Analysis for Wh-Licensing by Position

```
rm(list = ls())
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------
```

```
## ✓ ggplot2 2.2.1     ✓ purrr   0.2.4
## ✓ tibble  1.4.2     ✓ dplyr   0.7.5
## ✓ tidyr   0.8.1     ✓ stringr 1.3.1
## ✓ readr   1.1.1     ✓ forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(brms)
```

```
## Loading required package: Rcpp
```

```
## Loading 'brms' package (version 2.3.0). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').
## Run theme_set(theme_default()) to use the default bayesplot theme.
```

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
##
## Attaching package: 'lme4'
```

```
## The following object is masked from 'package:brms':
##
##     ngrps
```

```
library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##     lmer
```

```
## The following object is masked from 'package:stats':
##
##     step
```

```
library(plotrix)
library(stringr)
library(readxl)
```

```
remove_na = function(x) {
  x[!is.na(x)]
}


REGION_ORDER = c("prefix", "apositive", "NP1", "Verb", "NP2", "Prep", "NP3", "End", "EOS")
REGION_EXEMPLARS = c("I know that/wh", "despite protocol", "the CEO", "showed", "the presentation", "to
NUM_REGIONS = length(REGION_ORDER)

d = read_csv("tests/combined_results.csv") %>%
  select(-1, -2) %>%
  mutate(unk=unk == "True") %>%
  mutate(region = if_else(word == "." | word == "<eos>", "EOS", region)) %>%
  mutate(region=if_else(region == "Prefix" | region == "wh-subj" | region == "wh-obj" | region == "wh-p
         region=factor(region, levels=REGION_ORDER)) %>%
  separate(condition, sep="_", into=c("wh", "gap", "gap_position"))
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

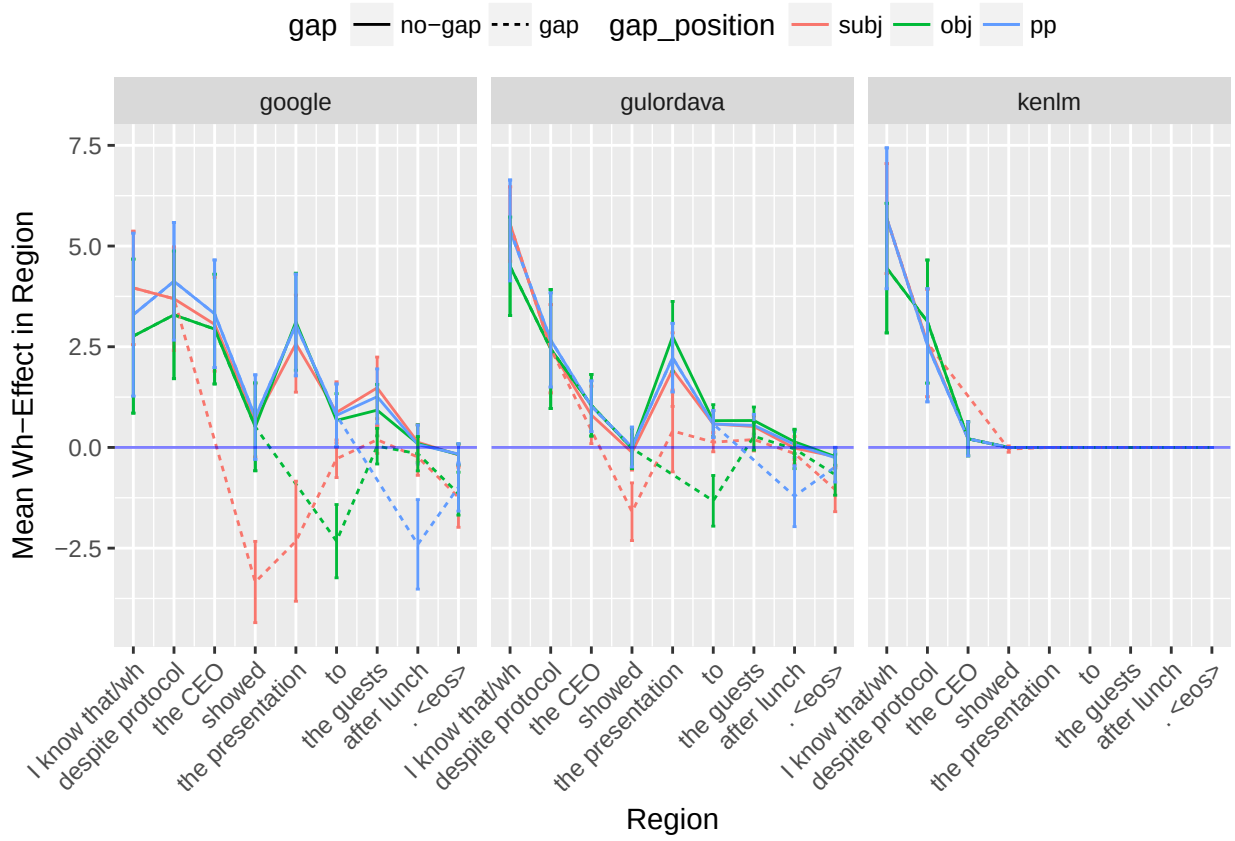
```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   `Unnamed: 0` = col_integer(),
##   sent_index = col_integer(),
##   word_index = col_integer(),
##   word = col_character(),
##   region = col_character(),
##   condition = col_character(),
##   model_word = col_character(),
##   surprisal = col_double(),
##   model = col_character(),
##   unk = col_character(),
##   final = col_character()
## )
```

```
d_agg = d %>%
  group_by(model, region, sent_index, wh, gap, gap_position) %>%
    summarise(surprisal=sum(surprisal),
              unk=any(unk)) %>%
    ungroup() %>%
  filter(!unk) %>%
  mutate(wh_numeric=if_else(wh == "who", 1, -1),
         wh=factor(wh, levels=c("what", "that")),
         gap=factor(gap, levels=c("no-gap", "gap")),
         gap_position=factor(gap_position, levels=c("subj", "obj", "pp")))
```

Visualization of wh-effect for all three models

```
d_agg %>%
  select(-wh_numeric) %>%
  spread(wh, surprisal) %>%
  mutate(wh_effect=what-`that`) %>%
```

```
  group_by(region, gap, gap_position, model) %>%
    summarise(m=mean(wh_effect),
              s=std.error(wh_effect),
              upper=m + 1.96*s,
              lower=m - 1.96*s) %>%
    ungroup() %>%
  mutate(region=as.numeric(region)) %>%
  ggplot(aes(x=region, y=m, ymax=upper, ymin=lower,linetype=gap, color=gap_position)) +
    geom_line() +
    geom_errorbar(linetype="solid", width=.1) +
    scale_x_continuous(breaks=seq(1, NUM_REGIONS), labels=REGION_EXEMPLARS) +
    theme(axis.text.x = element_text(angle=45, hjust=1)) +
    geom_hline(yintercept=0, color="blue", alpha=0.5) +
  facet_wrap(~model, ncol = 3) +
  ylab("Mean Wh-Effect in Region") +
  xlab("Region") +
  theme(legend.position="top", legend.margin=margin(c(0,0,0,0)))
```

