Concept Learning Data Analysis

Sahil Chopra

May 2, 2018

Introduction

Here, we present the preliminary analysis of the concept learning data collected April 30 - May 2, 2018. Before we delve into our analysis, we summurize the stimuli and concepts utilized in this experiment.

The stimuli have four axes of variablity, each with three possible values – allowing for 81 unique critters. 50 critters were used in training and 31 were held out for test. Teachers and Listeners, who were paired together, were provided the same critters during the test set.

The four axes of variability were as follows:

- Critter Type (Bug, Fish, Bird)
- Primary Color (Blue, Green, Orange)
- Secondary Color (Red, Yellow, Purple)
- Size (Small, Medium, Large)

We intentionally ommitted rules that relied on size. Unlike the other three properties, size is used described in relativeterms, e.g. "small", "medium", "large". Without visual grounding, it's unclear whether a listener would immediately understand what these terms refer to, when presented the test set. We will address this issue in later iterations of thise experiment.

We ran 5 concepts:

- 1 Single Feature Concept
- 2 Logical Conjunctions
- 2 Logical Disjunctions

For each concept, there were two lists. Each list comprised of the same stimuli, with different test / train splits and orderings of stimuli. We ran two lists to make sure that learning at similar rates was possible for a concept irrespective of the specific ordering of stimuli.

The 5 Concepts were:

- Primary Color == Orange
- Critter Type == Fish && Primary Color == Blue
- Primary Color == Orange && Secondary Color == Purple
- Critter Type == Bug || Secondary Color == Yellow
- Critter Type == Bird || Primary Color == Green

Data Processing

```
library(reshape2)
library(purrr)
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'

## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##     smiths
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(scales)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard
```

```
library(rwebppl)
```

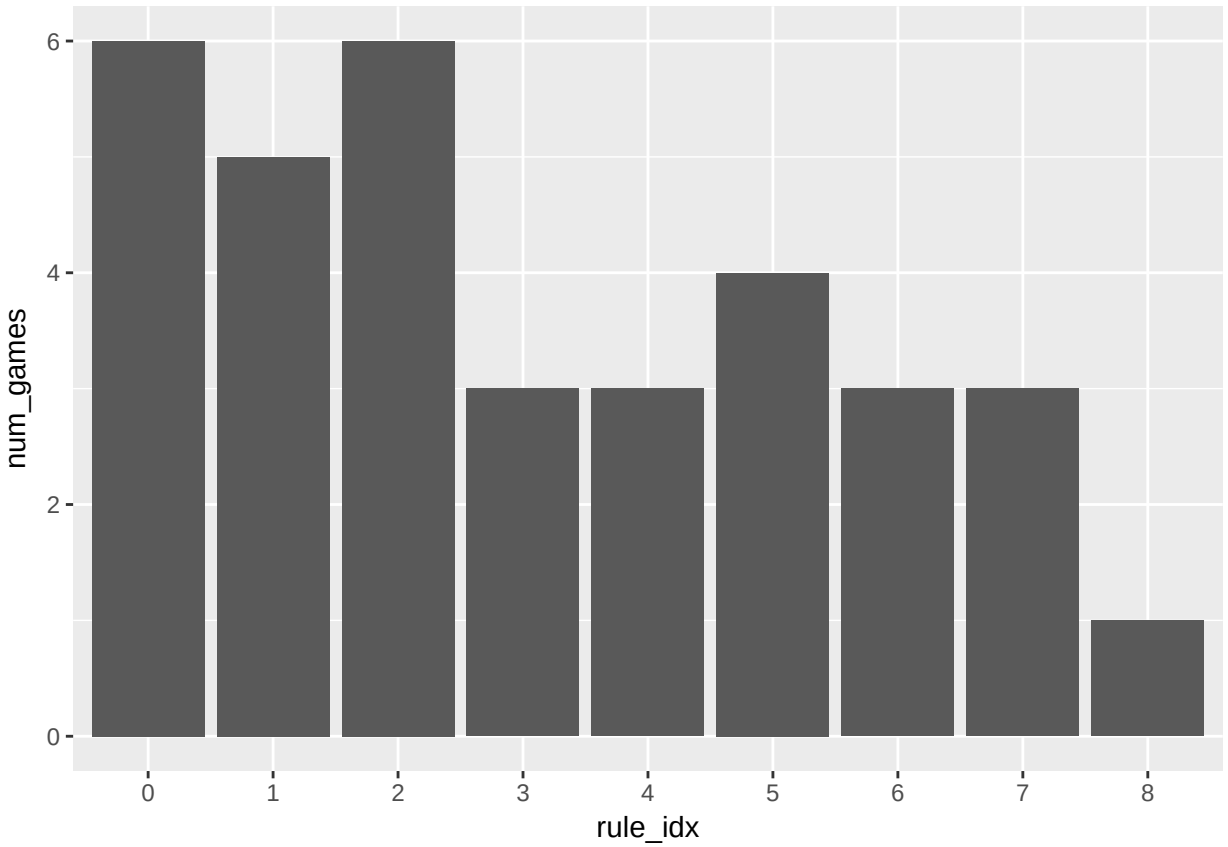
```
## using webppl version: v0.9.7 /Library/Frameworks/R.framework/Versions/3.3/Resources/library/rwebppl/
```

```
# Load Training Trial Responses
temp <- list.files(
  "../../mturk/mp-game-3/experiment_1/results-cleaned/train_trials",
  pattern="*.csv",
  full.names=TRUE
)
train_trials <- do.call(rbind, lapply(temp, read.csv))

# Load Test Trial Responses
temp <- list.files(
  "../../mturk/mp-game-3/experiment_1/results-cleaned/test_trials",
  pattern="*.csv",
  full.names=TRUE
)
test_trials <- do.call(rbind, lapply(temp, read.csv))

# Load Training Summary Stats
```

```
temp <- list.files(
  "../../mturk/mp-game-3/experiment_1/results-cleaned/train_summary_stats",
  pattern="*.csv",
  full.names=TRUE
)
train_stats <- do.call(rbind, lapply(temp, read.csv))

# Load Test Summary Stats
temp <- list.files(
  "../../mturk/mp-game-3/experiment_1/results-cleaned/test_summary_stats",
  pattern="*.csv",
  full.names=TRUE
)
test_stats <- do.call(rbind, lapply(temp, read.csv))

# Load Chat Logs
temp <- list.files(
  "../../mturk/mp-game-3/experiment_1/results-cleaned/chat_messages",
  pattern="*.csv",
  full.names=TRUE
)
chat_logs <- do.call(rbind, lapply(temp, read.csv))
```

