Initial Analysis for Few Shot Learning

```r
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------------------------------------------------
## v ggplot2 3.1.0     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.3.1
## v readr   1.2.1     v forcats 0.3.0
## Warning: package 'tibble' was built under R version 3.5.2
## Warning: package 'tidyr' was built under R version 3.5.2
## Warning: package 'purrr' was built under R version 3.5.2
## Warning: package 'dplyr' was built under R version 3.5.2
## -- Conflicts ---------------------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
library(brms)
```

```
## Loading required package: Rcpp
## Warning: package 'Rcpp' was built under R version 3.5.2
## Loading 'brms' package (version 2.6.0). 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.
```

```r
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
```

```r
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)
library(RColorBrewer)
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

##
## Attaching package: 'survival'

## The following object is masked from 'package:brms':
##
##     kidney

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

Verbal Argument Structure

Read in the data. Verb argument structure status (transitive, ambitransitive, intransitive) comes from CELEX2, distributed by the LDC.

```
d = read.csv("test_items_downsample/argstruct-downsample_items.csv", header=FALSE) %>%
  rename("word" = V1) %>%
  rename("verb" = V2) %>%
  rename("pos" = V3) %>%
  rename("freq" = V4) %>%
  rename("is_trans" = V5) %>%
  rename("is_intrans" = V6) %>%
  rename("gram" = V7) %>%
  rename("target_id" = V8) %>%
  rename("item_number" = V9) %>%
  rename("test" = V10)

lstm_results = read.csv("test_items_downsample/argstruct-downsample_lstm_output.txt", sep="\t", header=
  rename("word_1" = V1) %>%
  rename("surprisal" = V2) %>%
  mutate(sent = if_else(word_1 == "<eos>", 1, 0)) %>%
  mutate(sent = cumsum(sent)) %>%
  mutate(sent_pos = 1) %>%
  group_by(sent) %>%
```

```
    mutate(sent_pos = cumsum(sent_pos)) %>%
  ungroup() %>%
  select(-sent) %>%
  mutate(sent_pos = sent_pos - 2) %>%
  filter(word_1 != "<eos>") %>%
  mutate(model = "lstm")

ngram_results = read.csv("test_items_downsample/argstruct-downsample_ngram_output.txt", sep="\t", heade
  rename("word_1" = V1) %>%
  rename("surprisal" = V2) %>%
  mutate(sent = if_else(word_1 == ".", 1, 0)) %>%
  mutate(sent = cumsum(sent)) %>%
  mutate(sent_pos = 1) %>%
  group_by(sent) %>%
    mutate(sent_pos = cumsum(sent_pos)) %>%
  ungroup() %>%
  select(-sent) %>%
  mutate(sent_pos = sent_pos - 1) %>%
  mutate(model = "5gram")

v_counts = read_csv("data/v_counts.csv") %>%
  rename("freq" = `0`) %>%
  select(-X1, -LEMMA) %>%
  spread(XPOS, freq) %>%
  replace(is.na(.), 0) %>%
  mutate(total = VBD + VBN,
         percent_VBD = VBD/total,
         percent_VBN = VBN/total) %>%
  rename("word" = WORD)
```

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

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   WORD = col_character(),
##   LEMMA = col_character(),
##   XPOS = col_character(),
##   `0` = col_double()
## )
```

```
d_args = read_csv("data/wsj_proportions.csv") %>%
  slice(0:14139) %>%
  select(-X1, -lemma) %>%
  mutate(has_dobj = if_else(has_dobj==TRUE, "obj", "no-obj")) %>%
  mutate(has_nsubjpass = if_else(has_nsubjpass == TRUE, "passubj", "no-passsubj")) %>%
  unite("type", 3:4) %>%
  spread(type, count)
```

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

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   word = col_character(),
```

```
##   lemma = col_character(),
##   ptb_pos = col_character(),
##   has_dobj = col_logical(),
##   has_nsubjpass = col_logical(),
##   count = col_double()
## )

## Warning: 7 parsing failures.
##   row           col           expected        actual                        file
## 14138 X1            a double           =======       'data/wsj_proportions.csv'
## 14138 NA            7 columns          1 columns     'data/wsj_proportions.csv'
## 14139 has_dobj      1/0/T/F/TRUE/FALSE has_dobj      'data/wsj_proportions.csv'
## 14139 has_nsubjpass 1/0/T/F/TRUE/FALSE has_nsubjpass 'data/wsj_proportions.csv'
## 14139 count         a double           count         'data/wsj_proportions.csv'
## ..... ............. .................. ............. ..........................
## See problems(...) for more details.
```

```
  #select(-obj_passubj, -NA_NA, -`no-obj_no-passsubj`)
```

Add in factor values based on token frequency:

- "total_freq" counts the number of times the token shows up in the PTB
- "vbd_freq" counts the number of times it appears in transitive contexts
- "vbn_freq" counts the number of times it appears in passive contexts

```
d_ngram = merge(d, ngram_results, by=0, all=TRUE)
d_lstm = merge(d, lstm_results, by=0, all=TRUE)
d_agg = rbind(d_ngram, d_lstm)


d_agg = d_agg %>%
  arrange(as.numeric(Row.names)) %>%
  filter((target_id-1 == sent_pos) | (target_id-2 == sent_pos) | (target_id-3 == sent_pos)) %>%
  filter( !((target_id-3 == sent_pos) & (test=="base-nomod")) ) %>%
  filter( !((target_id-3 == sent_pos) & (test=="base-pres")) ) %>%
  group_by(model, verb, pos, is_trans, is_intrans, gram, test, freq, item_number) %>%
    summarise(surprisal = sum(surprisal)) %>%
  ungroup() %>%
  spread(gram, surprisal) %>%
  mutate(obj_exp = nobj-obj) %>%
  select(-obj, -nobj) %>%
  mutate(is_trans = if_else(((is_trans=="Y") & (is_intrans=="Y")), "Ambitrans", as.character(is_trans))
  mutate(is_trans = if_else(is_trans == "Y", "Trans", is_trans)) %>%
  mutate(is_trans = if_else(is_trans == "N", "Intrans", is_trans)) %>%
  select(-is_intrans, -freq, -pos)


d_agg =  merge(d_agg, v_counts, by.x="verb", by.y = "word") %>%
  mutate(total = if_else(total > 100, 100, total)) %>%
  mutate(VB = if_else(VB > 100, 100, VB)) %>%
  mutate(VBD = if_else(VBD > 100, 100, VBD)) %>%
  mutate(VBN = if_else(VBN > 100, 100, VBN)) %>%

  mutate(vb_freq = "100") %>%
  mutate(vb_freq = if_else(VB<=50, "50", vb_freq)) %>%
```