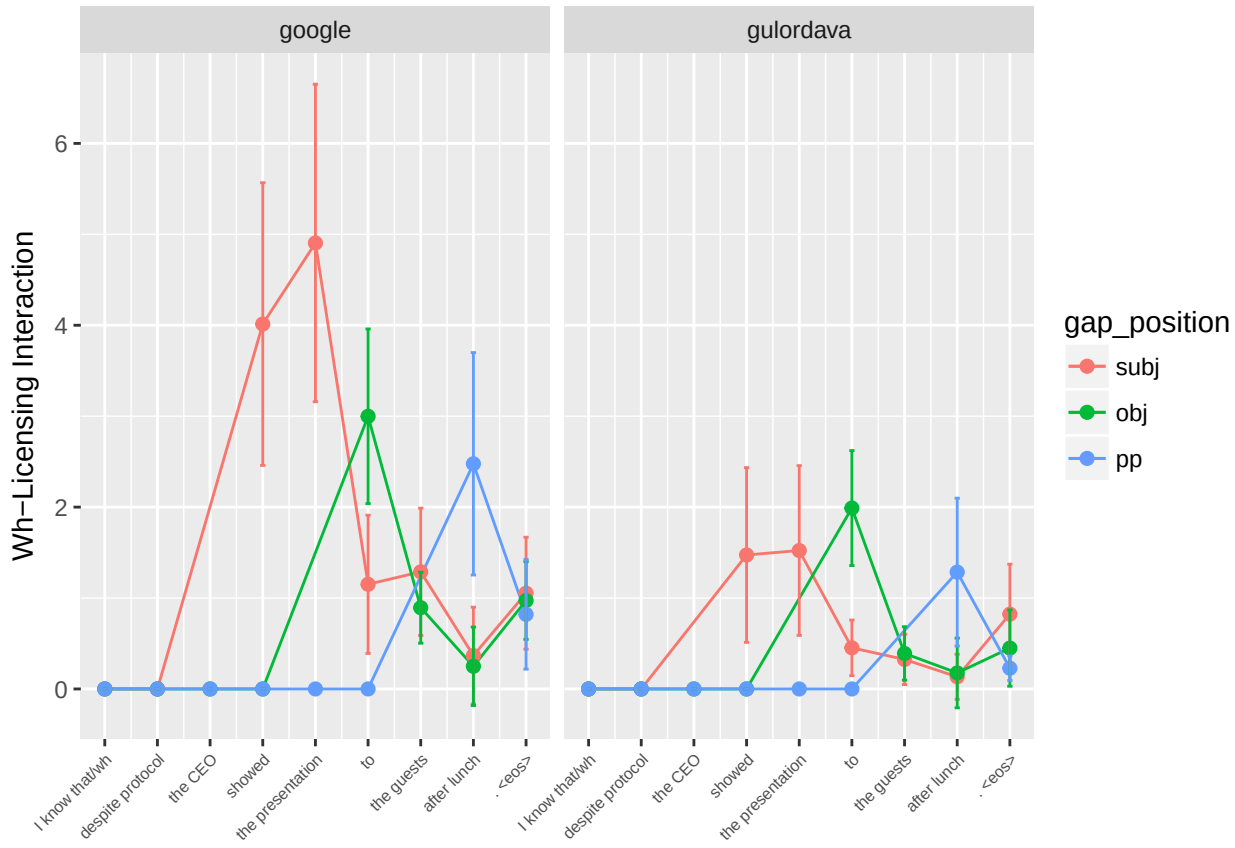



```
ggsave("~/Desktop/island-graphs/position-wordbyword.pdf",height=3,width=7)
```

These are positive visualizations. The drops of the dotted lines (gapped conditions) in the immediate post gap material in each variant indicates that the models are representing the filler-gap dependency.

Visualization of region-by-region wh-licensing interaction

```
d_agg %>%
  filter(model == "gulordava" | model == "google") %>%
  select(-wh_numeric) %>%
  spread(gap, surprisal) %>%
  mutate(gap_effect=`no-gap`-gap) %>%
  select(-unk, -gap, -`no-gap`) %>%
  spread(wh, gap_effect) %>%
  mutate(wh_interaction=what-`that`) %>%

  group_by(region, gap_position, model) %>%
    summarise(m=mean(wh_interaction),
              s=std.error(wh_interaction),
              upper=m + 1.96*s,
              lower=m - 1.96*s) %>%
    ungroup() %>%
  na.exclude() %>%
  mutate(region=as.numeric(region)) %>%
  ggplot(aes(x=region, y=m, ymax=upper, ymin=lower, color=gap_position)) +
    geom_line() +
    geom_point(size = 2) +
    geom_errorbar(linetype="solid", width=.1) +
    scale_x_continuous(breaks=seq(1, NUM_REGIONS), labels=REGION_EXEMPLARS) +
    theme(axis.text.x = element_text(angle=45, hjust=1, size = 6)) +
    theme(axis.title.x=element_blank()) +
  facet_wrap(~model, ncol = 2) +
  ylab("Wh-Licensing Interaction") +
  #xlab("Region") +
  theme(legend.position="right", legend.margin=margin(c(0,0,0,0)))
```



