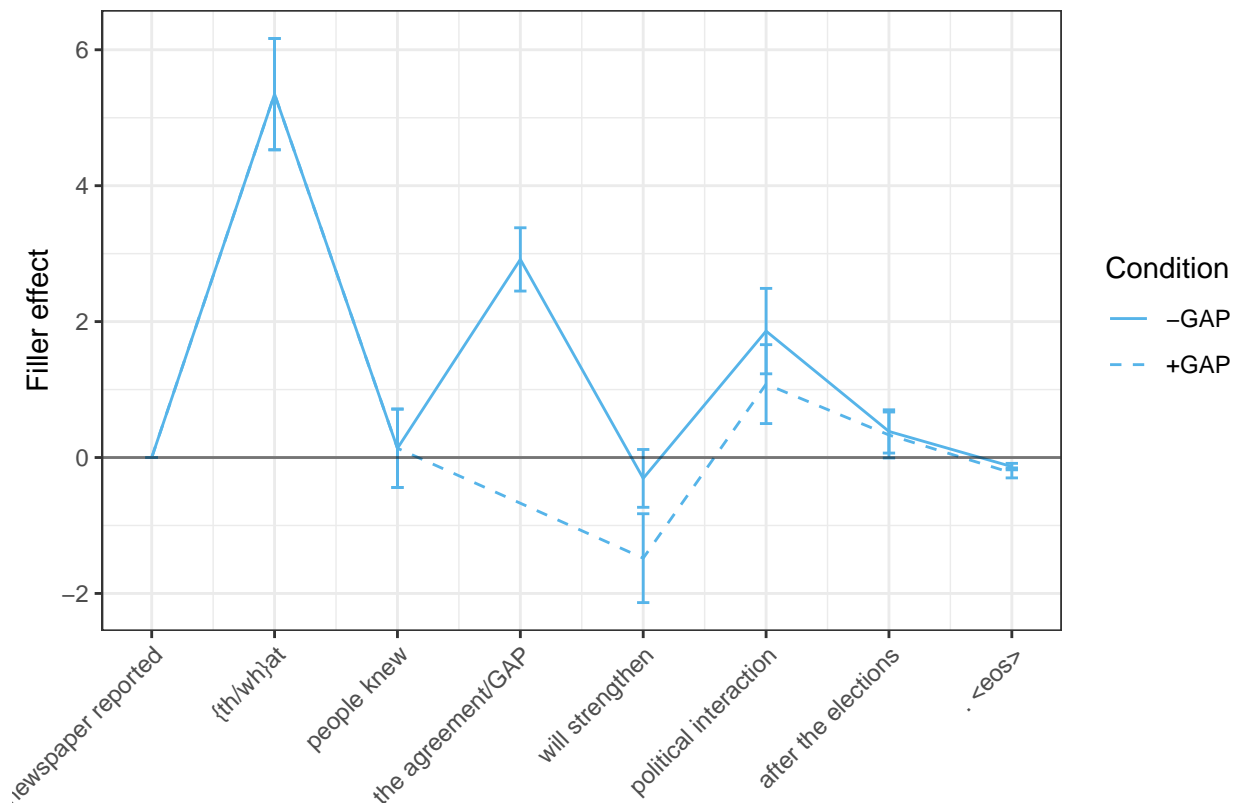
Aggregating the data and plotting raw surprisal values:

```
wh_embed = region.surprisal(data = wh_embed)
raw.surprisal.plot(data = wh_embed, name = "no-wh-dep-control-emb", path = regions_raw,
  regions = REGION_EXEMPLARS, color_choice = c("#56B4E9"))
```



Calculating filler effects and plotting them by region:

```
wh_embed_fe = fe.calculation(data = wh_embed)
fe.regions.plot(data = wh_embed_fe, name = "no-wh-dep-control-emb", path = regions_fe,
                regions = REGION_EXEMPLARS, color_choice = c("#56B4E9"))
```



```
wh_embed_fe = wh_embed_fe %>%
  filter(region == "subj" & gap == "no-gap" | region == "verb" & gap == "gap")

wh_embed_fe_roi = fe.roi.stats(wh_embed_fe)
wh_embed_fe_roi$dependency = "Wh"
wh_embed_fe_roi$language = "Norwegian"
```

