

# Analysis of Whether 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

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

filename_wh = sprintf("../data/results/%s/whether_wh_result.csv", model_type) # Norwegian wh
filename_rc = sprintf("../data/results/%s/whether_rc_result.csv", model_type) # Norwegian RC
filename_en = sprintf("../data/results/%s/whether_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

```
# 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. whether-EQ)
control = df[endsWith(df$condition, "that-comp"),] # decl that complementizer
island = df[endsWith(df$condition, "wh-comp"),] # eq whether complementizer
```

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

```
REGION_ORDER = c("prefix", "embed", "comp2", "subj", "verb", "obj", "end", "EOS")
REGION_EXEMPLARS = c("He knows {th/wh}at", "the professor could tell", "that",
  "the students", "hated", "the book/GAP", "from day one", ". <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")
```

```
# Changing the data according to the ROIs
```

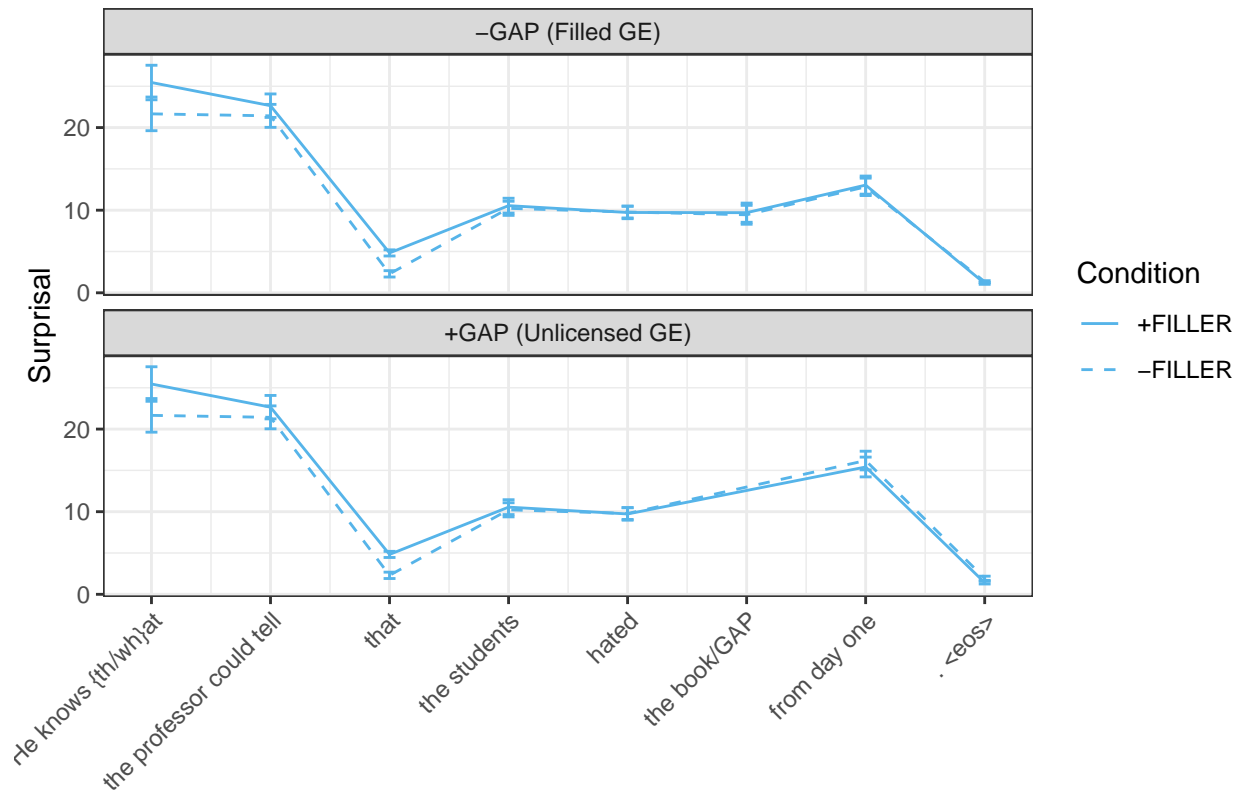
```
wh_control = wh_control %>%
  mutate(region = if_else(region == "that1" | region == "wh-obj", "prefix", region),
    region = if_else(region == "that2" | region == "whether", "comp2", region),
    region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

Norwegian, wh-dependency, control condition    Aggregating the data and plotting raw surprisal values:

```
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")
```

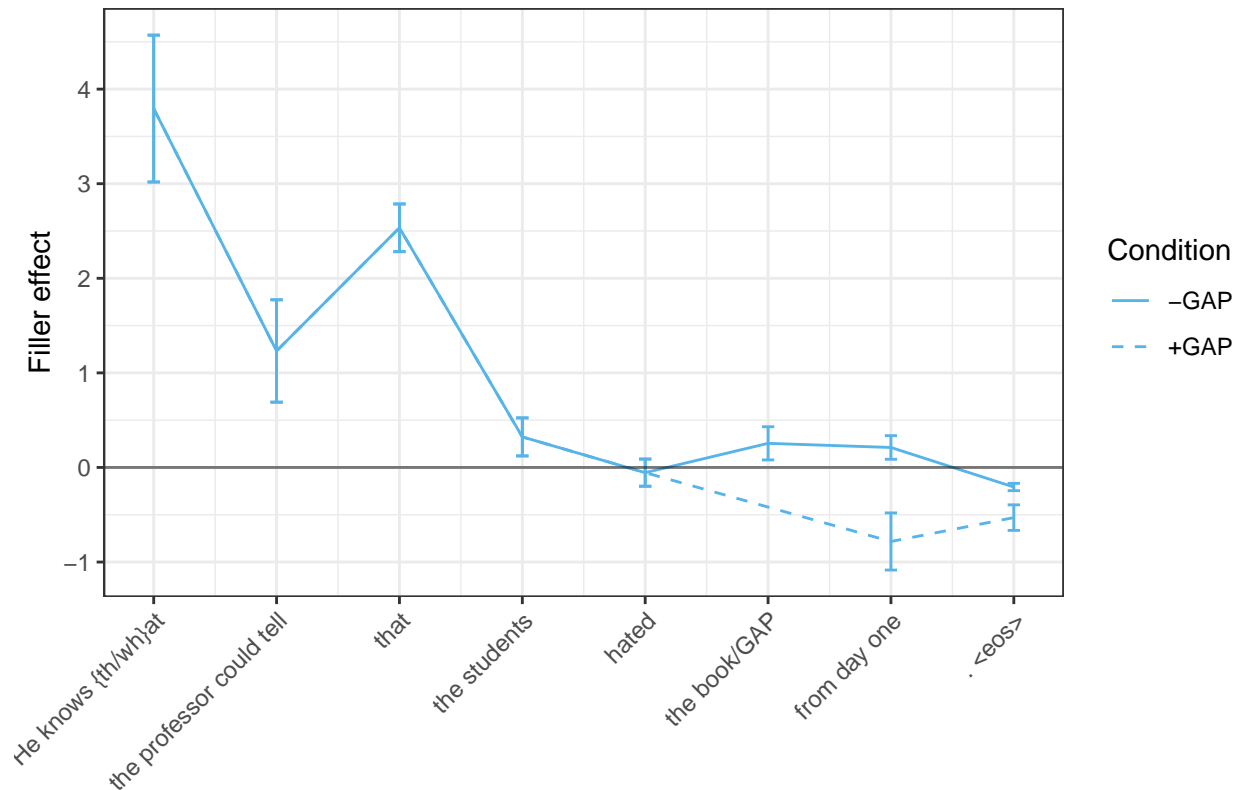
## Raw surprisal for wh-dependencies with embedded declaratives, NO



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")
```

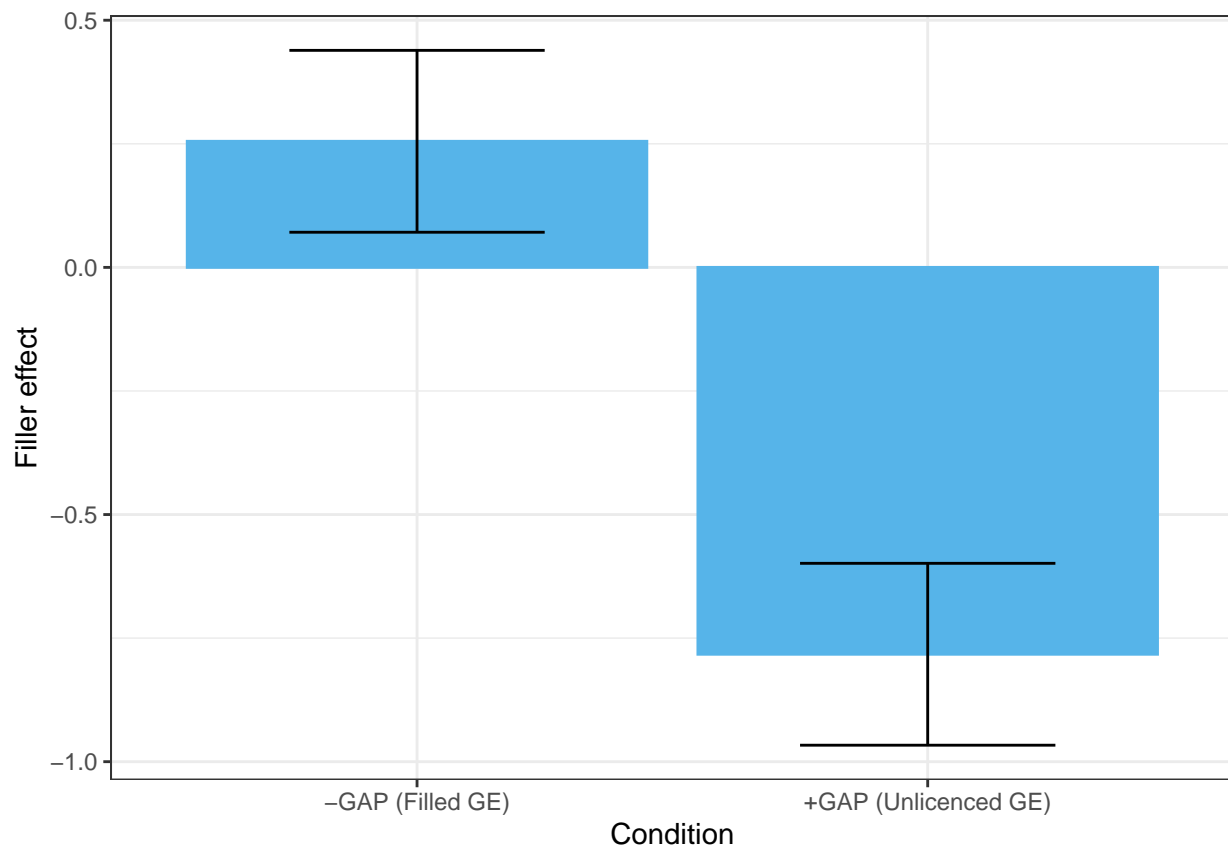
## Filler effects for wh-dependencies with embedded declaratives, NO