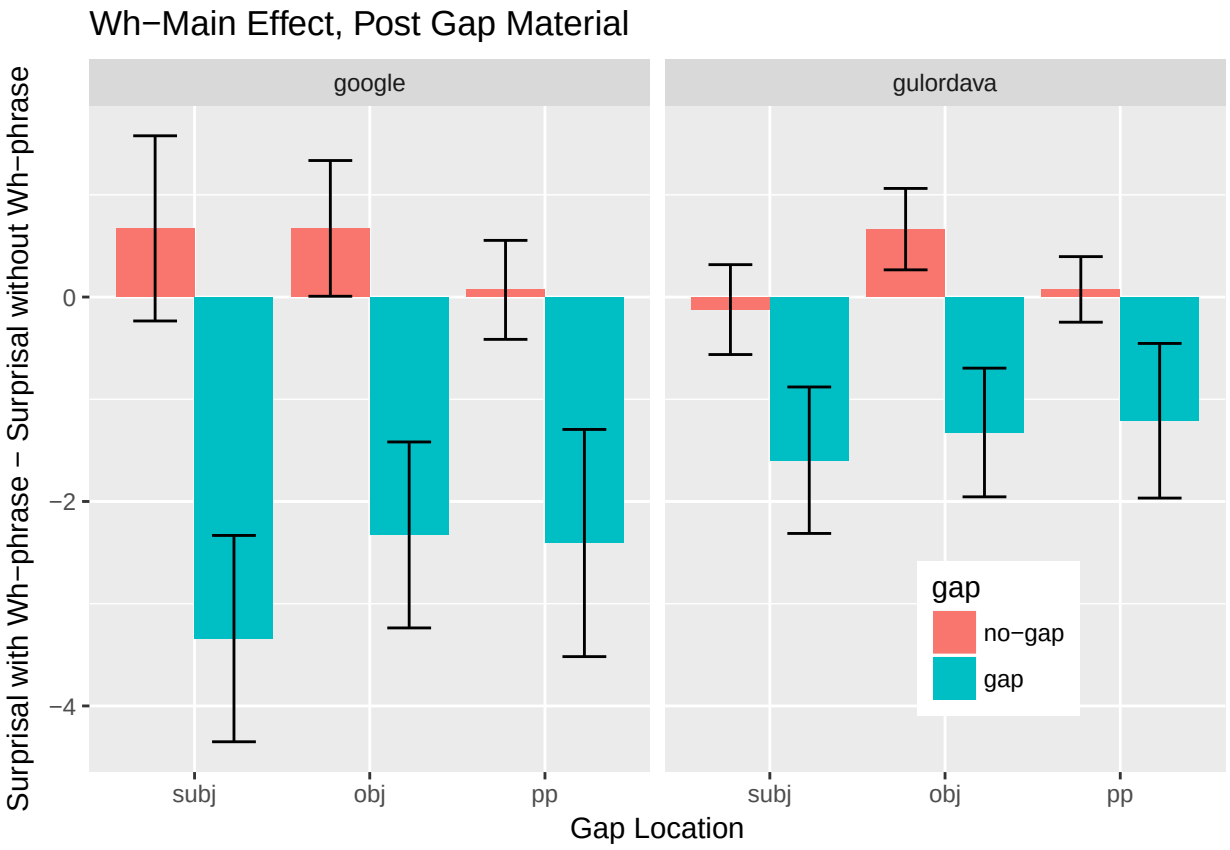

```
ggsave("~/Desktop/island-graphs/position-wordbyword-whinteraction.pdf",height=2,width=7)
```

Wh-Licensing Interaction, Post-gap Material

Let's start off with the wh-effect

```
d_whe_1 = d_agg %>%
  filter(region == "Verb", gap_position == "subj") %>%
  filter(model == "google" | model == "gulordava") %>%
  select(-wh_numeric) %>%
  spread(wh, surprisal) %>%
  mutate(wh_effect=what-`that`)

d_whe_2 = d_agg %>%
  filter(region == "Prep", gap_position == "obj") %>%
  filter(model == "google" | model == "gulordava") %>%
  select(-wh_numeric) %>%
  spread(wh, surprisal) %>%
  mutate(wh_effect=what-`that`)

d_whe_3 = d_agg %>%
  filter(region == "End", gap_position == "pp") %>%
  filter(model == "google" | model == "gulordava") %>%
  select(-wh_numeric) %>%
  spread(wh, surprisal) %>%
```

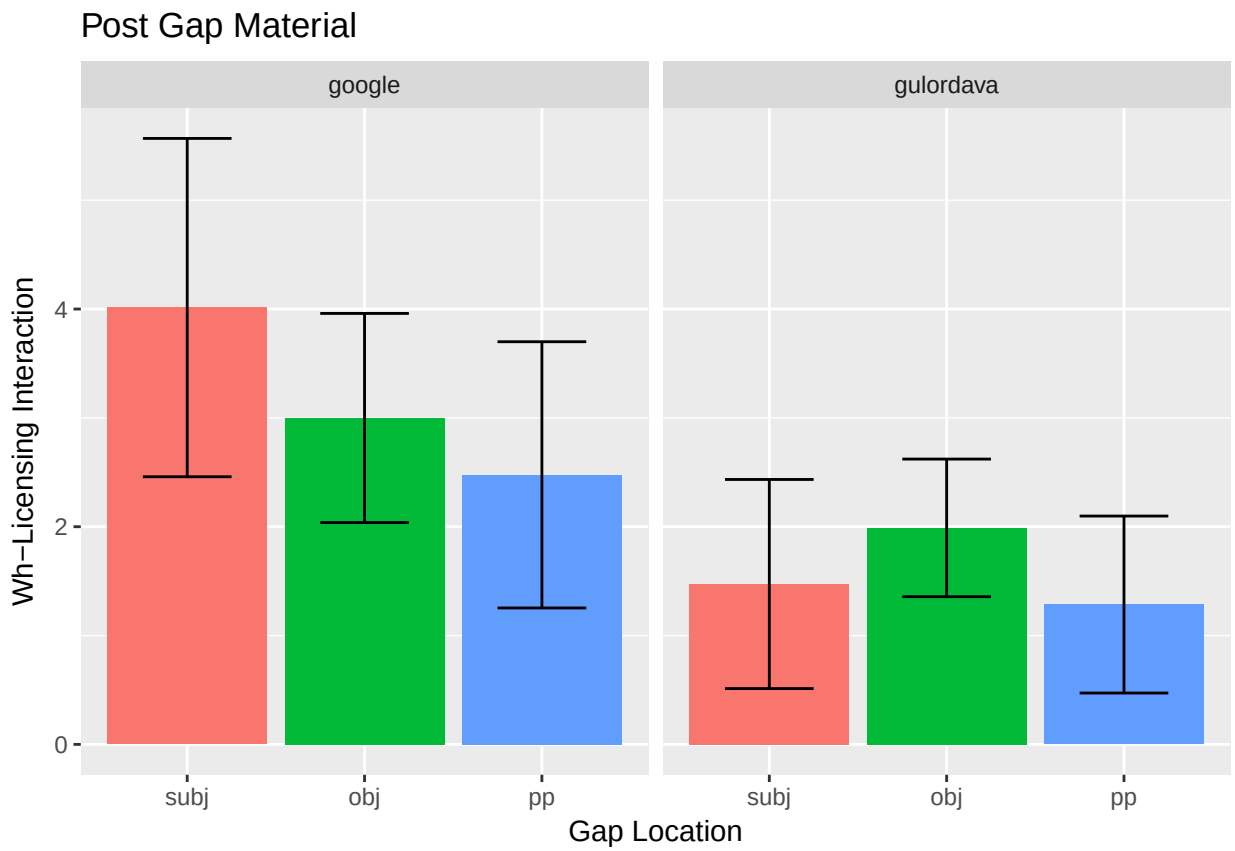
```
  mutate(wh_effect=what-`that`)

d_wh_effect = Reduce(function(x, y) merge(x, y, all=TRUE), list(d_whe_1, d_whe_2, d_whe_3))

d_wh_effect %>%
  group_by(model, gap, gap_position) %>%
    summarise(m=mean(wh_effect),
              s=std.error(wh_effect),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
      ungroup() %>%
  ggplot(aes(x=gap_position, y=m, ymin=lower, ymax=upper, fill=gap)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model) +
  ylab("Surprisal with Wh-phrase - Surprisal without Wh-phrase") +
  xlab("Gap Location") +
  theme(legend.position = c(0.8, 0.2)) +
  ggtitle("Wh-Main Effect, Post Gap Material")
```




```
ggsave("~/Desktop/island-graphs/position-wheffect-postgap.pdf",height=5,width=3.5)
```