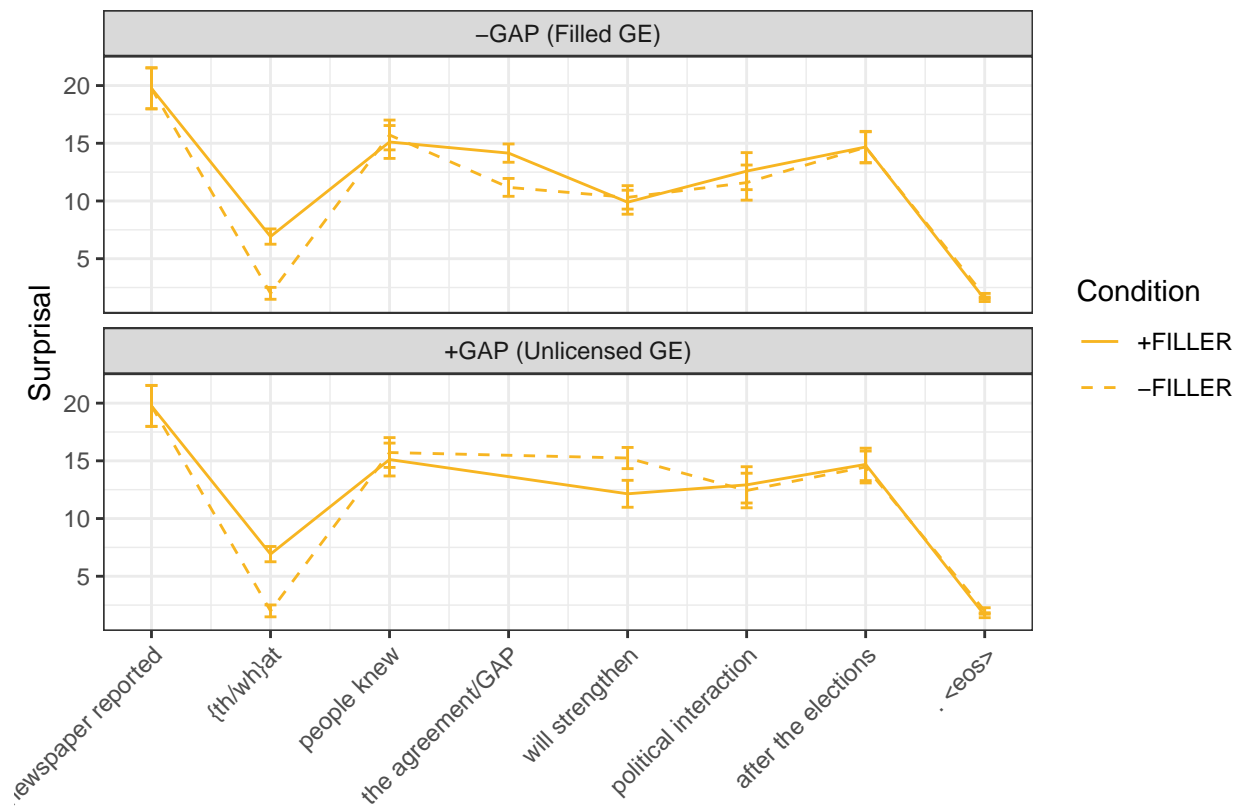
## Embed, English

```
# Changing the data according to the ROIs
en_embed = en_embed %>%
  mutate(region = if_else(region == "that" | region == "wh-subj" | region == "rp", "comp", region),
         region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

Aggregating the data and plotting raw surprisal values:

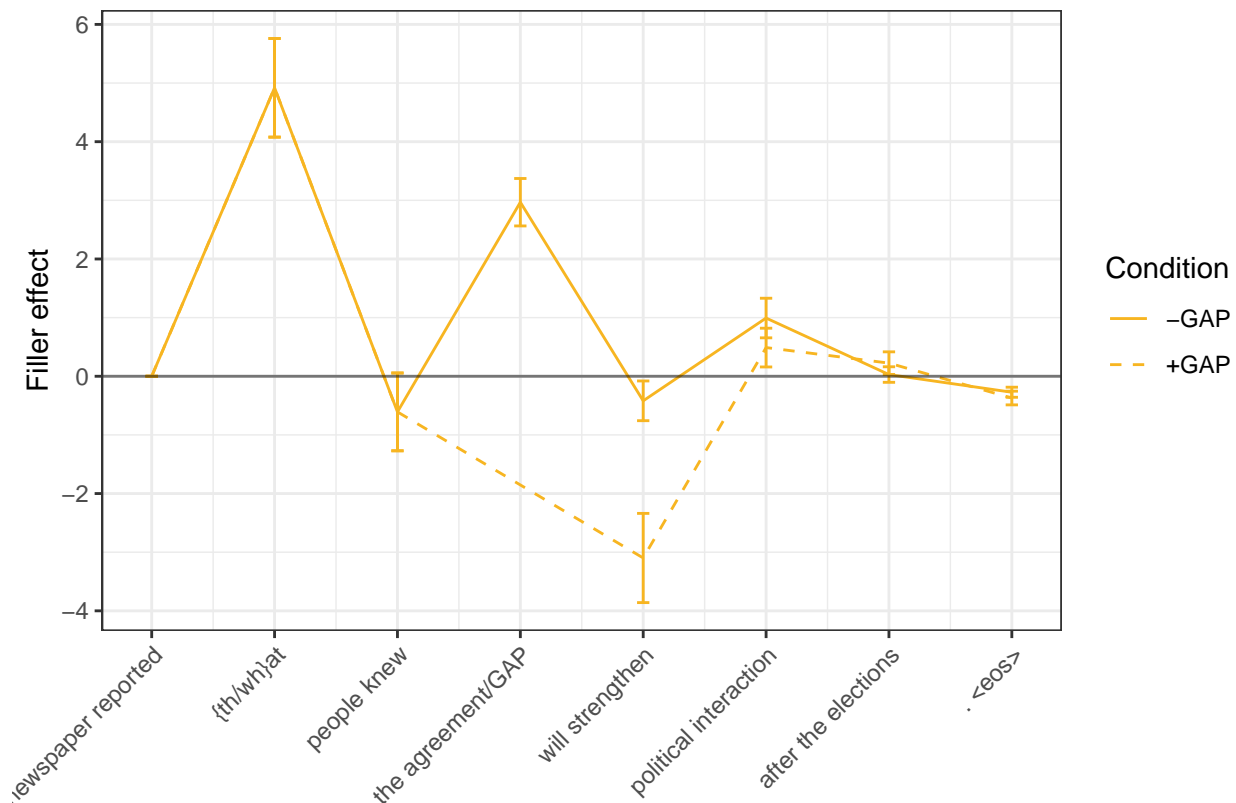
```
en_embed = region.surprisal(data = en_embed)
raw.surprisal.plot(data = en_embed, name = "en-wh-dep-control-emb", path = regions_raw,
                   regions = REGION_EXEMPLARS, color_choice = c("#F7B521"))
```





Calculating filler effects and plotting them by region:

```
en_embed_fe = fe.calculation(data = en_embed)
fe.regions.plot(data = en_embed_fe, name = "en-wh-dep-control-emb", path = regions_fe,
                regions = REGION_EXEMPLARS, color_choice = c("#F7B521"))
```



```
en_embed_fe = en_embed_fe %>%
  filter(region == "subj" & gap == "no-gap" | region == "verb" & gap == "gap")

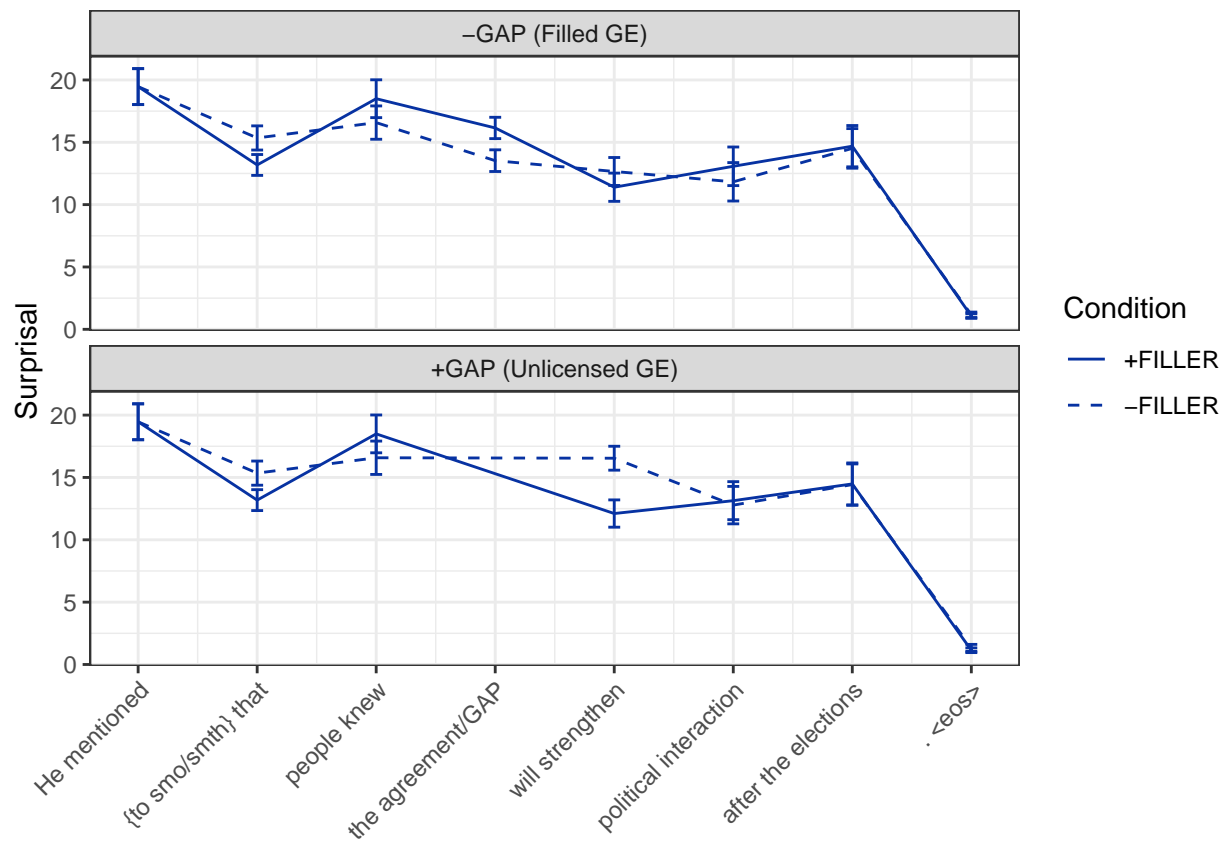
en_embed_fe_roi = fe.roi.stats(en_embed_fe)
en_embed_fe_roi$dependency = "Wh"
en_embed_fe_roi$language = "English"
```

### Embed, RC-dependency

