# Analysis of Adjunct EQ island results

Anastasia Kobzeva

## Data pre-processing and plotting

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

# 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

```
#model_type = "lstm"
model_type = "gpt2"

filename_wh = sprintf("../data/results/%s/eq_wh_result.csv", model_type) # Norwegian wh
filename_rc = sprintf("../data/results/%s/eq_rc_result.csv", model_type) # Norwegian RC
filename_en = sprintf("../data/results/%s/eq_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
```

### Loading in data and analysis functions

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

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

```
# Splitting by condition (embedded declarative clause vs. EQ)
control = df[endsWith(df$condition,"that-comp"),]  # decl that complementizer
no_comp = df[endsWith(df$condition,"no-comp"),]  # no complementizer
island = df[endsWith(df$condition,"wh-comp"),]  # eq wh-word complementizer (where/when etc.)
```

**Embedded declaratives (control) condition, wh-dependencies**

```
REGION_ORDER = c("prefix", "embed", "comp2", "subj", "verb", "adjunct", "end", "EOS")
REGION_EXEMPLARS = c("He said {th/wh}at", "the designer specified", "that",
                     "the shelf/GAP", "should be mounted", "in the hallway", "ASAP", ". <eos>")
```

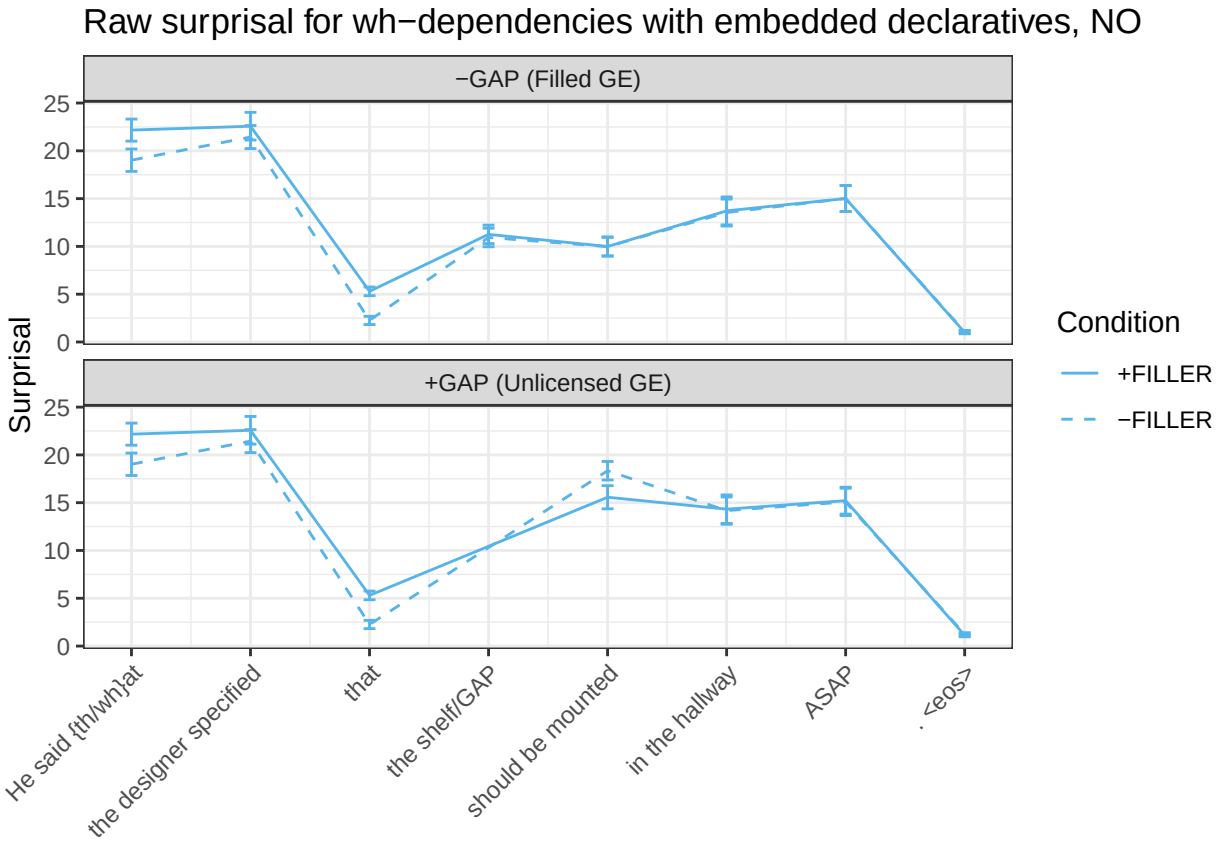
```
wh_control = control %>%
  filter(dependency == "Wh" & language == "Norwegian")
rc_control = control %>%
  filter(dependency == "RC" & language == "Norwegian")
wh_island = island %>%
  filter(dependency == "Wh" & language == "Norwegian")
rc_island = island %>%
  filter(dependency == "RC" & language == "Norwegian")
en_control = control %>%
  filter(language == "English")
en_island = island %>%
  filter(language == "English")
en_no_comp = no_comp %>%
  filter(language == "English")
rc_no_comp = no_comp %>%
  filter(language == "Norwegian" & dependency == "RC")
wh_no_comp = no_comp %>%
  filter(language == "Norwegian" & dependency == "Wh")
```

**Norwegian, wh-dependency, control condition**

```
# Changing the data according to the ROIs
wh_control = wh_control %>%
  mutate(region = if_else(region == "that1" | region == "wh-subj", "prefix", region),
         region = if_else(region == "that2" | region == "wh-word", "comp2", region),
         region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

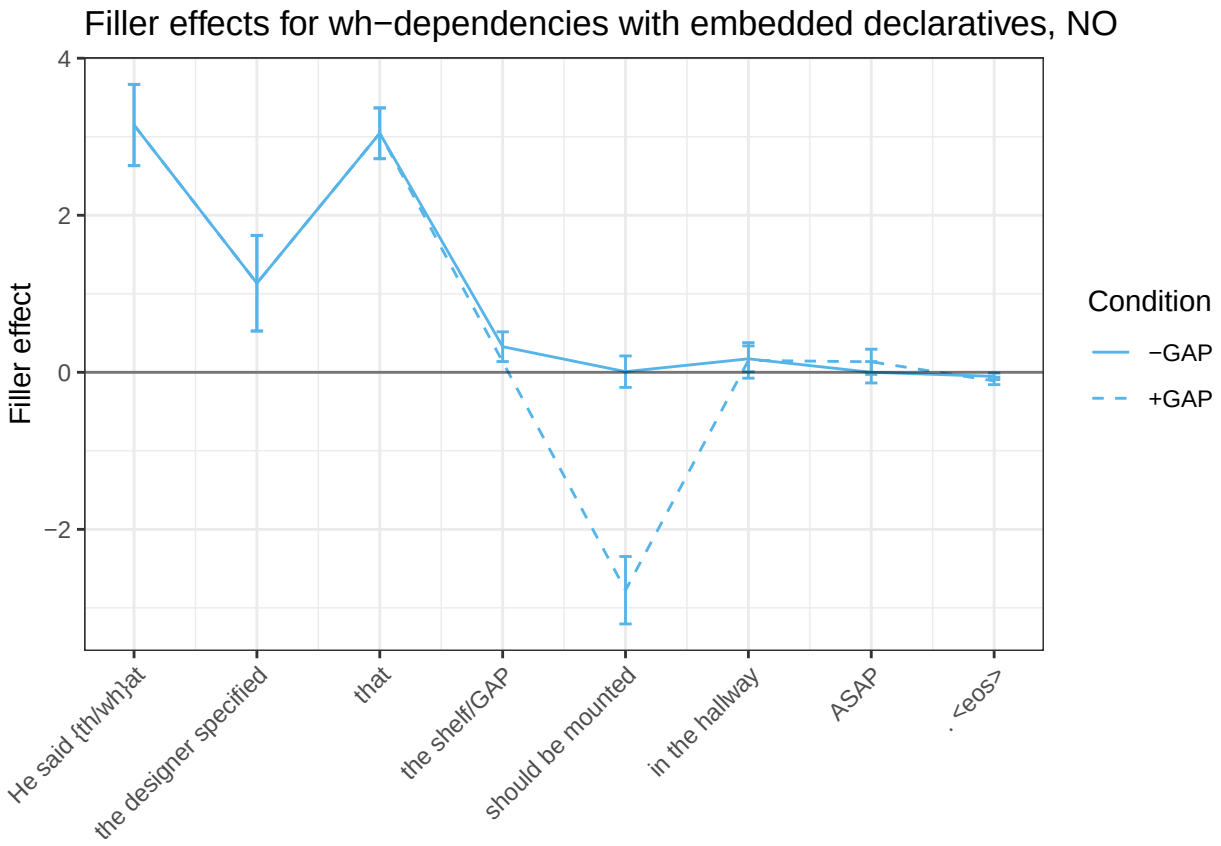
```
wh_control = region.surprisal(data = wh_control)
```

```
plot = raw.surprisal.plot(data = wh_control, name = "no-wh-dep-control",
                          path = regions_raw, regions = REGION_EXEMPLARS,
                          color_choice = c("#56B4E9"))
plot + ggtitle("Raw surprisal for wh-dependencies with embedded declaratives, NO")
```



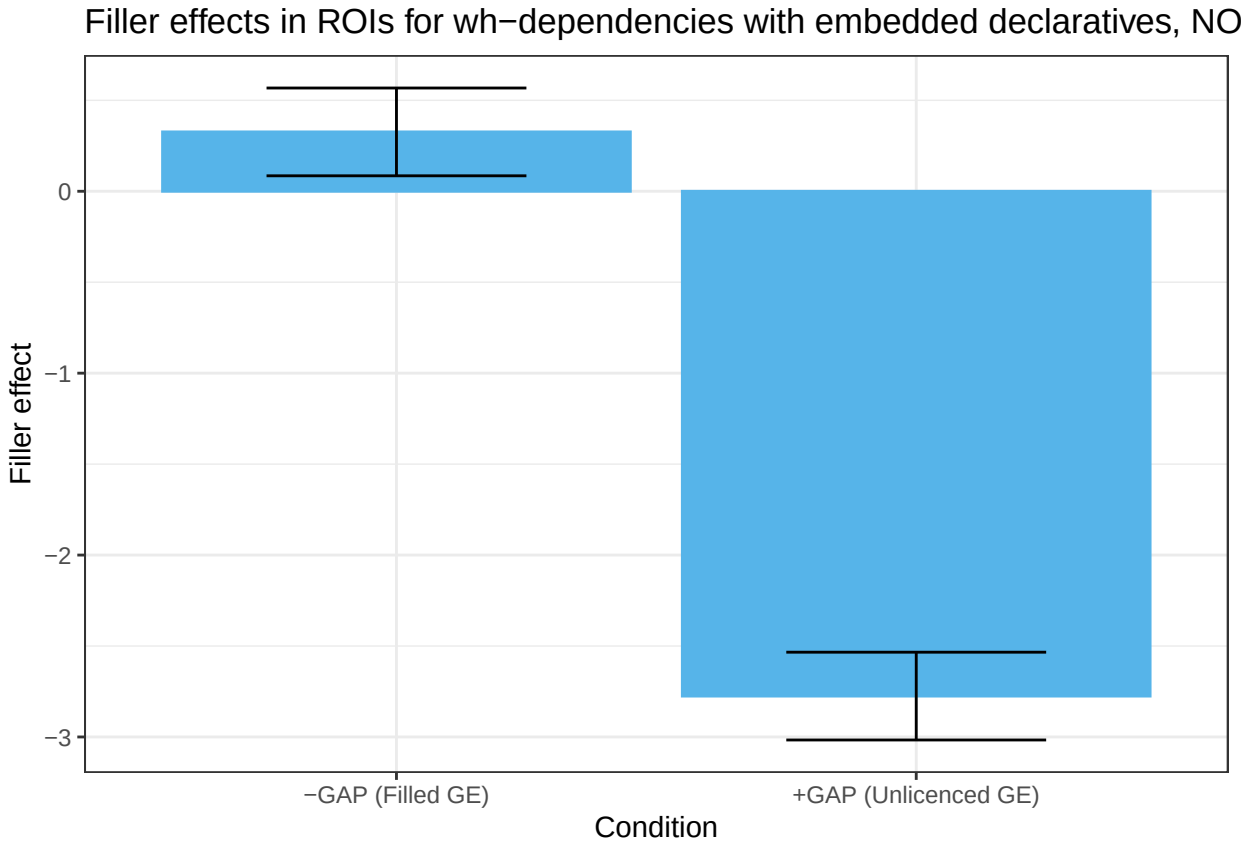

Calculating filler effects and plotting them by region:

```
wh_control_fe = fe.calculation(data = wh_control)
plot = fe.regions.plot(data = wh_control_fe, name = "no-wh-dep-control", path = regions_fe,
                       regions = REGION_EXEMPLARS, color_choice = c("#56B4E9"))
plot + ggtitle("Filler effects for wh-dependencies with embedded declaratives, NO")
```




Calculating and plotting filler effects in ROIs:

```
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"
plot = fe.roi.plot(data = wh_control_fe_roi, name = "no-wh-dep-control", path = fe_roi, color_choice = c
plot + ggtitle("Filler effects in ROIs for wh-dependencies with embedded declaratives, NO")
```



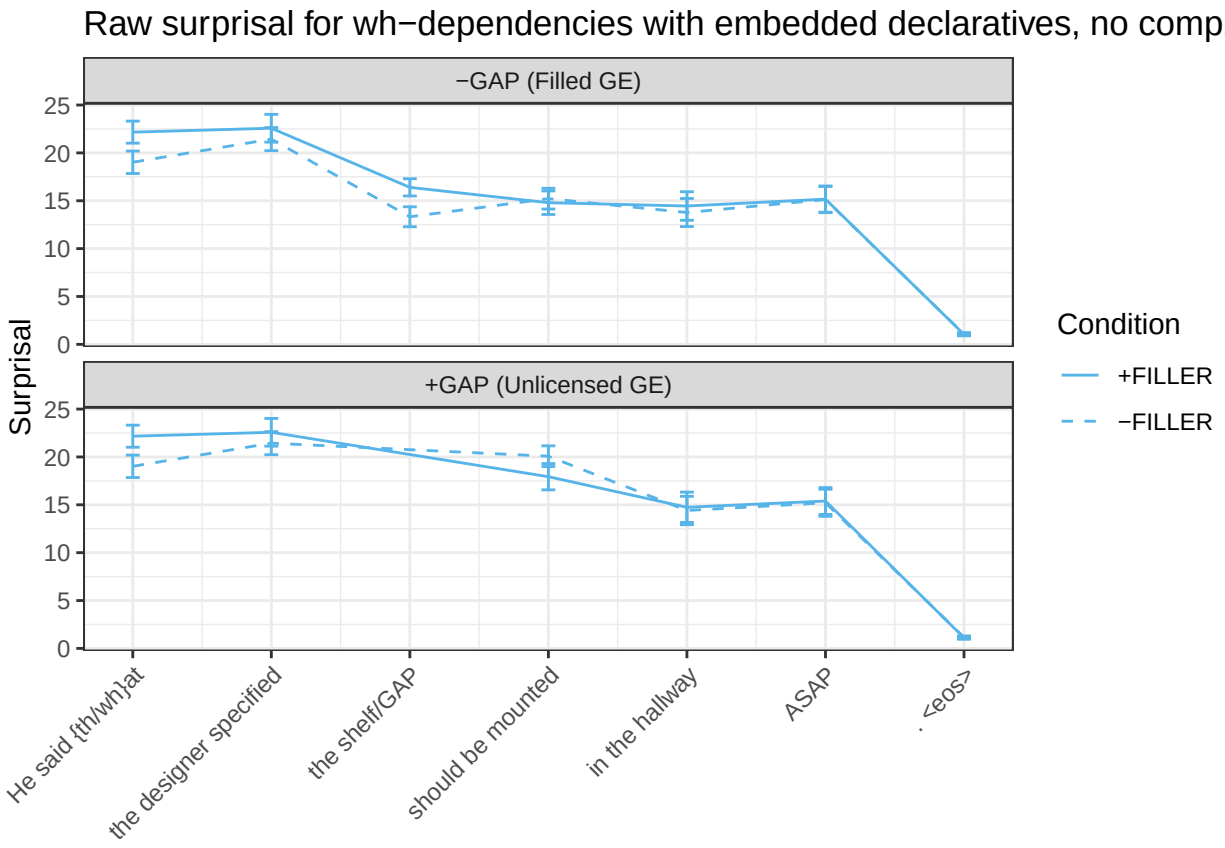