Now let's look at the full 2x2 interaction of wh-phrase presence and gap presence, for each of the gap positions. This is the strength of evidence for the filler-gap dependency.

```
d_full_interaction_1 = d_agg %>%
  filter(region == "Verb", gap_position == "subj") %>%
  select(-wh_numeric) %>%
  spread(gap, surprisal) %>%
  mutate(gap_effect=`no-gap`-gap) %>%
  select(-unk, -gap, -`no-gap`) %>%
  spread(wh, gap_effect) %>%
  mutate(wh_interaction=what-`that`)

d_full_interaction_2 = d_agg %>%
  filter(region == "Prep", gap_position == "obj") %>%
  select(-wh_numeric) %>%
  spread(gap, surprisal) %>%
  mutate(gap_effect=`no-gap`-gap) %>%
  select(-unk, -gap, -`no-gap`) %>%
  spread(wh, gap_effect) %>%
  mutate(wh_interaction=what-`that`)

d_full_interaction_3 = d_agg %>%
  filter(region == "End", gap_position == "pp") %>%
  select(-wh_numeric) %>%
  spread(gap, surprisal) %>%
  mutate(gap_effect=`no-gap`-gap) %>%
  select(-unk, -gap, -`no-gap`) %>%
  spread(wh, gap_effect) %>%
  mutate(wh_interaction=what-`that`)

d_full_interaction = Reduce(function(x, y) merge(x, y, all=TRUE), list(d_full_interaction_1, d_full_int

d_full_interaction %>%
  filter(model == "google" | model == "gulordava") %>%
  group_by(model, gap_position) %>%
    summarise(m=mean(wh_interaction, na.rm=T),
              s=std.error(wh_interaction, na.rm=T),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
  ggplot(aes(x=gap_position, y=m, ymin=lower, ymax=upper, fill=gap_position)) +
    geom_bar(stat="identity") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model) +
  ylab("Wh-Licensing Interaction") +
  xlab("Gap Location") +
  theme(legend.position="none") +
  ggtitle("Post Gap Material")
```




```
ggsave("~/Desktop/island-graphs/position-postgap.pdf",height=5,width=3.5)
```

There seems to be evidence for subject, object, and PP gaps in the full interaction. Neither model shows the expected decrease in strength with the object, subject and then PP gap. The Google model seems to have a robustly higher licensing interaction than the gulordava model.

Stats.

```
d_agg = d_agg %>%
  mutate(gap_numeric=if_else(gap == "gap", 1, -1)) %>%
  mutate(wh_numeric = if_else(wh == "what", 1, -1))

# For the subject condition
m_google = d_agg %>%
  filter(region == "Verb", gap_position == "subj") %>%
  filter(model == "google") %>%
  lmer(surprisal ~ wh_numeric * gap_numeric + (wh_numeric+gap_numeric|sent_index), data=.)
summary(m_google)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## surprisal ~ wh_numeric * gap_numeric + (wh_numeric + gap_numeric |
##     sent_index)
##    Data: .
##
## REML criterion at convergence: 469.4
##
```

```
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.96393 -0.45801 -0.08219  0.34762  2.28161
##
## Random effects:
##  Groups     Name        Variance Std.Dev. Corr
##  sent_index (Intercept) 11.75373 3.4284
##             wh_numeric   0.05279 0.2298    0.48
##             gap_numeric  1.83387 1.3542   -0.02 -0.89
##  Residual                2.75493 1.6598
## Number of obs: 96, groups:  sent_index, 24
##
## Fixed effects:
##                         Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)              16.9168     0.7200 23.0000  23.495  < 2e-16 ***
## wh_numeric               -0.6675     0.1758 40.2739  -3.797 0.000484 ***
## gap_numeric               2.5744     0.3242 23.0329   7.941 4.82e-08 ***
## wh_numeric:gap_numeric   -1.0033     0.1694 46.0000  -5.923 3.78e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) wh_nmr gp_nmr
## wh_numeric   0.124
## gap_numeric -0.014 -0.202
## wh_nmrc:gp_  0.000  0.000  0.000
```

```
m_gul = d_agg %>%
  filter(region == "Verb", gap_position == "subj") %>%
  filter(model == "gulordava") %>%
  lmer(surprisal ~ wh_numeric * gap_numeric + (wh_numeric+gap_numeric|sent_index), data=.)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge: degenerate Hessian with 1
## negative eigenvalues
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -1.0e+00
```

```
summary(m_gul)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## surprisal ~ wh_numeric * gap_numeric + (wh_numeric + gap_numeric |
##     sent_index)
##    Data: .
##
## REML criterion at convergence: 428.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.02363 -0.39177 -0.06235  0.43075  2.44738
##
```

```
## Random effects:
##  Groups     Name        Variance Std.Dev. Corr
##  sent_index (Intercept) 14.12028 3.7577
##             wh_numeric   0.01798 0.1341   1.00
##             gap_numeric  2.40106 1.5495   0.24 0.24
##  Residual                1.06768 1.0333
## Number of obs: 96, groups:  sent_index, 24
##
## Fixed effects:
##                         Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)              17.1654     0.7743 23.0000  22.170  < 2e-16 ***
## wh_numeric               -0.4297     0.1090 40.9833  -3.944 0.000306 ***
## gap_numeric               2.4124     0.3334 22.9964   7.236  2.3e-07 ***
## wh_numeric:gap_numeric   -0.3684     0.1055 45.9942  -3.493 0.001068 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) wh_nmr gp_nmr
## wh_numeric  0.249
## gap_numeric 0.226  0.057
## wh_nmrc:gp_ 0.000  0.000  0.000
## convergence code: 0
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

```
# For the object condition
m_google = d_agg %>%
  filter(region == "Prep", gap_position == "obj") %>%
  filter(model == "google") %>%
  lmer(surprisal ~ wh_numeric * gap_numeric + (wh_numeric+gap_numeric|sent_index), data=.)
summary(m_google)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## surprisal ~ wh_numeric * gap_numeric + (wh_numeric + gap_numeric |
##     sent_index)
##    Data: .
##
## REML criterion at convergence: 451.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.29533 -0.51952  0.05297  0.37979  1.95632
##
## Random effects:
##  Groups     Name        Variance Std.Dev. Corr
##  sent_index (Intercept) 21.4917  4.6359
##             wh_numeric   0.2699  0.5195    0.15
##             gap_numeric  1.5127  1.2299    0.12 -0.78
##  Residual                1.4412  1.2005
## Number of obs: 96, groups:  sent_index, 24
##
## Fixed effects:
```