Calculating and plotting filler effects in ROIs:

```
wh_control_fe = wh_control_fe %>%
  filter(region == "obj" & gap == "no-gap" | region == "end" & gap == "gap")

wh_control_fe_roi = fe.roi.stats(wh_control_fe)
wh_control_fe_roi$dependency = "Wh"
wh_control_fe_roi$language = "Norwegian"
fe.roi.plot(data = wh_control_fe_roi, name = "no-wh-dep-control", path = fe_roi, color_choice = c("#56B8C8"))
```



English, wh-dependency, control condition Regions as defined previously

*# Changing the data according to the ROIs*

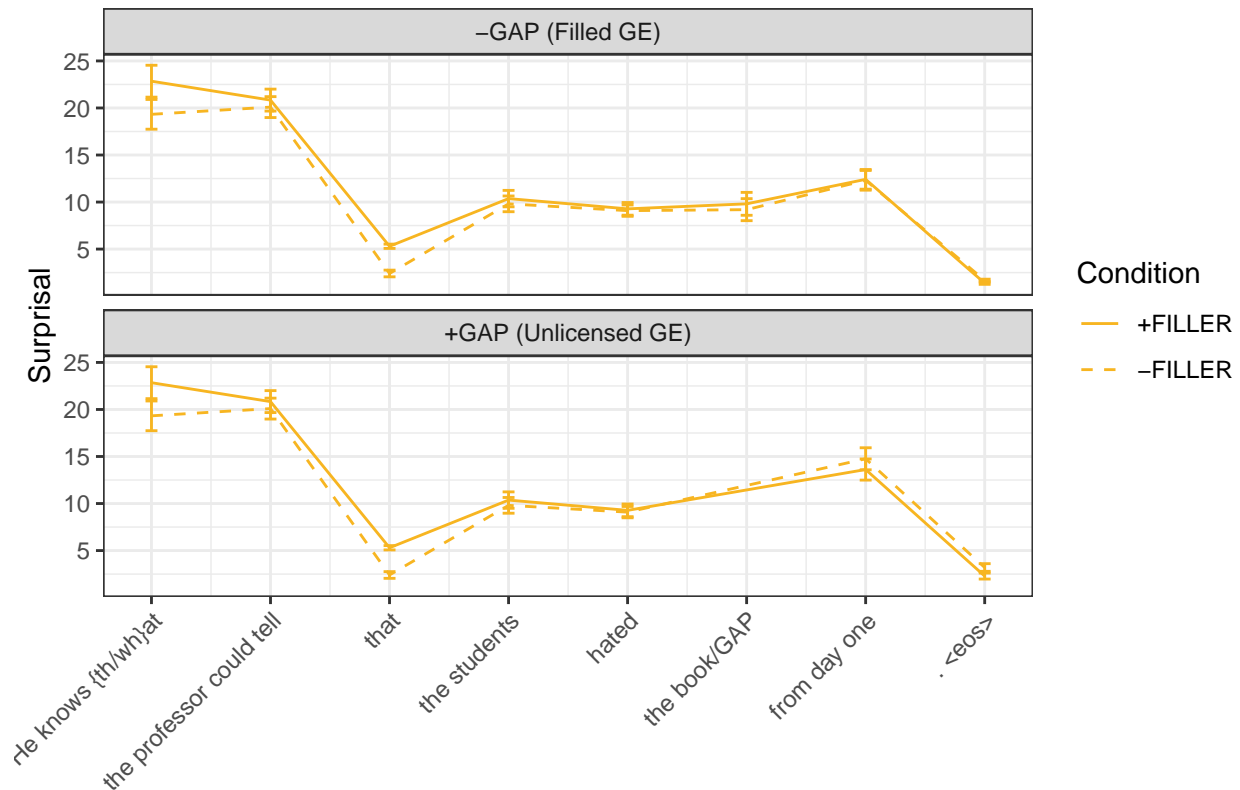
```
en_control = en_control %>%
  mutate(region = if_else(region == "that1" | region == "wh-obj", "prefix", region),
         region = if_else(region == "that2" | region == "whether", "comp2", region),
         region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "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")
```

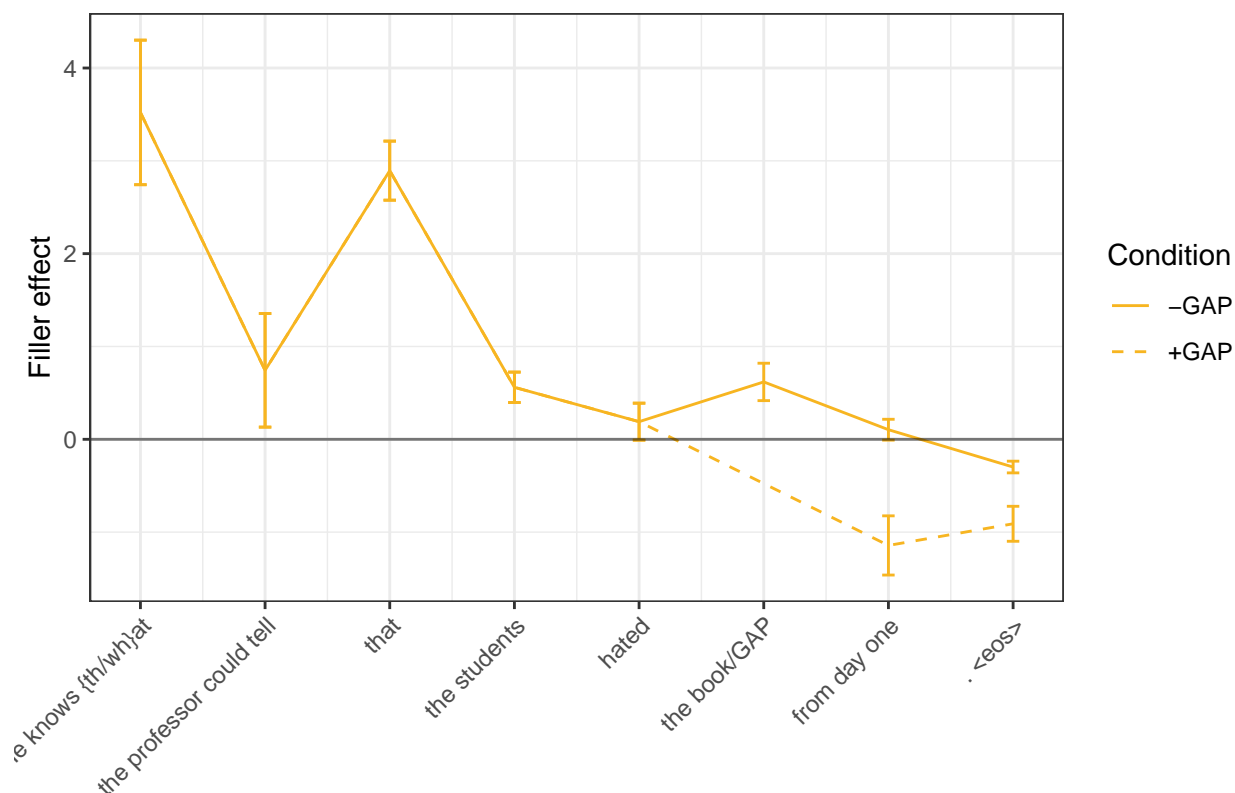
## Raw surprisal for wh-dependencies with embedded declaratives, EN