```r
  mutate(vb_freq = if_else(VB<=30, "30", vb_freq)) %>%
  mutate(vb_freq = if_else(VB<=20, "20", vb_freq)) %>%
  mutate(vb_freq = if_else(VB<=10, "10", vb_freq)) %>%
  mutate(vb_freq = if_else(VB<=5, "5", vb_freq)) %>%
  mutate(vb_freq = if_else(VB<=3, "3", vb_freq)) %>%
  mutate(vb_freq = if_else(VB<=2, "2", vb_freq)) %>%
  mutate(vb_freq = if_else(VB<=1, "1", vb_freq)) %>%
  mutate(vb_freq = if_else(VB<=0, "1", vb_freq)) %>%
  mutate(vb_freq = factor(vb_freq, levels = c("0", "1", "2", "3", "5", "10", "20", "30", "50", "100")))

  mutate(total_freq = "100") %>%
  mutate(total_freq = if_else(total<=50, "50", total_freq)) %>%
  mutate(total_freq = if_else(total<=30, "30", total_freq)) %>%
  mutate(total_freq = if_else(total<=20, "20", total_freq)) %>%
  mutate(total_freq = if_else(total<=10, "10", total_freq)) %>%
  mutate(total_freq = if_else(total<=5, "5", total_freq)) %>%
  mutate(total_freq = if_else(total<=3, "3", total_freq)) %>%
  mutate(total_freq = if_else(total<=2, "2", total_freq)) %>%
  mutate(total_freq = if_else(total<=1, "1", total_freq)) %>%
  mutate(total_freq = if_else(total<=0, "1", total_freq)) %>%
  mutate(total_freq = factor(total_freq, levels = c("0", "1", "2", "3", "5", "10", "20", "30", "50", "1

  mutate(vbd_freq = "100") %>%
  mutate(vbd_freq = if_else(VBD<=50, "50", vbd_freq)) %>%
  mutate(vbd_freq = if_else(VBD<=30, "30", vbd_freq)) %>%
  mutate(vbd_freq = if_else(VBD<=20, "20", vbd_freq)) %>%
  mutate(vbd_freq = if_else(VBD<=10, "10", vbd_freq)) %>%
  mutate(vbd_freq = if_else(VBD<=5, "5", vbd_freq)) %>%
  mutate(vbd_freq = if_else(VBD<=3, "3", vbd_freq)) %>%
  mutate(vbd_freq = if_else(VBD<=2, "2", vbd_freq)) %>%
  mutate(vbd_freq = if_else(VBD<=1, "1", vbd_freq)) %>%
  mutate(vbd_freq = if_else(VBD<=0, "0", vbd_freq)) %>%
  mutate(vbd_freq = factor(vbd_freq, levels = c("0", "1", "2", "3", "5", "10", "20", "30", "50", "100")