Dataset Composition

```
rule_freq <- as.data.frame(table(train_stats$rule_idx)) %>%
  rename(rule_idx = Var1) %>%
  rename(num_games = Freq)
ggplot(rule_freq, aes(x=rule_idx, y = num_games)) + geom_bar(stat="identity")
```



Due to some server issues early on we have a slight imbalance in the number of trials prescribed to each rule index. We intended to have ~3 per rule type originally.

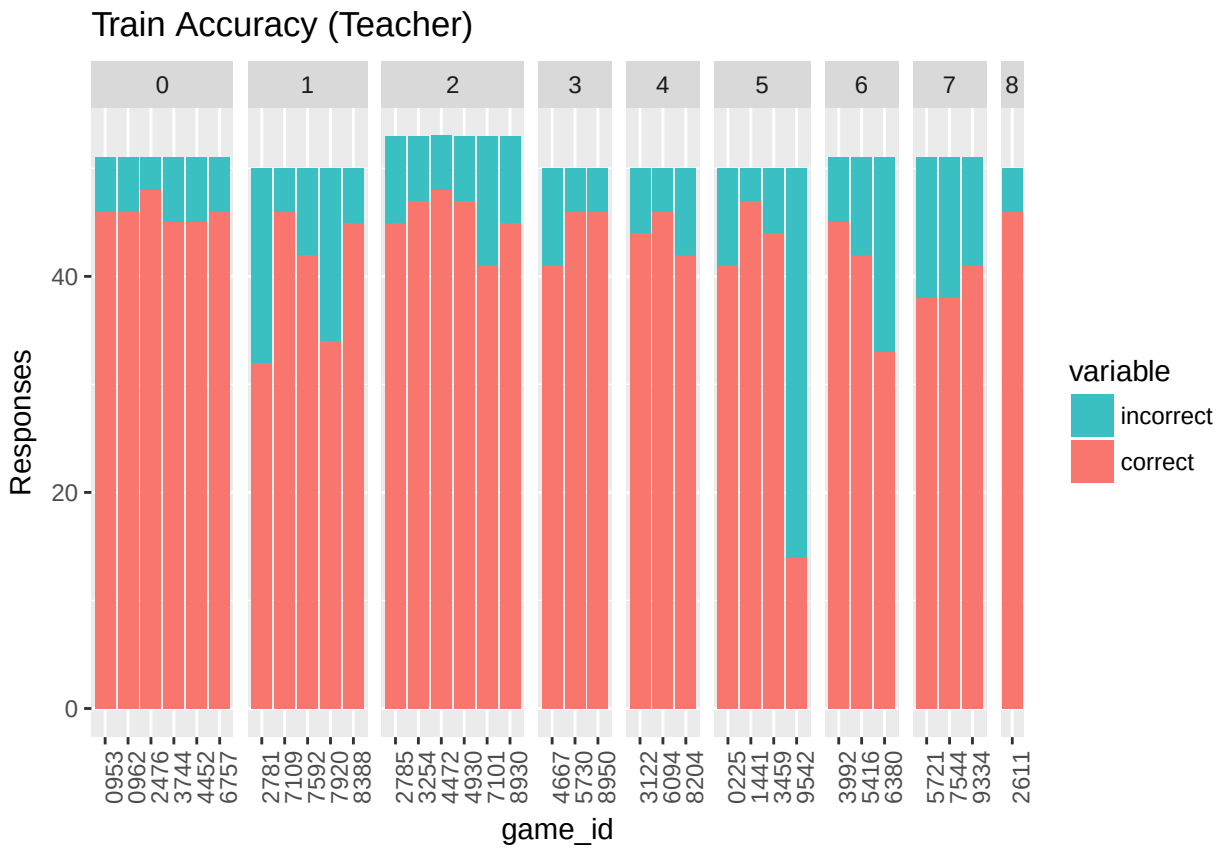
Analysis: Accuracy

First, we examine how people perform on the train and test splits, for each specific list.

```
num_train = 5
num_test = 31

train_acc <- train_stats %>%
  mutate(correct = hits + correct_rejections) %>%
  mutate(incorrect = misses + false_alarms) %>%
  select(game_id, rule_idx, correct, incorrect)

test_acc_teacher <- test_stats %>%
  filter(role == "explorer") %>%
  mutate(correct = hits + correct_rejections) %>%
  mutate(incorrect = misses + false_alarms) %>%
  select(game_id, rule_idx, correct, incorrect)

test_acc_listener <- test_stats %>%
  filter(role == "student") %>%
  mutate(correct = hits + correct_rejections) %>%
  mutate(incorrect = misses + false_alarms) %>%
  select(game_id, rule_idx, incorrect, correct)
```