```
##                         Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)              7.3475     0.9542 23.0000   7.700 8.21e-08 ***
## wh_numeric              -0.4139     0.1620 23.0000  -2.554   0.0177 *
## gap_numeric              1.4082     0.2794 23.0000   5.041 4.21e-05 ***
## wh_numeric:gap_numeric  -0.7497     0.1225 23.0000  -6.119 3.05e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) wh_nmr gp_nmr
## wh_numeric   0.099
## gap_numeric  0.102 -0.459
## wh_nmrc:gp_  0.000  0.000  0.000
```

```
m_gul = d_agg %>%
  filter(region == "Prep", gap_position == "obj") %>%
  filter(model == "gulordava") %>%
  lmer(surprisal ~ wh_numeric * gap_numeric + (wh_numeric+gap_numeric|sent_index), data=.)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge: degenerate Hessian with 1
## negative eigenvalues
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -1.2e+01
```

```
summary(m_gul)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## surprisal ~ wh_numeric * gap_numeric + (wh_numeric + gap_numeric |
##     sent_index)
##    Data: .
##
## REML criterion at convergence: 420.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.06202 -0.44244 -0.03954  0.35728  2.86670
##
## Random effects:
##  Groups     Name        Variance Std.Dev. Corr
##  sent_index (Intercept) 31.93092 5.6507
##             wh_numeric   0.06866 0.2620   1.00
##             gap_numeric  1.62168 1.2735   0.29 0.29
##  Residual                0.72887 0.8537
## Number of obs: 96, groups:  sent_index, 24
##
## Fixed effects:
##                         Estimate Std. Error       df t value Pr(>|t|)
## (Intercept)              7.88425    1.15674 22.79114   6.816 6.26e-07 ***
## wh_numeric              -0.16520    0.10224 30.71405  -1.616   0.1164
## gap_numeric              0.66616    0.27416 22.14576   2.430   0.0237 *
```

```
## wh_numeric:gap_numeric -0.49732    0.08713 44.86988  -5.708 8.54e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) wh_nmr gp_nmr
## wh_numeric  0.522
## gap_numeric 0.270  0.142
## wh_nmrc:gp_ 0.000  0.000  0.000
## convergence code: 0
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

```
# For the PP/Goal condition
m_google = d_agg %>%
  filter(region == "End", gap_position == "pp") %>%
  filter(model == "google") %>%
  lmer(surprisal ~ wh_numeric * gap_numeric + (wh_numeric+gap_numeric|sent_index), data=.)
summary(m_google)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## surprisal ~ wh_numeric * gap_numeric + (wh_numeric + gap_numeric |
##     sent_index)
##    Data: .
##
## REML criterion at convergence: 481.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.8266 -0.6168 -0.0376  0.4347  1.9146
##
## Random effects:
##  Groups     Name        Variance Std.Dev. Corr
##  sent_index (Intercept) 37.00981 6.084
##             wh_numeric   0.02993 0.173    -0.13
##             gap_numeric  1.66448 1.290     0.14 -1.00
##  Residual                2.23402 1.495
## Number of obs: 96, groups:  sent_index, 24
##
## Fixed effects:
##                         Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)              20.3078     1.2511 23.0000  16.231 4.33e-14 ***
## wh_numeric               -0.5841     0.1566 40.8169  -3.730 0.000582 ***
## gap_numeric               2.5405     0.3043 23.0033   8.348 2.05e-08 ***
## wh_numeric:gap_numeric   -0.6191     0.1525 46.0000  -4.059 0.000190 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) wh_nmr gp_nmr
## wh_numeric  -0.029
## gap_numeric  0.119 -0.195
## wh_nmrc:gp_  0.000  0.000  0.000
```

```
m_gul = d_agg %>%
  filter(region == "End", gap_position == "pp") %>%
  filter(model == "gulordava") %>%
  lmer(surprisal ~ wh_numeric * gap_numeric + (wh_numeric+gap_numeric|sent_index), data=.)
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -4.0e-01
```

```
summary(m_gul)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## surprisal ~ wh_numeric * gap_numeric + (wh_numeric + gap_numeric |
##     sent_index)
##    Data: .
##
## REML criterion at convergence: 459.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.42765 -0.42155 -0.00362  0.35964  2.60897
##
## Random effects:
##  Groups     Name        Variance Std.Dev. Corr
##  sent_index (Intercept) 28.16129 5.3067
##             wh_numeric   0.02539 0.1593    0.55
##             gap_numeric  5.36532 2.3163   -0.07 -0.87
##  Residual                1.00327 1.0016
## Number of obs: 96, groups:  sent_index, 24
##
## Fixed effects:
##                         Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)              25.3930     1.0880 23.0000  23.338  < 2e-16 ***
## wh_numeric               -0.2839     0.1073 38.9551  -2.647  0.01167 *
## gap_numeric               1.8223     0.4837 23.0000   3.767  0.00100 **
## wh_numeric:gap_numeric   -0.3212     0.1022 46.0000  -3.142  0.00293 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) wh_nmr gp_nmr
## wh_numeric   0.165
## gap_numeric -0.071 -0.259
## wh_nmrc:gp_  0.000  0.000  0.000
```

This says: when there is a gap, it makes things easier (because there's no filler to contribute to surprisal). And when there is a wh-word, the gaps get even easier, significantly so for all gap positions. This is the expected result, with the exception that the order of effect sizes is not as expected under the Google model (obj dependency weaker than PP dependency).