  mutate(vbn_freq = "100") %>%
  mutate(vbn_freq = if_else(VBN<=50, "50", vbn_freq)) %>%
  mutate(vbn_freq = if_else(VBN<=30, "30", vbn_freq)) %>%
  mutate(vbn_freq = if_else(VBN<=20, "20", vbn_freq)) %>%
  mutate(vbn_freq = if_else(VBN<=10, "10", vbn_freq)) %>%
  mutate(vbn_freq = if_else(VBN<=5, "5", vbn_freq)) %>%
  mutate(vbn_freq = if_else(VBN<=3, "3", vbn_freq)) %>%
  mutate(vbn_freq = if_else(VBN<=2, "2", vbn_freq)) %>%
  mutate(vbn_freq = if_else(VBN<=1, "1", vbn_freq)) %>%
  mutate(vbn_freq = if_else(VBN<=0, "0", vbn_freq)) %>%
  mutate(vbn_freq = factor(vbn_freq, levels = c("0", "1", "2", "3", "5", "10", "20", "30", "50", "100")
```

Transitive Context Few Shot Learning

Tests

The Object Expectation is the surprisal in the no-obj condition minus the surprisal in the obj condition. Positive means the model expects an object after the verb

Base-Past

- The lion devoured the gazelle today . [obj]
- The lion devoured today . [no-obj]

Base-Pres

- The lion can devour the gazelle today . [obj]
- The lion can devour today . [no-obj]

We measure the surprisal in the "yesterday ." portion of the sentence

Predictions:

(1) The object expectation for transitive should be greater than for intransitive verbs.

(2) The object expectation should be positive for transitive verbs (an object should be expected)

(3) The object expectation should be negative for intransitive verbs (no object should be expected)

(4) Ambitransitive verbs should be in between transitive and intransitive

Results

Plot the object expectation for transitive, intransitive and ambitransitive verbs against their frequency in active contexts (vbn_freq) in the corpus.

Base-Present

We see a couple of things:

NGram:

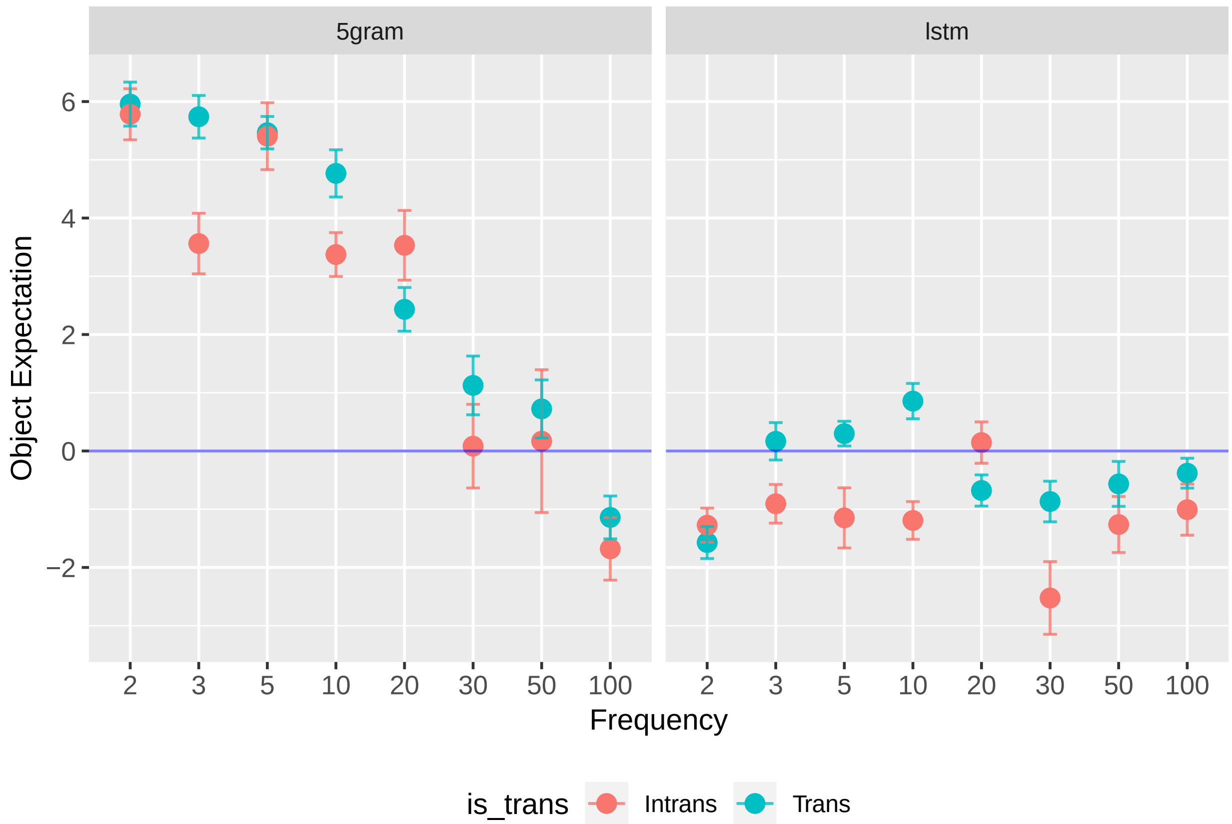
- The transitive verbs are not significantly above the intransitive verbs.
- There is a main effect of verb frequency. As the verb becomes more frequent, the model expects no object regardless of whether it's a transitive or intransitive verb.

LSTM:

- The transitive verbs show more object expectation than the intransitive verbs, except for 2 and 20 exposures.
- No increase in object expectation.

```
d_agg %>%
  filter(test == "base-pres" & is_trans != "Ambitrans") %>%
  group_by(model, is_trans, test, vb_freq) %>%
    summarise(m = mean(obj_exp),
              s=std.error(obj_exp),
              upper=m+1.96*s,
              lower=m-1.96*s)%>%
  ungroup() %>%
  ggplot(aes(x=vb_freq, y=m, ymin=lower, ymax=upper, color=is_trans)) +
    geom_point(stat="identity", position="dodge", size=3) +
    geom_errorbar(width=.2, alpha=0.8) +
    geom_hline(yintercept=0, color="blue", alpha=0.5) +
    ylab("Object Expectation") +
    xlab("Frequency") +
    facet_grid(~model) +
    theme(axis.text=element_text(size=10),
          legend.position = "bottom")
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```




```r
ggsave("./images/argstruct-downsample_base-pres.png",height=5,width=5)
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```

Base-Past

NGram:

- The intransitive verbs show higher object expectation than the transitive verbs. (This is odd..)
- There is a main effect of verb frequency. As the verb becomes more frequent, the model expects no object regardless of whether it's a transitive or intransitive verb.

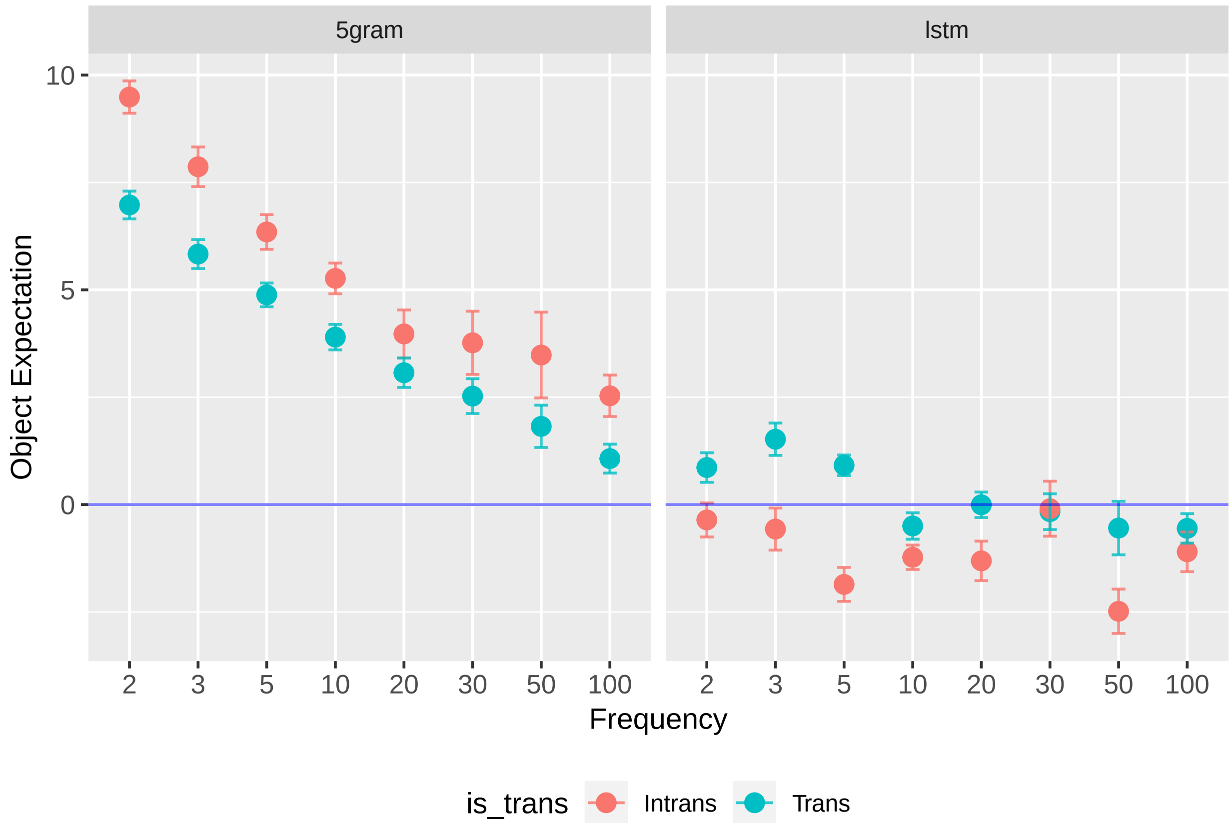
LSTM:

- The transitive verbs show more object expectation than the intransitive verbs, except for 30 exposures.