```
# New regions without the complementizer
REGION_ORDER_NC = c("prefix", "embed", "subj", "verb", "adjunct", "end", "EOS")
REGION_EXEMPLARS_NC = c("He said {th/wh}at", "the designer specified",
                        "the shelf/GAP", "should be mounted", "in the hallway", "ASAP", ". <eos>")
```

```
# Changing the data according to the ROIs
wh_no_comp = wh_no_comp %>%
  mutate(region = if_else(region == "that1" | region == "wh-subj", "prefix", region),
         region = if_else(region == "that2" | region == "wh-word", "comp2", region),
         region = factor(region, levels=REGION_ORDER_NC)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

**Norwegian, wh-dependency, no-complementizer condition** Aggregating the data and plotting raw surprisal values:

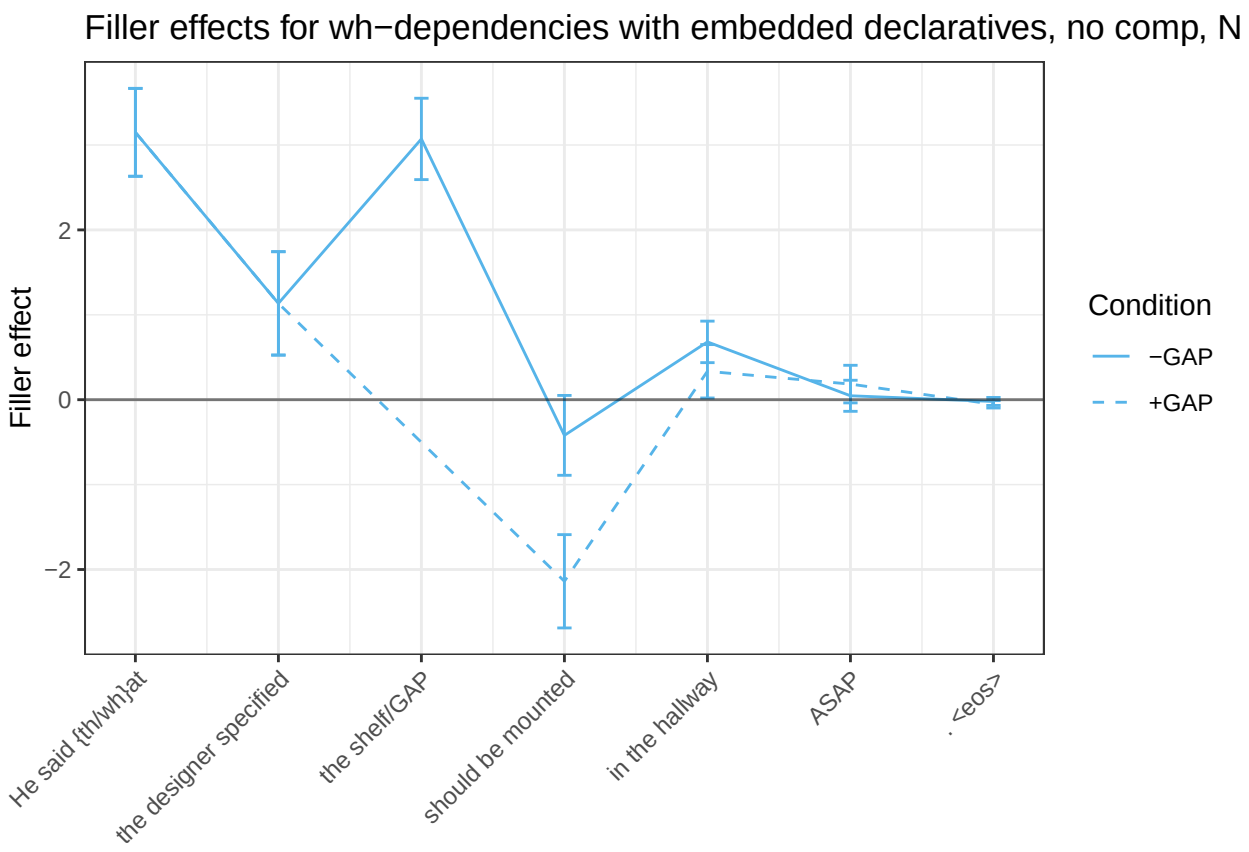
```
wh_no_comp = region.surprisal(data = wh_no_comp)
```

```
plot = raw.surprisal.plot(data = wh_no_comp, name = "no-wh-dep-no-comp",
                          path = regions_raw, regions = REGION_EXEMPLARS_NC,
                          color_choice = c("#56B4E9"))
plot + ggtitle("Raw surprisal for wh-dependencies with embedded declaratives, no comp, NO")
```



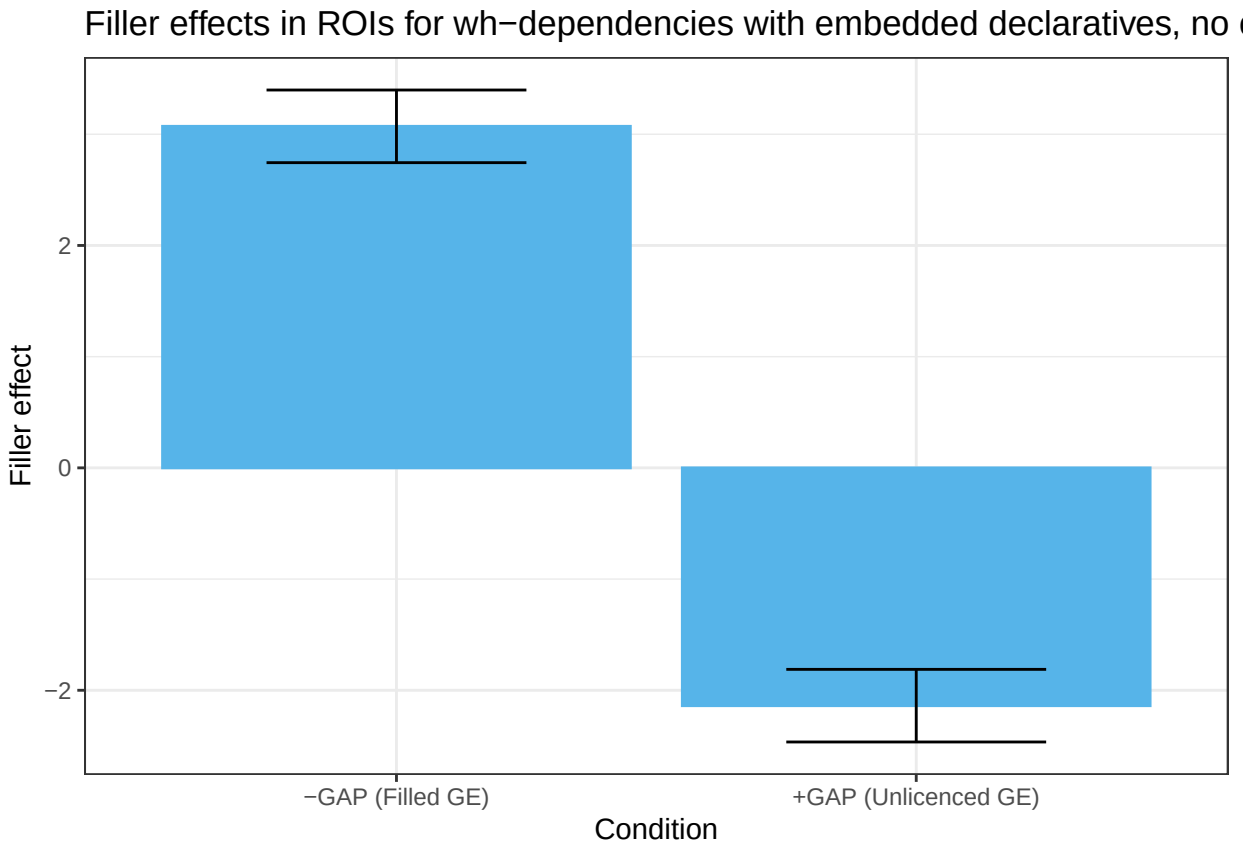

Calculating filler effects and plotting them by region:

```
wh_no_comp_fe = fe.calculation(data = wh_no_comp)
plot = fe.regions.plot(data = wh_no_comp_fe, name = "no-wh-dep-no-comp", path = regions_fe,
                       regions = REGION_EXEMPLARS_NC, color_choice = c("#56B4E9"))
plot + ggtitle("Filler effects for wh-dependencies with embedded declaratives, no comp, NO")
```




Calculating and plotting filler effects in ROIs:

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

wh_no_comp_fe_roi = fe.roi.stats(wh_no_comp_fe)
wh_no_comp_fe_roi$dependency = "Wh"
wh_no_comp_fe_roi$language = "Norwegian"
plot = fe.roi.plot(data = wh_no_comp_fe_roi, name = "no-wh-dep-no-comp", path = fe_roi, color_choice = c
plot + ggtitle("Filler effects in ROIs for wh-dependencies with embedded declaratives, no comp, NO")
```



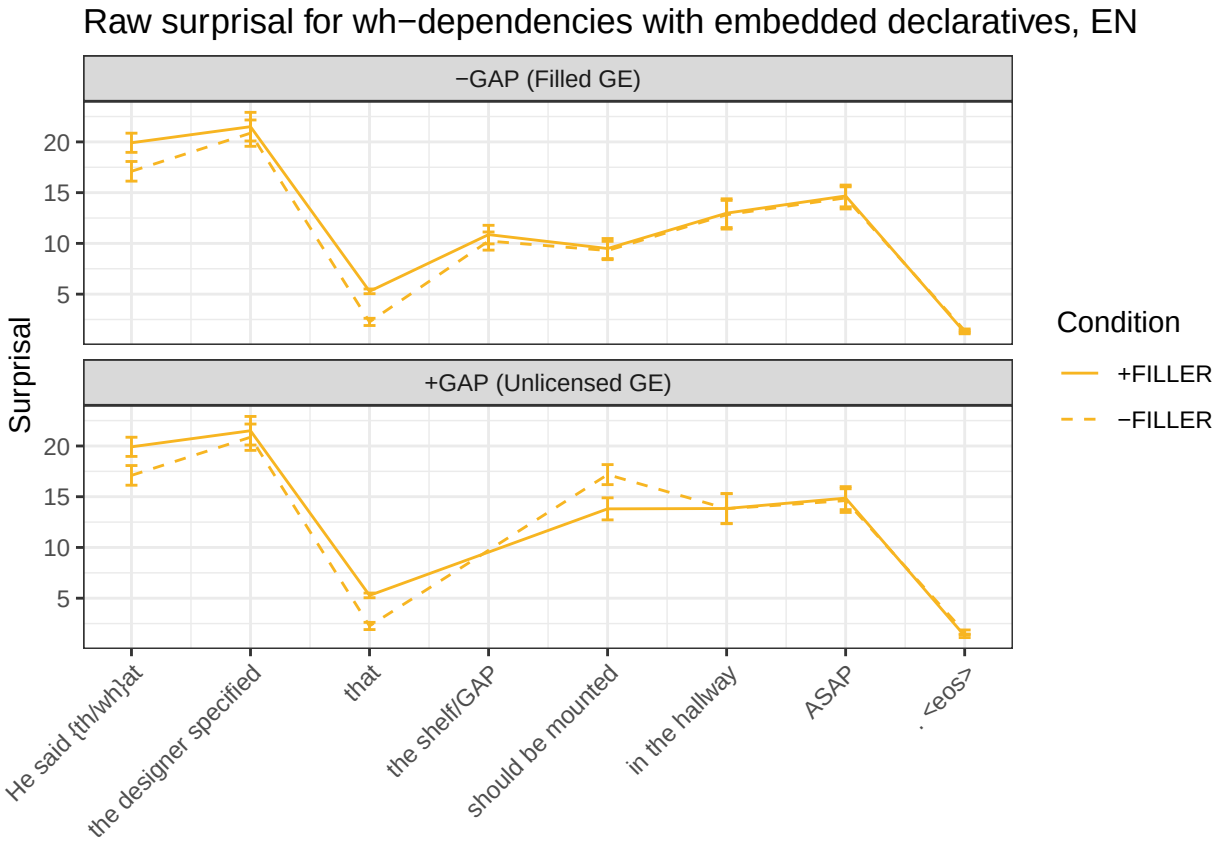

**English, wh-dependency, control condition**

```
# Changing the data according to the ROIs
en_control = en_control %>%
  mutate(region = if_else(region == "that1" | region == "wh-subj", "prefix", region),
         region = if_else(region == "that2" | region == "wh-word", "comp2", region),
         region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

Aggregating the data and plotting raw surprisal values:

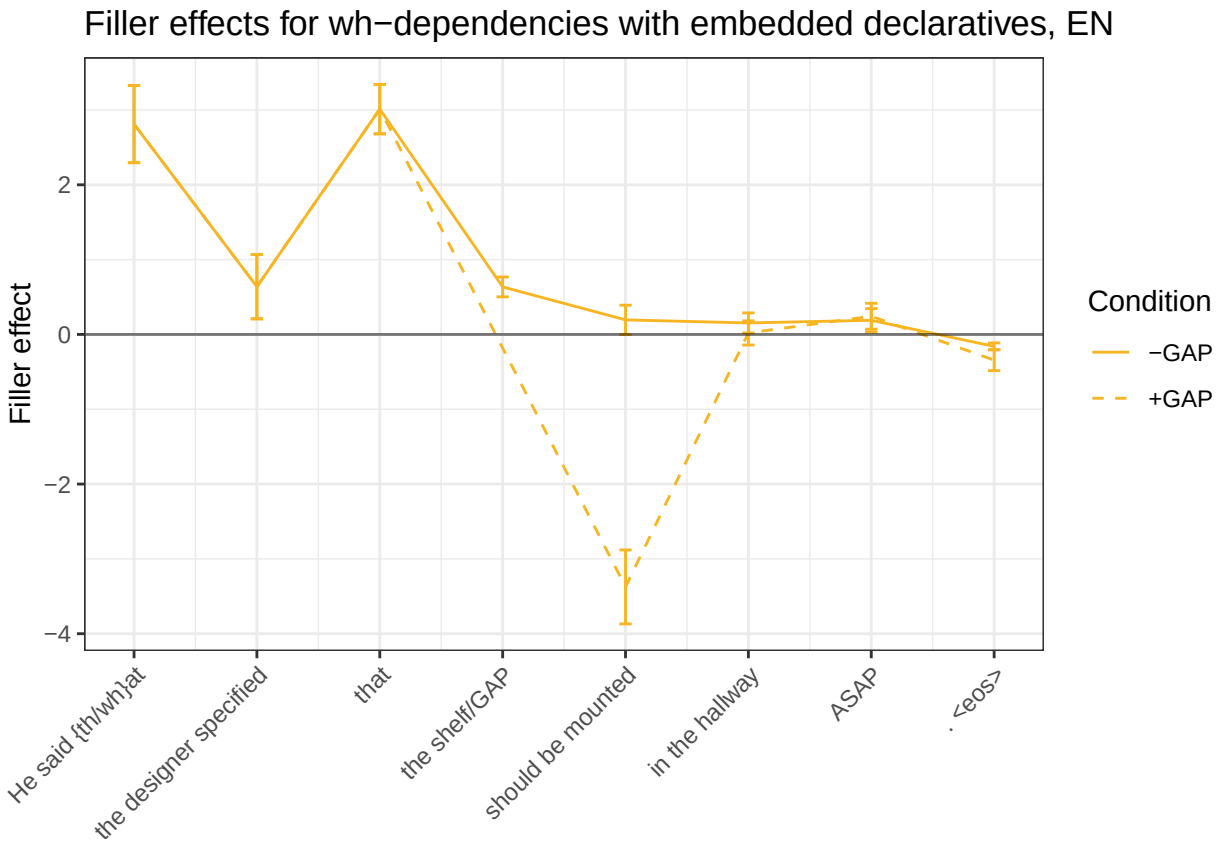
```
en_control = region.surprisal(data = en_control)
```

```
plot = raw.surprisal.plot(data = en_control, name = "en-wh-dep-control",
                          path = regions_raw, regions = REGION_EXEMPLARS,
                          color_choice = c("#F7B521"))
plot + ggtitle("Raw surprisal for wh-dependencies with embedded declaratives, EN")
```



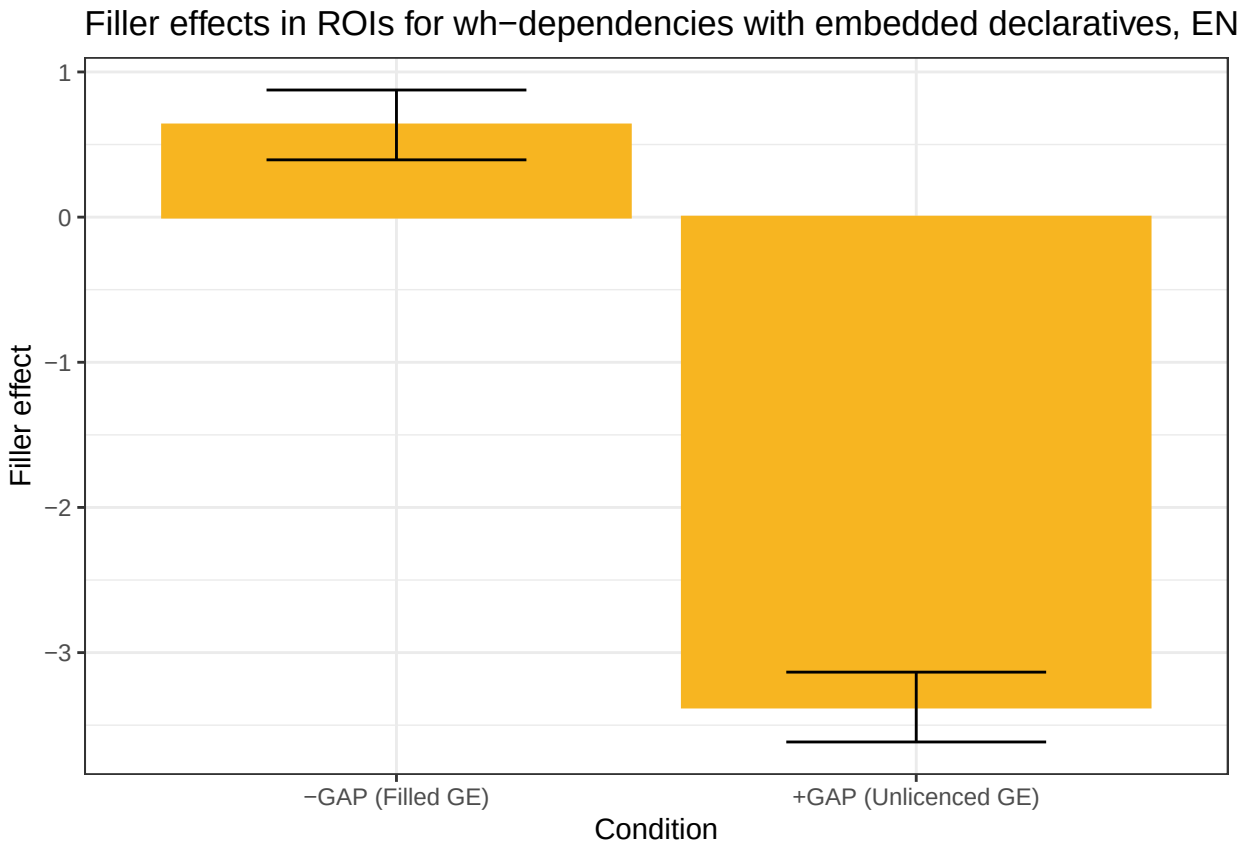

Calculating filler effects and plotting them by region:

```
en_control_fe = fe.calculation(data = en_control)
plot = fe.regions.plot(data = en_control_fe, name = "en-wh-dep-control", path = regions_fe,
                       regions = REGION_EXEMPLARS, color_choice = c("#F7B521"))
plot + ggtitle("Filler effects for wh-dependencies with embedded declaratives, EN")
```




Calculating and plotting filler effects in ROIs:

```
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"
plot = fe.roi.plot(data = en_control_fe_roi, name = "en-wh-dep-control", path = fe_roi, color_choice = c
plot + ggtitle("Filler effects in ROIs for wh-dependencies with embedded declaratives, EN")
```



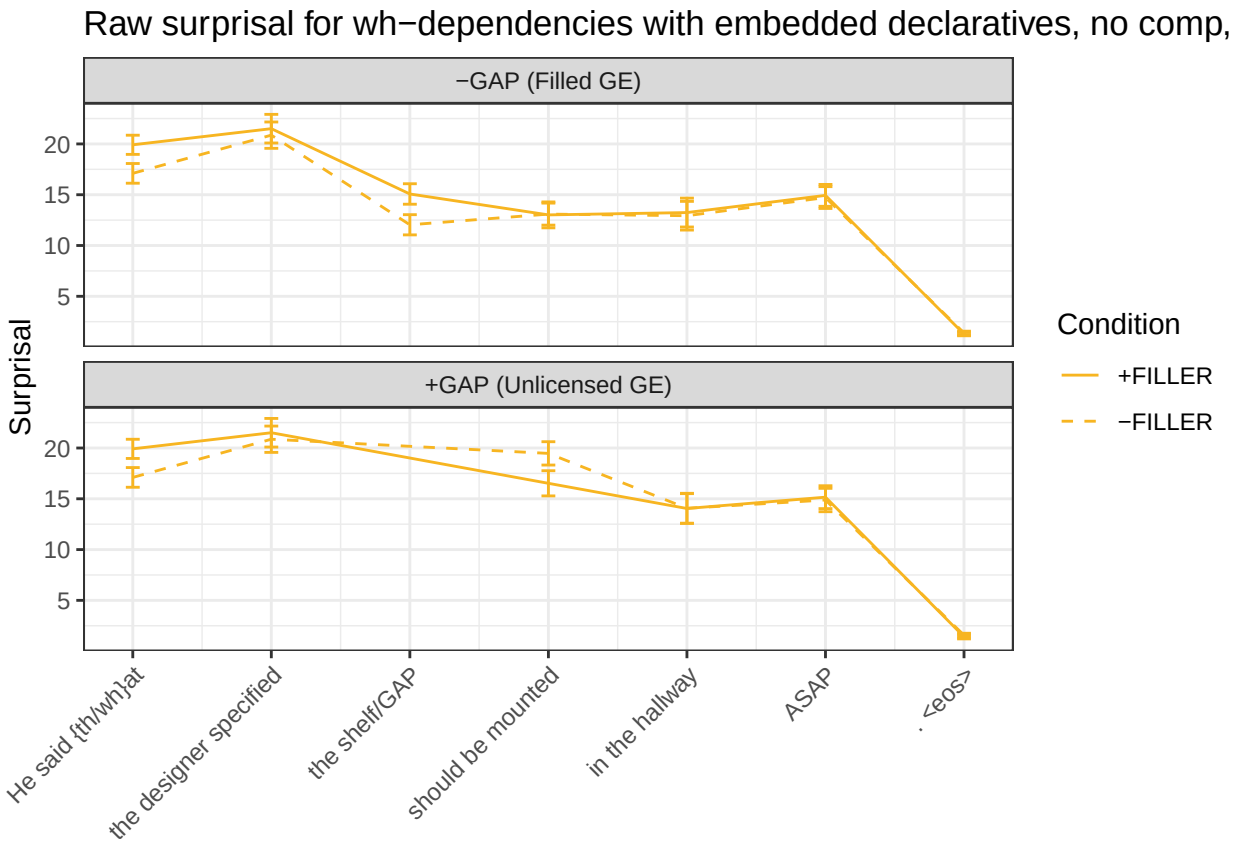

**English, wh-dependency, no-complementizer condition**

```
# Regions as before
# Changing the data according to the ROIs
en_no_comp = en_no_comp %>%
  mutate(region = if_else(region == "that1" | region == "wh-subj", "prefix", region),
         region = if_else(region == "that2" | region == "wh-word", "comp2", region),
         region = factor(region, levels=REGION_ORDER_NC)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

Aggregating the data and plotting raw surprisal values:

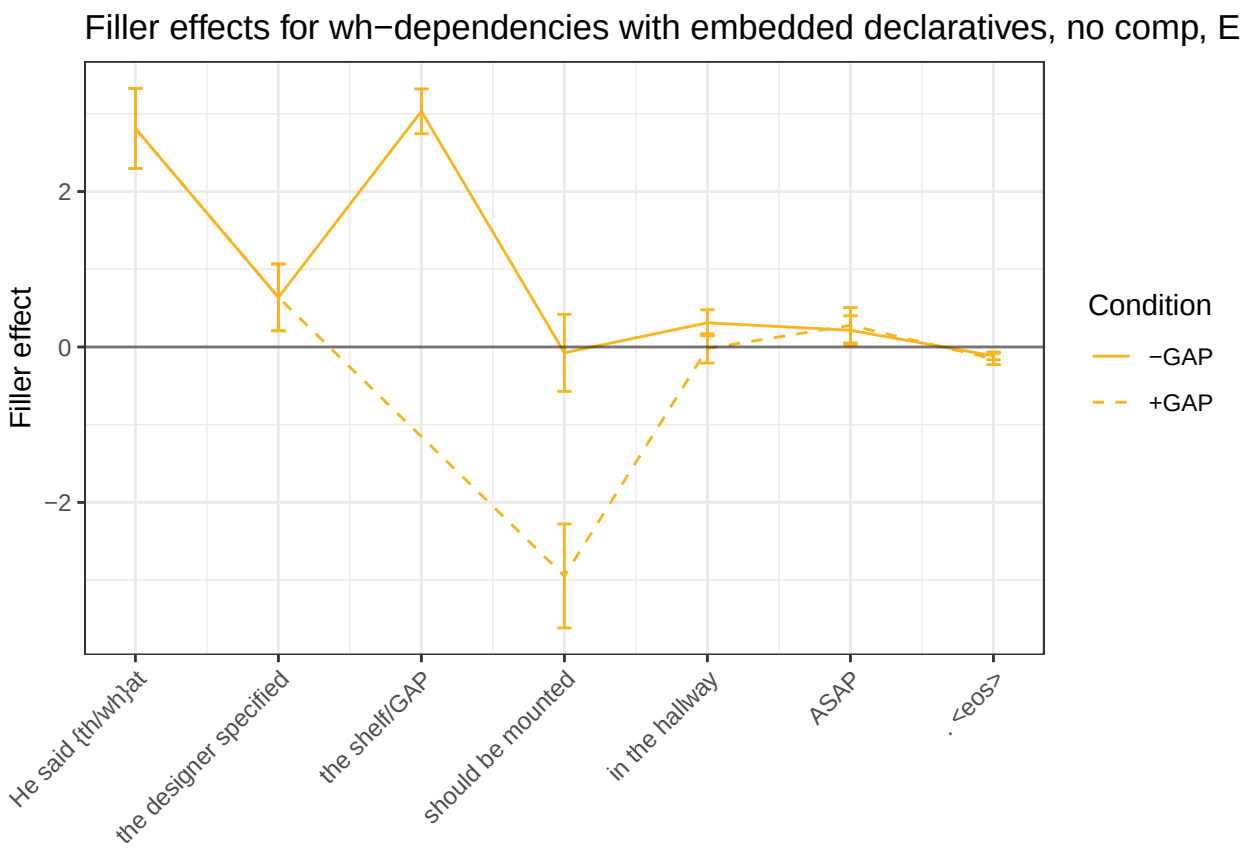
```
en_no_comp = region.surprisal(data = en_no_comp)
```