Training

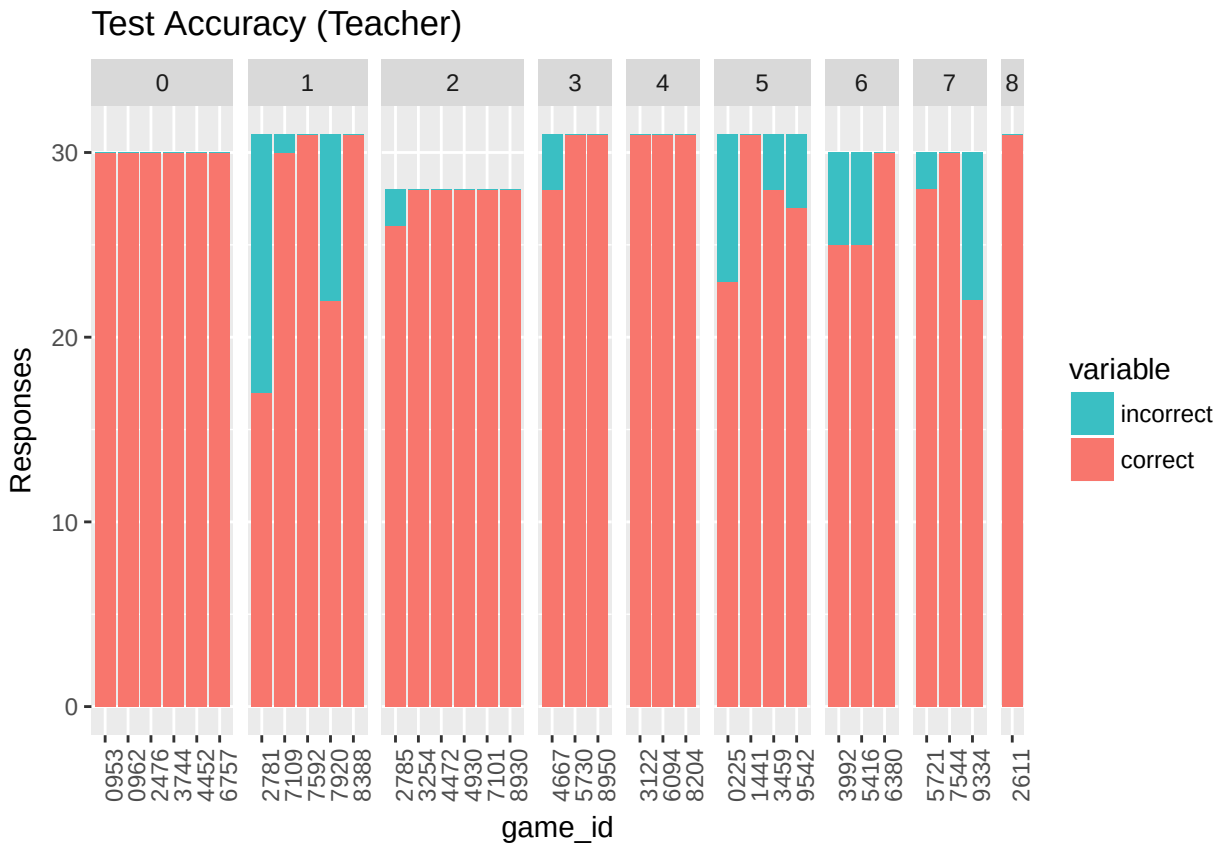
```
temp <- melt(train_acc, id.var= c("game_id", 'rule_idx'))
temp$variable <- factor(temp$variable, levels = c("incorrect", "correct"))
training_acc_plot <- ggplot(temp, aes(x=game_id, y=value, fill=variable)) +
  geom_bar(stat = "identity") +
  facet_grid(.~rule_idx, scales="free_x", space="free") +
  ylab("Responses") +
  theme(axis.text.x=element_text(angle=90)) +
  scale_x_discrete(label=abbreviate) +
  labs(title = "Train Accuracy (Teacher)") +
  scale_fill_manual( values = c("#3ABFC3", "#F8766D"))
plot(training_acc_plot)
```