```r
d_agg %>%
  filter(test == "base-nomod" & is_trans != "Ambitrans") %>%
  group_by(model, is_trans, test, vbd_freq) %>%
    summarise(m = mean(obj_exp),
              s=std.error(obj_exp),
              upper=m+1.96*s,
              lower=m-1.96*s)%>%
  ungroup() %>%
  ggplot(aes(x=vbd_freq, y=m, ymin=lower, ymax=upper, color=is_trans)) +
    geom_point(stat="identity", position="dodge", size=3) +
    geom_errorbar(width=.2, alpha=0.8) +
```

```
    geom_hline(yintercept=0, color="blue", alpha=0.5) +
    ylab("Object Expectation") +
    xlab("Frequency") +
    facet_wrap(~model) +
    theme(axis.text=element_text(size=10),
          legend.position = "bottom")
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```




```
ggsave("./images/argstruc-downsamplet_base.png",height=5,width=5)
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```

Passive Context Few Shot Learning

Test Items

Again, object expectation is the surprisal in the object condition minus the surprisal in the no-object condition. Now, it may be the case that all the model is doing is learning that some verbs can occur in a "was + VERB" context. If so, that's fine. This test is – in some sense – a control (the model would fail to generalize for verbs it hasn't seen in the passive context).

- The gazelle was devoured yesterday . [object]
- The gazelle devoured yesterday . [no-object]

Example with transitive verb

- The gazelle was slept yesterday . [object]

- The gazelle slept yesterday . [no-object]

Modifier

- The gazelle was quickly and rapidly devoured yesterday . [obj]
- The gazelle quickly and rapidly devoured yesterday . [no-obj]

Long Modifier

- The gazelle was quickly , rapidly , and totally devoured yesterday . [obj]
- The gazelle quickly , rapidly , and totally devoured yesterday . [no-obj]

We measure the object expectation in the "devoured yesterday ." portion of the sentences.

Predictions:

(1) The object expectation for transitive should be greater than for intransitive verbs.

(2) The object expectation should be positive for transitive verbs (it is more likely to occur with a passive object than without an object)

(3) The object expectation should be negative for intransitive verbs (it is more likely to occur without an object than with a passive object)

(4) Ambitransitive verbs should be in between transitive and intransitive

Results

Here we have to plot against the frequency in total contexts, becaues (theoretically) intransitive verbs don't occur in passive contexts. (Actually, according to the PTB there are a number of times intransitive verbs occur passivally. I'm investigating this further.)

We see a couple of things:

- For intransitive verbs: The object expectation stars out positive and becomes negative with more exposure, indicating that the models learn proper transitive behavior with greater exposure. Crucially, the learning rate is slower than for the active contexts. This is because the model has less (or no) exposure for these verbs in passive contexts.

- For transitive verbs: The object expectation starts off postivie, and increases with exposure.

- There is a significant difference between transitive and intransitive after only 2 exposures! Strong evidence of few-shot learning capeabilities

- The ambi-transitive verbs are between transitive and intransitives.

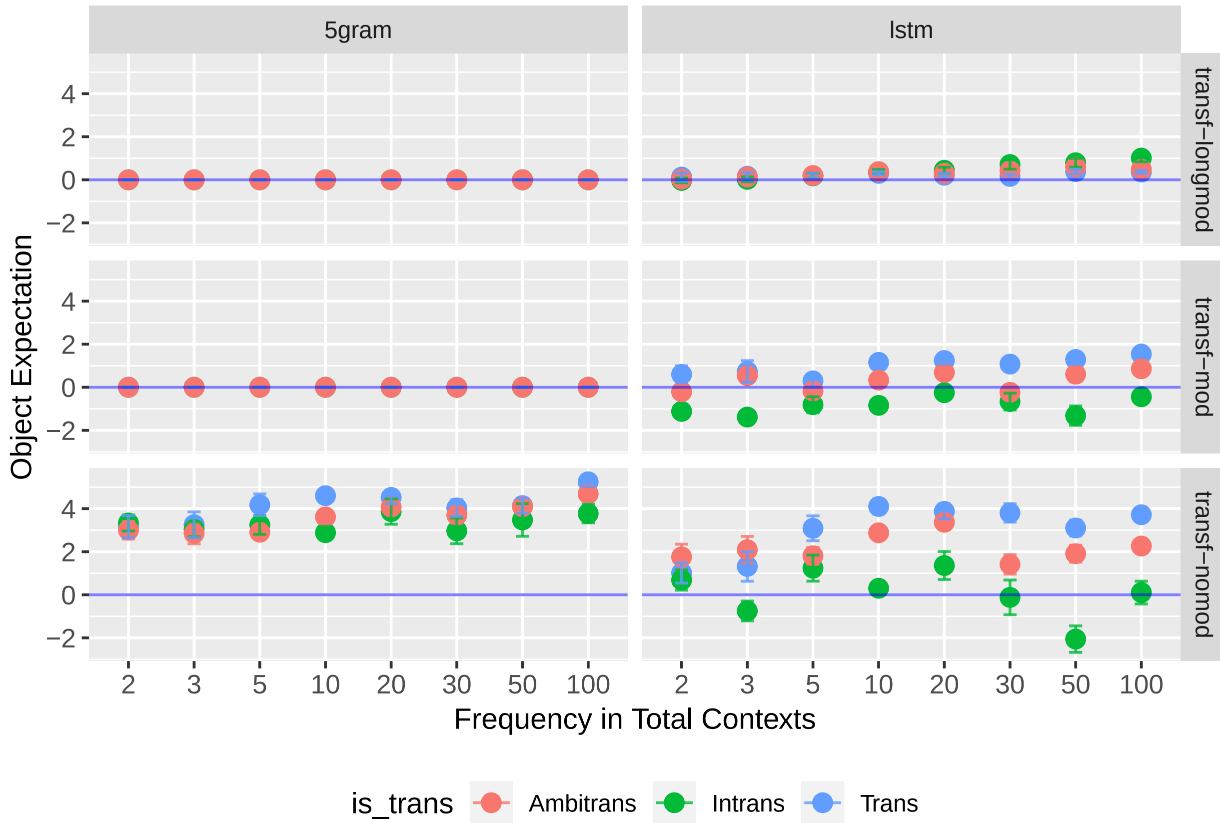
LSTM Baseline:

- We see no difference between the conditions for the modified sentences. For the no-modifier sentences, we see same pattern of behavior as with the LSTM model.