Now let's take a look at the whole-clause condition

```
d = read_csv("tests/combined_results.csv") %>%
  select(-1, -2) %>%
```

```
  mutate(unk=unk == "True") %>%
  mutate(region=if_else(region == "Prefix" | region == "wh-subj" | region == "wh-obj" | region == "wh-p
  mutate(region = if_else(region == "apositive" | region == "NP1" | region == "Verb" | region == "NP2"
  separate(condition, sep="_", into=c("wh", "gap", "gap_position"))
```

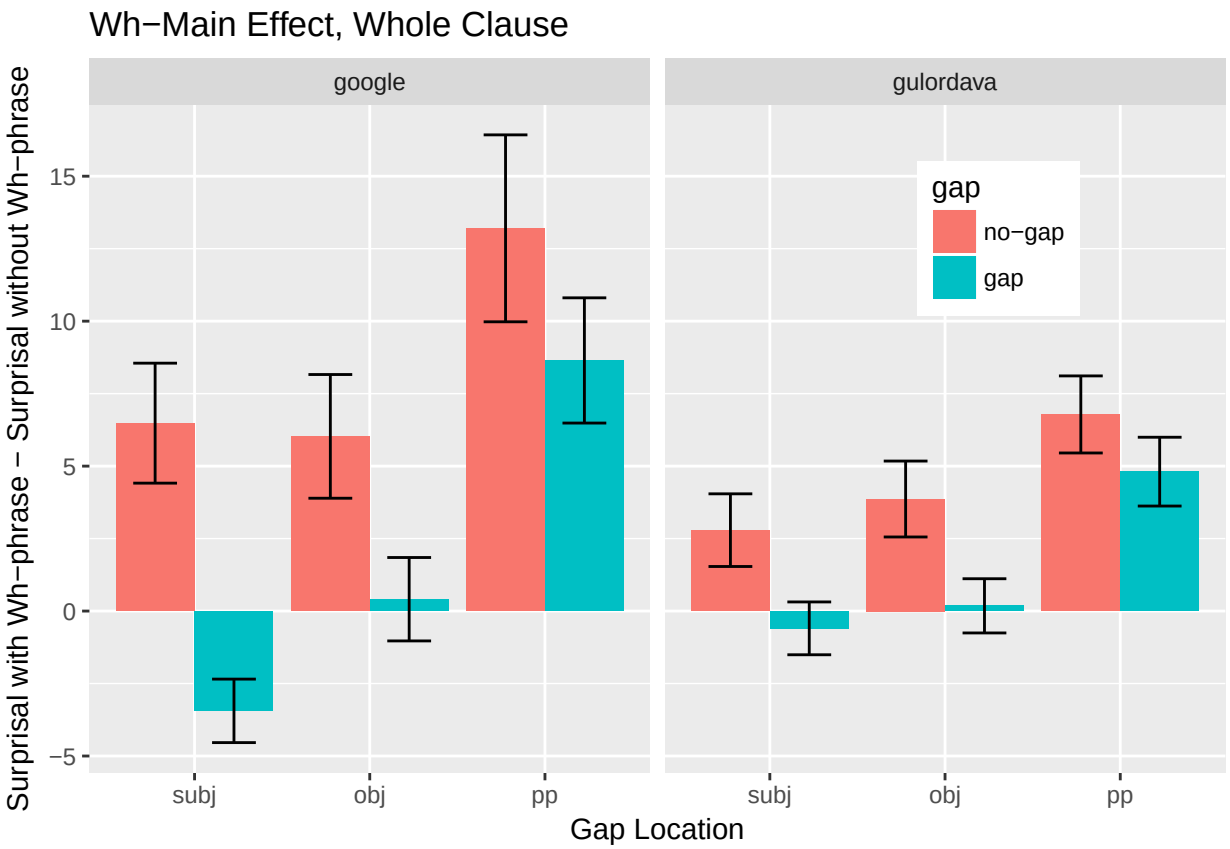
```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   `Unnamed: 0` = col_integer(),
##   sent_index = col_integer(),
##   word_index = col_integer(),
##   word = col_character(),
##   region = col_character(),
##   condition = col_character(),
##   model_word = col_character(),
##   surprisal = col_double(),
##   model = col_character(),
##   unk = col_character(),
##   final = col_character()
## )
```

```
d_agg = d %>%
  group_by(model, region, sent_index, wh, gap, gap_position) %>%
    summarise(surprisal=sum(surprisal),
              unk=any(unk)) %>%
    ungroup() %>%
  filter(!unk) %>%
  mutate(wh_numeric=if_else(wh == "what", 1, -1),
         wh=factor(wh, levels=c("what", "that")),
         gap=factor(gap, levels=c("no-gap", "gap")),
         gap_position=factor(gap_position, levels=c("subj", "obj", "pp")))
```

```
d_whe_3 = d_agg %>%
  filter(region == "embed") %>%
  filter(model == "google" | model == "gulordava") %>%
  select(-wh_numeric) %>%
  spread(wh, surprisal) %>%
  mutate(wh_effect=what-`that`)

d_wh_effect = Reduce(function(x, y) merge(x, y, all=TRUE), list(d_whe_1, d_whe_2, d_whe_3))

d_wh_effect %>%
  group_by(model, gap, gap_position) %>%
    summarise(m=mean(wh_effect),
              s=std.error(wh_effect),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
      ungroup() %>%
  ggplot(aes(x=gap_position, y=m, ymin=lower, ymax=upper, fill=gap)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model) +
  ylab("Surprisal with Wh-phrase - Surprisal without Wh-phrase") +
```

```
  xlab("Gap Location") +
  theme(legend.position = c(0.8, 0.8)) +
  ggtitle("Wh-Main Effect, Whole Clause")
```

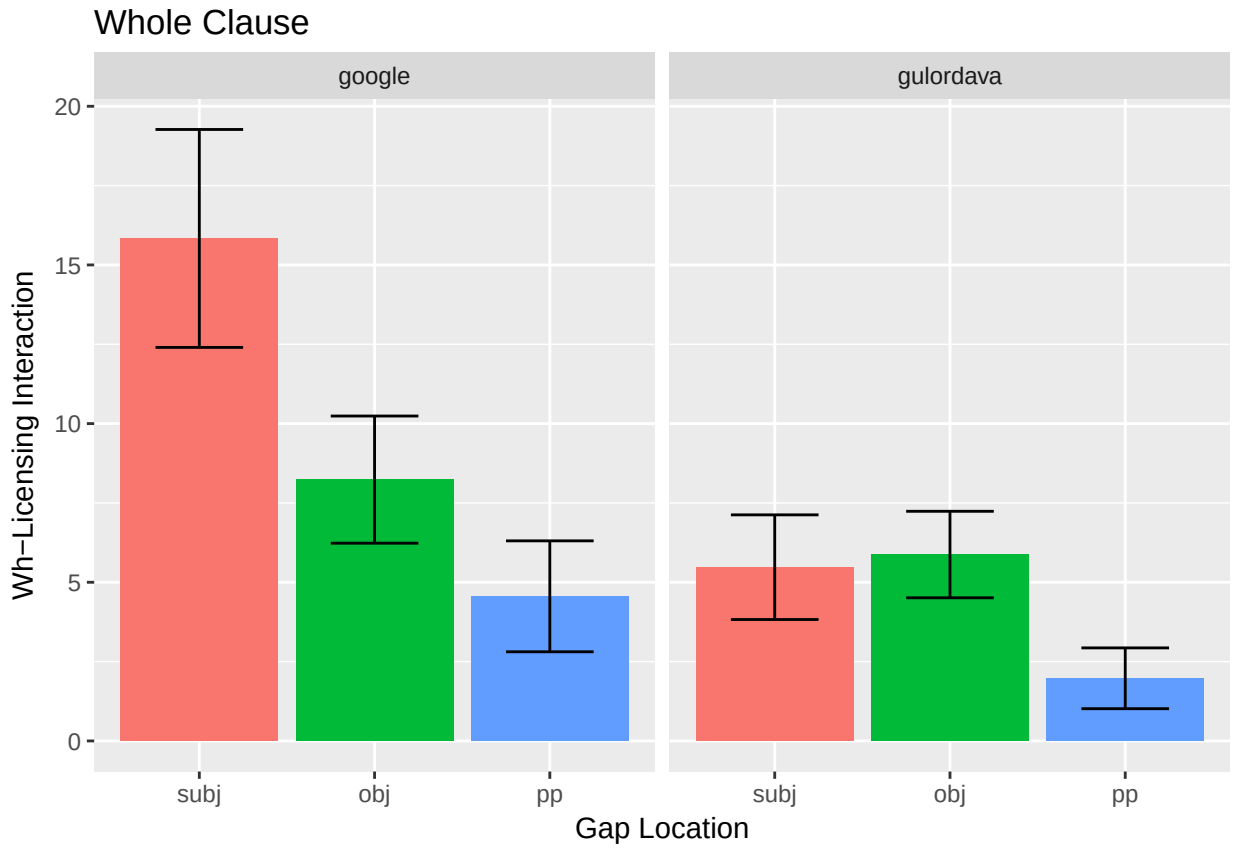



```
ggsave("~/Desktop/island-graphs/position-wheffect-wholeclause.pdf",height=5,width=3.5)
```

