

Experiment

Mohammed Al Owayyed & Adarsh Denga

2025-04-11

Contents

1	Introduction	1
2	Paired-Samples T-Test	2
2.1	Frequentist Analysis	2
2.2	Means and SDS	3
2.3	Visualizing Construct Comparisons	4
2.4	ROPE	6
2.5	Interpretation	6
2.6	Visualise Posterior Distributions	7
3	Preference Test	9
3.1	Visualising preference	10
4	Thematic analysis	11
4.1	Cohen's Kappa	11
4.2	Thematic analysis results	12

1 Introduction

This document gives an overview of the analysis of the measures used for evaluating the integration of Large Language Models into a rule-based chatbot counsellor training system for child helplines. The document includes an analysis of:

- Paired t-tests (frequentist and Bayesian) for five key constructs.
- Cohen's Kappa for coder agreement on thematic analysis.
- Binomial and Bayesian tests for system preference.
- Distribution plots and interpretative visuals.

We need the following files for our analysis:

- `data_raw_llm.csv`: The raw experiment scores for our constructs for each participant from using the LLM-based system.
- `data_raw_rbs.csv`: The raw experiment scores for our constructs for each participant from using the rule-based system.

The two above files have the following columns: `PROLIFIC_PID` (an anonymised participant ID), and `Q1-Q34` (participant scores for each question in our questionnaire). We use the script in `construct_split.py` to transform the data into the data in the two next files:

- `constructs_averaged_llm.csv`: The average scores for five constructs (Overall Experience, Human-Like Behaviour, Natural Behaviour, Engagement, Attitude) for each participant from using the LLM-based system.

- `constructs_averaged_rbs.csv`: The average scores for five constructs (Overall Experience, Human-Like Behaviour, Natural Behaviour, Engagement, Attitude) for each participant from using the rule-based system.

The two above files have the following columns: `PROLIFIC_PID` (an anonymised participant ID), `OVERALL` (average score for overall experience construct), `BELIEVABILITY1` (averaged score for human-like behaviour construct), `BELIEVABILITY2` (averaged score for natural behaviour construct), `ENGAGEMENT` (averaged score for engagement construct), and `ATTITUDE` (overall score for attitude construct).

- `qual.csv`: The raw thematic feedback from our participants for the LLM and RBS systems respectively.

The above file has the following columns: `PROLIFIC_PID` (an anonymised participant ID), `CONDITION_ORDER` (a flag which determines which order the participants interacted with the system), `Coder1` (assigned themes), `Coder2` (assignes themes), `Route` (which one does the quote belong to, i.e., LLM or RBS), `FinalCoding` (final code assigned to the quote)

```
# Load necessary libraries
library(dplyr)           # Data manipulation
library(tidyr)           # Data reshaping
library(ggplot2)         # Plots
library(patchwork)       # Plot composition
library(ggdists)         # Visualizing distributions
library(e1071)           # For skewness
library(BayesFactor)     # Bayesian t-tests and proportion tests
library(coda)            # For HDI computation
library(ggrridges)       # For additional plot types
library(purrr)           # List handling
library(irr)             # Cohen's Kappa and agreement measures
library(tidyverse)

# Commented-out libraries that were not used directly:
# library(loo)
# library(brms)
# library(lmerTest)
# library(bayestestR)
# library(bayesplot)
# library(rstan)
# library(see)
# library(tidyverse)

# Ensure reproducibility
set.seed(1234)
```

2 Paired-Samples T-Test

2.1 Frequentist Analysis

We begin by testing if the LLM-based and rule-based systems significantly differ across constructs using paired t-tests.

```
# Read data files for the LLM and rule-based conditions
df1 <- read.csv("constructs_averaged_llm.csv")
df2 <- read.csv("constructs_averaged_rbs.csv")

# Combine data from both files based on participant ID
merged_df <- inner_join(df1, df2, by = "PROLIFIC_PID", suffix = c("_1", "_2"))
```

```

# Perform T-Tests for the five constructs
tests <- list(
  overall = t.test(merged_df$OVERALL_1, merged_df$OVERALL_2, paired = TRUE),
  #AllBeliev = t.test(merged_df$AllBeliev_1, merged_df$AllBeliev_2, paired = TRUE),
  believability1 = t.test(merged_df$BELIEVABILITY1_1, merged_df$BELIEVABILITY1_2, paired = TRUE),
  believability2 = t.test(merged_df$BELIEVABILITY2_1, merged_df$BELIEVABILITY2_2, paired = TRUE),
  engagement = t.test(merged_df$ENGAGEMENT_1, merged_df$ENGAGEMENT_2, paired = TRUE),
  attitude = t.test(merged_df$ATTITUDE_1, merged_df$ATTITUDE_2, paired = TRUE)
)

# Gather relevant results
results_df <- data.frame(
  Measure = names(tests),
  t_statistic = sapply(tests, function(t) round(t$statistic, 3)),
  p_value = sapply(tests, function(t) formatC(t$p.value)),
  mean_difference = sapply(tests, function(t) round(t$estimate, 3)),
  conf_low = sapply(tests, function(t) round(t$conf.int[1], 3)),
  conf_high = sapply(tests, function(t) round(t$conf.int[2], 3))
)

# Print results
print(results_df, row.names = FALSE)