```
d_agg %>%
  filter( test=="transf-nomod" | test == "transf-mod" | test == "transf-longmod") %>%
  group_by(model, is_trans, test, total_freq) %>%
    summarise(m = mean(obj_exp),
              s=std.error(obj_exp),
              upper=m+1.96*s,
              lower=m-1.96*s)%>%
  ungroup() %>%
  ggplot(aes(x=total_freq, y=m, ymin=lower, ymax=upper, color=is_trans)) +
    geom_point(stat="identity", position="dodge", size=3) +
    geom_errorbar(width=.2, alpha=0.8) +
```

```r
    geom_hline(yintercept=0, color="blue", alpha=0.5) +
    ylab("Object Expectation") +
    xlab("Frequency in Total Contexts") +
    facet_grid(test ~ model) +
    theme(axis.text=element_text(size=10),
          legend.position = "bottom")
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```




```r
ggsave("./images/argstruct-downsample_transf.png",height=5,width=5)
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```

Transformations

Test Items

Now, we look at the learning rate for the transformed test, for verbs that occur *only* in the active contexts in the training data. I have to clip for verbs that occur only 10 or fewer times, because there weren't many intransitive and transitive verbs that occured only in the active context (the PTB has a lot of passive voice!).

- The gazelle was (quickly and rapidly) devoured yesterday . [object]
- The gazelle (quickly and rapidly) devoured yesterday . [no-object]

Predictions: If the model is able to learn something about verbal argument structure, then we expect our previos predictions to hold

(1) The object expectation for transitive should be greater than for intransitive verbs.

(2) The object expectation should be positive for transitive verbs (it is more likely to occur with a passive object than without an object)

(3) The object expectation should be negative for intransitive verbs (it is more likely to occur without an object than with a passive object)

Results

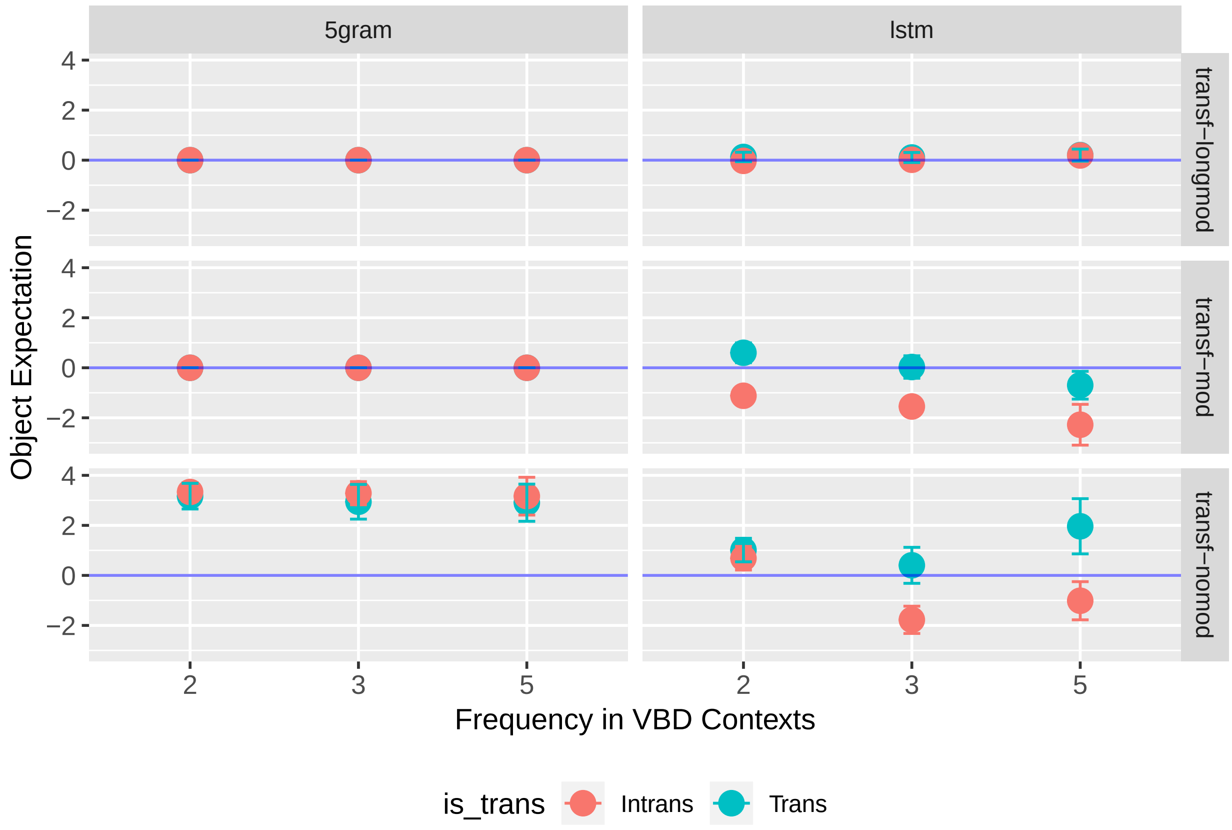
- For intransitive verbs: The object expectation stars out positive and becomes negative with more exposure.
- For transitive verbs: The object expectation starts off postivie, and remains positive.
- There is a significant difference between transitive and intransitive after only 2 exposures!

LSTM Baseline:

- We see no difference between the two conditions.