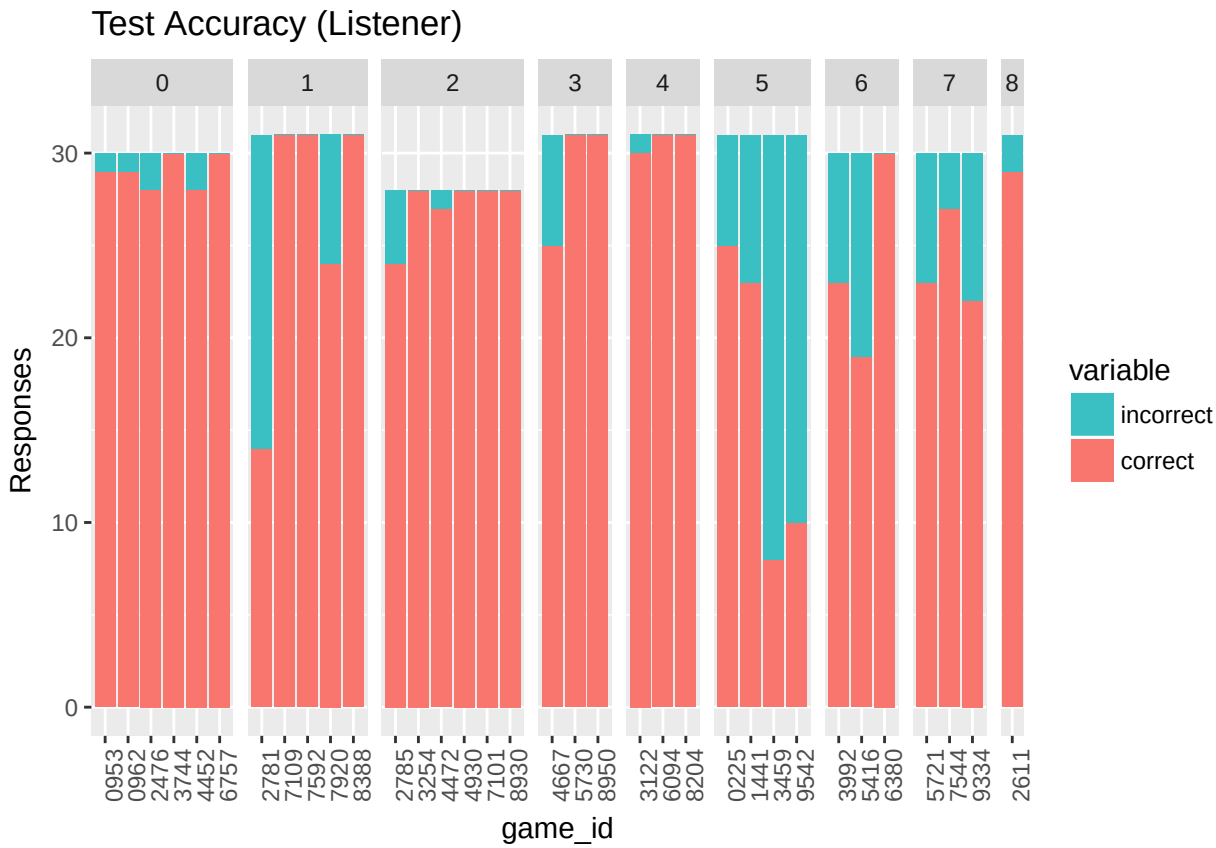
Test

```
temp <- melt(test_acc_teacher, id.var= c("game_id", 'rule_idx'))
temp$variable <- factor(temp$variable, levels = c("incorrect", "correct"))
test_acc_teacher_plot <- ggplot(temp, aes(x=game_id, y=value, fill=variable)) +
  geom_bar(stat = "identity") +
  facet_grid(.~rule_idx, scales="free_x", space="free") +
  ylab("Responses") +
  theme(axis.text.x=element_text(angle=90)) +
  scale_x_discrete(label=abbreviate) +
  labs(title = "Test Accuracy (Teacher)") +
```

```
  scale_fill_manual( values = c("#3ABFC3", "#F8766D"))
plot(test_acc_teacher_plot)
```




```
temp <- melt(test_acc_listener, id.var= c("game_id", 'rule_idx'))
temp$variable <- factor(temp$variable, levels = c("incorrect", "correct"))
test_acc_listener_plot <- ggplot(temp, aes(x=game_id, y=value, fill=variable)) +
  geom_bar(stat = "identity") +
  facet_grid(.~rule_idx, scales="free_x", space="free") +
  ylab("Responses") +
  theme(axis.text.x=element_text(angle=90)) +
  scale_x_discrete(label=abbreviate) +
  labs(title = "Test Accuracy (Listener)") +
  scale_fill_manual( values = c("#3ABFC3", "#F8766D"))
plot(test_acc_listener_plot)
```

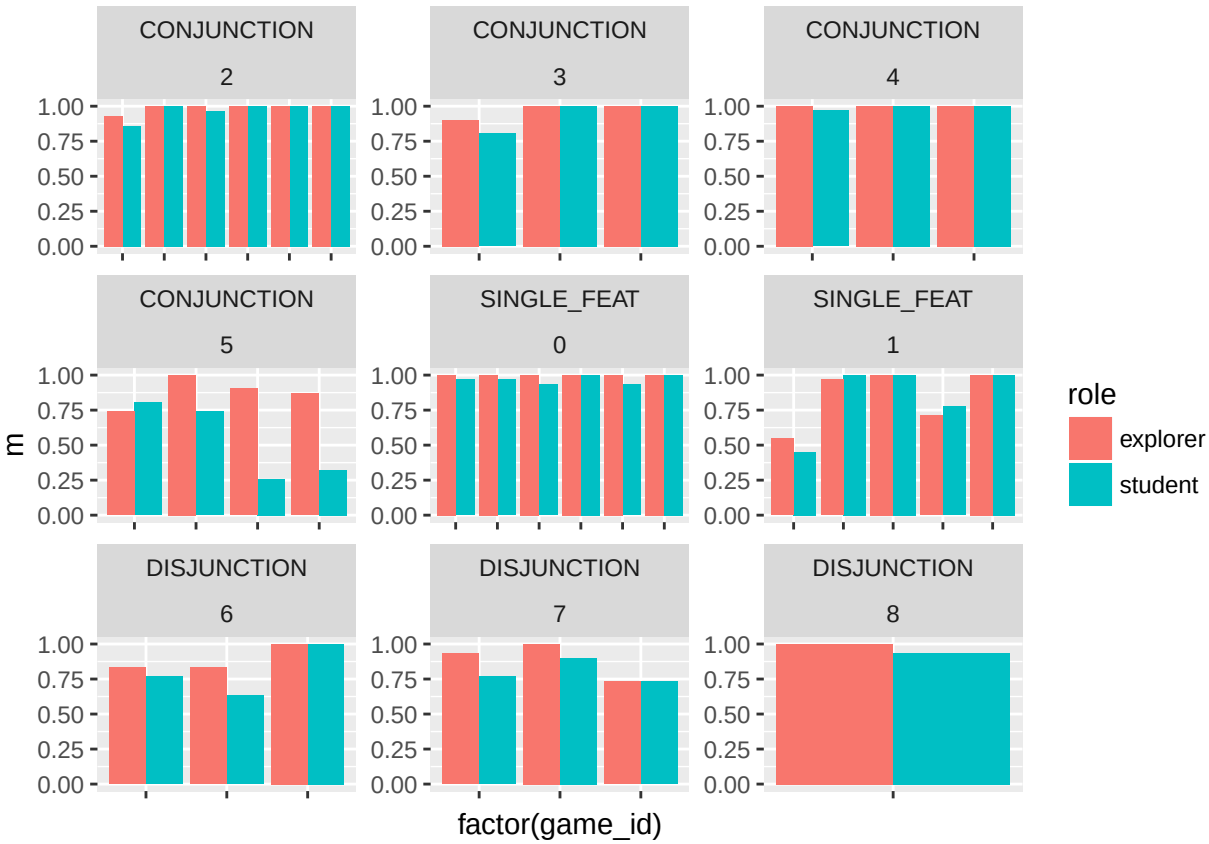


Note that test accuracy of listeners is often very similar to the test accuracy of teacher. There are few notable exceptions.