```

##	Measure	t_statistic	p_value	mean_difference	conf_low	conf_high
##	overall	2.567	0.01457	0.478	0.100	0.855
##	believability1	2.098	0.04294	0.438	0.015	0.861
##	believability2	1.381	0.1757	0.333	-0.156	0.823
##	engagement	1.069	0.2921	0.171	-0.154	0.496
##	attitude	2.456	0.01899	0.739	0.129	1.349

2.2 Means and SDS

```

construct_means_sds_clean <- data.frame(
  Measure = c("Human-Like Behaviour", "Natural Behaviour", "Engagement", "Attitude", "Overall Experience"),
  LLM_Mean = round(c(mean(merged_df$BELIEVABILITY1_1, na.rm = TRUE),
    mean(merged_df$BELIEVABILITY2_1, na.rm = TRUE),
    mean(merged_df$ENGAGEMENT_1, na.rm = TRUE),
    mean(merged_df$ATTITUDE_1, na.rm = TRUE),
    mean(merged_df$OVERALL_1, na.rm = TRUE)), 2),
  LLM_SD = round(c(sd(merged_df$BELIEVABILITY1_1, na.rm = TRUE),
    sd(merged_df$BELIEVABILITY2_1, na.rm = TRUE),
    sd(merged_df$ENGAGEMENT_1, na.rm = TRUE),
    sd(merged_df$ATTITUDE_1, na.rm = TRUE),
    sd(merged_df$OVERALL_1, na.rm = TRUE)), 2),
  RBS_Mean = round(c(mean(merged_df$BELIEVABILITY1_2, na.rm = TRUE),
    mean(merged_df$BELIEVABILITY2_2, na.rm = TRUE),
    mean(merged_df$ENGAGEMENT_2, na.rm = TRUE),
    mean(merged_df$ATTITUDE_2, na.rm = TRUE),
    mean(merged_df$OVERALL_2, na.rm = TRUE)), 2),
  RBS_SD = round(c(sd(merged_df$BELIEVABILITY1_2, na.rm = TRUE),
    sd(merged_df$BELIEVABILITY2_2, na.rm = TRUE),
    sd(merged_df$ENGAGEMENT_2, na.rm = TRUE),
    sd(merged_df$ATTITUDE_2, na.rm = TRUE),

```

```

sd(merged_df$OVERALL_2, na.rm = TRUE)), 2)
)

print(construct_means_sds_clean)

```

```

##           Measure LLM_Mean LLM_SD RBS_Mean RBS_SD
## 1 Human-Like Behaviour    0.79   1.62    0.35   1.57
## 2   Natural Behaviour    0.14   1.55   -0.19   1.57
## 3           Engagement    2.01   1.05    1.84   0.77
## 4             Attitude    0.86   1.53    0.13   1.76
## 5   Overall Experience    0.46   1.25   -0.02   1.28