```
REGION_ORDER = c("prefix", "comp", "embed", "subj", "verb", "obj", "end", "EOS")
REGION_EXEMPLARS = c("He mentioned", "{to smt/somth} that", "people knew", "the agreement/GAP", "will st",
  "political interaction", "after the elections", ". <eos>")
```

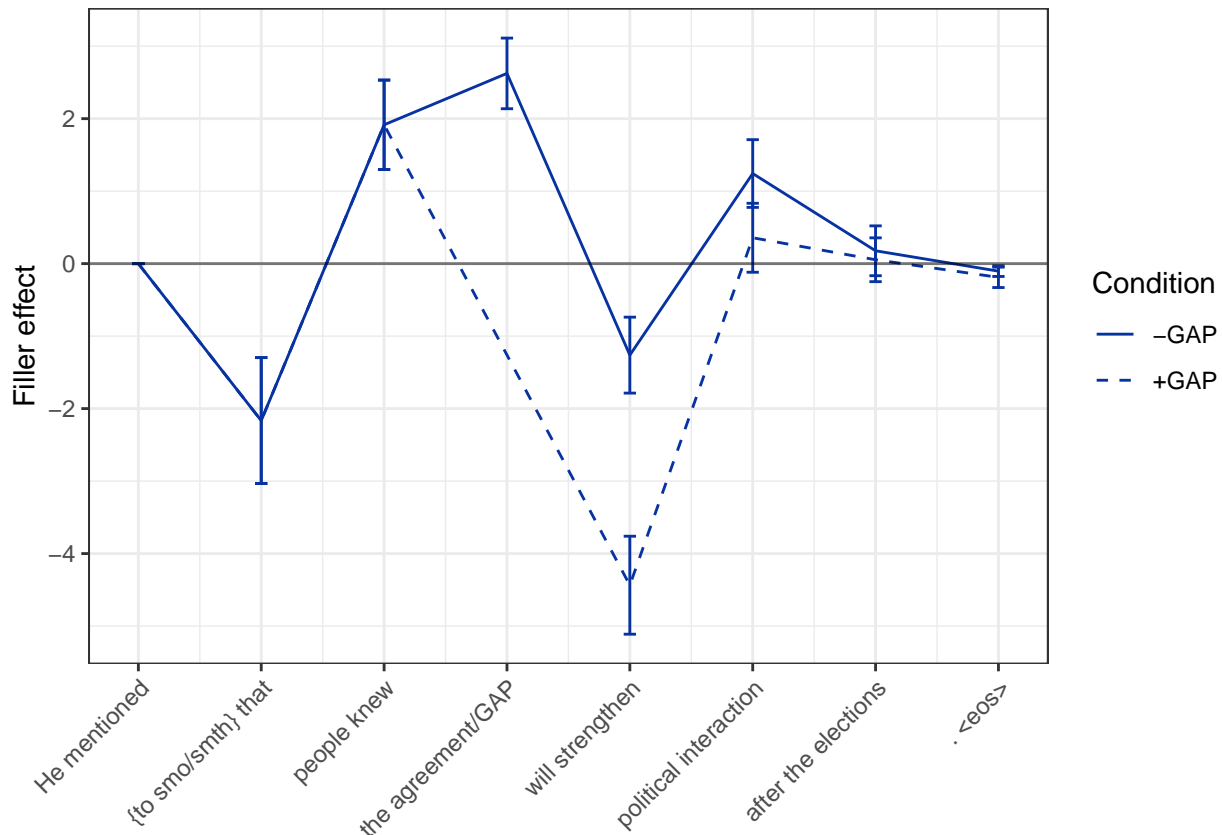
```
rc_embed = rc_embed %>%
  mutate(region = if_else(region == "pp-add" | region == "that" | region == "rp"
    | region == "pr-subj" | region == "about", "comp", region),
    region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

```
rc_embed = region.surprisal(data = rc_embed)
raw.surprisal.plot(data = rc_embed, name = "no-rc-dep-control-emb", path = regions_raw,
  regions = REGION_EXEMPLARS, color_choice = c("#0732A2"))
```



Calculating filler effects and plotting them by region:

```
rc_embed_fe = fe.calculation(data = rc_embed)
fe.regions.plot(data = rc_embed_fe, name = "no-rc-dep-control-emb", path = regions_fe,
                regions = REGION_EXEMPLARS,color_choice = c("#0732A2"))
```



```
rc_embed_fe = rc_embed_fe %>%
  filter(region == "subj" & gap == "no-gap" | region == "verb" & gap == "gap")

rc_embed_fe_roi = fe.roi.stats(rc_embed_fe)
rc_embed_fe_roi$dependency = "RC"
rc_embed_fe_roi$language = "Norwegian"
```