For rule_idx 3 game 1264, we see that the student performs much worse than the teacher. Upon examination of the chat logs, it seems like the student "hit" continue without conversing with the explorer. They only said "hello" to each other, before moving on to the test set. Thus, their performance is indicative of random guessing.

We also see significant discrepancies in performance on rule_idx 5. The rule here was Primary Color == Orange && Secondary Color == Purple. In Game 34589, the teacher correctly understood the concept, but said they also thought that fish couldn't be wudsy, mentioning that he had never seen orange and purple fish during training. In Game 9542, the teacher told the student that "I didn't notice any patterns at all." – so again the student was left with no information, i.e. an uninformative prior.

```
test_stats %>%
  mutate(acc = (hits + correct_rejections) / (hits + correct_rejections + false_alarms + misses)) %>%
  group_by(rule_type, rule_idx, game_id, role) %>%
  summarize(m = mean(acc)) %>%
  ggplot(., aes(x = factor(game_id), y = m, fill = role))+
  geom_col(position = position_dodge())+
  facet_wrap(~rule_type + rule_idx, scales = 'free')+
  theme(axis.text.x = element_blank())
```



Here we see relative accuracy of dyad pairs, grouped by rule type and rule index. Note that 0,1 are the same concept but with different lists. Same goes for 2,3; 4,5; 6,7; etc.

Analysis: Hits/Misses and Correct Rejections/False Alarms

Nexr, we break down how people perform on the train and test splits, for each specific list.

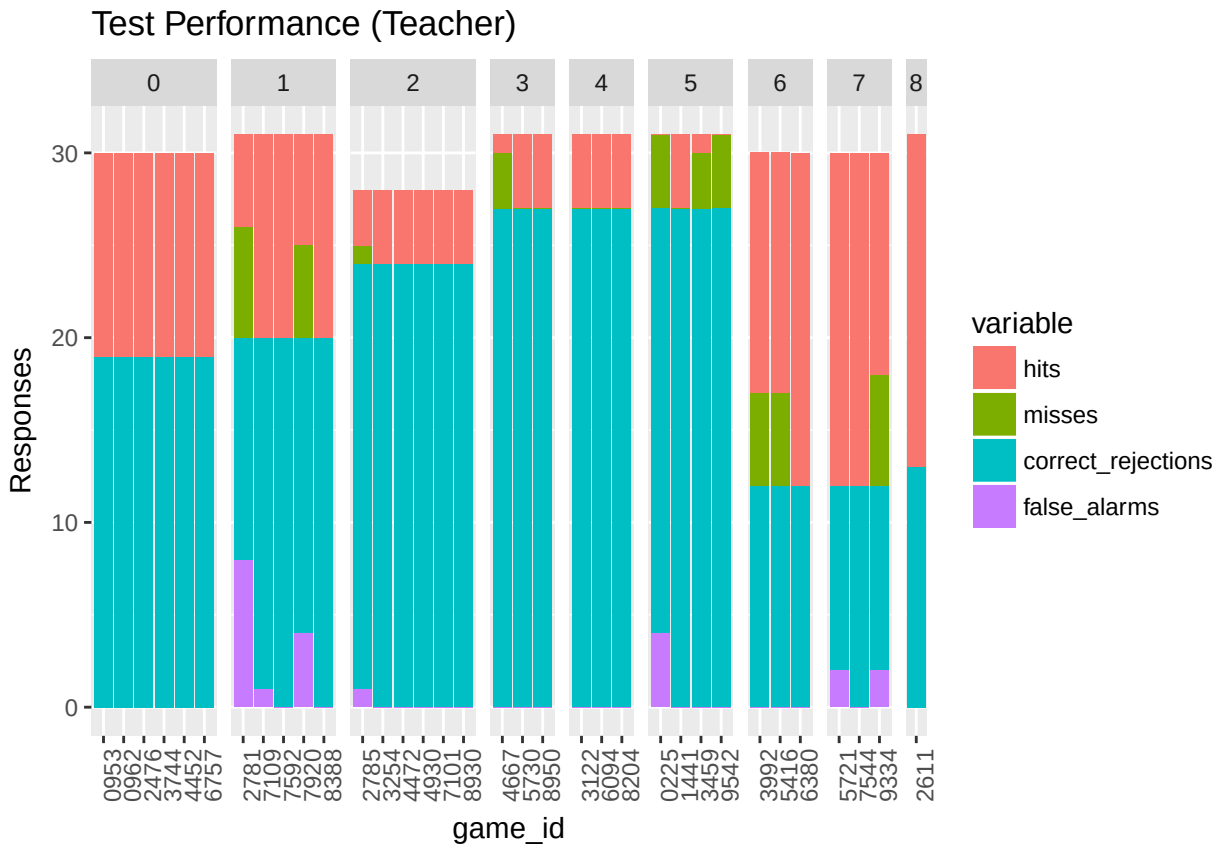
```
num_train = 5
num_test = 31

train_perf <- train_stats %>%
  select(game_id, rule_idx, hits, misses, correct_rejections, false_alarms)

test_perf_teacher <- test_stats %>%
  filter(role == "explorer") %>%
  select(game_id, rule_idx, hits, misses, correct_rejections, false_alarms)

test_perf_listener <- test_stats %>%
  filter(role == "student") %>%
  select(game_id, rule_idx, hits, misses, correct_rejections, false_alarms)
```

```
temp <- melt(train_perf, id.var= c("game_id", 'rule_idx'))
temp$variable <- factor(temp$variable, levels = c("hits", "misses", "correct_rejections", "false_alarms
training_perf_plot <- ggplot(temp, aes(x=game_id, y=value, fill=variable)) +
  geom_bar(stat = "identity") +
  facet_grid(.~rule_idx, scales="free_x", space="free") +
  ylab("Responses") +
```