```

2.3 Visualizing Construct Comparisons

The following boxplot shows how participant ratings differed between LLM and rule-based systems.

```

# Prepare the long-format data for all constructs
df_all <- merged_df %>%
  select(PROLIFIC_PID,
         OVERALL_1, OVERALL_2,
         BELIEVABILITY1_1, BELIEVABILITY1_2,
         BELIEVABILITY2_1, BELIEVABILITY2_2,
         ENGAGEMENT_1, ENGAGEMENT_2,
         ATTITUDE_1, ATTITUDE_2) %>%
  pivot_longer(-PROLIFIC_PID, names_to = "measure", values_to = "value") %>%
  mutate(
    Construct = case_when(
      grepl("OVERALL", measure) ~ "Overall Experience",
      grepl("BELIEVABILITY1", measure) ~ "Human-like\nBehaviour (H1)",
      grepl("BELIEVABILITY2", measure) ~ "Natural Behaviour (H1)",
      grepl("ENGAGEMENT", measure) ~ "Engagement (H2)",
      grepl("ATTITUDE", measure) ~ "Attitude (H3)"
    ),
    Condition = ifelse(grepl("_1", measure), "LLM", "Rule-based"),
    Condition = factor(Condition, levels = c("Rule-based", "LLM")), # REVERSED ORDER
    Construct = factor(Construct, levels = c(
      "Human-like\nBehaviour (H1)", "Natural Behaviour (H1)",
      "Engagement (H2)", "Attitude (H3)",
      "Overall Experience"
    ))
  )

# Plot with fixed y-axis and slightly wider boxes
p_all_facet <- ggplot(df_all, aes(x = Condition, y = value, fill = Condition)) +
  geom_boxplot(width = 0.5, outlier.size = 0.8) + # slightly wider
  facet_wrap(~Construct, scales = "fixed", nrow = 1) +
  scale_y_continuous(limits = c(-3, 3), breaks = seq(-3, 3, by = 1))
) +
  labs(x = NULL, y = NULL, title = "Construct Comparison Across Conditions") +
  theme_minimal(base_size = 10) +
  theme(
    legend.position = "none",
    plot.title = element_text(hjust = 0.5),
    strip.text = element_text(size = 10),

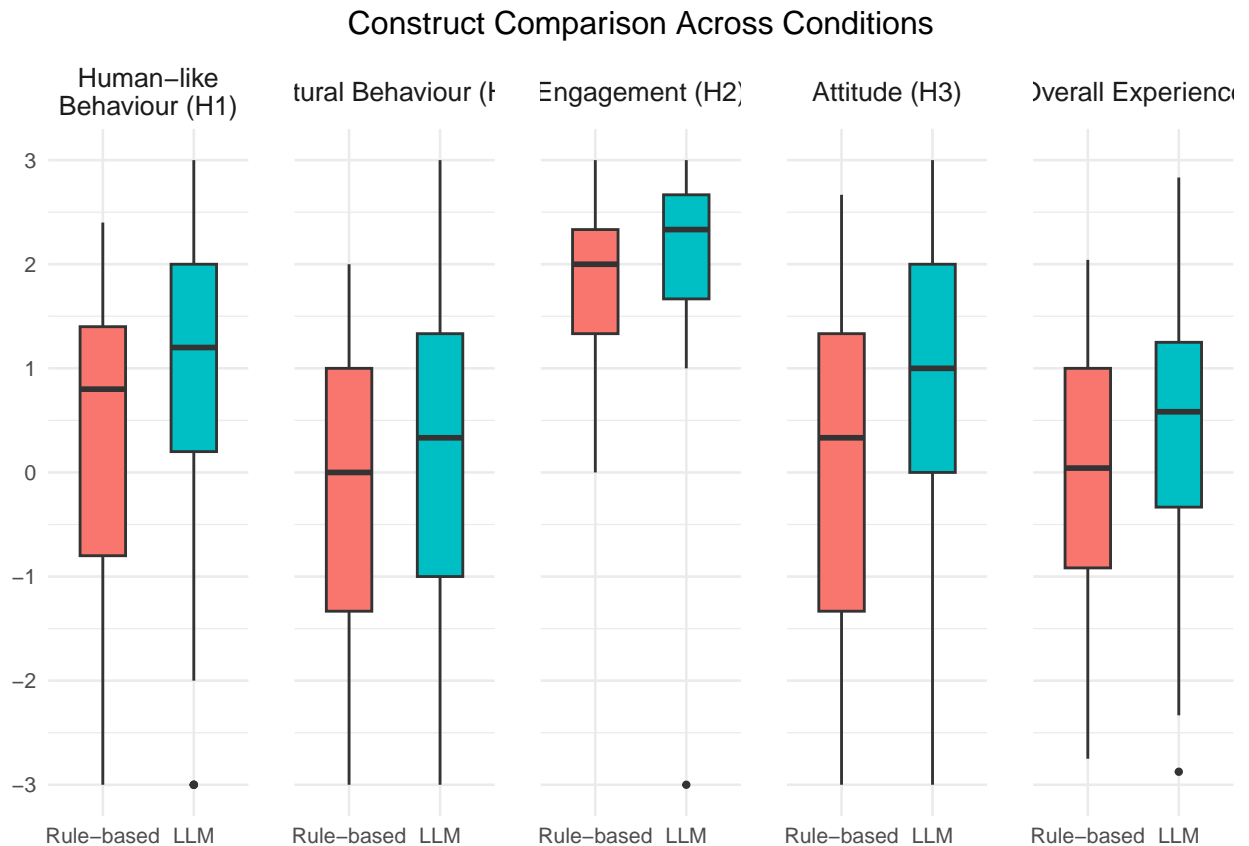
```

```

axis.text.x = element_text(size = 8),
panel.spacing = unit(1.2, "lines")
)

# Print the figure
print(p_all_facet)