### Plot with all filler effects

```
d_filler_effect <- Reduce(function(x, y) merge(x, y, all=TRUE),
  list(wh_control_fe_roi, wh_island_fe_roi,
        rc_control_fe_roi, rc_island_fe_roi,
        en_control_fe_roi, en_island_fe_roi,
        wh_embed_fe_roi, rc_embed_fe_roi,
        en_embed_fe_roi))

d_filler_effect[d_filler_effect == "pp-subj"] <- "subject island"
d_filler_effect[d_filler_effect == "subj-emb"] <- "embed. control"
d_filler_effect[d_filler_effect == "subj"] <- "subject control"

d_filler_effect$gap_position <- factor(d_filler_effect$gap_position,
  levels = c('subject control', 'embed. control', 'subject island'), ordered = TRUE)

d_filler_effect = d_filler_effect %>%
  mutate(lang_dep = paste(language, dependency, sep = " - "))

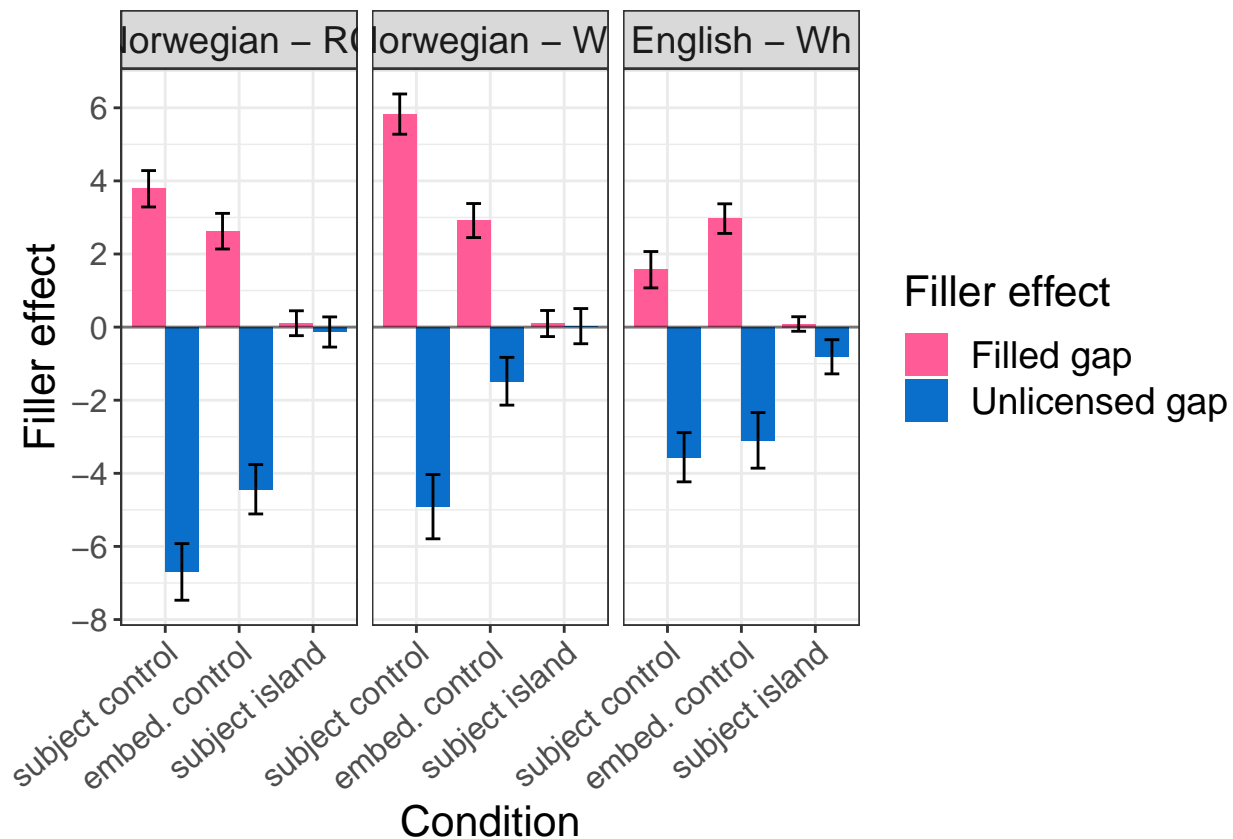
d_filler_effect$lang_dep <- factor(d_filler_effect$lang_dep,
```

```

levels = c('Norwegian - RC', 'Norwegian - Wh', 'English - Wh'), ordered = TRUE)

customs_two <- c("#FF5B97", "#096FCA")
d_filler_effect %>%
  group_by(gap, gap_position, lang_dep) %>%
  summarise(m = mean(filler_effect),
            n = n(),
            sd = sd(filler_effect),
            se = sd/sqrt(n),
            upper = m + 1.96*se,
            lower = m - 1.96*se) %>%
  ungroup() %>%
  ggplot(aes(x = gap_position, y = m, ymin = lower, ymax = upper, fill = gap)) +
  theme_bw() +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~lang_dep) +
  geom_errorbar(color = "black", width = .4, position=position_dodge(width = 0.9)) +
  ylab("Filler effect") + xlab("Condition") +
  scale_y_continuous(breaks = scales::pretty_breaks(n = 8)) +
  theme(axis.text.x = element_text(angle=40, hjust = 1, size = 12),
        strip.text = element_text(size = 14),
        axis.text.y = element_text(size = 12),
        legend.text = element_text(size = 14),
        legend.title = element_text(size = 16),
        axis.title = element_text(size = 16)) +
  geom_hline(yintercept = 0, color = "black", alpha = 0.5) +
  scale_fill_manual(values = customs_two, name = "Filler effect",
                    labels = c("Filled gap", "Unlicensed gap"))