```
plot = raw.surprisal.plot(data = en_no_comp, name = "en-wh-dep-no-comp",
                          path = regions_raw, regions = REGION_EXEMPLARS_NC,
                          color_choice = c("#F7B521"))
plot + ggtitle("Raw surprisal for wh-dependencies with embedded declaratives, no comp, EN")
```



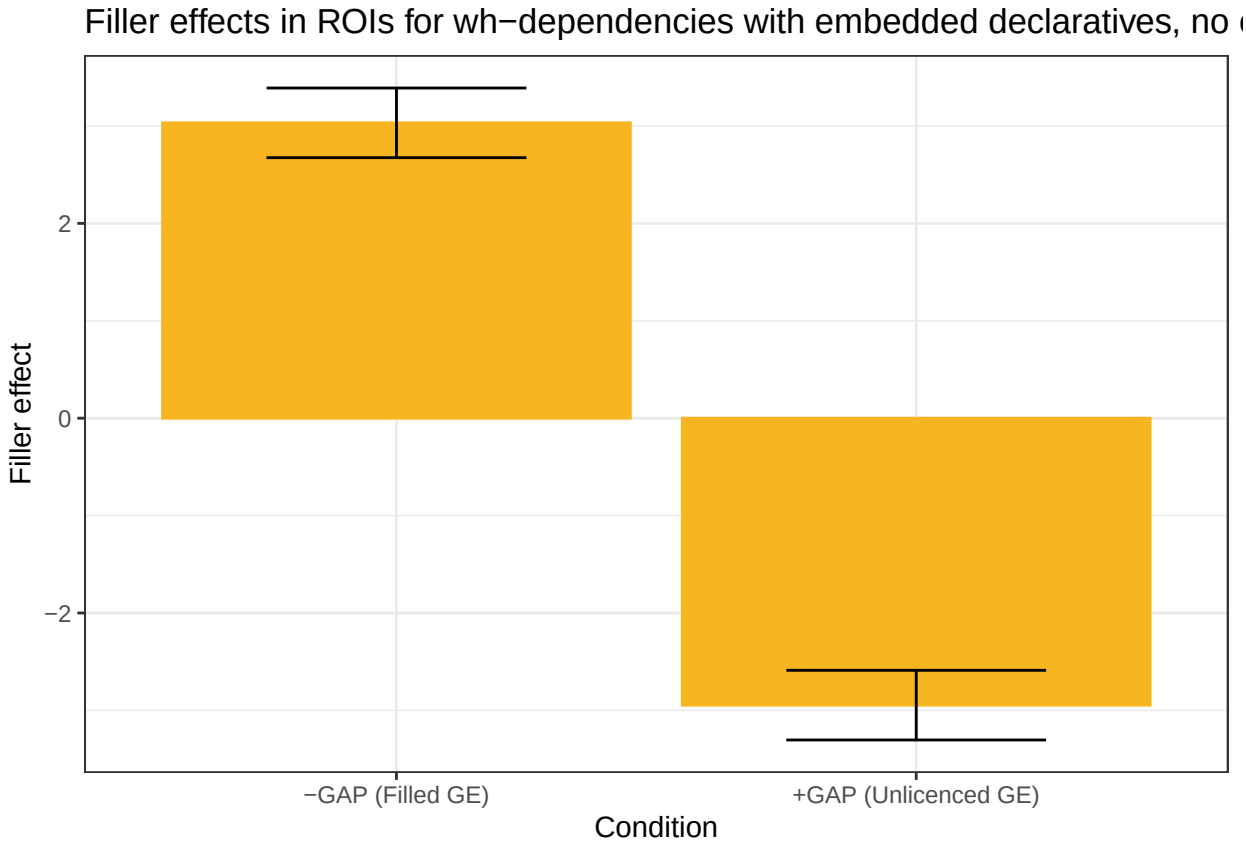

Calculating filler effects and plotting them by region:

```
en_no_comp_fe = fe.calculation(data = en_no_comp)
plot = fe.regions.plot(data = en_no_comp_fe, name = "en-wh-dep-no-comp", path = regions_fe,
                       regions = REGION_EXEMPLARS_NC, color_choice = c("#F7B521"))
plot + ggtitle("Filler effects for wh-dependencies with embedded declaratives, no comp, EN")
```




Calculating and plotting filler effects in ROIs:

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

en_no_comp_fe_roi = fe.roi.stats(en_no_comp_fe)
en_no_comp_fe_roi$dependency = "Wh"
en_no_comp_fe_roi$language = "English"
plot = fe.roi.plot(data = en_no_comp_fe_roi, name = "en-wh-dep-no-comp", path = fe_roi, color_choice = c
plot + ggtitle("Filler effects in ROIs for wh-dependencies with embedded declaratives, no comp, EN")
```




**Embedded questions (island) condition, Norwegian wh-dependencies**

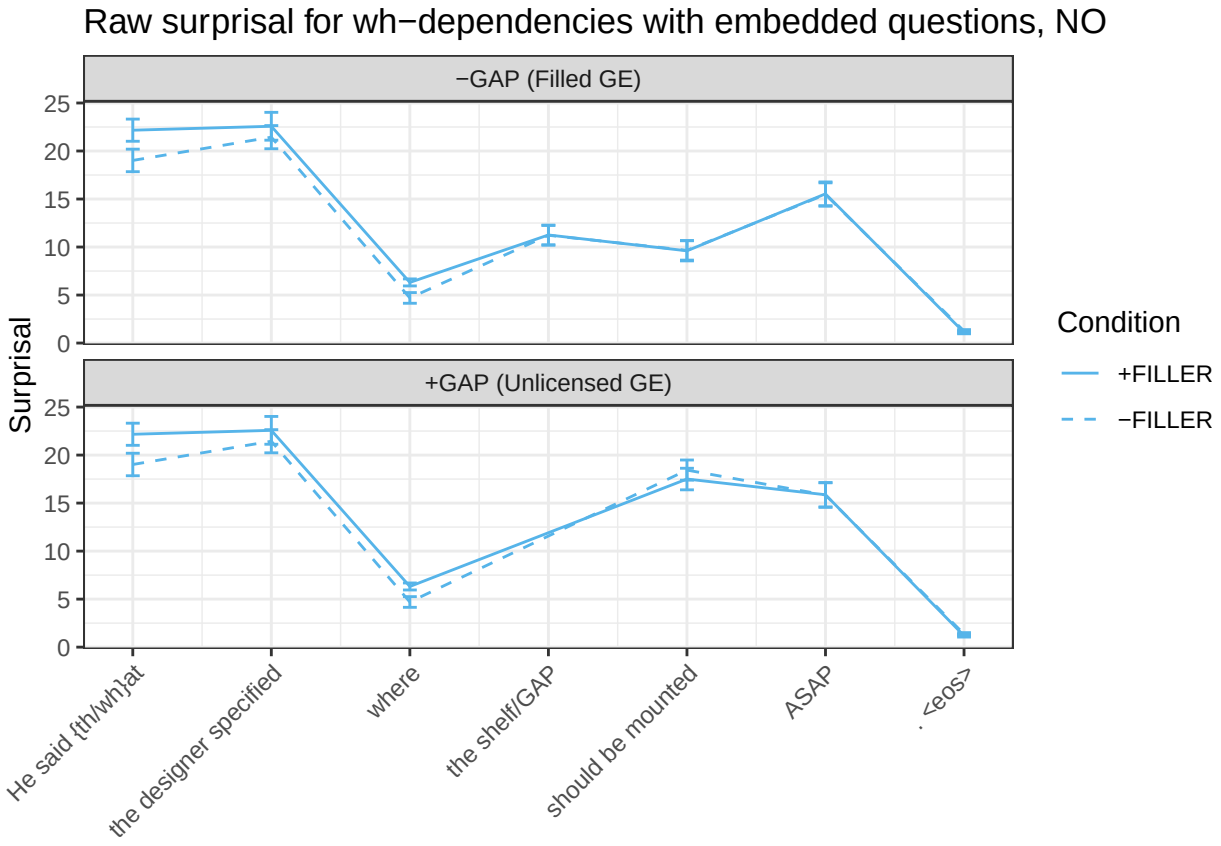
New regions:

```
REGION_ORDER = c("prefix", "embed", "comp2", "subj", "verb", "end", "EOS")
REGION_EXEMPLARS = c("He said {th/wh}at", "the designer specified", "where",
                     "the shelf/GAP", "should be mounted", "ASAP", ". <eos>")
```

```
# Changing the data according to the ROIs
wh_island = wh_island %>%
  mutate(region = if_else(region == "that1" | region == "wh-subj", "prefix", region),
         region = if_else(region == "that2" | region == "wh-word", "comp2", region),
         region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

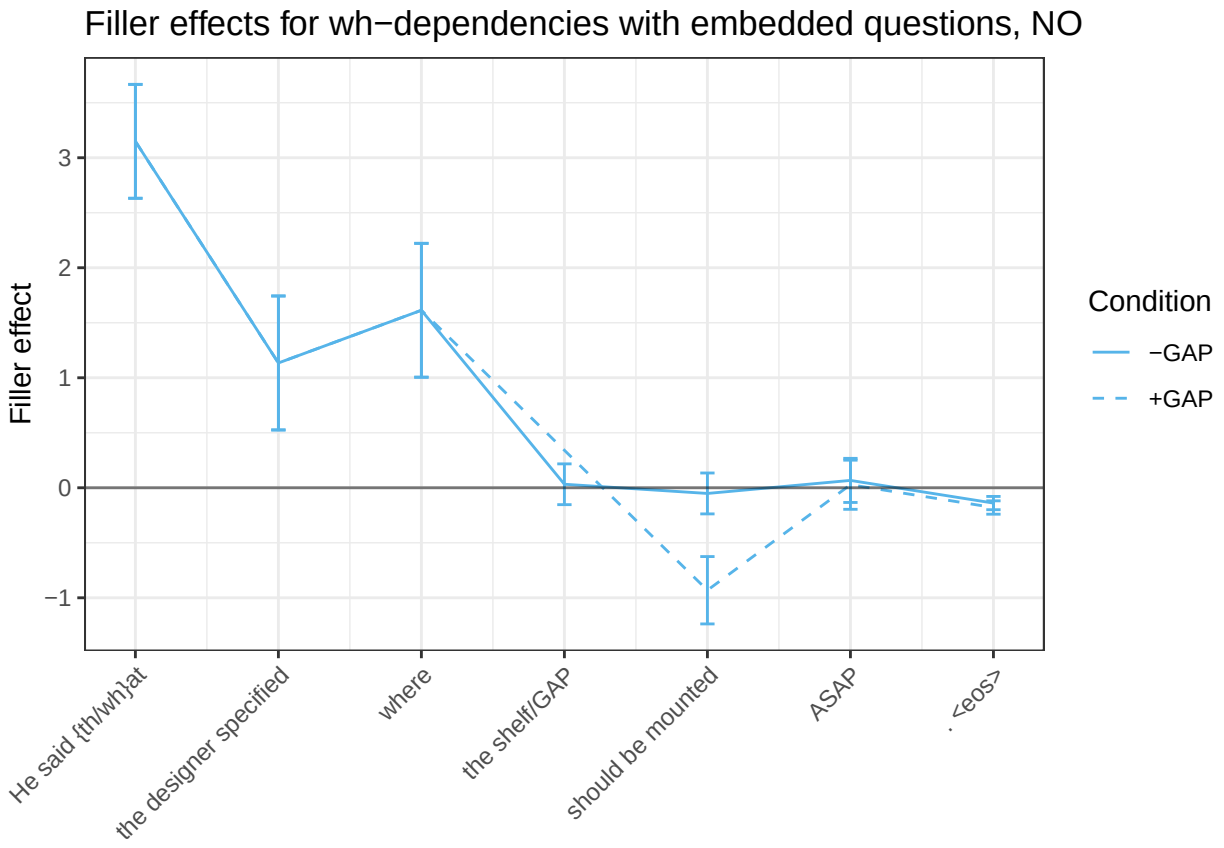
Aggregating the data and plotting raw surprisal values:

```
wh_island = region.surprisal(data = wh_island)
plot = raw.surprisal.plot(data = wh_island, name = "no-wh-dep-island", regions = REGION_EXEMPLARS, path
plot + ggtitle("Raw surprisal for wh-dependencies with embedded questions, NO")
```



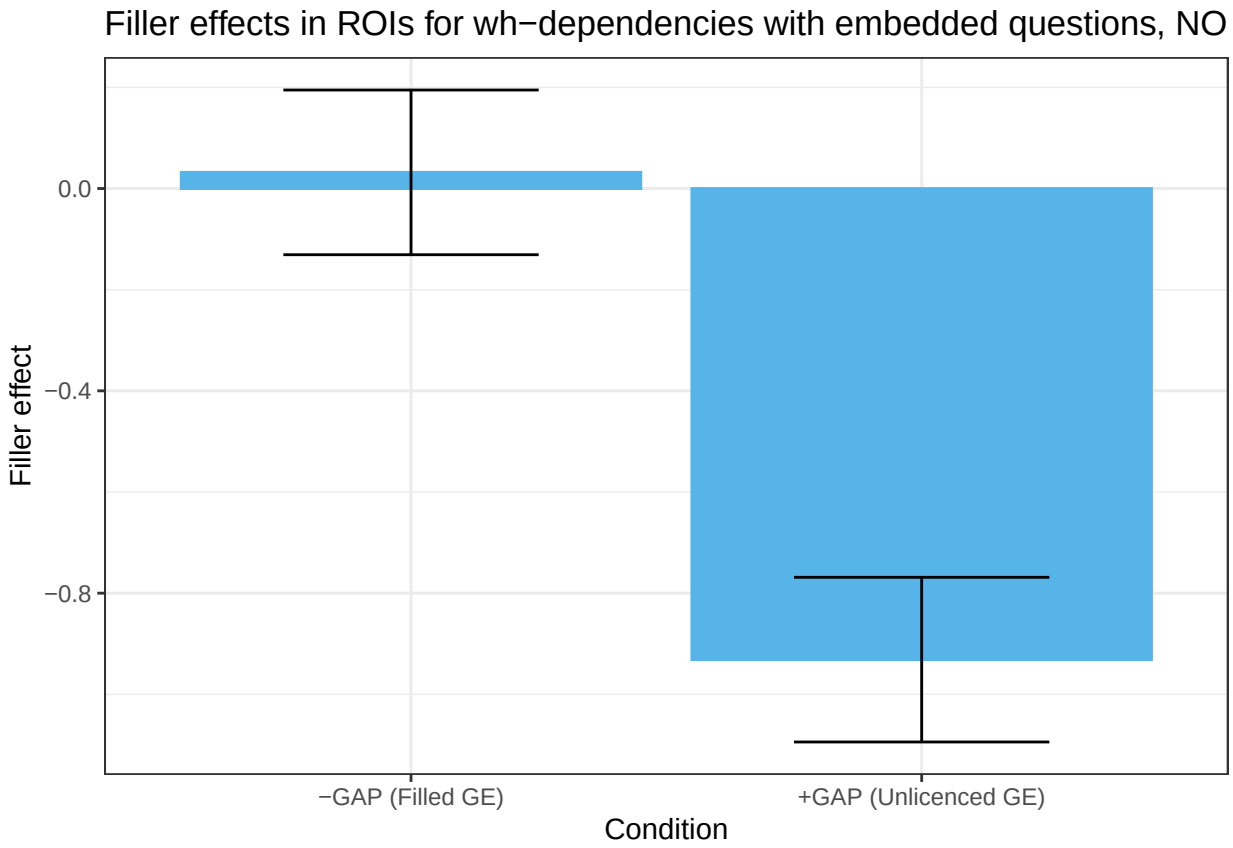

Calculating filler effects and plotting them by region:

```
wh_island_fe = fe.calculation(data = wh_island)
plot = fe.regions.plot(data = wh_island_fe, name = "no-wh-dep-island", path = regions_fe,
                       regions = REGION_EXEMPLARS, color_choice = "#56B4E9")
plot + ggtitle("Filler effects for wh-dependencies with embedded questions, NO")
```




Calculating and plotting filler effects in ROIs:

```
wh_island_fe = wh_island_fe %>%
  filter(region == "subj" & 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"
plot = fe.roi.plot(data = wh_island_fe_roi, name = "no-wh-dep-island", path = fe_roi, color_choice = c(
plot + ggtitle("Filler effects in ROIs for wh-dependencies with embedded questions, NO")
```

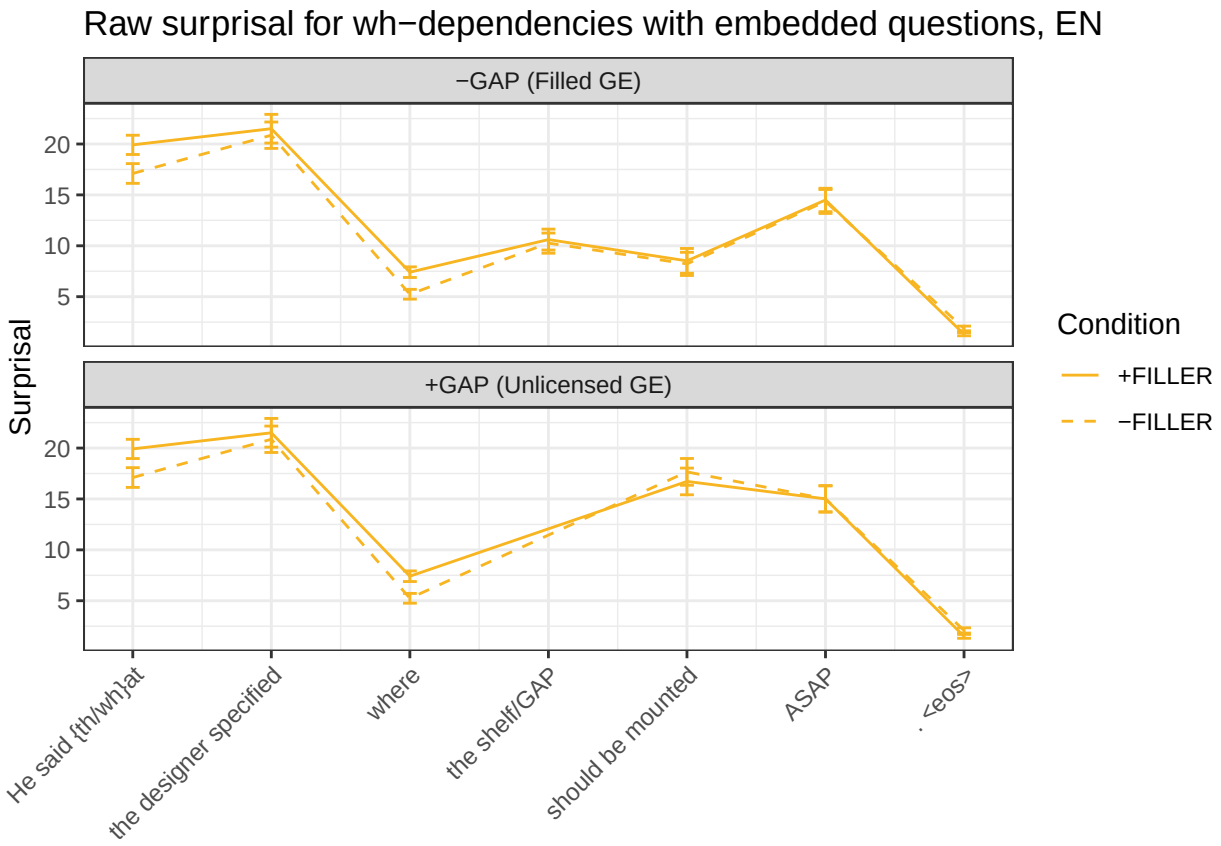



**English wh-island**

```
# Changing the data according to the ROIs
en_island = en_island %>%
  mutate(region = if_else(region == "that1" | region == "wh-subj", "prefix", region),
         region = if_else(region == "that2" | region == "wh-word", "comp2", region),
         region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

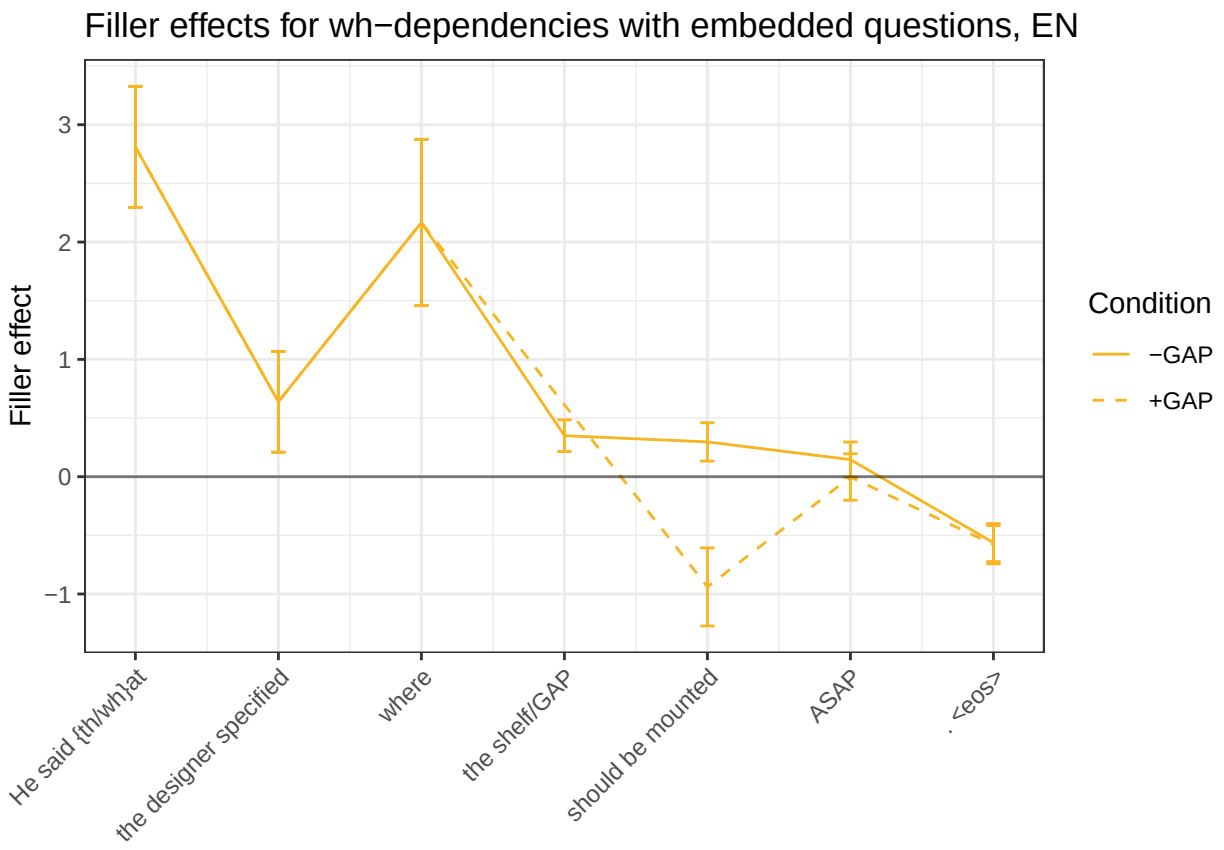
Aggregating the data and plotting raw surprisal values:

```
en_island = region.surprisal(data = en_island)
plot = raw.surprisal.plot(data = en_island, name = "en-wh-dep-island", regions = REGION_EXEMPLARS, path
plot + ggtitle("Raw surprisal for wh-dependencies with embedded questions, EN")
```



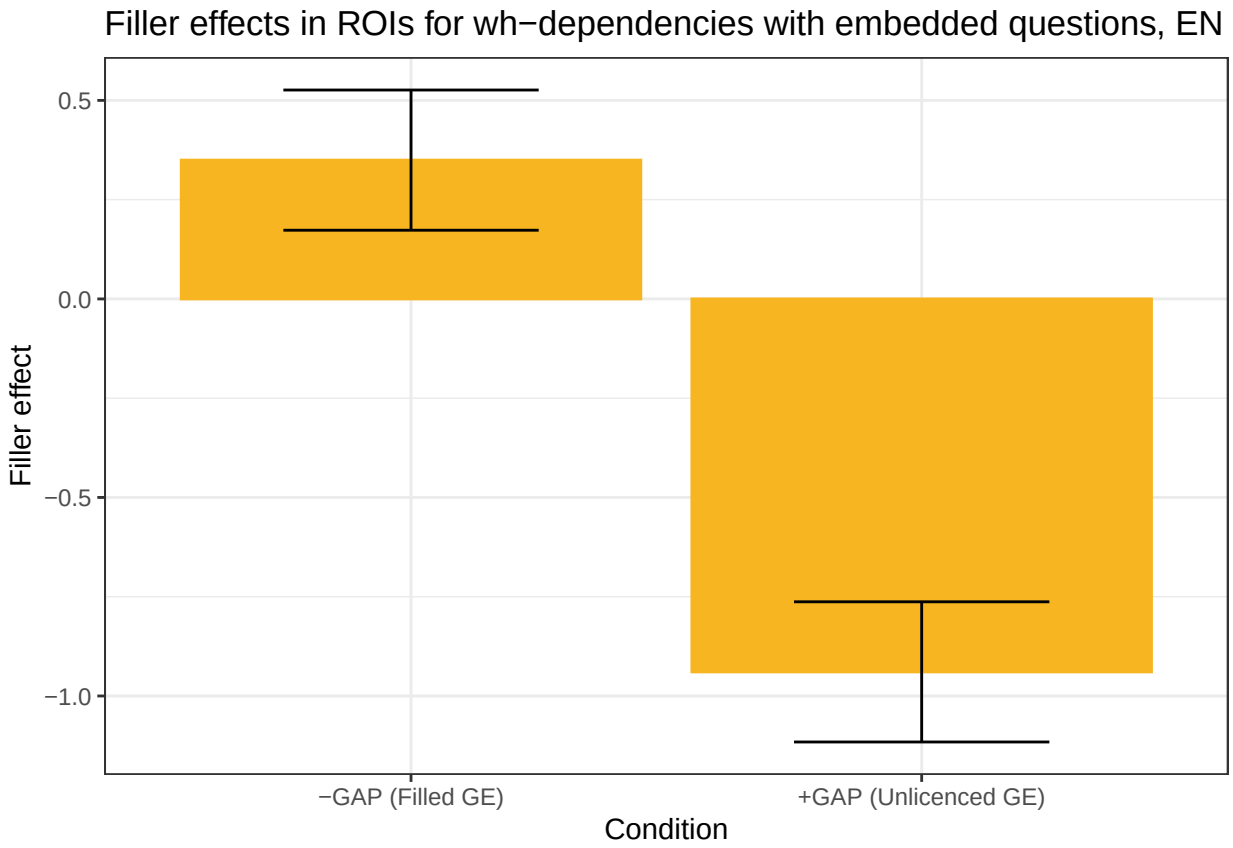

Calculating filler effects and plotting them by region:

```
en_island_fe = fe.calculation(data = en_island)
plot = fe.regions.plot(data = en_island_fe, name = "en-wh-dep-island", path = regions_fe,
                       regions = REGION_EXEMPLARS, color_choice = "#F7B521")
plot + ggtitle("Filler effects for wh-dependencies with embedded questions, EN")
```




Calculating and plotting filler effects in ROIs:

```
en_island_fe = en_island_fe %>%
  filter(region == "subj" & 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"
plot = fe.roi.plot(data = en_island_fe_roi, name = "en-wh-dep-island", path = fe_roi, color_choice = c(
plot + ggtitle("Filler effects in ROIs for wh-dependencies with embedded questions, EN")
```