```



Bayesian Paired T-Test Now, we fit Bayesian t-tests to compare LLM and rule-based scores

```

# Compute Bayesian t-tests

```

```

tests2 <- list(
  #AllBeliev = ttestBF(x = merged_df$AllBeliev_1, y = merged_df$AllBeliev_2, paired = TRUE), #uncomment
  believability1 = ttestBF(x = merged_df$BELIEVABILITY1_1, y = merged_df$BELIEVABILITY1_2, paired = TRUE),
  believability2 = ttestBF(x = merged_df$BELIEVABILITY2_1, y = merged_df$BELIEVABILITY2_2, paired = TRUE),
  Engagement = ttestBF(x = merged_df$ENGAGEMENT_1, y = merged_df$ENGAGEMENT_2, paired = TRUE),
  Attitude = ttestBF(x = merged_df$ATTITUDE_1, y = merged_df$ATTITUDE_2, paired = TRUE),
  overall = ttestBF(x = merged_df$OVERALL_1, y = merged_df$OVERALL_2, paired = TRUE)
)

```

2.3.1 Extract Bayes Factors, posterior distributions, and HDIs.

```

# Extract Bayes Factors
results_df <- data.frame(
  Measure = names(tests2),
  BayesFactor = sapply(tests2, function(t) extractBF(t)[, "bf"])
)

```

```

# Posterior sampling
set.seed(1234)
posterior_samples <- lapply(tests2, function(t) posterior(t, iterations = 100000))

# Probabilities and HDIs
probability_B_higher <- sapply(posterior_samples, function(samples) mean(samples[, "mu"] > 0))
set.seed(1234)
hdi_intervals <- t(sapply(posterior_samples, function(samples) {
  hdi <- HPDinterval(mcmc(samples[, "mu"]), prob = 0.95)
  c(HDI_Lower = hdi[1], HDI_Upper = hdi[2])
}))

results_df$Probability_B_Higher <- probability_B_higher
results_df$HDI_Lower <- hdi_intervals[, "HDI_Lower"]
results_df$HDI_Upper <- hdi_intervals[, "HDI_Upper"]
results_df$HDI_95 <- paste0("[", round(results_df$HDI_Lower, 3), ", ", round(results_df$HDI_Upper, 3), "]")

```

2.4 ROPE

We compute Region of Practical Equivalence (ROPE) bounds

```

rope_bounds <- data.frame(
  Measure = names(tests2),
  SD_Diff = c(
    sd(merged_df$BELIEVABILITY1_1 - merged_df$BELIEVABILITY1_2, na.rm = TRUE),
    sd(merged_df$BELIEVABILITY2_1 - merged_df$BELIEVABILITY2_2, na.rm = TRUE),
    sd(merged_df$ENGAGEMENT_1 - merged_df$ENGAGEMENT_2, na.rm = TRUE),
    sd(merged_df$ATTITUDE_1 - merged_df$ATTITUDE_2, na.rm = TRUE),
    sd(merged_df$OVERALL_1 - merged_df$OVERALL_2, na.rm = TRUE)
  )
)
rope_bounds$ROPE_Lower <- -0.1 * rope_bounds$SD_Diff
rope_bounds$ROPE_Upper <- 0.1 * rope_bounds$SD_Diff
rope_bounds$ROPE <- paste0("[", round(rope_bounds$ROPE_Lower, 3), ", ", round(rope_bounds$ROPE_Upper, 3), "]")

results_df <- merge(results_df, rope_bounds[, c("Measure", "ROPE_Lower", "ROPE_Upper", "ROPE")], by = "Measure")

```

2.5 Interpretation

We apply a probabilistic interpretation guide.

```

label_map <- c(
  believability1 = "Human-Like Behaviour",
  believability2 = "Natural Behaviour",
  Engagement = "Engagement",
  Attitude = "Attitude",
  overall = "Overall Experience"
)

results_df$Measure <- recode(results_df$Measure, !!!label_map)
results_df$Measure <- factor(results_df$Measure, levels = c("Human-Like Behaviour", "Natural Behaviour", "Engagement", "Attitude", "Overall Experience"))

interpret_probability <- function(p) {