This shows the wh-licensing effect (not interaciton) for each of the syntactic positions. For a better picture of what's going on, let's take a look at the whole licensing interaction below:

```
d_full_interaction = d_agg %>%
  filter(region == "embed") %>%
  select(-wh_numeric) %>%
  spread(gap, surprisal) %>%
  mutate(gap_effect=`no-gap`-gap) %>%
  select(-unk, -gap, -`no-gap`) %>%
  spread(wh, gap_effect) %>%
  mutate(wh_interaction=what-`that`)

d_full_interaction %>%
  filter(model == "google" | model == "gulordava") %>%
  group_by(model, gap_position) %>%
    summarise(m=mean(wh_interaction, na.rm=T),
              s=std.error(wh_interaction, na.rm=T),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
  ggplot(aes(x=gap_position, y=m, ymin=lower, ymax=upper, fill=gap_position)) +
```

```
    geom_bar(stat="identity") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model) +
    ylab("Wh-Licensing Interaction") +
    xlab("Gap Location") +
    theme(legend.position="none") +
    ggtitle("Whole Clause")
```




```
ggsave("~/Desktop/island-graphs/position-wholeclause.pdf",height=5,width=3.5)
```

This shows that there is a licensing interaction going on for all of our syntactic positions. However, it looks like for the google model there is a significantly larger licensing interaction that's going on in object position over and above the other two positions. Looking back at the region-by-region plots this seems to be due to the spillover effect, where there's a strong licensing interaction in both the embedded clause verb and also the direct obejct. BUT the size of these interactions isn't significantly taller than the size of the object or PP licensing interactions. Does this mean that the model "Expects" a subject gap more than the others? It's hard to tell, and an interprative question that we should discuss together.

```
d_agg = d_agg %>%
  mutate(gap_numeric=if_else(gap == "gap", 1, -1)) %>%
  mutate(wh_numeric = if_else(wh == "what", 1, -1))

m_google = d_agg %>%
  filter(model == "google", region == "embed") %>%
  lmer(surprisal ~ gap * wh_numeric * gap_position + (gap+wh_numeric+gap_position|sent_index), data=.)
summary(m_google)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## surprisal ~ gap * wh_numeric * gap_position + (gap + wh_numeric +
##     gap_position | sent_index)
##    Data: .
##
## REML criterion at convergence: 1843.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.58729 -0.57190  0.07977  0.51416  2.75419
##
## Random effects:
##  Groups     Name            Variance Std.Dev. Corr
##  sent_index (Intercept)     331.536  18.208
##             gapgap           41.451   6.438   -0.77
##             wh_numeric        4.201   2.050    0.15 -0.46
##             gap_positionobj  20.391   4.516   -0.29 -0.20  0.50
##             gap_positionpp   23.794   4.878   -0.38 -0.18  0.37  0.97
##  Residual                    19.550   4.422
## Number of obs: 288, groups:  sent_index, 24
##
## Fixed effects:
##                                  Estimate Std. Error        df t value
## (Intercept)                     125.67393    3.77111  23.44538  33.325
## gapgap                          -11.80237    1.59427  36.87348  -7.403
## wh_numeric                        6.14589    0.76311 106.17230   8.054
## gap_positionobj                  -0.45577    1.29004  40.44468  -0.353
## gap_positionpp                    0.45503    1.34388  38.46023   0.339
## gapgap:wh_numeric                -7.91758    0.90254 183.99997  -8.773
## gapgap:gap_positionobj            0.02632    1.27639 183.99997   0.021
## gapgap:gap_positionpp             3.25583    1.27639 183.99997   2.551
## wh_numeric:gap_positionobj       -0.45577    0.90254 183.99997  -0.505
## wh_numeric:gap_positionpp         0.45503    0.90254 183.99997   0.504
## gapgap:wh_numeric:gap_positionobj   3.79998    1.27639 183.99997   2.977
## gapgap:wh_numeric:gap_positionpp    5.63898    1.27639 183.99997   4.418
##                                 Pr(>|t|)
## (Intercept)                      < 2e-16 ***
## gapgap                          8.46e-09 ***
## wh_numeric                      1.26e-12 ***
## gap_positionobj                   0.7257
## gap_positionpp                    0.7368
## gapgap:wh_numeric               1.17e-15 ***
## gapgap:gap_positionobj            0.9836
## gapgap:gap_positionpp             0.0116 *
## wh_numeric:gap_positionobj        0.6142
## wh_numeric:gap_positionpp         0.6148
## gapgap:wh_numeric:gap_positionobj   0.0033 **
## gapgap:wh_numeric:gap_positionpp  1.70e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
```

```
##                       (Intr) gapgap wh_nmr gp_pstnb gp_pstnp gpgp:w_
## gapgap                -0.693
## wh_numeric             0.083 -0.207
## gap_postnbj           -0.289  0.082  0.196
## gap_postnpp           -0.357  0.078  0.150  0.747
## gpgp:wh_nmr            0.000  0.000 -0.591  0.000    0.000
## gpgp:gp_pstnb          0.085 -0.400  0.000 -0.495   -0.237    0.000
## gpgp:gp_pstnp          0.085 -0.400  0.000 -0.247   -0.475    0.000
## wh_nmrc:gp_pstnb       0.000  0.000 -0.591  0.000    0.000    0.500
## wh_nmrc:gp_pstnp       0.000  0.000 -0.591  0.000    0.000    0.500
## gpgp:wh_nmrc:gp_pstnb  0.000  0.000  0.418  0.000    0.000   -0.707
## gpgp:wh_nmrc:gp_pstnp  0.000  0.000  0.418  0.000    0.000   -0.707
##                       gpgp:gp_pstnb gpgp:gp_pstnp wh_nmrc:gp_pstnb
## gapgap
## wh_numeric
## gap_postnbj
## gap_postnpp
## gpgp:wh_nmr
## gpgp:gp_pstnb
## gpgp:gp_pstnp          0.500
## wh_nmrc:gp_pstnb       0.000         0.000
## wh_nmrc:gp_pstnp       0.000         0.000         0.500
## gpgp:wh_nmrc:gp_pstnb  0.000         0.000        -0.707
## gpgp:wh_nmrc:gp_pstnp  0.000         0.000        -0.354
##                       wh_nmrc:gp_pstnp gpgp:wh_nmrc:gp_pstnb
## gapgap
## wh_numeric
## gap_postnbj
## gap_postnpp
## gpgp:wh_nmr
## gpgp:gp_pstnb
## gpgp:gp_pstnp
## wh_nmrc:gp_pstnb
## wh_nmrc:gp_pstnp
## gpgp:wh_nmrc:gp_pstnb -0.354
## gpgp:wh_nmrc:gp_pstnp -0.707           0.500
```