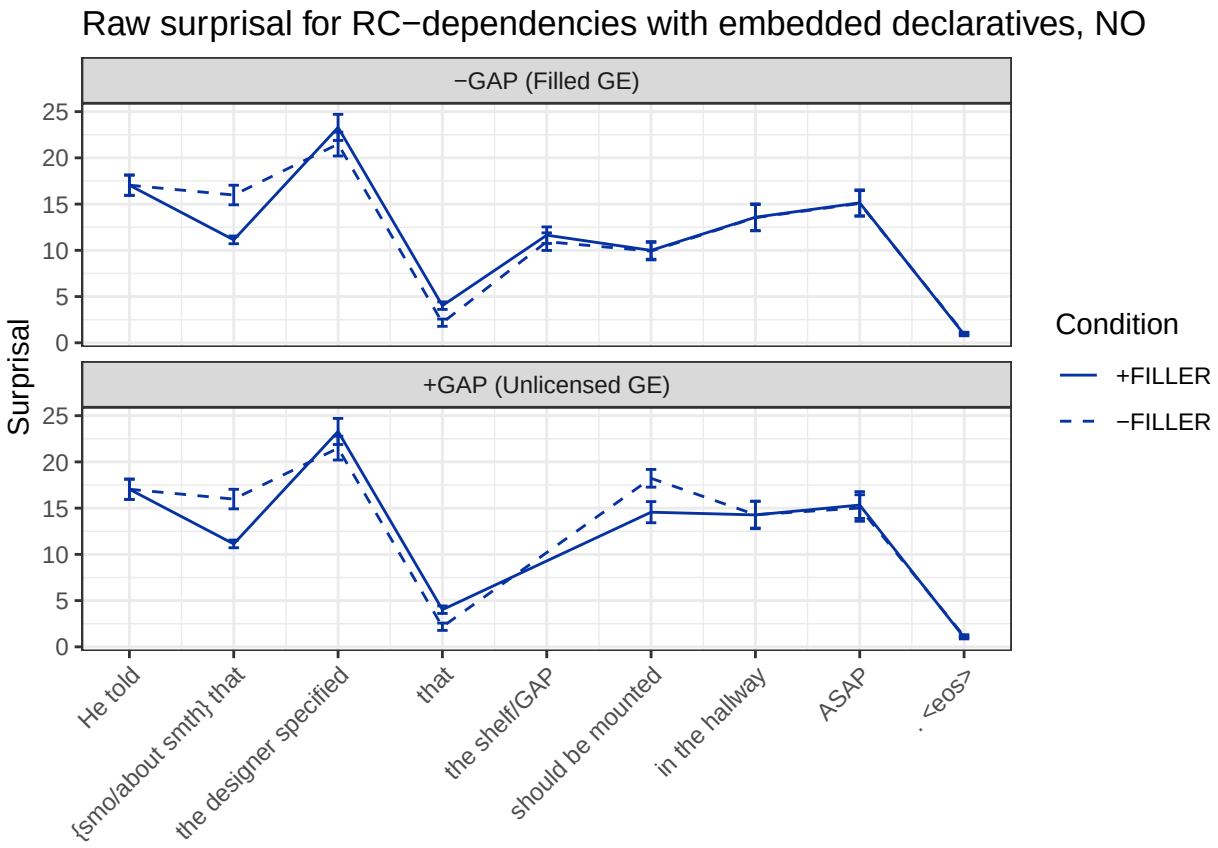

**Embedded declaratives (control) condition, RC-dependencies**

```
REGION_ORDER = c("prefix", "pp", "embed", "comp2", "subj", "verb", "adjunct", "end", "EOS")
REGION_EXEMPLARS = c("He told", "{smo/about smth} that", "the designer specified", "that",
                     "the shelf/GAP", "should be mounted", "in the hallway", "ASAP", ". <eos>")
```

```
# Changing the data according to the ROIs
rc_control = rc_control %>%
  mutate(region = if_else(region == "that1" | region == "rc-head-subj" |
                            region == "pp-add" | region == "rp", "pp", region),
         region = if_else(region == "that2" | region == "wh-word", "comp2", region),
         region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

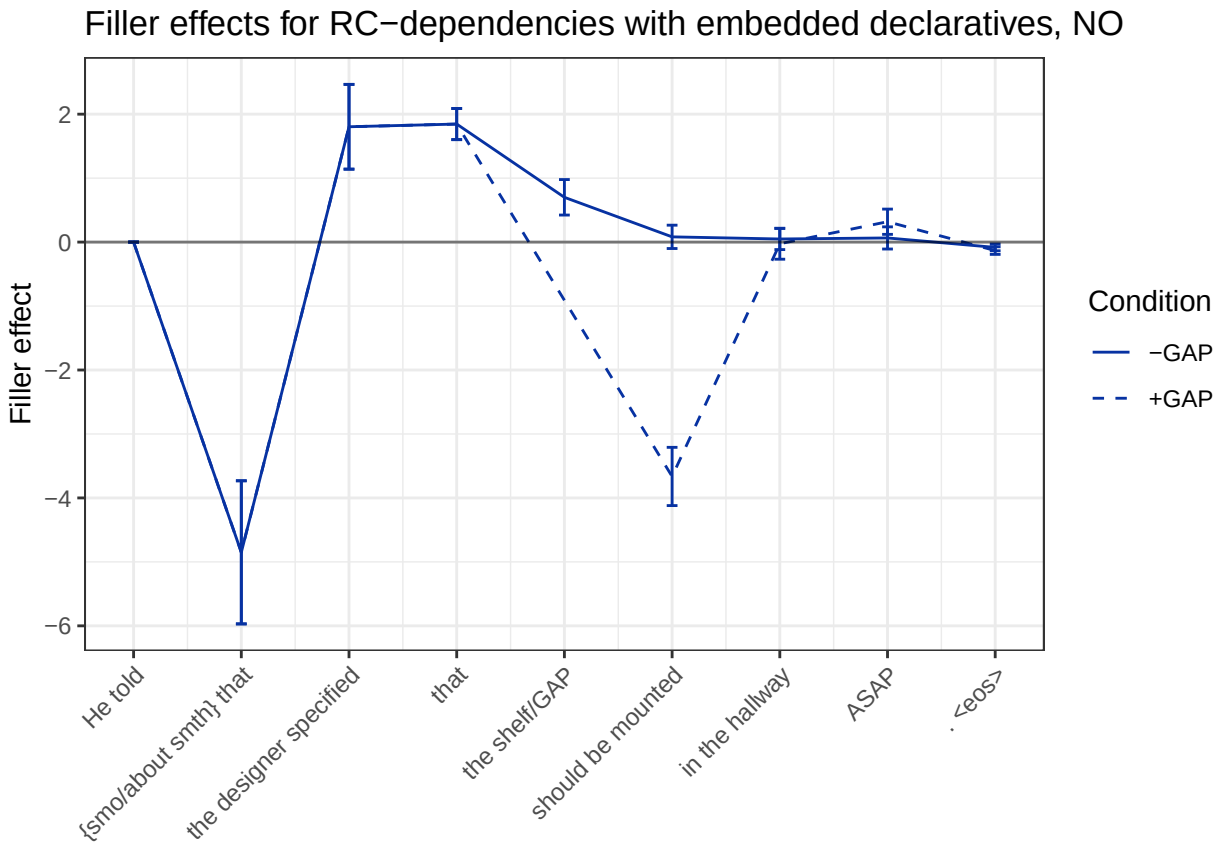
Aggregating the data and plotting raw surprisal values:

```
rc_control = region.surprisal(data = rc_control)
plot = raw.surprisal.plot(data = rc_control, name = "no-rc-dep-control", path = regions_raw,
                      regions = REGION_EXEMPLARS, color_choice = c("#0732A2"))
plot + ggtitle("Raw surprisal for RC-dependencies with embedded declaratives, NO")
```



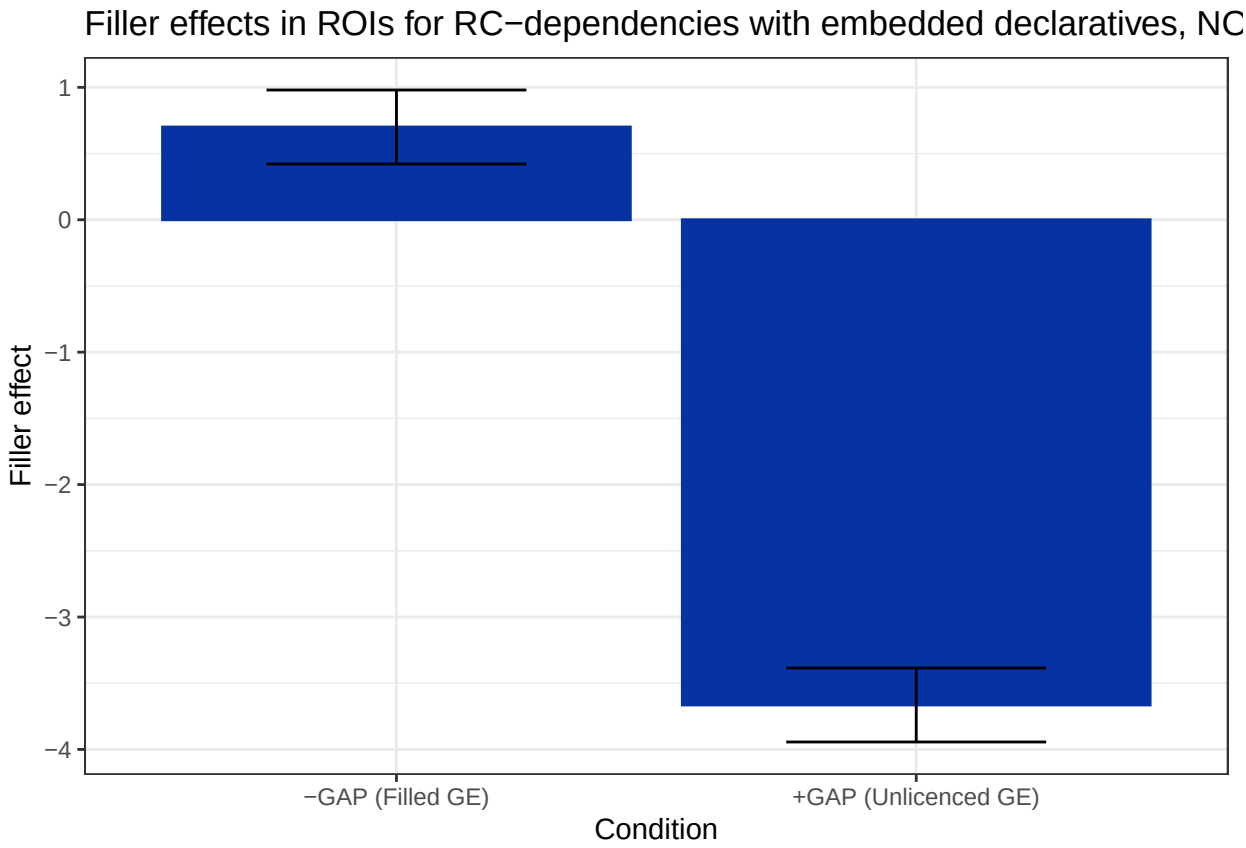

Calculating filler effects and plotting them by region:

```
rc_control_fe = fe.calculation(data = rc_control)
plot = fe.regions.plot(data = rc_control_fe, name = "no-rc-dep-control", path = regions_fe,
                       regions = REGION_EXEMPLARS, color_choice = c("#0732A2"))
plot + ggtitle("Filler effects for RC-dependencies with embedded declaratives, NO")
```




Calculating and plotting filler effects in ROIs:

```
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"
plot = fe.roi.plot(data = rc_control_fe_roi, name = "no-rc-dep-control", path = fe_roi, color_choice = c
plot + ggtitle("Filler effects in ROIs for RC-dependencies with embedded declaratives, NO")
```




**Norwegian, RC-dependency, no-complementizer condition**

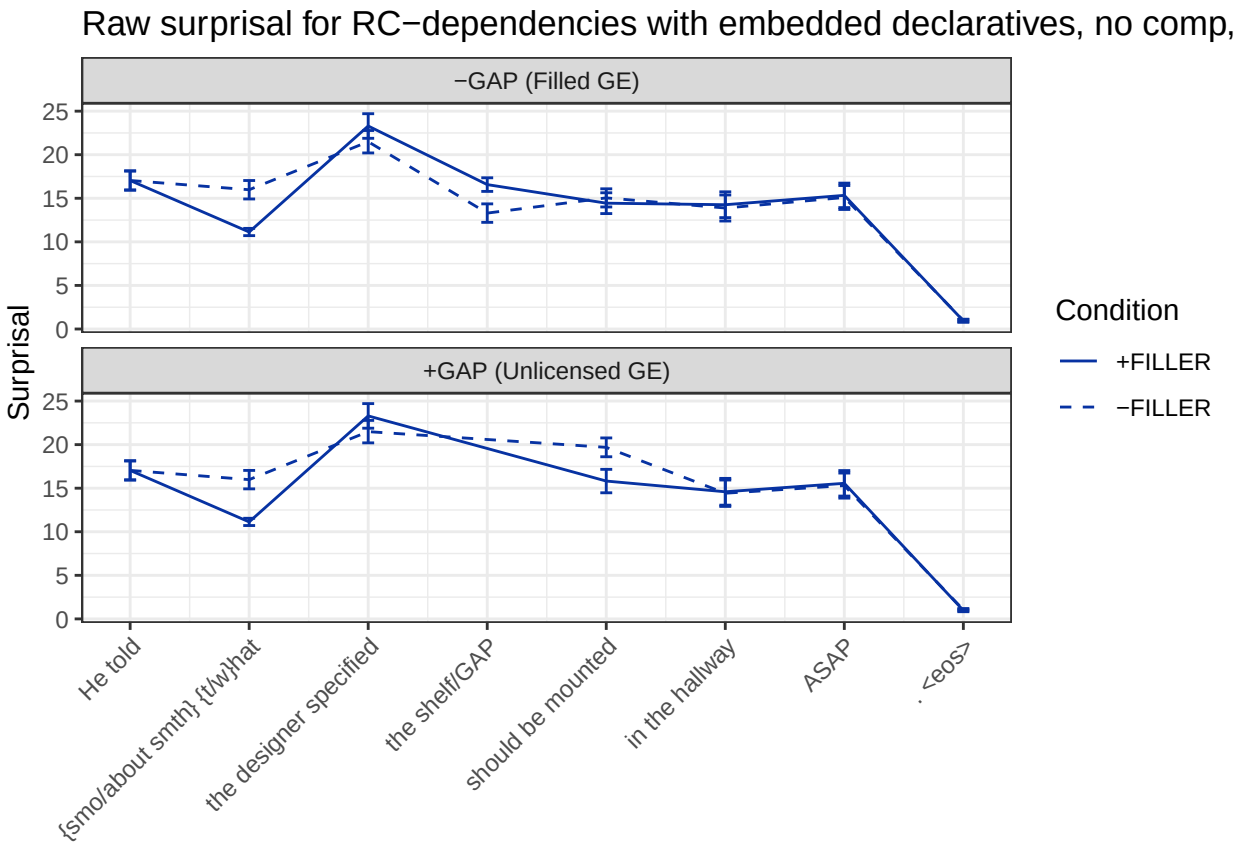
```
# New regions without the complementizer
REGION_ORDER_NC = c("prefix", "pp", "embed", "subj", "verb", "adjunct", "end", "EOS")
REGION_EXEMPLARS_NC = c("He told", "{smo/about smth} {t/w}hat", "the designer specified",
                        "the shelf/GAP", "should be mounted", "in the hallway", "ASAP", ". <eos>")
```

```
# Changing the data according to the ROIs
rc_no_comp = rc_no_comp %>%
  mutate(region = if_else(region == "that1" | region == "rc-head-subj" |
                            region == "pp-add" | region == "rp", "pp", region),
         region = factor(region, levels=REGION_ORDER_NC)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

Aggregating the data and plotting raw surprisal values:

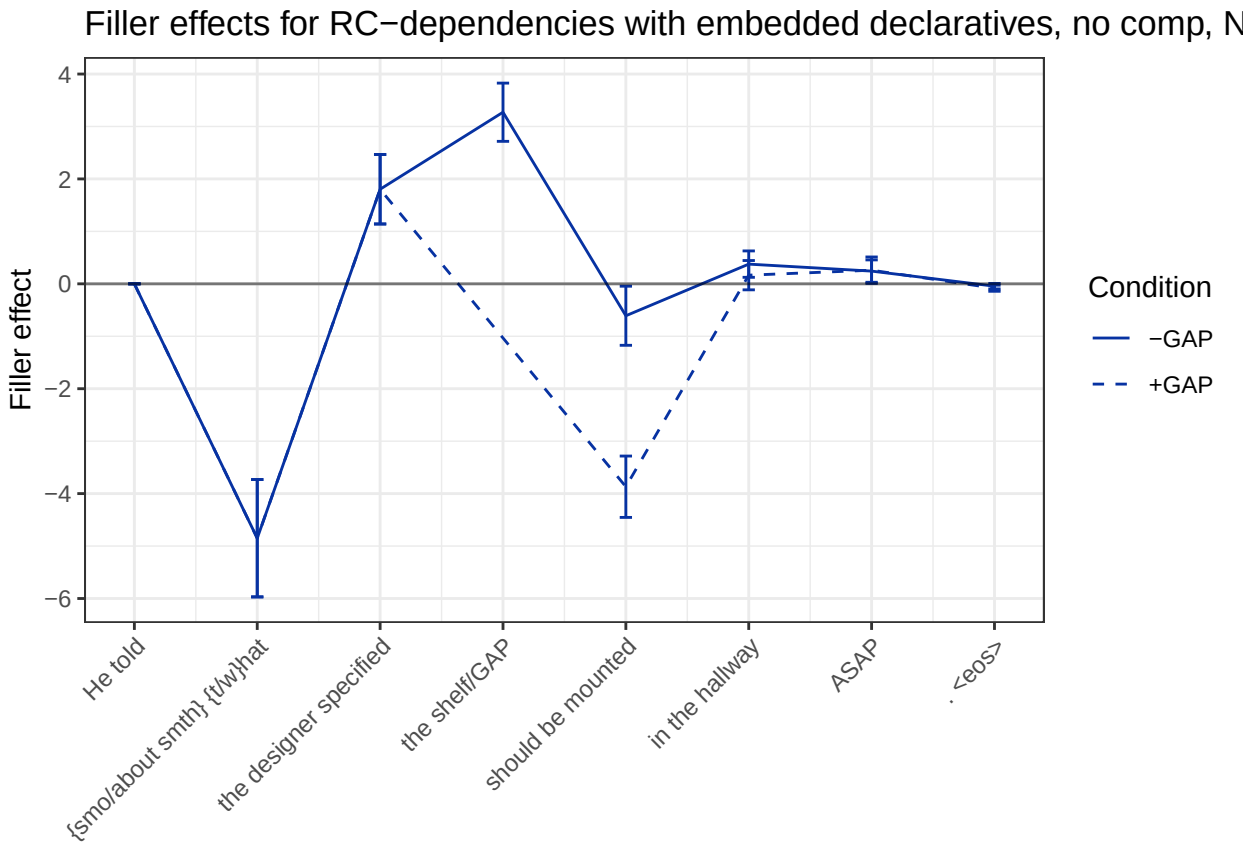
```
rc_no_comp = region.surprisal(data = rc_no_comp)
```

```
plot = raw.surprisal.plot(data = rc_no_comp, name = "no-rc-dep-no-comp",
                          path = regions_raw, regions = REGION_EXEMPLARS_NC,
                          color_choice = c("#0732A2"))
plot + ggtitle("Raw surprisal for RC-dependencies with embedded declaratives, no comp, NO")
```