```

```

if (p >= 0.99995 && p < 1) return("Virtually certain for A")
else if (p >= 0.9995) return("Nearing certainty for A")
else if (p >= 0.995) return("Very strong bet on A")
else if (p >= 0.99) return("Strong bet - irresponsible to avoid A")
else if (p >= 0.95) return("Good bet - too good to disregard A")
else if (p >= 0.9) return("A promising but risky bet for A")
else if (p >= 0.75) return("Only a casual bet for A")
else if (p >= 0.5) return("Not worth betting on A")
else if (p >= 0.25) return("Not worth betting on B")
else if (p >= 0.1) return("Only a casual bet for B")
else if (p >= 0.05) return("A promising but risky bet for B")
else if (p >= 0.01) return("Good bet - too good to disregard B")
else if (p >= 0.005) return("Strong bet - irresponsible to avoid B")
else if (p >= 0.0005) return("Very strong bet on B")
else if (p >= 0.00005) return("Nearing certainty for B")
else return("Virtually certain for B")
}
results_df$Interpretation <- sapply(results_df$Probability_B_Higher, interpret_probability)

# Print final table
print(results_df[order(results_df$Measure), ], row.names = FALSE)

##           Measure BayesFactor Probability_B_Higher   HDI_Lower HDI_Upper
## Human-Like Behaviour   1.2464629           0.97525  0.002044738 0.8157133
##   Natural Behaviour    0.4232408           0.90503 -0.160611267 0.7705683
##       Engagement      0.2998277           0.84496 -0.148512532 0.4649976
##         Attitude      2.4383069           0.98824  0.083183947 1.2657205
## Overall Experience     3.0459649           0.99122  0.075176689 0.8071565
##           HDI_95  ROPE_Lower ROPE_Upper           ROPE
## [0.002, 0.816] -0.12691536 0.12691536 [-0.127, 0.127]
## [-0.161, 0.771] -0.14678636 0.14678636 [-0.147, 0.147]
## [-0.149, 0.465] -0.09738499 0.09738499 [-0.097, 0.097]
## [0.083, 1.266] -0.18293768 0.18293768 [-0.183, 0.183]
## [0.075, 0.807] -0.11317140 0.11317140 [-0.113, 0.113]
##           Interpretation
##   Good bet - too good to disregard A
##     A promising but risky bet for A
##       Only a casual bet for A
##   Good bet - too good to disregard A
## Strong bet - irresponsible to avoid A

```

2.6 Visualise Posterior Distributions

```

set.seed(1234)
generate_posterior <- function(mean, lower, upper) {
  sd_est <- (upper - lower) / (2 * 1.96)
  rnorm(5000, mean = mean, sd = sd_est)
}
posterior_samples_long <- results_df %>%
  rowwise() %>%
  mutate(samples = list(generate_posterior((HDI_Lower + HDI_Upper)/2, HDI_Lower, HDI_Upper))) %>%
  unnest(cols = c(samples))

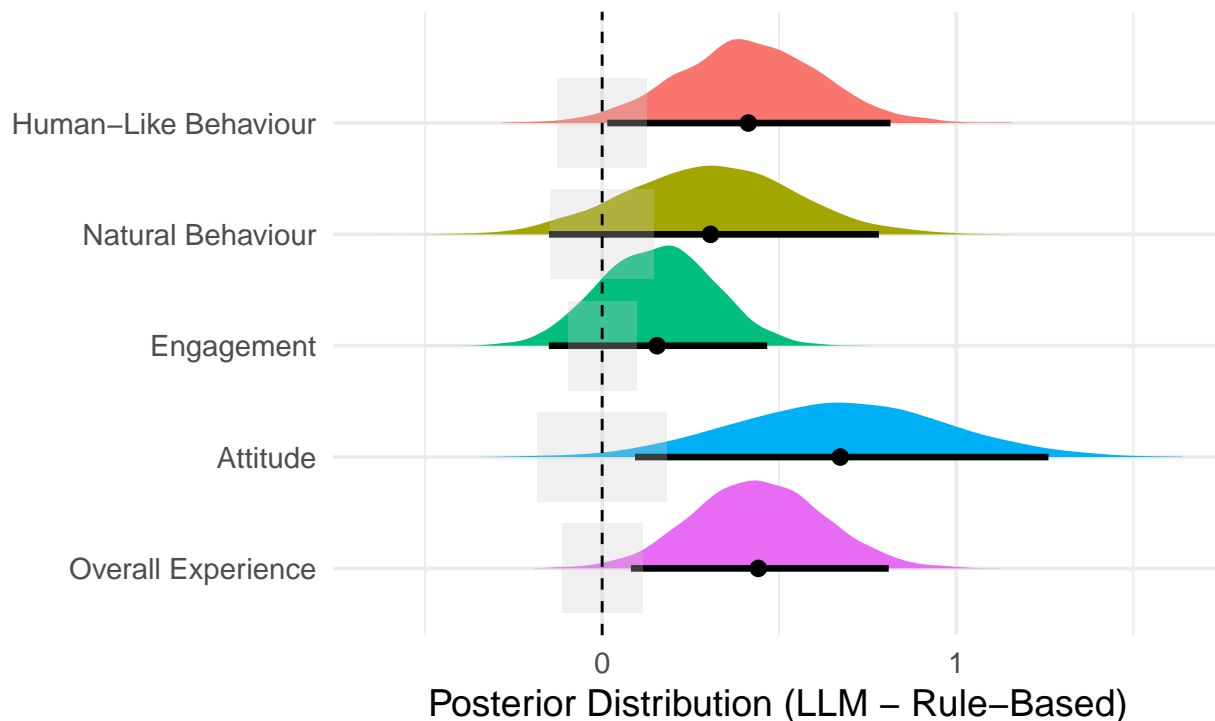
```

```
posterior_samples_long$Measure <- factor(posterior_samples_long$Measure,
  levels = c("Human-Like Behaviour", "Natural Behaviour", "Engagement", "Attitude", "Overall Experience"))

ggplot(posterior_samples_long, aes(x = samples, y = fct_rev(Measure), fill = Measure)) +
  stat_halfeye(.width = 0.95, show.legend = FALSE, point_interval = mean_qi) +
  geom_rect(data = results_df,
    aes(xmin = ROPE_Lower, xmax = ROPE_Upper,
      ymin = as.numeric(fct_rev(Measure)) - 0.4,
      ymax = as.numeric(fct_rev(Measure)) + 0.4),
    fill = "gray80", alpha = 0.3, inherit.aes = FALSE) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  labs(title = "Posterior Distributions by Measure",
    subtitle = "Shaded area = practical equivalence; horizontal bars = 95% HDIs",
    x = "Posterior Distribution (LLM - Rule-Based)", y = NULL) +
  theme_minimal(base_size = 14)
```