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")
```

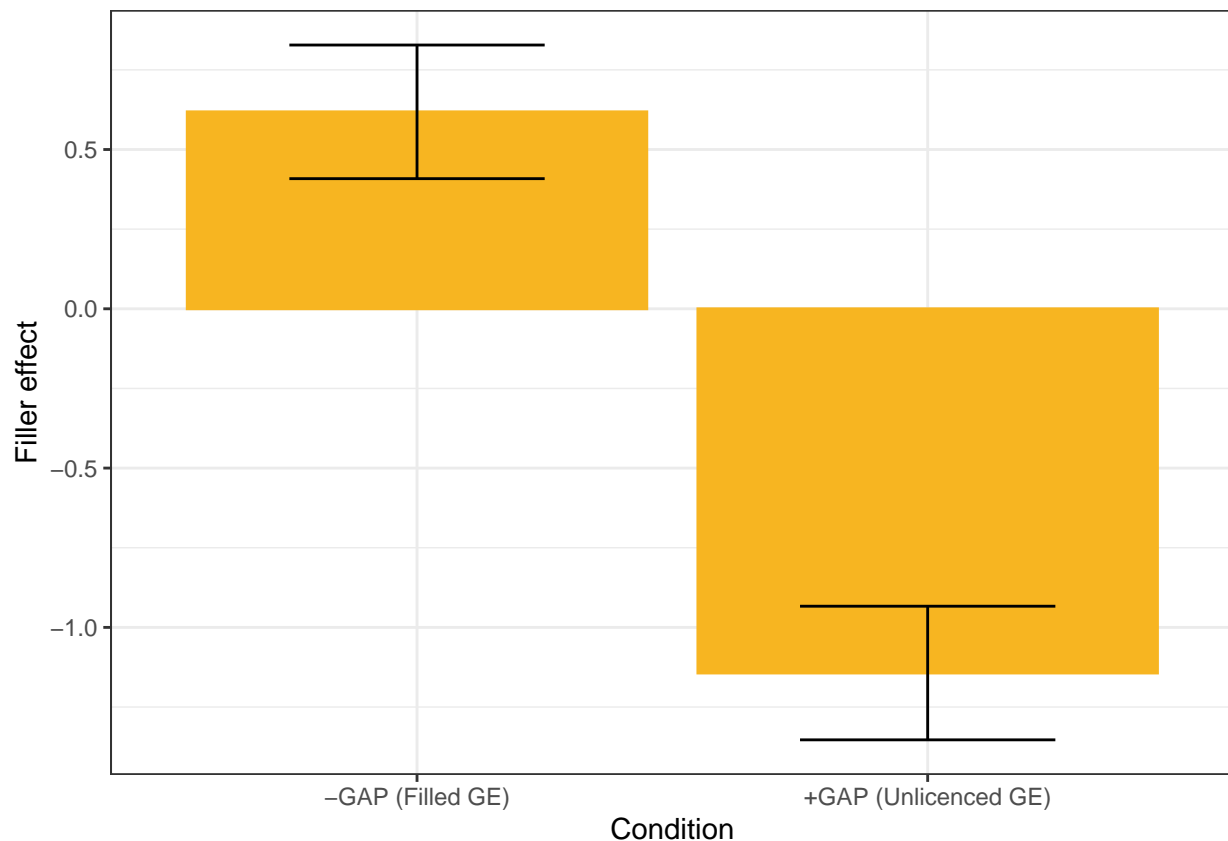
## Filler effects for wh-dependencies with embedded declaratives, EN



Calculating and plotting filler effects in ROIs:

```
en_control_fe = en_control_fe %>%
  filter(region == "obj" & gap == "no-gap" | region == "end" & gap == "gap")

en_control_fe_roi = fe.roi.stats(en_control_fe)
en_control_fe_roi$dependency = "Wh"
en_control_fe_roi$language = "English"
fe.roi.plot(data = en_control_fe_roi, name = "en-wh-dep-control", path = fe_roi, color_choice = c("#F7B6D2", "#F7B6D2"))
```



### Embedded whether-questions (island) condition, wh-dependencies

Norwegian New regions:

```
REGION_ORDER = c("prefix", "embed", "comp2", "subj", "verb", "obj", "end", "EOS")
REGION_EXEMPLARS = c("He knows {th/wh}at", "the professor could tell", "whether",
                     "the students", "hated", "the book/GAP", "from day one", ". <eos>")
```

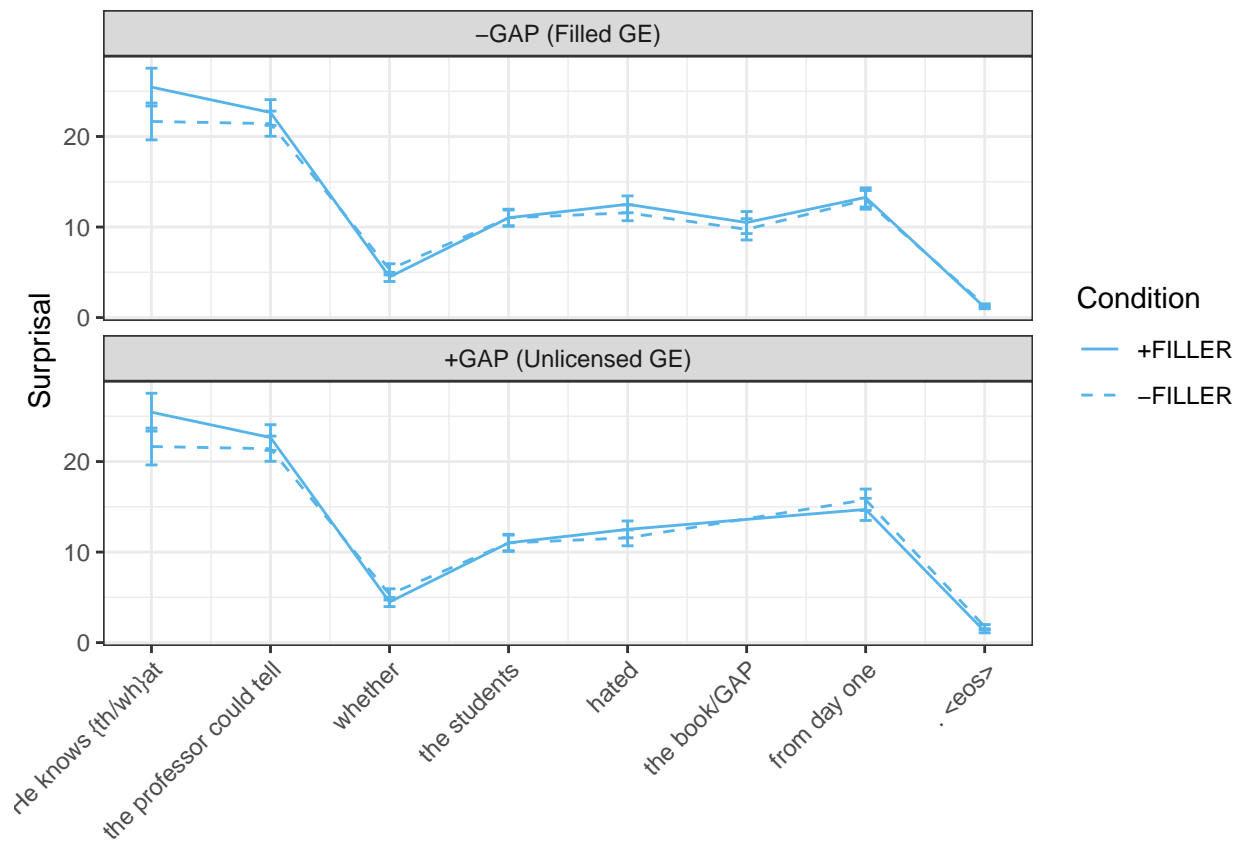
*# Changing the data according to the ROIs*

```
wh_island = wh_island %>%
  mutate(region = if_else(region == "that1" | region == "wh-obj", "prefix", region),
         region = if_else(region == "that2" | region == "whether", "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:

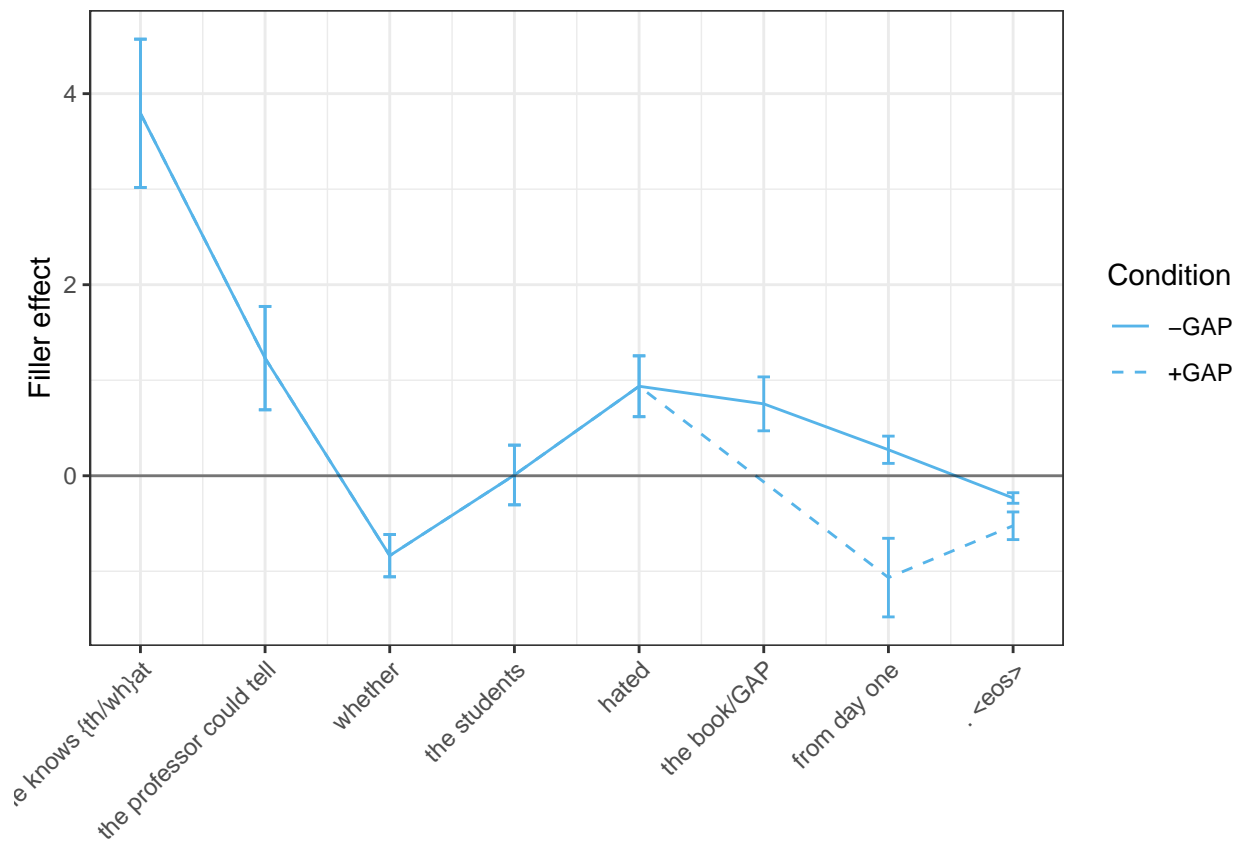
```
wh_island = region.surprisal(data = wh_island)
raw.surprisal.plot(data = wh_island, name = "no-wh-dep-island", regions = REGION_EXEMPLARS,
                  path = regions_raw, color_choice = "#56B4E9")
```