```
d_agg %>%
  filter(VBN == 0, test=="transf-nomod" | test == "transf-mod" | test == "transf-longmod", is_trans !=
  filter(vbd_freq != "20" & vbd_freq != "100" & vbd_freq != "10") %>%
  group_by(model, is_trans, test, vbd_freq) %>%
    summarise(m = mean(obj_exp),
              s=std.error(obj_exp),
              upper=m+1.96*s,
              lower=m-1.96*s)%>%
  ungroup() %>%
  ggplot(aes(x=vbd_freq, y=m, ymin=lower, ymax=upper, color=is_trans)) +
    geom_point(stat="identity", position="dodge", size=4) +
    geom_errorbar(width=.1) +
    geom_hline(yintercept=0, color="blue", alpha=0.5) +
    ylab("Object Expectation") +
    xlab("Frequency in VBD Contexts") +
    facet_grid(test ~ model) +
    theme(axis.text=element_text(size=10),
          legend.position = "bottom")
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```




```
ggsave("./images/argstruct-downsample_invar.png",height=5,width=5)
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`
```

Nominal Number Learning

Read in the data.

```
d = read.csv("test_items_downsample/number-downsample_items.csv", header=FALSE) %>%
  rename("word" = V1) %>%
  rename("verb" = V2) %>%
  rename("pos" = V3) %>%
  rename("freq" = V4) %>%
  rename("gram" = V5) %>%
  rename("target_id" = V6) %>%
  rename("item_number" = V7) %>%
  rename("test" = V8) %>%
  filter(word != ".")

lstm_results = read.csv("test_items_downsample/number-downsample_lstm_output.txt", sep="\t", header=FALS
  rename("word_1" = V1) %>%
  rename("surprisal" = V2) %>%
  mutate(sent = if_else(word_1 == "<eos>", 1, 0)) %>%
  mutate(sent = cumsum(sent)) %>%
  mutate(sent_pos = 1) %>%
  group_by(sent) %>%
```

```
    mutate(sent_pos = cumsum(sent_pos)) %>%
  ungroup() %>%
  select(-sent) %>%
  mutate(sent_pos = sent_pos - 1) %>%
  filter(word_1 != "<eos>") %>%
  mutate(model = "lstm")

ngram_results = read.csv("test_items_downsample/number-downsample_ngram_output.txt", sep="\t", header=F
  rename("word_1" = V1) %>%
  rename("surprisal" = V2) %>%
  mutate(sent = if_else(word_1 == ".", 1, 0)) %>%
  mutate(sent = cumsum(sent)) %>%
  mutate(sent_pos = 1) %>%
  group_by(sent) %>%
    mutate(sent_pos = cumsum(sent_pos)) %>%
  ungroup() %>%
  select(-sent) %>%
  mutate(sent_pos = sent_pos - 1) %>%
  filter(word_1 != ".") %>%
  mutate(model = "5gram")

rnng_results = read.csv("test_items_downsample/number-downsample_rnng_output.txt", sep="\t", header=FALS
  rename("word_1" = V1) %>%
  rename("surprisal" = V2) %>%
  mutate(sent = if_else(word_1 == "<eos>", 1, 0)) %>%
  mutate(sent = cumsum(sent)) %>%
  mutate(sent_pos = 1) %>%
  group_by(sent) %>%
    mutate(sent_pos = cumsum(sent_pos)) %>%
  ungroup() %>%
  select(-sent) %>%
  mutate(sent_pos = sent_pos - 1) %>%
  filter(word_1 != "<eos>") %>%
  mutate(model = "rnng")

d_ngram = merge(d, ngram_results, by=0, all=TRUE)
d_lstm = merge(d, lstm_results, by=0, all=TRUE)
d_rnng = merge(d, rnng_results, by=0, all=TRUE)
d_all = rbind(d_ngram, d_lstm, d_rnng)

d_agg = d_all %>%
  arrange(as.numeric(Row.names)) %>%
  filter(target_id == sent_pos) %>%
  select(-Row.names, -word, -target_id, -word_1, -sent_pos) %>%
  spread(gram, surprisal) %>%
  mutate(pl_exp = sing-pl) %>%
  select(-sing, -pl) %>%
  mutate(freq = if_else(freq>100, 100, freq/1)) %>%
  mutate(freq_cat = "100") %>%
  #mutate(freq_cat = if_else(freq<=50, "50", freq_cat)) %>%
  #mutate(freq_cat = if_else(freq<=20, "20", freq_cat)) %>%
  mutate(freq_cat = if_else(freq<=10, "10", freq_cat)) %>%
  mutate(freq_cat = if_else(freq<=5, "5", freq_cat)) %>%
```

```r
  mutate(freq_cat = if_else(freq<=4, "4", freq_cat)) %>%
  mutate(freq_cat = if_else(freq<=3, "3", freq_cat)) %>%
  mutate(freq_cat = if_else(freq<=2, "2", freq_cat)) %>%
  mutate(freq_cat = if_else(freq<=1, "1", freq_cat)) %>%
  mutate(freq_cat = factor(freq_cat, levels = c("1", "2", "3", "4", "5", "10", "100")))
```