Posterior Distributions by Measure

Shaded area = practical equivalence; horizontal bars =



Skewness check

We check for skewness in the engagement measure.

```
combined_engagement <- c(merged_df$ENGAGEMENT_1, merged_df$ENGAGEMENT_2)

skewness(combined_engagement, na.rm = TRUE)
```

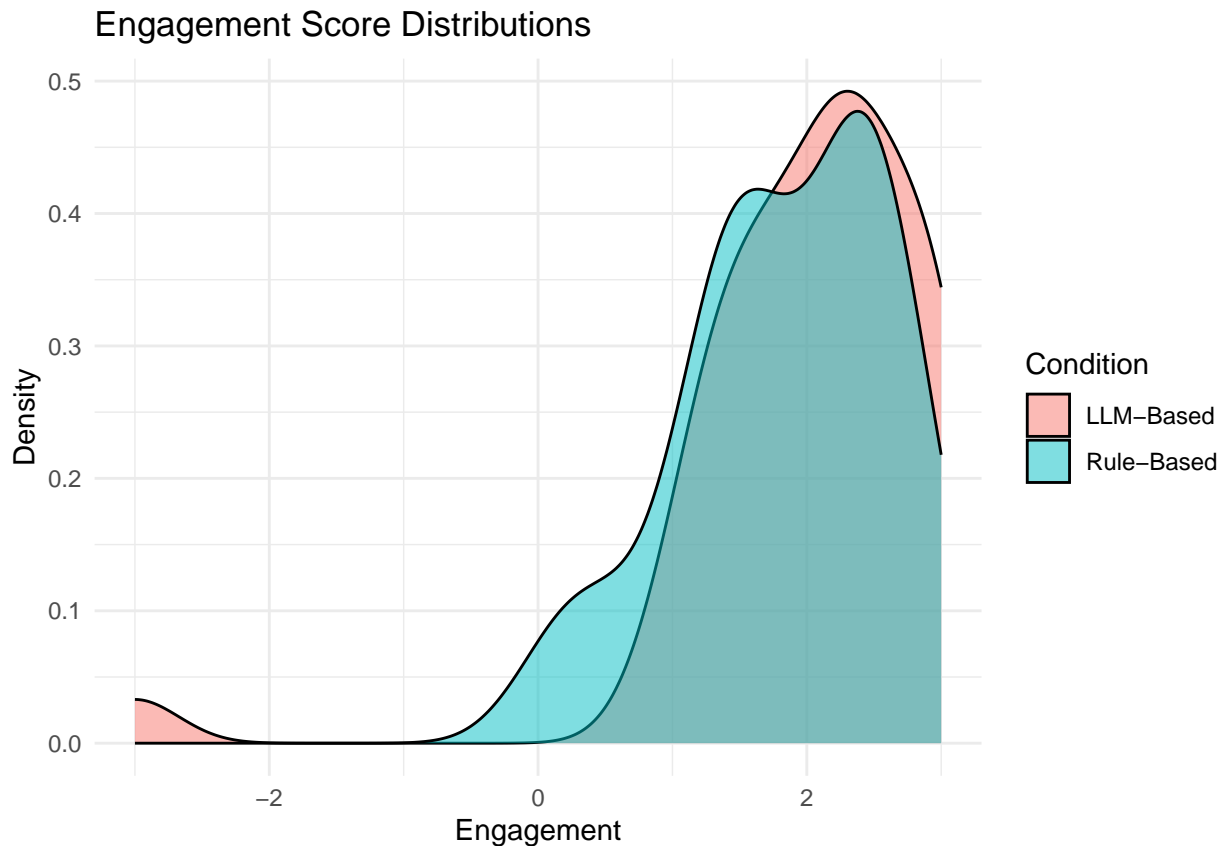
```
## [1] -2.209805
```

```
engagement_df <- data.frame(
  Engagement = c(merged_df$ENGAGEMENT_1, merged_df$ENGAGEMENT_2),
  Condition = rep(c("LLM-Based", "Rule-Based"), each = nrow(merged_df))
```



```
)

ggplot(engagement_df, aes(x = Engagement, fill = Condition)) +
  geom_density(alpha = 0.5) +
  labs(title = "Engagement Score Distributions", x = "Engagement", y = "Density") +
  theme_minimal()
```