Calculating filler effects and plotting them by region:

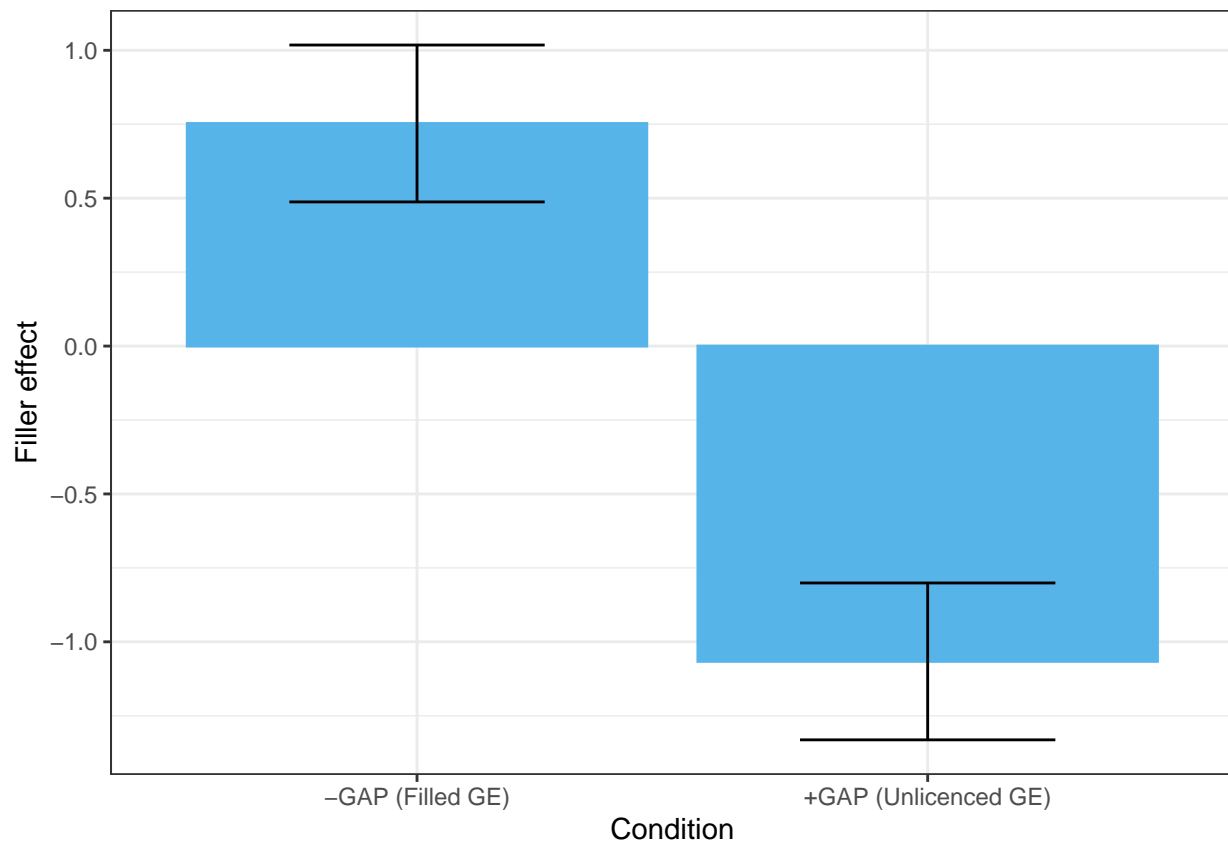
```
wh_island_fe = fe.calculation(data = wh_island)
fe.regions.plot(data = wh_island_fe, name = "no-wh-dep-island", path = regions_fe,
                regions = REGION_EXEMPLARS, color_choice = "#56B4E9")
```



Calculating and plotting filler effects in ROIs:

```
wh_island_fe = wh_island_fe %>%
  filter(region == "obj" & gap == "no-gap" | region == "end" & gap == "gap")

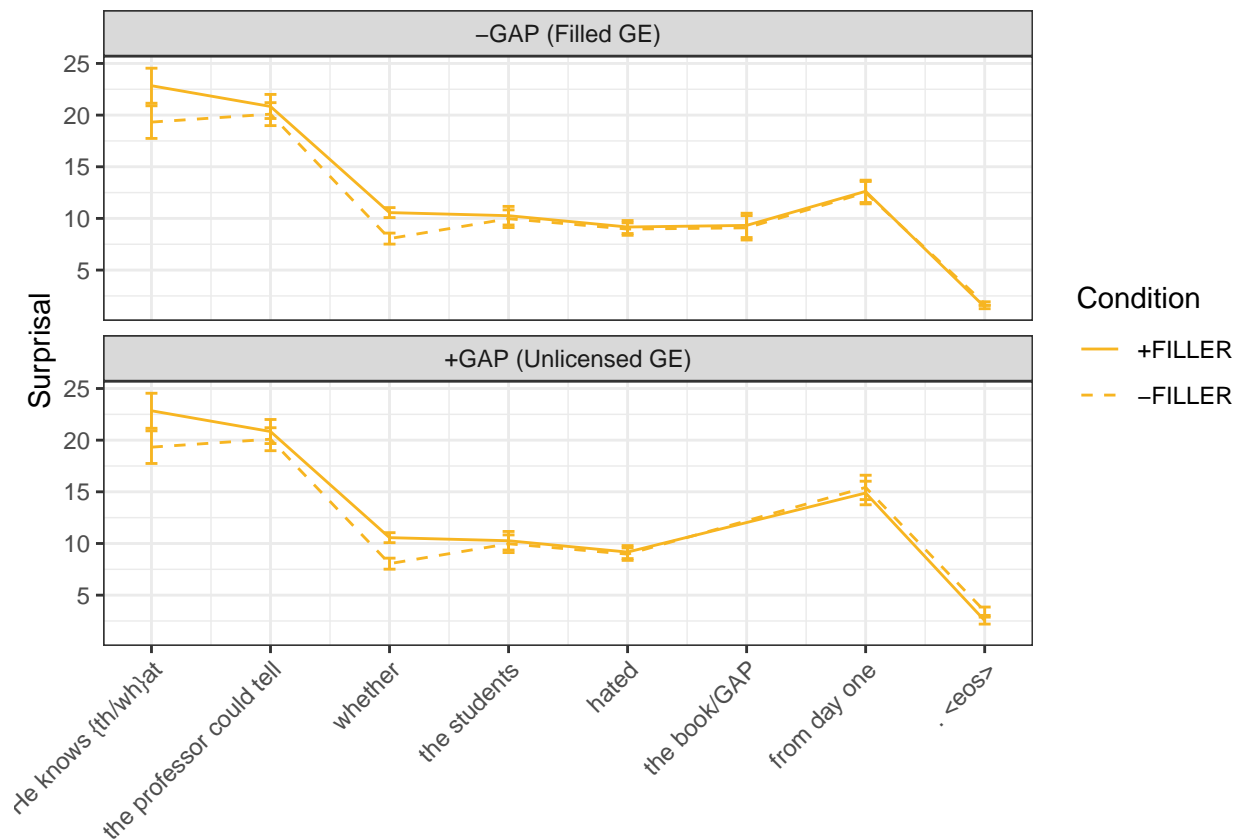
wh_island_fe_roi = fe.roi.stats(wh_island_fe)
wh_island_fe_roi$dependency = "Wh"
wh_island_fe_roi$language = "Norwegian"
fe.roi.plot(data = wh_island_fe_roi, name = "no-wh-dep-island", path = fe_roi, color_choice = c("#56B4E9", "#E69138"))
```