Base Context Few Shot Learning

For these results I plot learning against frequency in both base and transformed contexts.

Test Items

We measure the plural expectation by taking the surprisal at "is" minus the surprisal at "are" for two classes of nouns: Singular nouns (NN) and Plural nouns (NNS)

Base simple

- The president is [sing]
- The president are [pl]

Base_PP: A nominal distractor of the opposite number

- The president with the documents is…
- The president with the documents are…

Predictions:

(1) For singular nouns (NN): Negative plural expectation
(2) For plural nouns (NNS): Positive plural expectation
(3) Difference in expectation between NN and NNS

Results

Simple Condition

- We find a significant difference between NN and NNS with even one exposure
- We find a positive plural expectation for NNS after two exposures

Base_PP Condition

- We find a significant difference between NN and NNS after two exposures
- The singular expectation grows with exposure for NNS, but is never greater than zero

Model Comparison:

- We see the largest difference between the two conditions with the RNNG

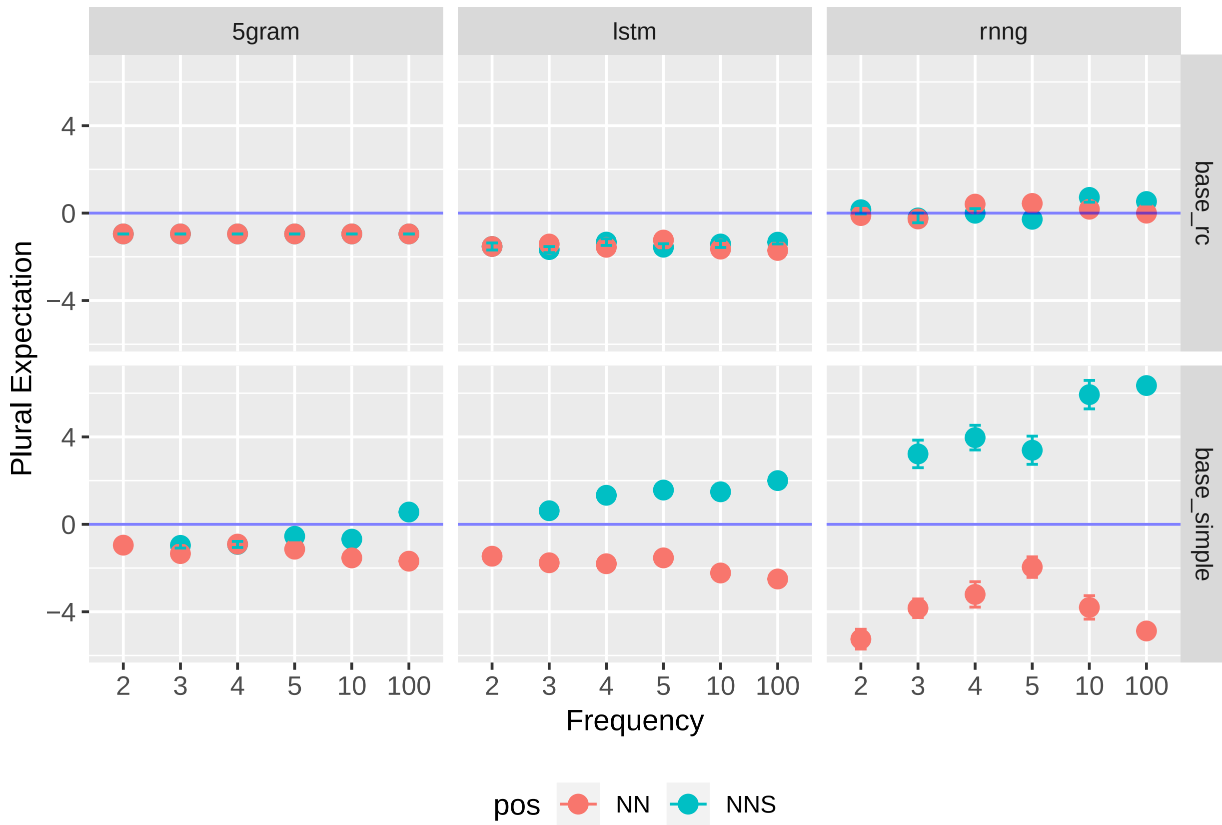
```r
d_agg %>%
  filter(test == "base_pp" | test == "base_simple" | test == "base_rc") %>%
  group_by(model, pos, freq_cat, test) %>%
    summarise(m = mean(pl_exp),
              s=std.error(pl_exp),
              upper=m+1.96*s,
              lower=m-1.96*s)%>%
  ungroup() %>%
  ggplot(aes(x=freq_cat, y=m, ymin=lower, ymax=upper, color=pos)) +
    geom_point(stat="identity", position="dodge", size=3, alpha=1) +
```

```
    geom_errorbar(width=.2, alpha=1) +
    geom_hline(yintercept=0, color="blue", alpha=0.5) +
    ylab("Plural Expectation") +
    xlab("Frequency") +
    facet_grid(test ~ model) +
    #ylim(-1, 1) +
    theme(axis.text=element_text(size=10),
          legend.position = "bottom")
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`

## Warning: Removed 3 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_errorbar).
```




```
ggsave("./images/number-downsample_base.png",height=5,width=5)
```

```
## Warning: Width not defined. Set with `position_dodge(width = ?)`

## Warning: Removed 3 rows containing missing values (geom_point).

## Warning: Removed 3 rows containing missing values (geom_errorbar).
```