The engagement results showed substantial negative skewness (skewness = -2.21), suggesting a ceiling effect that may have limited our ability to detect further improvements in engagement.

3 Preference Test

To test the preference, we conducted a binomial test. First, is the frequentist approach:

```
# Perform binomial test
binom_results <- binom.test(26, 37, p = 0.5, alternative = "two.sided")

# Print results
print(binom_results)

##
## Exact binomial test
##
## data: 26 and 37
## number of successes = 26, number of trials = 37, p-value = 0.02007
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
## 0.5302005 0.8412746
```

```
## sample estimates:
## probability of success
## 0.7027027
```

Second is the Bayesian binomial test.

```
# Compute Bayesian binomial test
binom_results_bay <- proportionBF(26, 37, p = 0.5)

# Print results
print(binom_results_bay)

## Bayes factor analysis
## -----
## [1] Alt., p0=0.5, r=0.5 : 5.395049 ±0%
##
## Against denominator:
## Null, p = 0.5
## ---
## Bayes factor type: BFproportion, logistic

# Extract posterior samples (10,000 iterations)
set.seed(1234)
posterior_samplesBinom <- posterior(binom_results_bay, iterations = 100000)

# Compute probability that p > 0.5
prob_p_greater_0.5 <- mean(posterior_samplesBinom[, "p"] > 0.5)

# Print probability
print(paste("P(p > 0.5) =", round(prob_p_greater_0.5, 5)))

## [1] "P(p > 0.5) = 0.99019"
```

The posterior probability that this preference is above chance is 0.99, which is a strong bet that the LLM-based agent is more favourable.