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

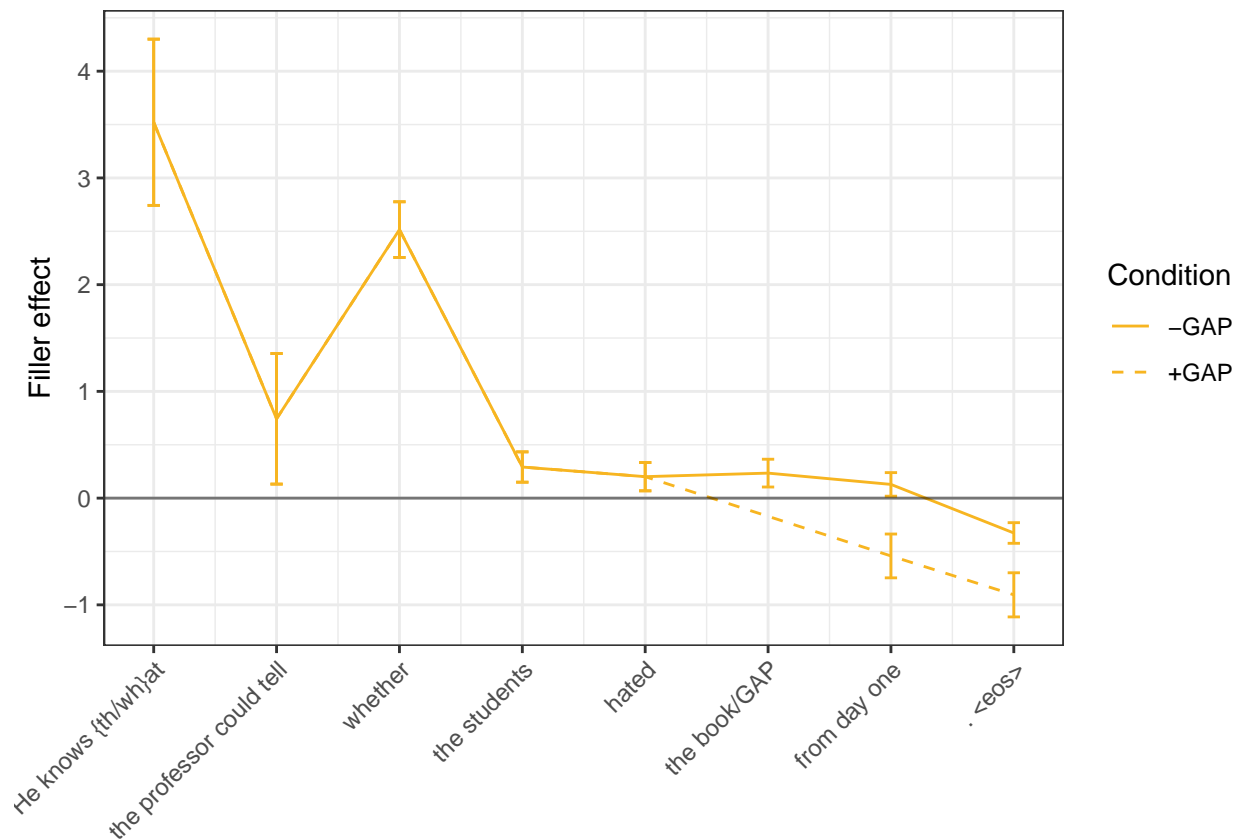
**English** Aggregating the data and plotting raw surprisal values:

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



Calculating filler effects and plotting them by region:

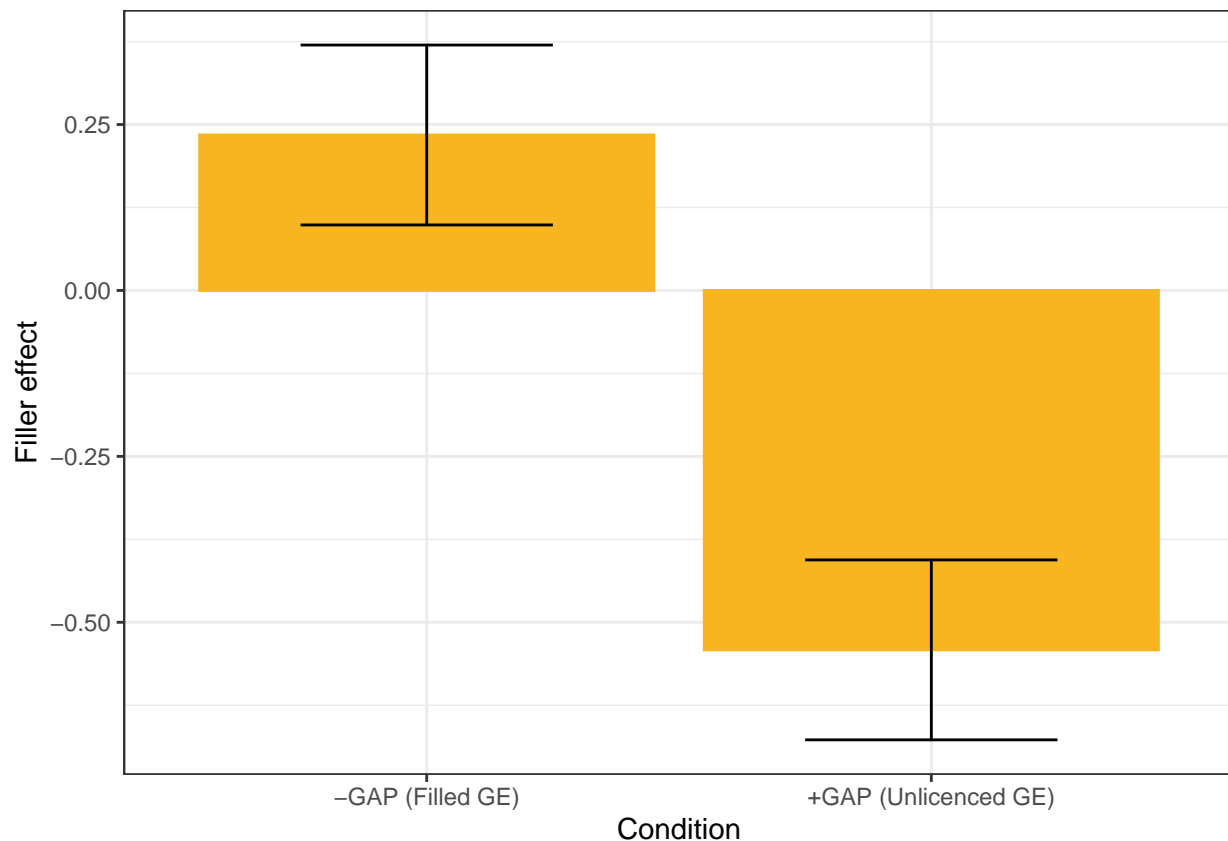
```
en_island_fe = fe.calculation(data = en_island)
fe.regions.plot(data = en_island_fe, name = "en-wh-dep-island", path = regions_fe,
                regions = REGION_EXEMPLARS, color_choice = "#F7B521")
```



Calculating and plotting filler effects in ROIs:

```
en_island_fe = en_island_fe %>%
  filter(region == "obj" & gap == "no-gap" | region == "end" & gap == "gap")

en_island_fe_roi = fe.roi.stats(en_island_fe)
en_island_fe_roi$dependency = "Wh"
en_island_fe_roi$language = "English"
fe.roi.plot(data = en_island_fe_roi, name = "en-wh-dep-island", path = fe_roi, color_choice = c("#F7B52", "#F7B52"))
```



Embedded declaratives (control) condition, RC-dependencies

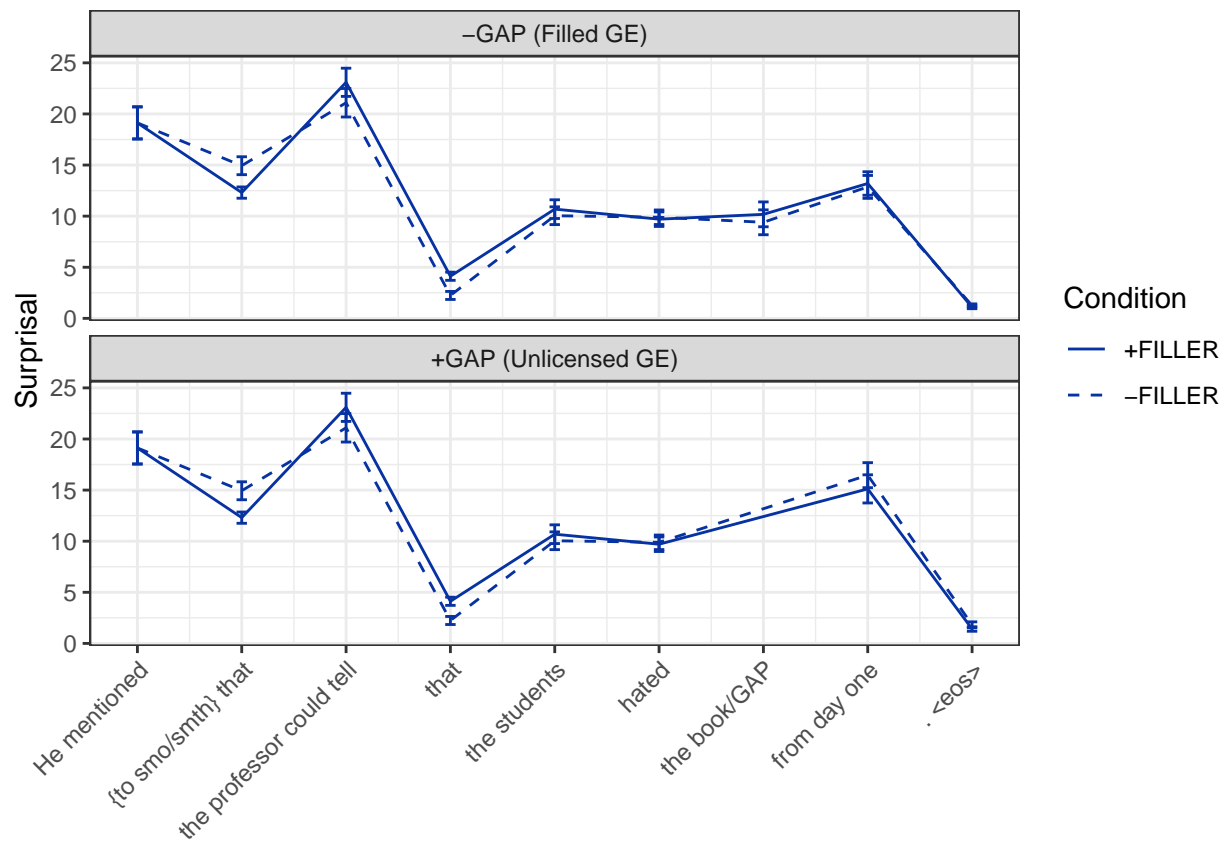
```
REGION_ORDER = c("prefix", "pp", "embed", "comp2", "subj", "verb", "obj", "end", "EOS")
REGION_EXEMPLARS = c("He mentioned", "{to smo/smith} that", "the professor could tell", "that",
  "the students", "hated", "the book/GAP", "from day one", ". <eos>")
```