Transformed Context Few Shot Learning

Test Items

We measure the plural expectation by taking the surprisal difference between the two conditions at the noun. We use singular nouns (NN) and plural nouns (NNS)

Transf_simple:

- Is/was the president… [sing]
- Are/were the president… [pl]

Transf_mod:

- Is/was the very big and important president…[sing]
- Are/were the very big and important president… [pl]

Predictions:

(1) For singular nouns (NN): Negative plural expectation
(2) For plural nouns (NNS): Positive plural expectation
(3) Difference in expectation between NN and NNS

Results

Note: The scale here is *really small*. Because we have so many items the difference is probably significant (I haven't ran the stats yet, but the error bars are tiny). However, the diffence is like 1/5 of a bit of surprisal.

Simple Condition

- We find a significant difference between NN and NNS after two exposures, except for nouns that occur 3 times in training.
- The NN never has a negative plural expectation.

Modifier Condition

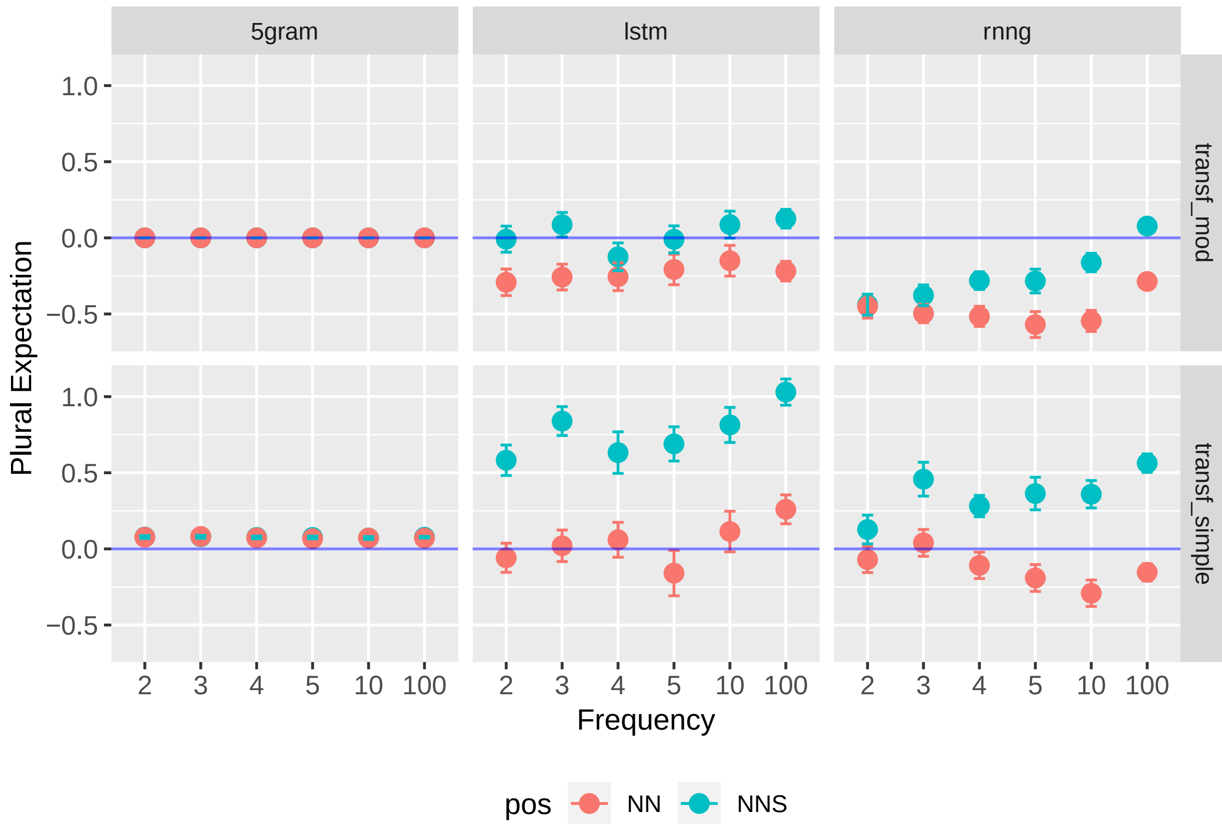
- We find a significant difference between NN and NNS after two exposures, but again for nouns that occur 3 timse the difference does not look significant.
- Again, the plural expectation for NN doesn't really go below zero (except for maybe one bucket)

Model Comparison:

- The RNNG is the only model that, for the simple test, shows positive expectation for plurals and negatie expectation for the singulars.

```
d_agg %>%
  filter(test == "transf_simple" | test == "transf_mod") %>%
  group_by(model, pos, freq_cat, test) %>%
    summarise(m = mean(pl_exp),
              s=std.error(pl_exp),
              upper=m+1.96*s,
              lower=m-1.96*s)%>%
  ungroup() %>%
  ggplot(aes(x=freq_cat, y=m, ymin=lower, ymax=upper, color=pos)) +
    geom_point(stat="identity", position="dodge", size=3) +
    geom_errorbar(width=.2) +
    geom_hline(yintercept=0, color="blue", alpha=0.5) +
    ylab("Plural Expectation") +
    xlab("Frequency") +
    facet_grid(test ~ model) +
    #ylim(-1, 1) +
    theme(axis.text=element_text(size=10),
          legend.position = "bottom")
```

Warning: Width not defined. Set with `position_dodge(width = ?)`




```
ggsave("./images/number-downsample_transf.png",height=5,width=5)
```

Warning: Width not defined. Set with `position_dodge(width = ?)`

To Do: Invarience to Transformation