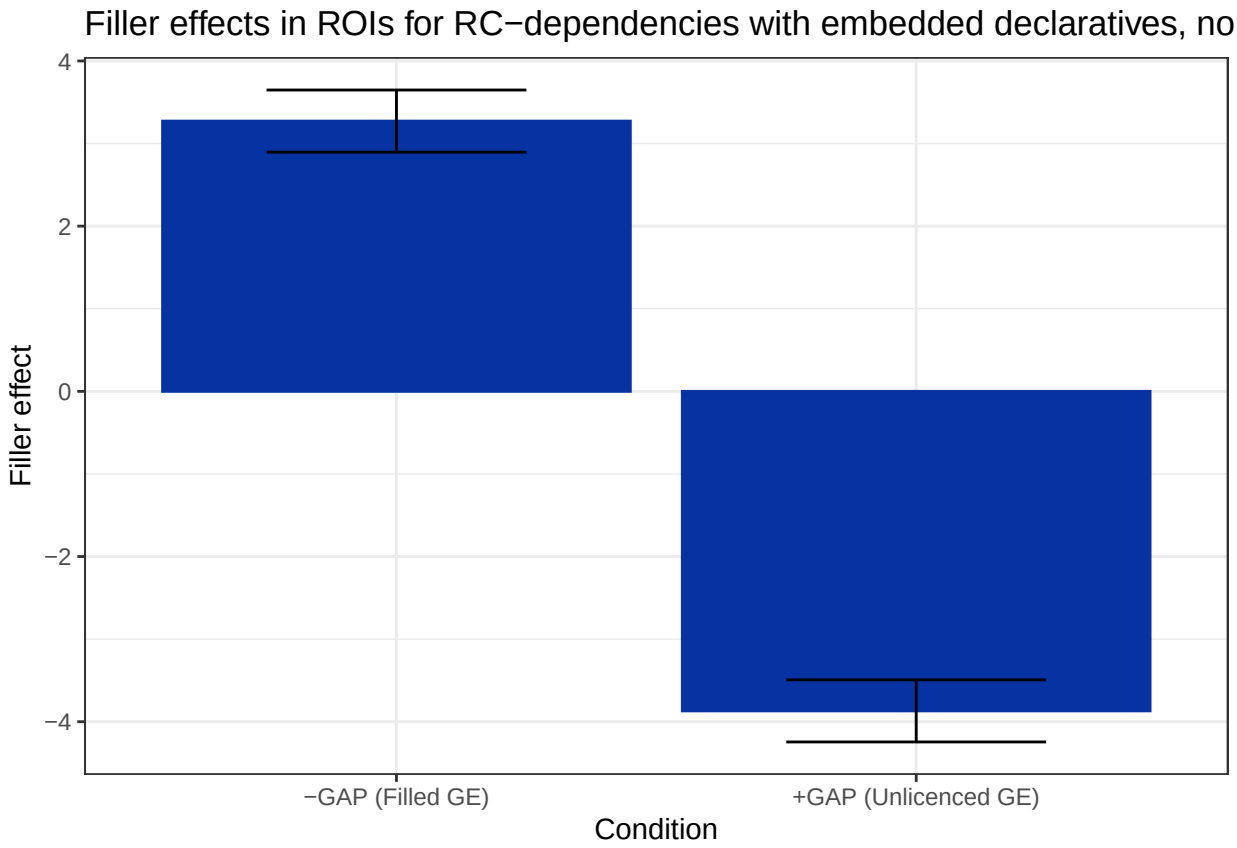

Calculating filler effects and plotting them by region:

```
rc_no_comp_fe = fe.calculation(data = rc_no_comp)
plot = fe.regions.plot(data = rc_no_comp_fe, name = "no-rc-dep-no-comp", path = regions_fe,
                       regions = REGION_EXEMPLARS_NC, color_choice = c("#0732A2"))
plot + ggtitle("Filler effects for RC-dependencies with embedded declaratives, no comp, NO")
```




Calculating and plotting filler effects in ROIs:

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

rc_no_comp_fe_roi = fe.roi.stats(rc_no_comp_fe)
rc_no_comp_fe_roi$dependency = "RC"
rc_no_comp_fe_roi$language = "Norwegian"
plot = fe.roi.plot(data = rc_no_comp_fe_roi, name = "no-rc-dep-no-comp", path = fe_roi, color_choice = c
plot + ggtitle("Filler effects in ROIs for RC-dependencies with embedded declaratives, no comp, NO")
```



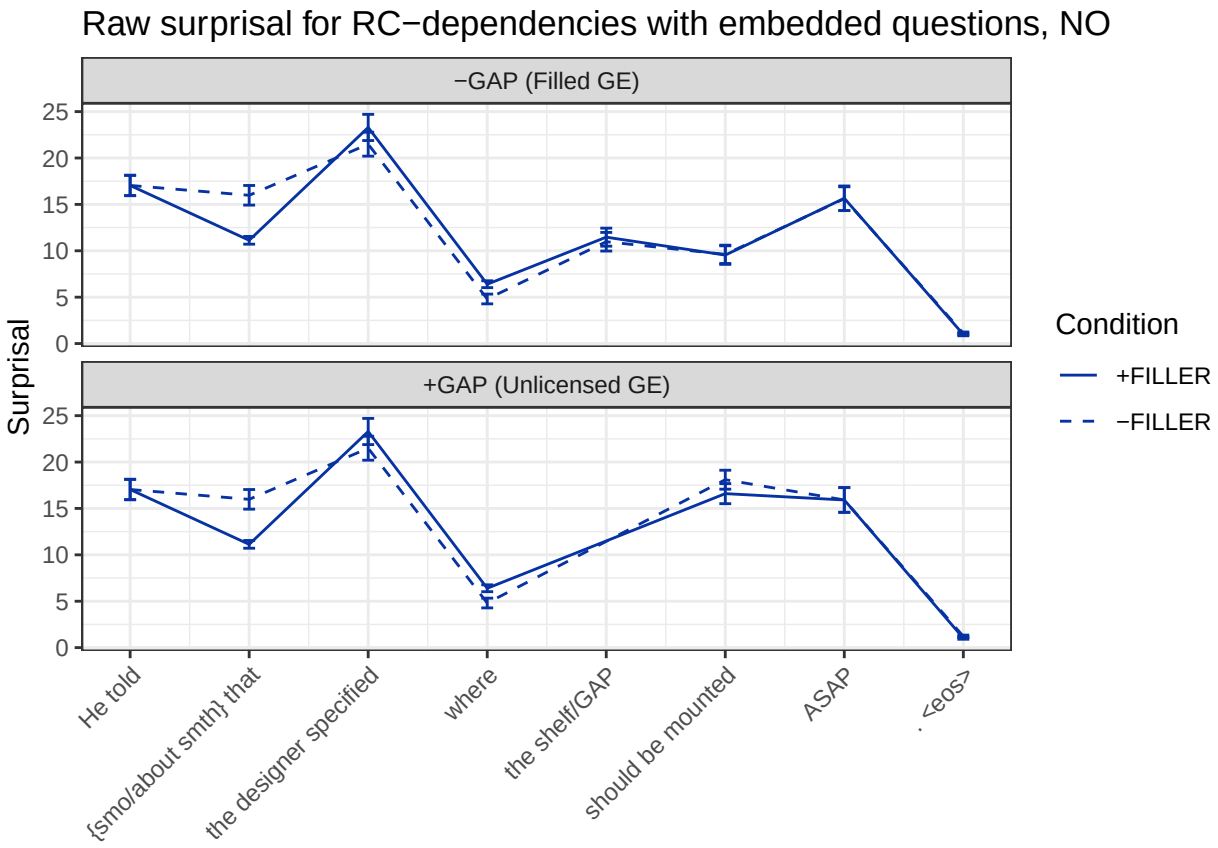

**Embedded whether-questions (island) condition, RC-dependencies**

```
REGION_ORDER = c("prefix", "pp", "embed", "comp2", "subj", "verb", "end", "EOS")
REGION_EXEMPLARS = c("He told", "{smo/about smth} that", "the designer specified", "where",
                     "the shelf/GAP", "should be mounted", "ASAP", ". <eos>")
```

```
# Changing the data according to the ROIs
rc_island = rc_island %>%
  mutate(region = if_else(region == "that1" | region == "rc-head-subj" |
                          region == "pp-add" | region == "rp", "pp", region),
         region = if_else(region == "that2" | region == "wh-word", "comp2", region),
         region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

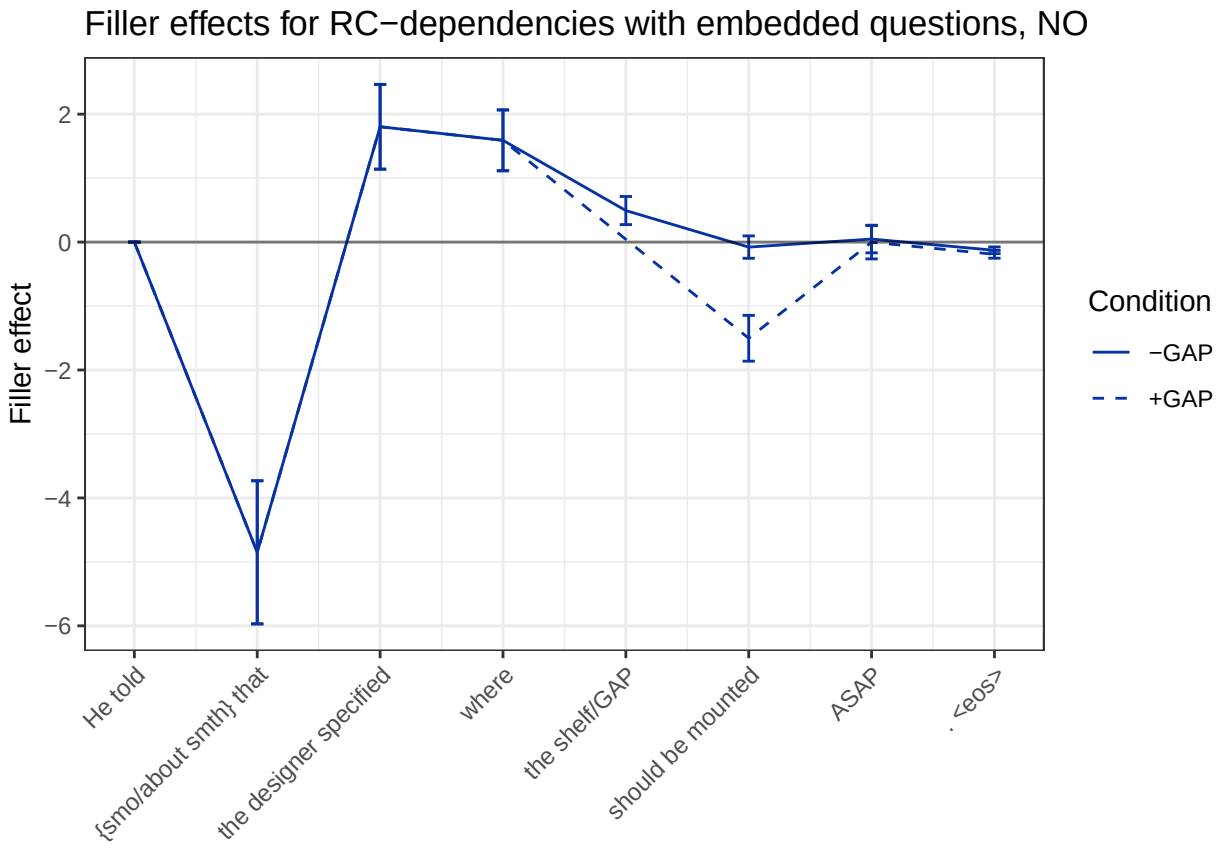
Aggregating the data and plotting raw surprisal values:

```
rc_island = region.surprisal(data = rc_island)
plot = raw.surprisal.plot(data = rc_island, name = "no-rc-dep-island", path = regions_raw,
                     regions = REGION_EXEMPLARS, color_choice = c("#0732A2"))
plot + ggtitle("Raw surprisal for RC-dependencies with embedded questions, NO")
```

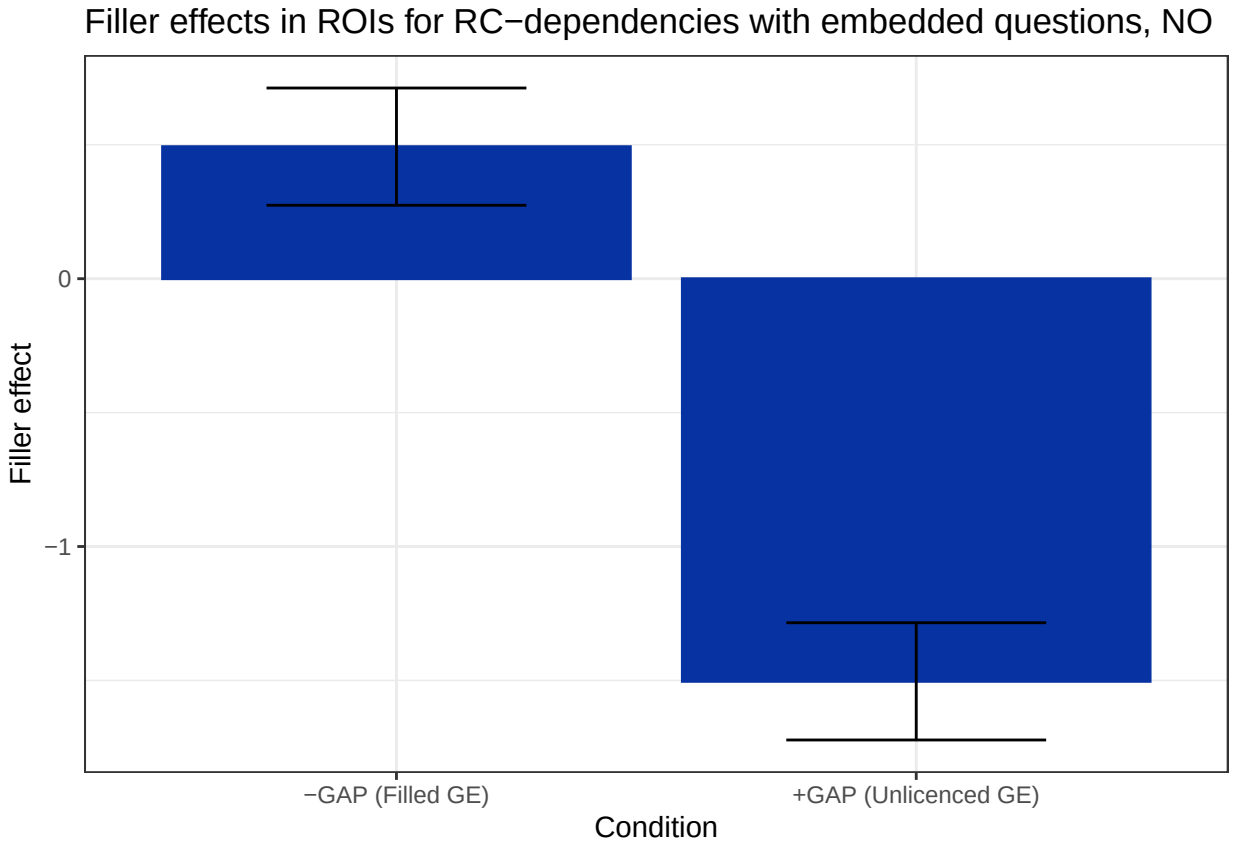



Calculating filler effects and plotting them by region:

```
rc_island_fe = fe.calculation(data = rc_island)
plot = fe.regions.plot(data = rc_island_fe, name = "no-rc-dep-island", path = regions_fe,
                       regions = REGION_EXEMPLARS, color_choice = c("#0732A2"))
plot + ggtitle("Filler effects for RC-dependencies with embedded questions, NO")
```




```
rc_island_fe = rc_island_fe %>%
  filter(region == "subj" & 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"
plot = fe.roi.plot(data = rc_island_fe_roi, name = "no-rc-dep-island", path = fe_roi, color_choice = c(
plot + ggtitle("Filler effects in ROIs for RC-dependencies with embedded questions, NO")
```