```
m_gul = d_agg %>%
  filter(model == "gulordava", region == "embed") %>%
  lmer(surprisal ~ gap * wh_numeric * gap_position + (gap+wh_numeric+gap_position|sent_index), data=.)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : unable to evaluate scaled gradient
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =
## control$checkConv, : Model failed to converge: degenerate Hessian with 1
## negative eigenvalues
```

```
## Warning: Model failed to converge with 1 negative eigenvalue: -7.3e-04
```

```
summary(m_gul)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## surprisal ~ gap * wh_numeric * gap_position + (gap + wh_numeric +
```

```
##     gap_position | sent_index)
##    Data: .
##
## REML criterion at convergence: 1650.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.49197 -0.48512 -0.00132  0.51589  2.79815
##
## Random effects:
##  Groups     Name            Variance Std.Dev. Corr
##  sent_index (Intercept)     409.040  20.225
##             gapgap           39.874   6.315   -0.85
##             wh_numeric        1.466   1.211    0.40  0.00
##             gap_positionobj  15.630   3.954   -0.22 -0.12 -0.97
##             gap_positionpp   41.048   6.407   -0.40  0.20 -0.87  0.89
##  Residual                    14.311   3.783
## Number of obs: 268, groups:  sent_index, 24
##
## Fixed effects:
##                                    Estimate Std. Error       df t value
## (Intercept)                        132.9875     4.1934  23.4219  31.713
## gapgap                             -13.6573     1.5423  33.6850  -8.855
## wh_numeric                           3.0228     0.6250 136.0717   4.836
## gap_positionobj                      0.6197     1.1623  37.8431   0.533
## gap_positionpp                       0.2544     1.5789  28.3158   0.161
## gapgap:wh_numeric                   -2.7377     0.8065 166.2076  -3.394
## gapgap:gap_positionobj              -2.3828     1.1404 166.3290  -2.089
## gapgap:gap_positionpp                0.8162     1.1404 166.3402   0.716
## wh_numeric:gap_positionobj           0.6940     0.8065 166.2076   0.860
## wh_numeric:gap_positionpp            0.4066     0.8065 166.2076   0.504
## gapgap:wh_numeric:gap_positionobj   -0.1006     1.1354 166.7947  -0.089
## gapgap:wh_numeric:gap_positionpp     1.7463     1.1354 166.8037   1.538
##                                    Pr(>|t|)
## (Intercept)                         < 2e-16 ***
## gapgap                             2.59e-10 ***
## wh_numeric                         3.53e-06 ***
## gap_positionobj                     0.59703
## gap_positionpp                      0.87314
## gapgap:wh_numeric                   0.00086 ***
## gapgap:gap_positionobj              0.03819 *
## gapgap:gap_positionpp               0.47516
## wh_numeric:gap_positionobj          0.39075
## wh_numeric:gap_positionpp           0.61485
## gapgap:wh_numeric:gap_positionobj   0.92950
## gapgap:wh_numeric:gap_positionpp    0.12595
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                       (Intr) gapgap wh_nmr gp_pstnb gp_pstnp gpgp:w_
## gapgap                -0.765
## wh_numeric             0.161  0.004
## gap_postnbj           -0.229  0.105 -0.287
```

```
## gap_postnpp           -0.392  0.280 -0.305  0.727
## gpgp:wh_nmr            0.000  0.000 -0.645  0.000    0.000
## gpgp:gp_pstnb          0.064 -0.367  0.000 -0.491   -0.180    0.000
## gpgp:gp_pstnp          0.064 -0.367  0.000 -0.246   -0.361    0.000
## wh_nmrc:gp_pstnb       0.000  0.000 -0.645  0.000    0.000    0.500
## wh_nmrc:gp_pstnp       0.000  0.000 -0.645  0.000    0.000    0.500
## gpgp:wh_nmrc:gp_pstnb -0.001 -0.002  0.455  0.006    0.006   -0.710
## gpgp:wh_nmrc:gp_pstnp -0.002 -0.001  0.455  0.006    0.006   -0.710
##                       gpgp:gp_pstnb gpgp:gp_pstnp wh_nmrc:gp_pstnb
## gapgap
## wh_numeric
## gap_postnbj
## gap_postnpp
## gpgp:wh_nmr
## gpgp:gp_pstnb
## gpgp:gp_pstnp          0.500
## wh_nmrc:gp_pstnb       0.000         0.000
## wh_nmrc:gp_pstnp       0.000         0.000         0.500
## gpgp:wh_nmrc:gp_pstnb  0.000         0.000        -0.710
## gpgp:wh_nmrc:gp_pstnp  0.000         0.000        -0.355
##                       wh_nmrc:gp_pstnp gpgp:wh_nmrc:gp_pstnb
## gapgap
## wh_numeric
## gap_postnbj
## gap_postnpp
## gpgp:wh_nmr
## gpgp:gp_pstnb
## gpgp:gp_pstnp
## wh_nmrc:gp_pstnb
## wh_nmrc:gp_pstnp
## gpgp:wh_nmrc:gp_pstnb -0.355
## gpgp:wh_nmrc:gp_pstnp -0.710           0.505
## convergence code: 0
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
```

What this says is that the licensing interaction is significant in both models, but in the google model there is a reduction in licensing interaction in the goal and pp positions, as it looked like from the bar chart.