*# Changing the data according to the ROIs*

```
rc_control = rc_control %>%
  mutate(region = if_else(region == "that1" | region == "rc-head-obj" |
    region == "pp-add" | region == "rp", "pp", region),
    region = if_else(region == "that2" | region == "whether", "comp2", region),
    region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

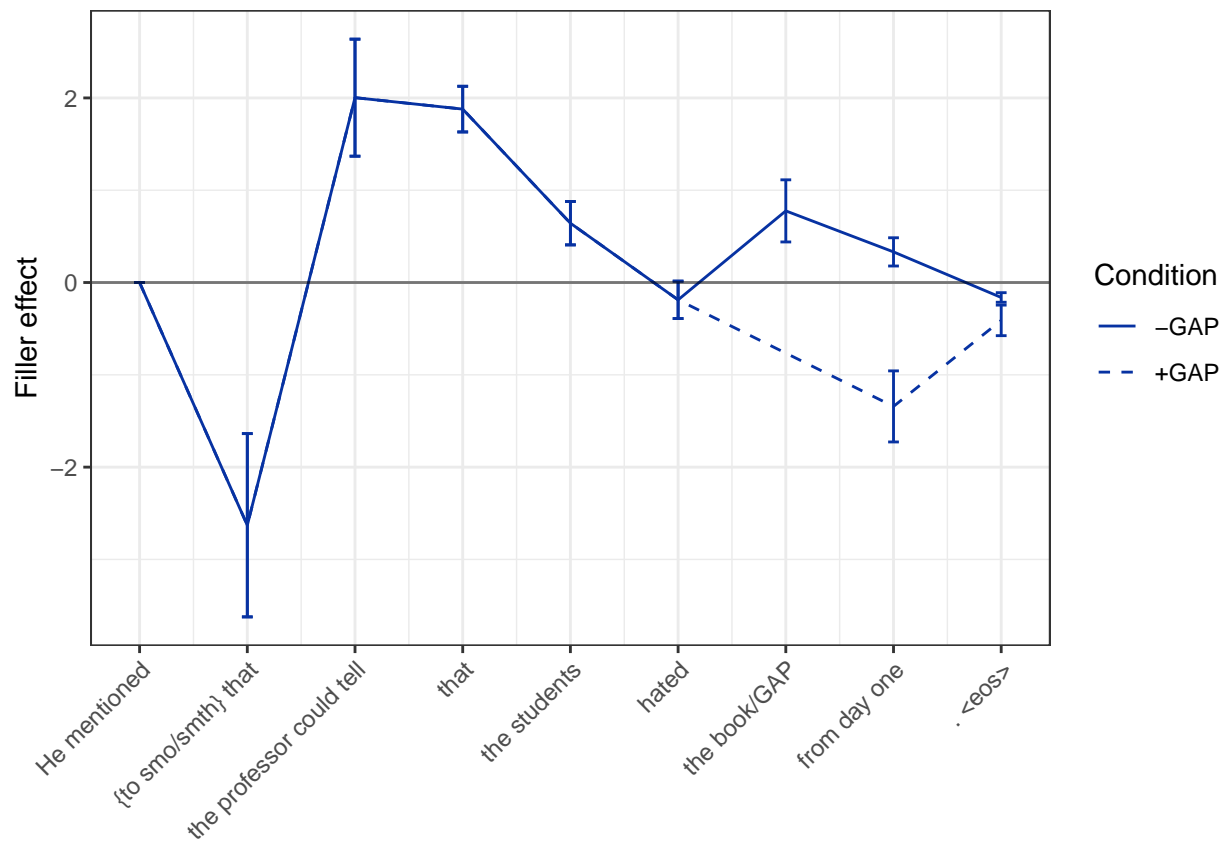
Aggregating the data and plotting raw surprisal values:

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



Calculating filler effects and plotting them by region:

```
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"))
```

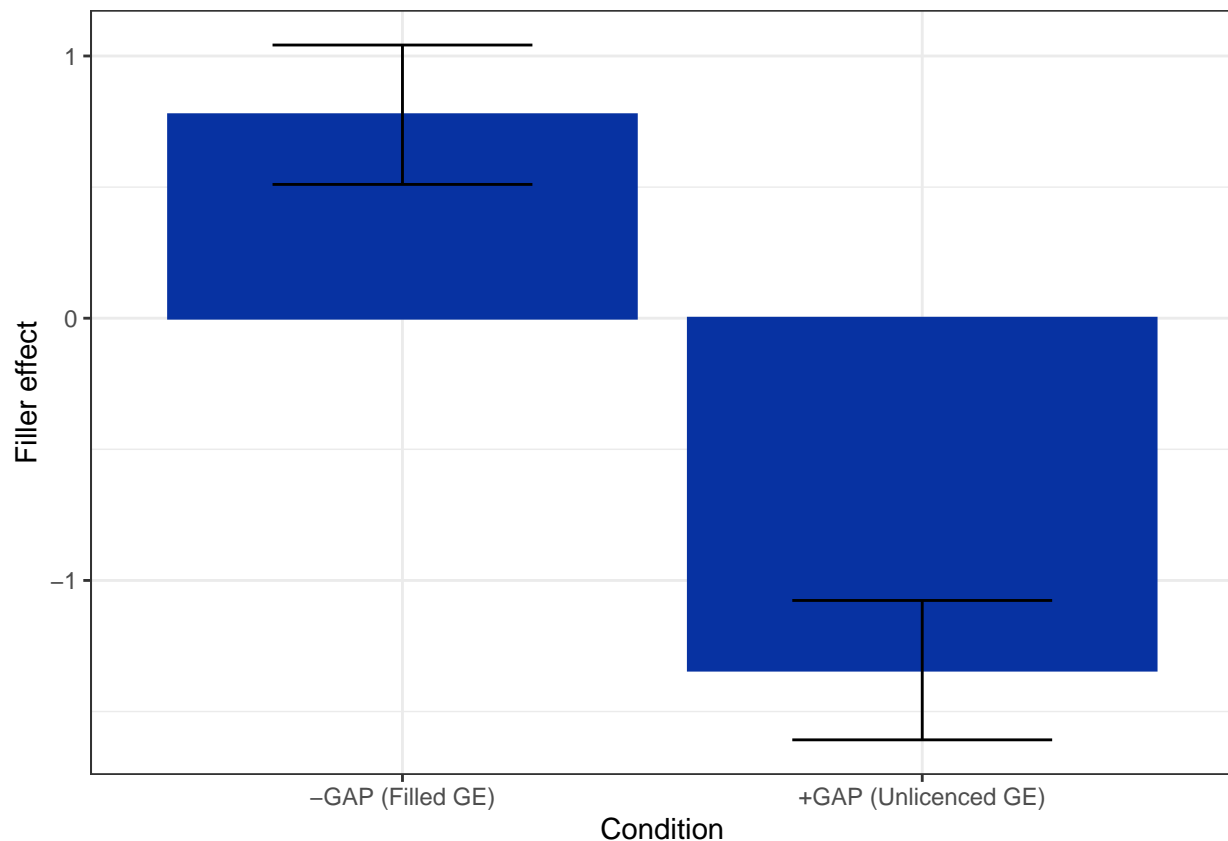


Calculating and plotting filler effects in ROIs:

```
rc_control_fe = rc_control_fe %>%
  filter(region == "obj" & gap == "no-gap" | region == "end" & gap == "gap")

rc_control_fe_roi = fe.roi.stats(rc_control_fe)
rc_control_fe_roi$dependency = "RC"
rc_control_fe_roi$language = "Norwegian"
fe.roi.plot(data = rc_control_fe_roi, name = "no-rc-dep-control", path = fe_roi, color_choice = c("#073763", "#E69F00"))
```





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

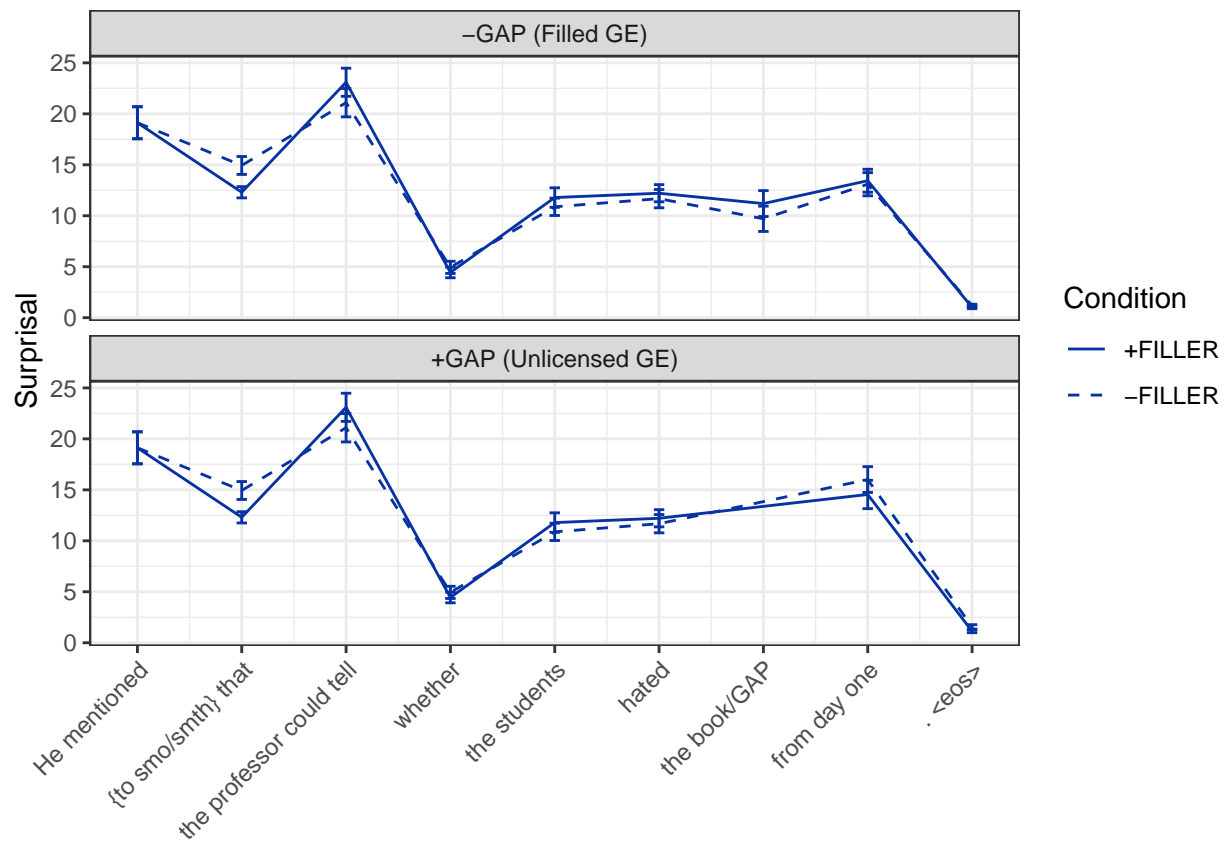
```
REGION_ORDER = c("prefix", "pp", "embed", "comp2", "subj", "verb", "obj", "end", "EOS")
REGION_EXEMPLARS = c("He mentioned", "{to smo/smith} that", "the professor could tell", "whether",
  "the students", "hated", "the book/GAP", "from day one", ". <eos>")
```

*# Changing the data according to the ROIs*