3.1 Visualising preference

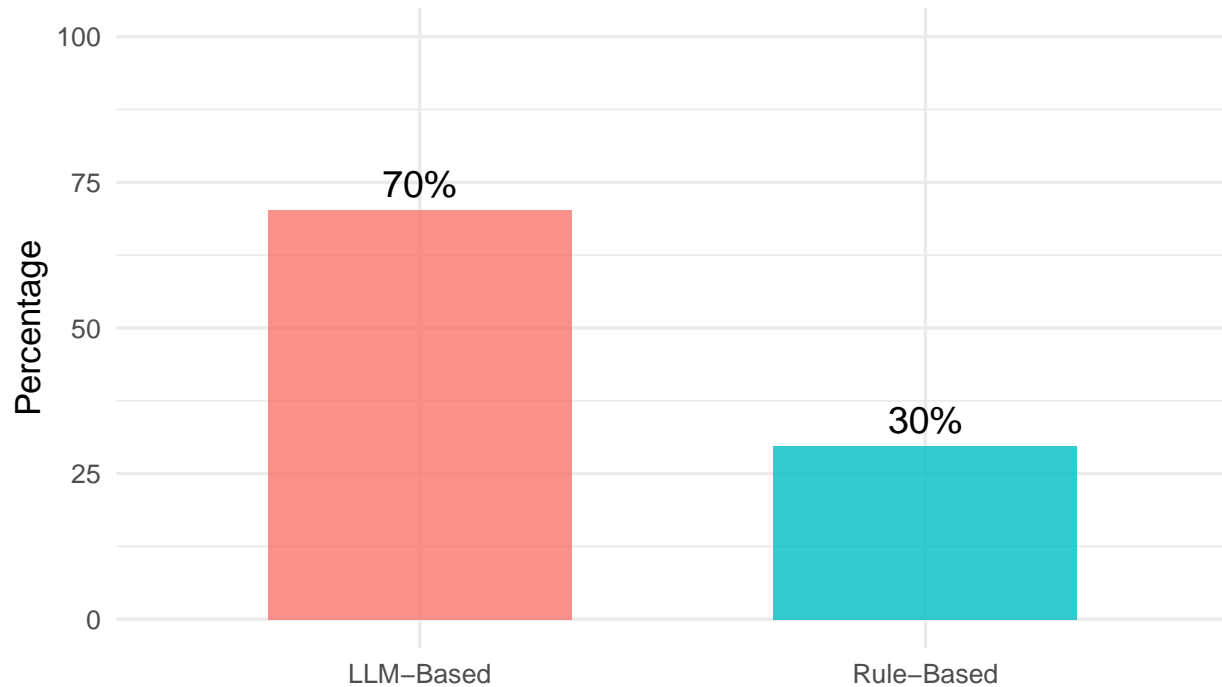
```
# Create preference count data
preference_df <- data.frame(
  System = c("LLM-Based", "Rule-Based"),
  Count = c(26, 11)
) %>%
  mutate(Percentage = Count / sum(Count) * 100)

# Create bar plot
ggplot(preference_df, aes(x = System, y = Percentage, fill = System)) +
  geom_bar(stat = "identity", width = 0.6, alpha = 0.8) +
  geom_text(aes(label = paste0(round(Percentage), "%"),
    vjust = -0.5, size = 5) +
  ylim(0, 100) +
  labs(
    title = "Participant Preference for Virtual Child",
    subtitle = paste0("BF = ", round(binom_results_bay@bayesFactor$bf, 2),
      " | P(p > 0.5) = ", round(prob_p_greater_0.5, 3)),
    x = "", y = "Percentage"
```

```
) +
theme_minimal(base_size = 13) +
theme(legend.position = "none")
```

Participant Preference for Virtual Child

BF = 1.69 | $P(p > 0.5) = 0.99$



4 Thematic analysis

4.1 Cohen's Kappa

```
dfCoding <- read_csv("qual.csv")

# 1. Calculate Cohen's Kappa between Coder1 and Coder2
kappa_result <- kappa2(dfCoding[, c("Coder1", "Coder2")])
print("Cohen's Kappa:")

## [1] "Cohen's Kappa:"
print(kappa_result)

## Cohen's Kappa for 2 Raters (Weights: unweighted)
##
## Subjects = 74
## Raters = 2
## Kappa = 0.316
##
## z = 8.12
## p-value = 4.44e-16
```

4.2 Thematic analysis results

```
# 2. Count quotes per theme in FinalCoding column
quotes_per_theme <- dfCoding %>%
  count(FinalCoding, name = "QuoteCount")
```

```
print(quotes_per_theme)
```

```
## # A tibble: 9 x 2
##   FinalCoding      QuoteCount
##   <chr>          <int>
## 1 Abrupt Ending      10
## 2 Boring Task        2
## 3 Depth of Conversation  4
## 4 Emotional Engagement  7
## 5 Human-Like Responses  6
## 6 Positive Experience  17
## 7 Scripted Responses   1
## 8 Slow Responses     16
## 9 Unnatural Responses  11
```

```
# Create comparison table: Themes per Route
comparison_table <- dfCoding %>%
  group_by(FinalCoding, Route) %>%
  summarise(Count = n(), .groups = "drop") %>%
  pivot_wider(
    names_from = Route,
    values_from = Count,
    values_fill = 0 # Fill missing combinations with 0
  )
```

```
# View the table
print(comparison_table)
```

```
## # A tibble: 9 x 3
##   FinalCoding      LLM    RBS
##   <chr>          <int> <int>
## 1 Abrupt Ending      3      7
## 2 Boring Task        1      1
## 3 Depth of Conversation  4      0
## 4 Emotional Engagement  6      1
## 5 Human-Like Responses  4      2
## 6 Positive Experience  9      8
## 7 Scripted Responses   0      1
## 8 Slow Responses     6     10
## 9 Unnatural Responses  4      7
```

```
# Group and count themes per Route
themes_per_route <- dfCoding %>%
  group_by(Route, FinalCoding) %>%
  summarise(Count = n(), .groups = "drop")
```

```
# Plot grouped bar chart
ggplot(themes_per_route, aes(x = FinalCoding, y = Count, fill = Route)) +
  geom_bar(stat = "identity", position = "dodge") +
```

```
labs(
  title = "Themes per Route",
  x = "Final Coding Theme",
  y = "Number of Quotes"
) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