```



```
fname = sprintf("plots/subject-plots/%s-subject-all.png", model_type)
ggsave(fname, width = 10)
```

```
## Saving 10 x 4.5 in image
```

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

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/subject_lstm_agg.csv", fileEncoding = "UTF-8-BOM")
gpt2 = read.csv("../data/results/gpt2/subject_gpt2_agg.csv", fileEncoding = "UTF-8-BOM")
d = rbind(lstm, gpt2)
```

```
d$model = factor(d$model, levels = c("LSTM", "GPT2"), ordered = TRUE)
d$lang_dep = factor(d$lang_dep, levels = c('Norwegian - RC', 'Norwegian - Wh',
                                           'English - Wh'), ordered = TRUE)
d$gap = factor(d$gap, levels = c("no-gap", "gap"), ordered = TRUE)
gap.labs <- c("-GAP (Filled gap effect)", "+GAP (Unlicensed gap effect)")
names(gap.labs) <- c("no-gap", "gap")
d$gap_position <- factor(d$gap_position,
                        levels = c('subject control', 'embed. control', 'subject island'), ordered = TRUE)
```

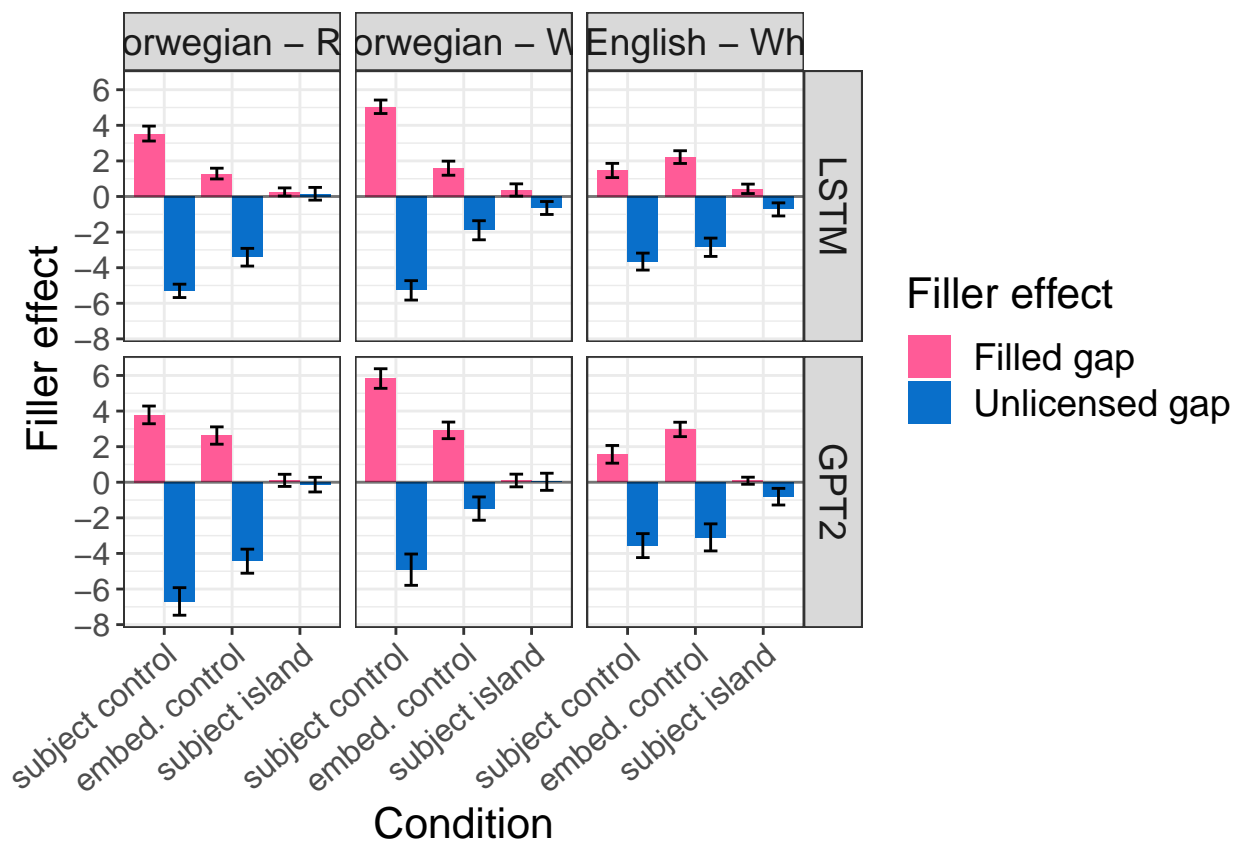
```
d %>%
  group_by(model, gap, gap_position, lang_dep) %>%
  summarise(m = mean(filler_effect),
```