```
rc_island = rc_island %>%
  mutate(region = if_else(region == "that1" | region == "rc-head-obj" |
    region == "pp-add" | region == "rp", "pp", region),
    region = if_else(region == "that2" | region == "whether", "comp2", region),
    region = factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("comp", "gap", "gap_position"))
```

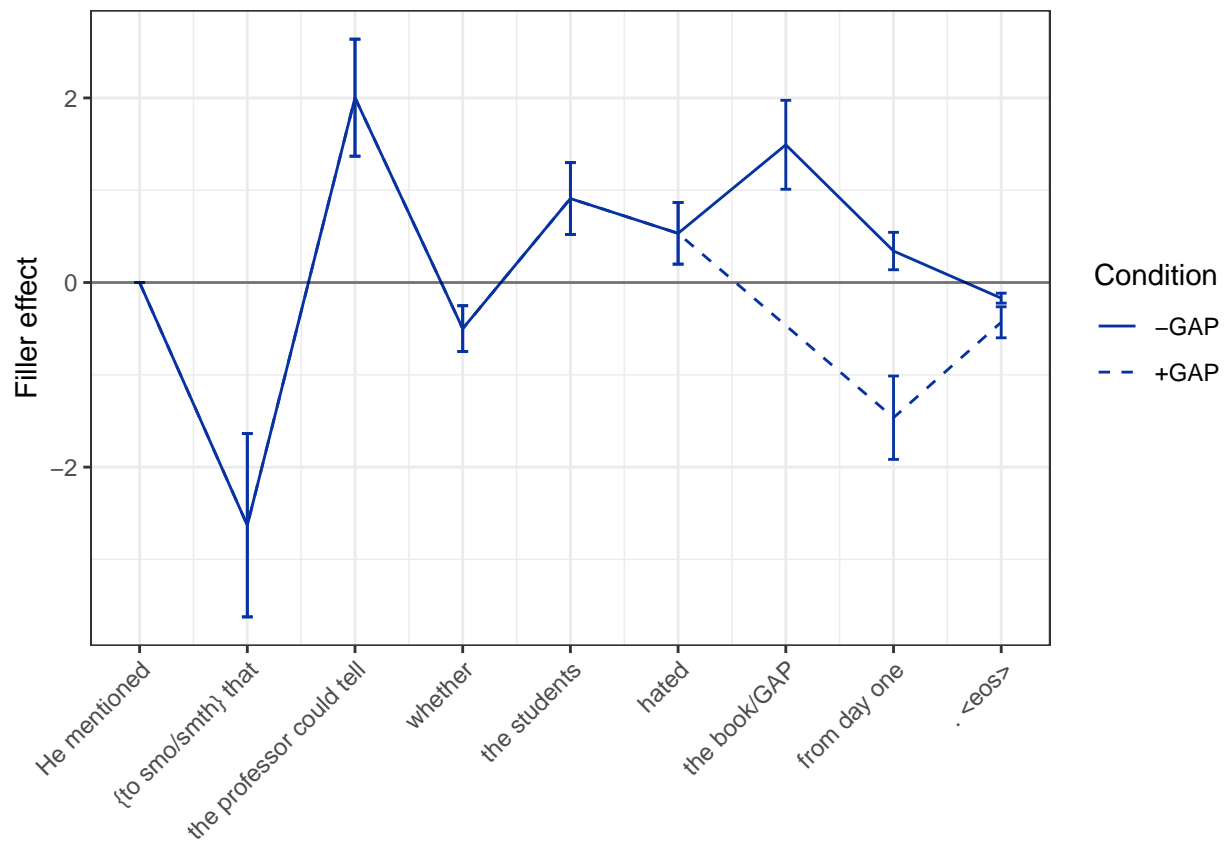
Aggregating the data and plotting raw surprisal values:

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



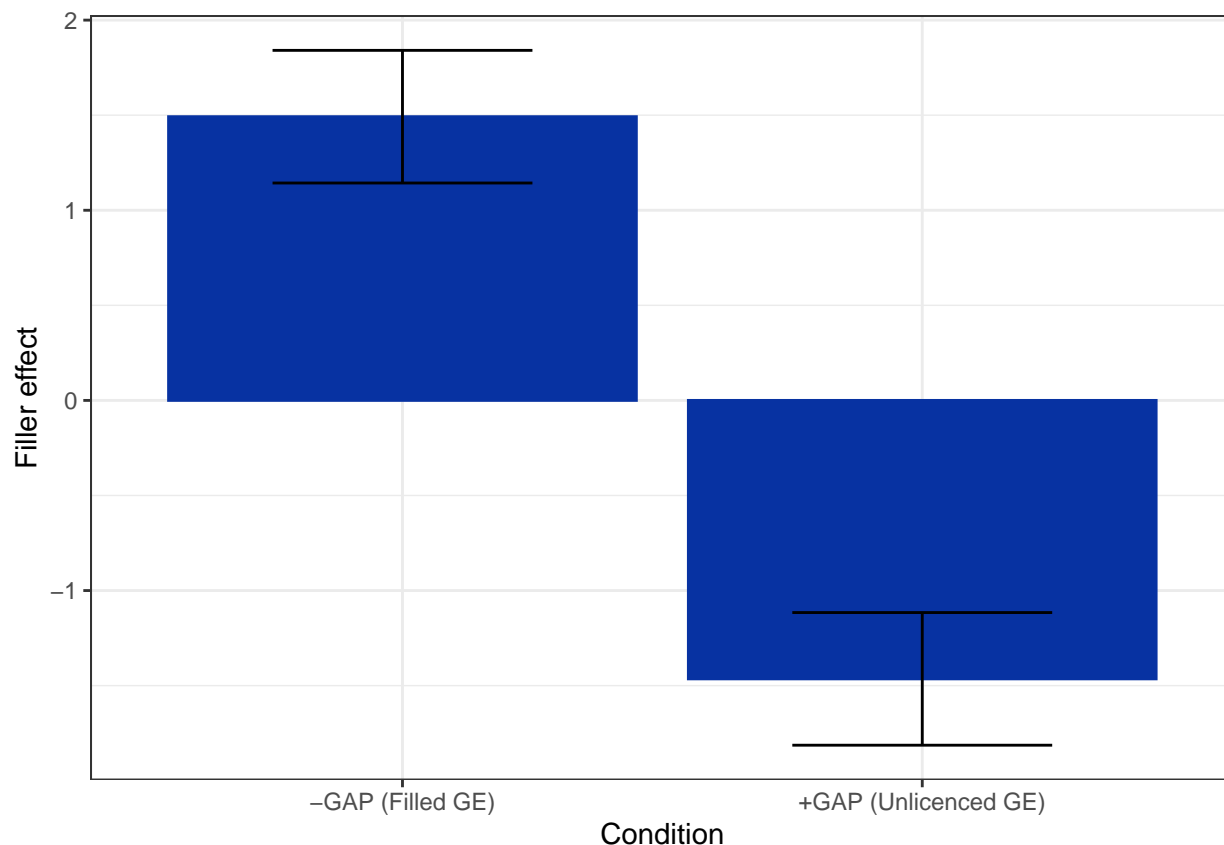
Calculating filler effects and plotting them by region:

```
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 == "obj" & gap == "no-gap" | region == "end" & gap == "gap")

rc_island_fe_roi = fe.roi.stats(rc_island_fe)
rc_island_fe_roi$dependency = "RC"
rc_island_fe_roi$language = "Norwegian"
fe.roi.plot(data = rc_island_fe_roi, name = "no-rc-dep-island", path = fe_roi, color_choice = c("#0732A7"))
```



### 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_control_fe_roi, en_island_fe_roi))

d_filler_effect[d_filler_effect == "wh-comp"] <- "whether-comp"
d_filler_effect[d_filler_effect == "that-comp"] <- "decl-comp"

d_filler_effect$gap_position <- factor(d_filler_effect$gap_position,
  levels = c("decl-comp", "whether-comp"), ordered = TRUE)

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

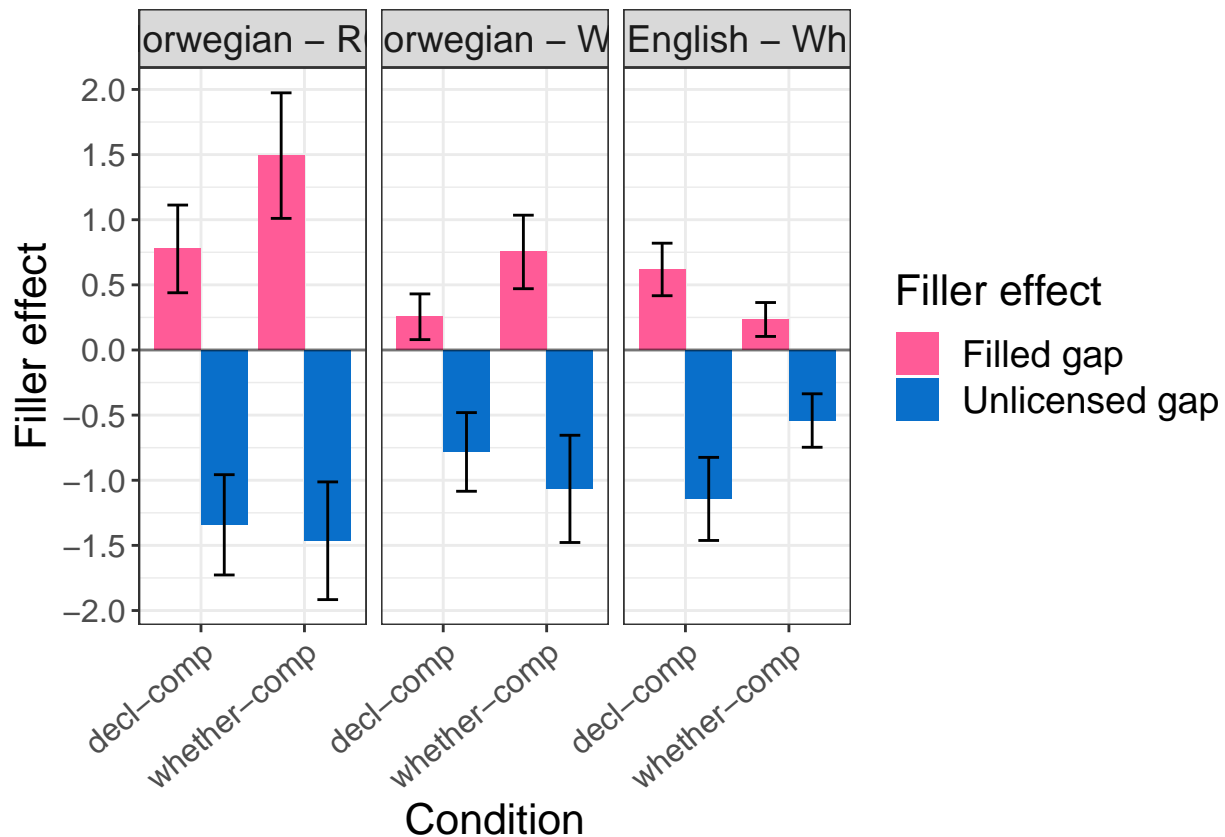
d_filler_effect$lang_dep <- factor(d_filler_effect$lang_dep,
  levels = c('Norwegian - RC', 'Norwegian - Wh', 'English - Wh'), ordered = TRUE)

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/whether-plots/%s-whether-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/whether_%s_agg.csv", model_type, model_type)

```

```

write.csv(d_filler_effect, dfname, row.names=FALSE)

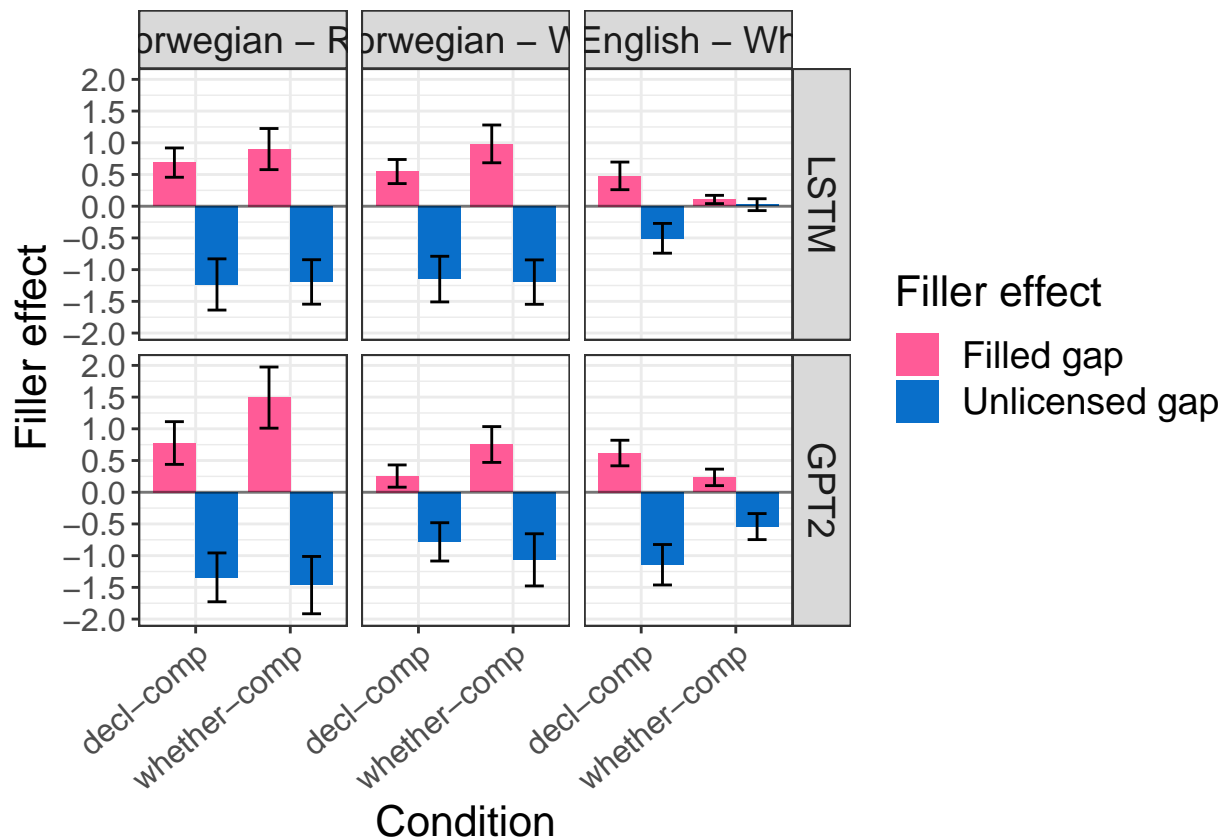
# Read in the aggregated data for both models
lstm = read.csv("../data/results/lstm/whether_lstm_agg.csv", fileEncoding = "UTF-8-BOM")
gpt2 = read.csv("../data/results/gpt2/whether_gpt2_agg.csv", fileEncoding = "UTF-8-BOM")
d = rbind(lstm, gpt2)

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('decl-comp', 'whether-comp'), 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/whether-plots/whether-two-models.png", width = 10, height = 7)
ggsave("plots/whether-plots/whether-two-models.pdf", width = 10, height = 7)
```

## Stats

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

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

```

## Between-language comparison

```

d_wh = d %>%
  filter(dependency == "Wh")
d_wh$gap_position <- factor(d_wh$gap_position,
  levels = c('decl-comp', 'whether-comp'), ordered = TRUE)
contrasts(d_wh$gap_position) <- c(0.5, -0.5) # control first, so expecting a bigger effect
d_wh$language <- factor(d_wh$language,
  levels = c('English', 'Norwegian'), ordered = TRUE)
contrasts(d_wh$language) <- c(-0.5, 0.5) # En first, so expecting a smaller effect

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

```



```
## boundary (singular) fit: see help('isSingular')
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: 906.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.6046 -0.5761 -0.0932  0.4301  4.0446
##
## Random effects:
##      Groups      Name      Variance Std.Dev.
## sent_index (Intercept) 0.05755  0.2399
## model      (Intercept) 0.00000  0.0000
## Residual                0.50885  0.7133
## Number of obs: 400, groups: sent_index, 50; model, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      0.49670    0.04922  10.090
## gap_position1     -0.04432    0.07133  -0.621
## language1         0.27505    0.07133   3.856
## gap_position1:language1 -0.84476    0.14267  -5.921
##
## 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
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')

fge_table = modelsummary(fge, output = "gt", stars = TRUE, gof_omit = ".*",
  estimate = "{estimate}{stars} ({std.error})",
  statistic = "t = {statistic}", fmt = 1,
  coef_rename = c("gap_position1" = "condition",
    "language1" = "language"))
fname_fge = sprintf("stats/whether-stats/whether-%s-between-lang-fge.html", model_type)
fge_table |> gtsave(fname_fge)

# FGE
uge = d_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: 1159.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.9747 -0.4572  0.0349  0.6030  2.4428
##
## Random effects:
##   Groups      Name      Variance Std.Dev.
##  sent_index (Intercept) 0.33869  0.5820
##   model      (Intercept) 0.01122  0.1059
##   Residual                0.87420  0.9350
## Number of obs: 400, groups:  sent_index, 50; model, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)    -0.7951    0.1207  -6.587
## gap_position1   -0.2003    0.0935  -2.142
## language1      -0.5070    0.0935  -5.423
## gap_position1:language1  0.7314    0.1870   3.911
##
## 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 = 1,
  coef_rename = c("gap_position1" = "condition",
    "language1" = "language"))
fname_uge = sprintf("stats/whether-stats/whether-%s-between-lang-uge.html", model_type)
uge_table |> gtsave(fname_uge)

```