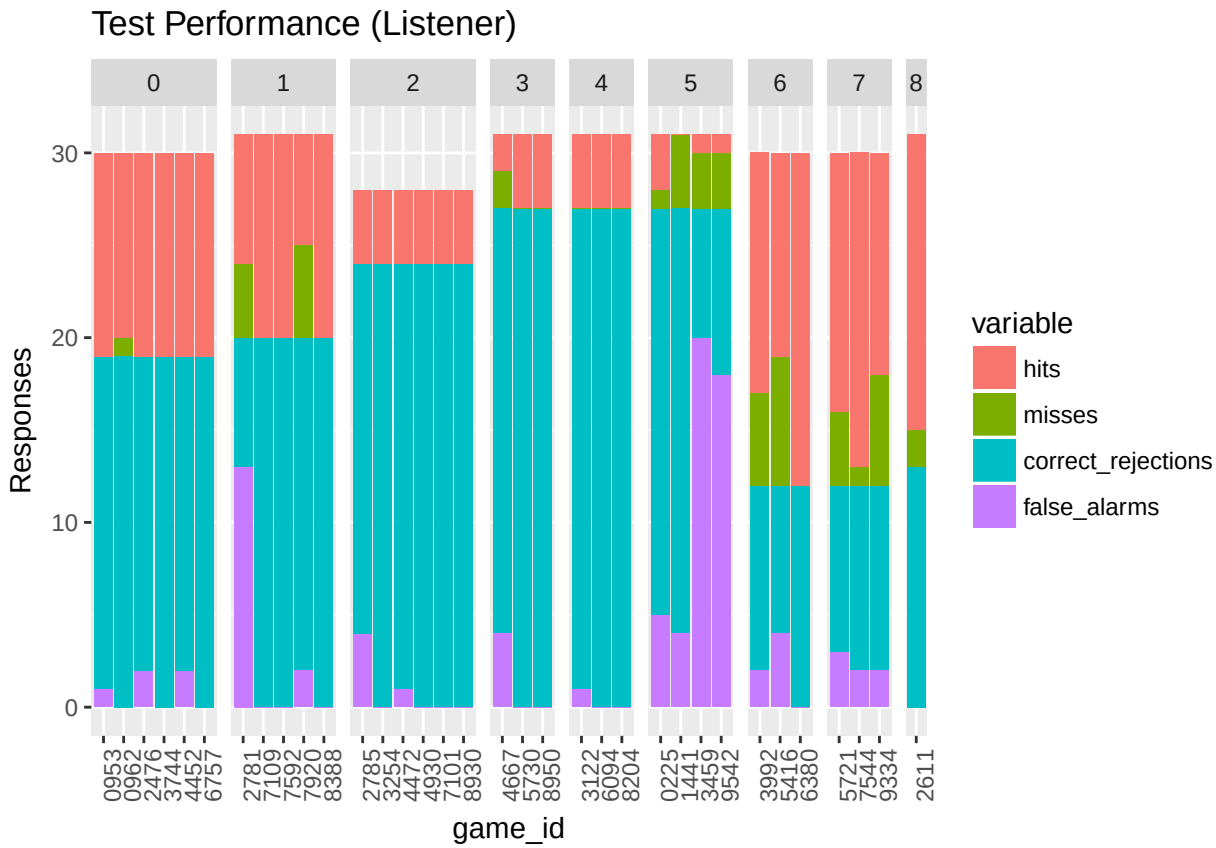
```
  theme(axis.text.x=element_text(angle=90)) +
  scale_x_discrete(label=abbreviate) +
  labs(title = "Train Performance (Teacher)")
plot(training_perf_plot)
```

Note that the bars aren't always split evenly across games with the same rule – except rule 7. I have to investigate why that's the case, as this shouldn't happen. We don't see this below for the test data.

```
temp <- melt(test_perf_teacher, id.var= c("game_id", 'rule_idx'))
temp$variable <- factor(temp$variable, levels = c("hits", "misses", "correct_rejections", "false_alarms
test_perf_teacher_plot <- ggplot(temp, aes(x=game_id, y=value, fill=variable)) +
  geom_bar(stat = "identity") +
  facet_grid(.~rule_idx, scales="free_x", space="free") +
  ylab("Responses") +
  theme(axis.text.x=element_text(angle=90)) +
  scale_x_discrete(label=abbreviate) +
  labs(title = "Test Performance (Teacher)")
plot(test_perf_teacher_plot)
```




```
temp <- melt(test_perf_listener, id.var= c("game_id", 'rule_idx'))
temp$variable <- factor(temp$variable, levels = c("hits", "misses", "correct_rejections", "false_alarms
test_perf_listener_plot <- ggplot(temp, aes(x=game_id, y=value, fill=variable)) +
  geom_bar(stat = "identity") +
  facet_grid(.~rule_idx, scales="free_x", space="free") +
  ylab("Responses") +
  theme(axis.text.x=element_text(angle=90)) +
  scale_x_discrete(label=abbreviate) +
  labs(title = "Test Performance (Listener)")
plot(test_perf_listener_plot)
```



Performance for the listeners generally tracked the performance of the teachers on the test set. There are few exemplars that stand out though. Game 1264 in rule 3 had no dialogue, as discussed earlier. Games 3459 and 9452 have uninformative or poorly interpreted dialogue, as discussed earlier.