```

n = n(),
sd = sd(filler_effect),
se = sd/sqrt(n),
upper = m + 1.96*se,
lower = m - 1.96*se) %>%

ungroup() %>%
ggplot(aes(x = gap_position, y = m, ymin = lower, ymax = upper, fill = gap)) +
theme_bw() +
geom_bar(stat = "identity", position = "dodge") +
facet_grid(model~lang_dep) +
geom_errorbar(color = "black", width = .4, position=position_dodge(width = 0.9)) +
ylab("Filler effect") + xlab("Condition") +
scale_y_continuous(breaks = scales::pretty_breaks(n = 8)) +
theme(axis.text.x = element_text(angle=40, hjust = 1, size = 12),
      strip.text = element_text(size = 14),
      axis.text.y = element_text(size = 12),
      legend.text = element_text(size = 14),
      legend.title = element_text(size = 16),
      axis.title = element_text(size = 16)) +
geom_hline(yintercept = 0, color = "black", alpha = 0.5) +
scale_fill_manual(values = customs_two, name = "Filler effect",
                  labels = c("Filled gap", "Unlicensed gap"))

```



```

ggsave("plots/subject-plots/subject-two-models.png", width = 10, height = 7)
ggsave("plots/subject-plots/subject-two-models.pdf", width = 10, height = 7)

```

## Stats

```

forw_coding = matrix(data = c(0.5,-0.5,0,0,0.5,-0.5) , nrow = 3, ncol=2)
d_filler_effect$gap_position <- as.factor(d_filler_effect$gap_position)
d_filler_effect$gap_position <- ordered(d_filler_effect$gap_position, levels = c("subject control", "em
contrasts(d_filler_effect$gap_position) <- forw_coding

regressions = list()

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

for (i in models) {
  fge_model = d_filler_effect %>%
    filter(gap == "no-gap" & lang_dep == i) %>%
    lmer(filler_effect ~ gap_position + (1 |sent_index), data=.)
  uge_model = d_filler_effect %>%
    filter(gap == "gap" & lang_dep == i) %>%
    lmer(filler_effect ~ gap_position + (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 = modelsummary(regressions, output = "gt", stars = TRUE, gof_omit = ".*",
  estimate = "{estimate}{stars}", statistic = NULL, fmt = 1,
  coef_rename = c("gap_position1" = "controlCntrst",
    "gap_position2" = "islandCntrst")) %>%

  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 = modelsummary(regressions, output = "gt", stars = TRUE, gof_omit = ".*",
  estimate = "{estimate}{stars} ({std.error})",
  statistic = "t = {statistic}", fmt = 1,
  coef_rename = c("gap_position1" = "controlCntrst",
    "gap_position2" = "islandCntrst")) %>%

  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)

stats_fname_html = sprintf("stats/subject-stats/subject-%s.html", model_type)
stats_fname_tex = sprintf("stats/subject-stats/subject-%s.tex", model_type)
html_table |> gtsave(stats_fname_html)
latex_table |> gtsave(stats_fname_tex)

```