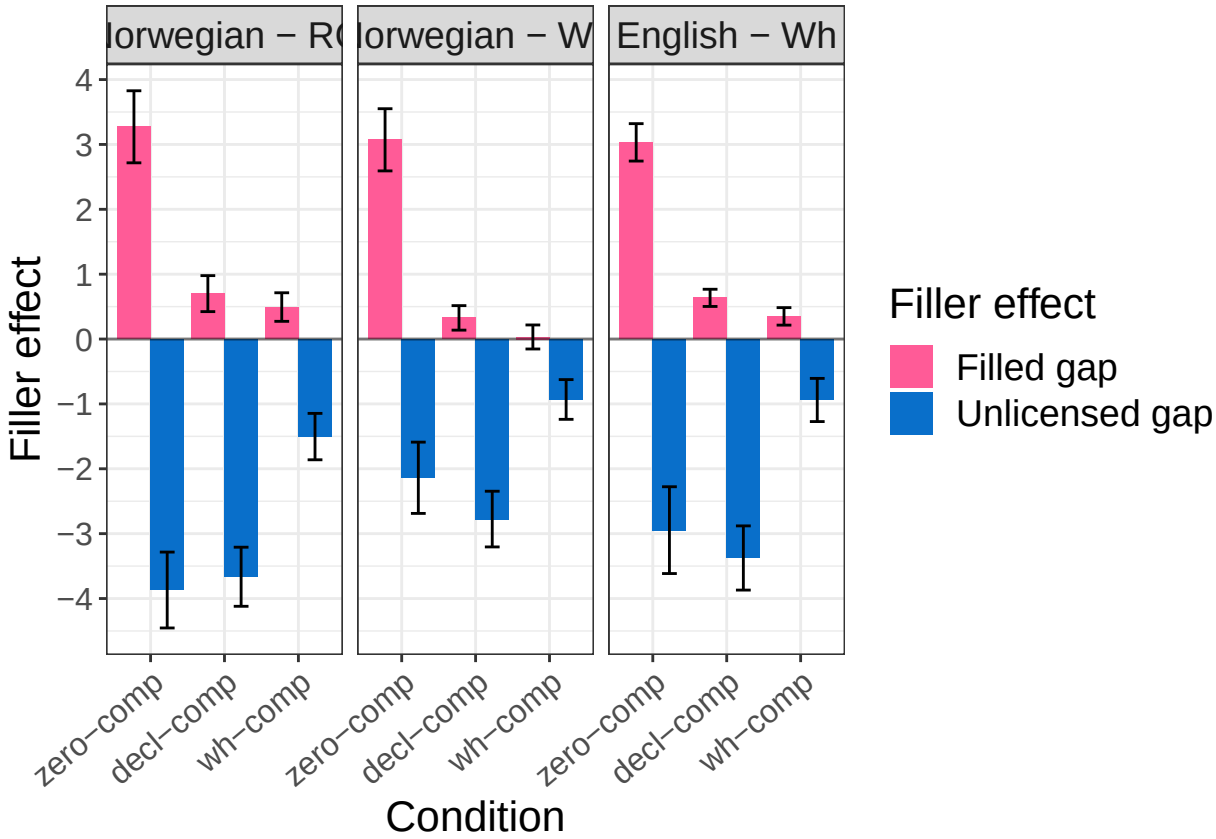



**Common plot**

```
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_no_comp_fe_roi, en_island_fe_roi,
                          wh_no_comp_fe_roi, rc_no_comp_fe_roi,
                          en_control_fe_roi))

d_filler_effect[d_filler_effect == "that-comp"] <- "decl-comp"
d_filler_effect[d_filler_effect == "no-comp"] <- "zero-comp"

d_filler_effect$gap_position <- factor(d_filler_effect$gap_position,
    levels = c('zero-comp', 'decl-comp', 'wh-comp'), 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/eq-plots/%s-eq-all.png", model_type)
ggsave(fname, width = 10)
```

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

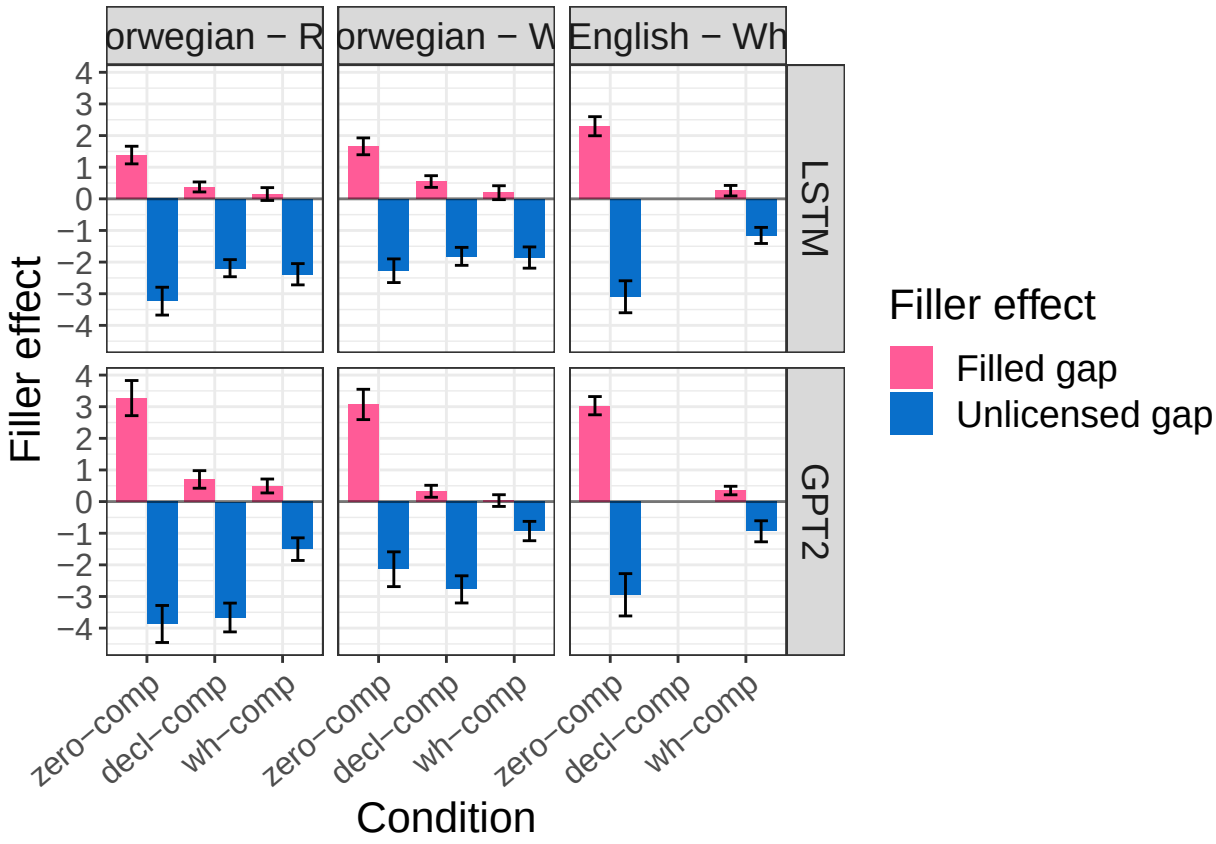
```
# Add model column and save the aggregated data
d_filler_effect['model'] = toupper(model_type)
```

```r
dfname = sprintf("../data/results/%s/eq_%s_agg.csv", model_type, model_type)
write.csv(d_filler_effect, dfname, row.names=FALSE)
```

```r
# Read in the aggregated date for both models
lstm = read.csv("../data/results/lstm/eq_lstm_agg.csv", fileEncoding = "UTF-8-BOM")
gpt2 = read.csv("../data/results/gpt2/eq_gpt2_agg.csv", fileEncoding = "UTF-8-BOM")
d = rbind(lstm, gpt2)
```

```r
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('zero-comp', 'decl-comp', 'wh-comp'), ordered = TRUE)
```

```r
d %>%
  filter(!(lang_dep == "English - Wh" & gap_position == "decl-comp")) %>%
  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/eq-plots/eq-two-models.png", width = 10, height = 7)
ggsave("plots/eq-plots/eq-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("zero-comp", "decl-comp
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" = "declCntrst",
                             "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" = "declCntrst",
                             "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/eq-stats/eq-%s.html", model_type)
stats_fname_tex = sprintf("stats/eq-stats/eq-%s.tex", model_type)
html_table |> gtsave(stats_fname_html)
latex_table |> gtsave(stats_fname_tex)
```

```
# English only
d_english = Reduce(function(x, y) merge(x, y, all=TRUE),
                   list(en_no_comp_fe_roi, en_island_fe_roi))

d_english$gap_position <- factor(d_english$gap_position,
    levels = c("no-comp", "wh-comp"), ordered = TRUE)
```

```
d_english$gap_position <- as.factor(d_english$gap_position)
contrasts(d_english$gap_position) <- c(0.5, -0.5) # control first, so expecting a bigger effect
```

```
# FGE
fge = d_english %>%
    filter(gap == "no-gap") %>%
```

```
    lmer(filler_effect ~ gap_position + (1 |sent_index), data=.)
summary(fge)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: filler_effect ~ gap_position + (1 | sent_index)
##    Data: .
##
## REML criterion at convergence: 241.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.43885 -0.48250 -0.06705  0.61362  2.32169
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  sent_index (Intercept) 0.1689   0.4110
##  Residual               0.4904   0.7003
## Number of obs: 100, groups:  sent_index, 50
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    1.69081    0.09101   18.58
## gap_position1  2.68248    0.14005   19.15
##
## Correlation of Fixed Effects:
##             (Intr)
## gap_positn1 0.000
```

```
fge_table = modelsummary(fge, output = "gt", stars = TRUE, gof_omit = ".*",
             estimate = "{estimate}{stars} ({std.error})",
             statistic = "t = {statistic}", fmt = 2,
             coef_rename = c("gap_position1" = "islandCntrst"))
stats_fname = sprintf("stats/eq-stats/eq-%s-", model_type)
fge_table |> gtsave(paste0(stats_fname, "english-fge.html"))
```

```
# UGE
uge = d_english %>%
    filter(gap == "gap") %>%
    lmer(filler_effect ~ gap_position + (1 |sent_index), data=.)
summary(uge)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: filler_effect ~ gap_position + (1 | sent_index)
##    Data: .
##
## REML criterion at convergence: 405.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2590 -0.3562  0.0178  0.4241  2.9940
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  sent_index (Intercept) 1.341    1.158
##  Residual               2.292    1.514
```

```
## Number of obs: 100, groups:  sent_index, 50
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept)    -1.9427     0.2230  -8.711
## gap_position1  -2.0068     0.3028  -6.627
##
## Correlation of Fixed Effects:
##             (Intr)
## gap_positn1 0.000
```

```
uge_table = modelsummary(uge, output = "gt", stars = TRUE, gof_omit = ".*",
             estimate = "{estimate}{stars} ({std.error})",
             statistic = "t = {statistic}", fmt = 2,
             coef_rename = c("gap_position1" = "islandCntrst"))
uge_table |> gtsave(paste0(stats_fname, "english-uge.html"))
```

### Between-language comparison

```
d_zero_wh = d %>% filter(dependency == "Wh" & (gap_position == "wh-comp" |
                                                 gap_position == "zero-comp"))
```

```
d_zero_wh$gap_position <- factor(d_zero_wh$gap_position,
    levels = c('zero-comp', 'wh-comp'), ordered = TRUE)
contrasts(d_zero_wh$gap_position) <- c(0.5, -0.5) # control first, so expecting a bigger effect
d_zero_wh$language <- factor(d_zero_wh$language,
    levels = c('English', 'Norwegian'), ordered = TRUE)
contrasts(d_zero_wh$language) <- c(-0.5, 0.5) # En first, so expecting a smaller effect
```

```
# FGE
fge = d_zero_wh %>%
    filter(gap == "no-gap") %>%
    lmer(filler_effect ~ gap_position*language + (1|sent_index) +(1|model), data=.)
summary(fge)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: filler_effect ~ gap_position * language + (1 | sent_index) +
##     (1 | model)
##    Data: .
##
## REML criterion at convergence: 1167.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0687 -0.6206 -0.0177  0.5275  5.7202
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  sent_index (Intercept) 0.06212  0.2492
##  model      (Intercept) 0.12986  0.3604
##  Residual               1.00618  1.0031
## Number of obs: 400, groups:  sent_index, 50; model, 2
##
## Fixed effects:
##                         Estimate Std. Error t value
```

```
## (Intercept)               1.3615     0.2621   5.195
## gap_position1             2.3064     0.1003  22.993
## language1                -0.2443     0.1003  -2.435
## gap_position1:language1  -0.1075     0.2006  -0.536
##
## Correlation of Fixed Effects:
##             (Intr) gp_ps1 langg1
## gap_positn1 0.000
## language1   0.000  0.000
## gp_pstn1:l1 0.000  0.000  0.000
```

```
fge_table = modelsummary(fge, output = "gt", stars = TRUE, gof_omit = ".*",
             estimate = "{estimate}{stars} ({std.error})",
             statistic = "t = {statistic}", fmt = 2,
             coef_rename = c("gap_position1" = "condition",
                             "language1" = "language"))
fge_table |> gtsave(paste0(stats_fname, "between-lang-fge.html"))
```

```
# FGE
uge = d_zero_wh %>%
    filter(gap == "gap") %>%
    lmer(filler_effect ~ gap_position*language + (1|sent_index) +(1|model), data=.)
summary(uge)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: filler_effect ~ gap_position * language + (1 | sent_index) +
##     (1 | model)
##    Data: .
##
## REML criterion at convergence: 1487
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.8211 -0.5702  0.0004  0.5921  4.2831
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  sent_index (Intercept) 0.35092  0.5924
##  model      (Intercept) 0.05287  0.2299
##  Residual               2.14386  1.4642
## Number of obs: 400, groups:  sent_index, 50; model, 2
##
## Fixed effects:
##                          Estimate Std. Error t value
## (Intercept)               -1.9172     0.1970  -9.731
## gap_position1             -1.3925     0.1464  -9.511
## language1                  0.2346     0.1464   1.602
## gap_position1:language1    1.1607     0.2928   3.964
##
## Correlation of Fixed Effects:
##             (Intr) gp_ps1 langg1
## gap_positn1 0.000
## language1   0.000  0.000
## gp_pstn1:l1 0.000  0.000  0.000
```

```
uge_table = modelsummary(uge, output = "gt", stars = TRUE, gof_omit = ".*",
             estimate = "{estimate}{stars} ({std.error})",
             statistic = "t = {statistic}", fmt = 2,
             coef_rename = c("gap_position1" = "condition",
                             "language1" = "language"))
uge_table |> gtsave(paste0(stats_fname, "between-lang-uge.html"))
```