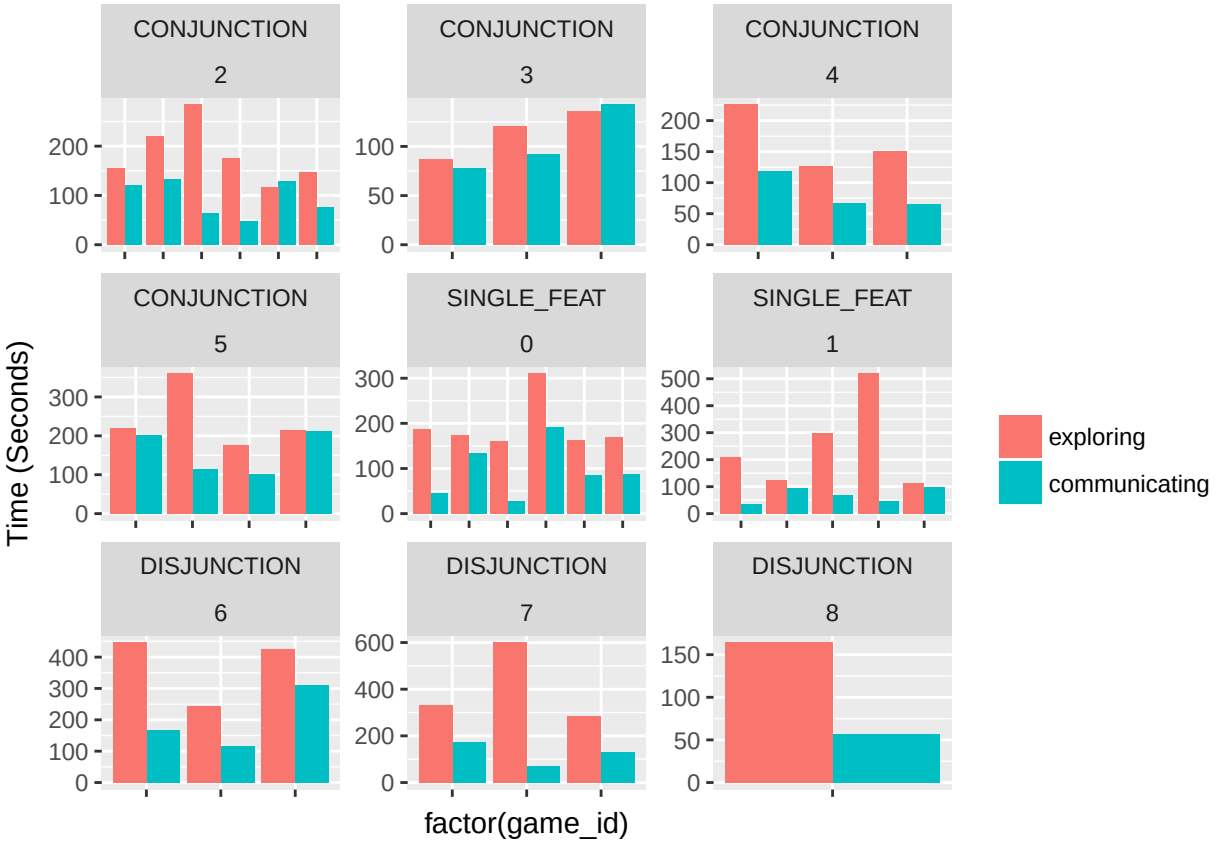
Analysis: Time Spent Learning Versus Communicating

There is a general hypothesis that communication allows one to learn with reasonable accuracy, with far less time expended. We've seen above that the teachers and listeners have comparable performance to one another, even though the the listener has no training samples. Here, we examine the the latter portion of the hypothesis by looking at time spent on learning versus communicating.

First, we compare without normalizing for time penalties during training.

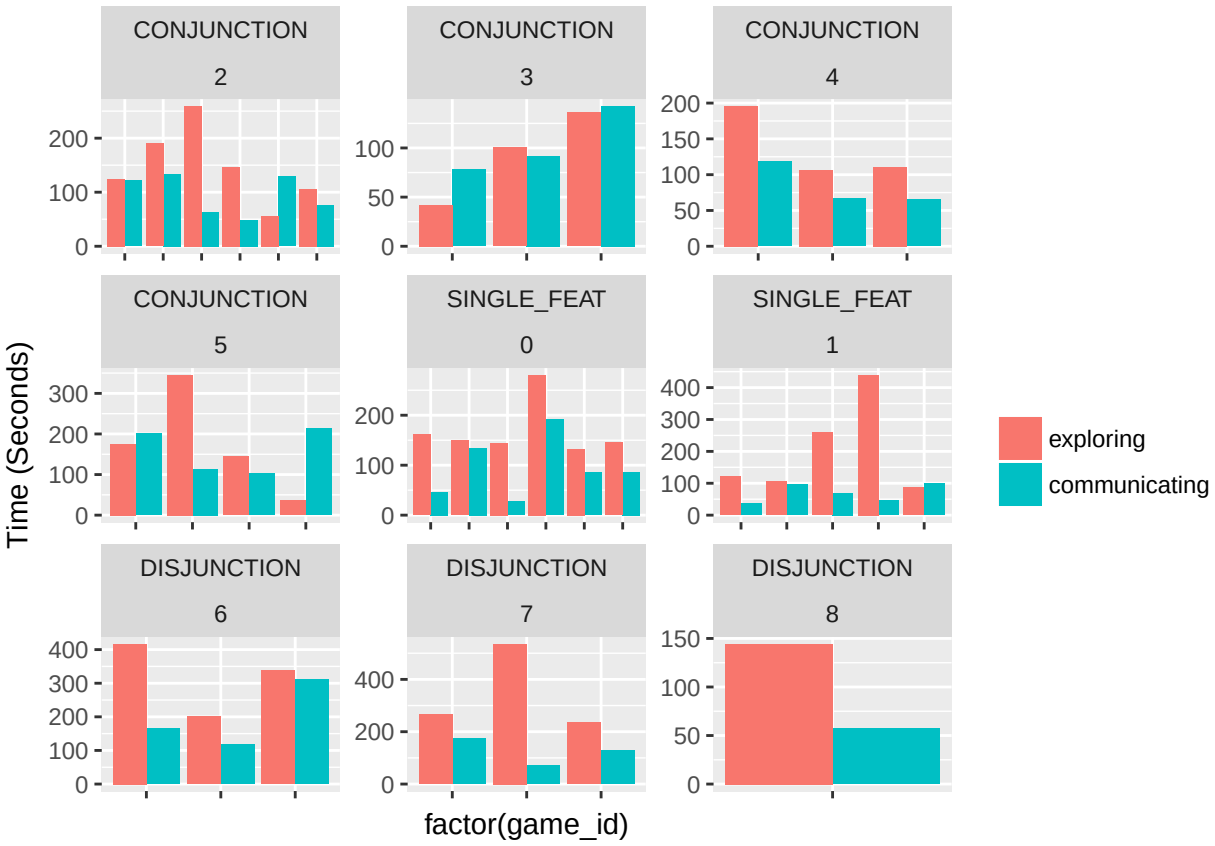
```
train_trials %>%
  select(rule_type, rule_idx, game_id, trial_num, time_in_seconds) %>%
  group_by(rule_type, rule_idx, game_id) %>%
  summarize(m = sum(time_in_seconds)) %>%
  mutate(phase='exploring') %>%
  bind_rows(
    chat_logs %>%
      select(rule_type, rule_idx, game_id, reactionTime) %>%
      group_by(rule_type, rule_idx, game_id) %>%
      summarize(m = sum(reactionTime)) %>%
      mutate(phase='communicating')
  ) %>%
  ggplot(., aes(x = factor(game_id), y = m, fill=factor(phase, levels=c('exploring', 'communicating'))))
  geom_col(position = position_dodge())+
```

```
  facet_wrap(~rule_type + rule_idx, scales = 'free')+
  ylab('Time (Seconds)') +
  theme(axis.text.x = element_blank()) +
  scale_fill_discrete("")
```



Now, we normalize for time penalties during training.

```
time_penalty = 5
train_trials %>%
  select(rule_type, rule_idx, game_id, trial_num, time_in_seconds, is_correct) %>%
  group_by(rule_type, rule_idx, game_id) %>%
  summarize(m = sum(time_in_seconds) - sum(is_correct == 'False') * time_penalty) %>%
  mutate(phase='exploring') %>%
  bind_rows(
    chat_logs %>%
      select(rule_type, rule_idx, game_id, reactionTime) %>%
      group_by(rule_type, rule_idx, game_id) %>%
      summarize(m = sum(reactionTime)) %>%
      mutate(phase='communicating')
  ) %>%
  ggplot(., aes(x = factor(game_id), y = m, fill=factor(phase, levels=c('exploring', 'communicating'))))
  geom_col(position = position_dodge())+
  facet_wrap(~rule_type + rule_idx, scales = 'free')+
  ylab('Time (Seconds)') +
  theme(axis.text.x = element_blank()) +
  scale_fill_discrete("")
```



Analysis: Rational Rules (List Pooled)

```
game_summary <- read.csv('../../mturk/mp-game-3/experiment_1/results-cleaned/game_summary.csv')

rational_rules <- function(r_idx, game_summary, incremental, num_train, max_trial_num) {
  # Filter accordin to list idx
  temp <- filter(game_summary, rule_idx == r_idx)
  base_dir_1 = '../../mturk/mp-game-3/experiment_1/results-cleaned'
  base_dir_2 = '../../experiments/mp-game-3/js'

  # Collect necessary file paths
  human_predictives_fp <- file.path(base_dir_1, 'predictives_pooled_list', paste(r_idx, '.csv', sep=''))
  training_stim_fp <- file.path(base_dir_2, basename(as.character(temp[1,"training_data_fn"])))
  test_stim_fp <- file.path(base_dir_2, basename(as.character(temp[1,"test_data_fn"])))

  train_human_responses <- purrr::map_chr(temp$game_id, function(game_id) {
    file.path(base_dir_1, 'train_trials', paste(game_id, '_explorer.csv', sep=''))
  })

  test_human_responses <-purrr::map_chr(temp$game_id, function(game_id) {
    file.path(base_dir_1, 'test_trials', paste(game_id, '_explorer.csv', sep=''))
  })
```

```
  # Construct datframe to contain variables (passed as input to webppl script)
  df = data.frame(human_predictives_fp, training_stim_fp, test_stim_fp)
  df$train_human_responses = list(train_human_responses)
  df$test_human_responses = list(test_human_responses)

  df$incremental = incremental
  df$num_train = num_train

  result <- as.data.frame(
    webppl(
      program_file = "./rational_rules.wppl",
      packages=c("webppl-json", "webppl-csv"),
      data=df,
      data_var="params"
    )
  ) %>%
  rename(critter = V1) %>%
  rename(rational_rules= V2) %>%
  rename(human = V3) %>%
  mutate(rule_idx = r_idx) %>%
  mutate(trial_num = (1:81))
}

plot_results <- function(results) {
  results$rational_rules <- as.numeric(as.character(results$rational_rules))
  results$human <- as.numeric(as.character(results$human))
  ggplot(results, aes(x=rational_rules, y=human)) +
    geom_point() +
    scale_x_continuous(limits = c(0,1), breaks = seq(0,1, .1)) +
    facet_wrap(~rule_idx, scales = 'free')
}
```