LLM-Extraction Project

Data Preparation

Johannes Schwenke

2025-11-19

Table of contents

1 Setup

```
library(tidyverse)
library(marginaleffects)
library(REDCapR)
library(lme4)
library(brms)
library(tinytable)
library(irr)
library(glue)
library(scales)
library(gt)
library(patchwork)
library(writexl)
library(here)
library(arrow)
```

2 Download Data from REDCap

```
redownload <- TRUE

if(redownload){
  data <- redcap_read(
    redcap_uri = Sys.getenv("redcap_fxdb_url"),
    token = Sys.getenv("llm_radio_api"),
    raw_or_label = "label"
  )$data

# saveRDS(data, "./data/oncology/redcap_export_complete.rds")
} else {
  data <- readRDS("./data/oncology/redcap-2025-10-09-labels.rds")$data
}

logs <- redcap_log_read(
  redcap_uri = Sys.getenv("redcap_fxdb_url"),
  token = Sys.getenv("llm_radio_api"),
```

```
  log_begin_date = as.Date("2025-05-01"),
  log_end_date = Sys.Date(),
  record = NULL,
  user = NULL,
  http_response_encoding = "UTF-8",
  locale = readr::default_locale(),
  verbose = TRUE,
  config_options = NULL,
  handle_httr = NULL
)
```

3 Format Data for Analysis

```
long_data <- data |>
  group_by(redcap_event_name) |>
  pivot_longer(
    cols = c(
      primary_tumor,
      bone_metastasis,
      cns_metastasis,
      meningeal_metastasis,
      lung_metastasis,
      pleural_metastasis,
      adrenal_metastasis,
      kidney_metastasis,
      liver_metastasis,
      spleen_metastasis,
      pancreas_metastasis,
      ovarian_metastasis,
      peritoneal_metastasis,
      lymph_node_metastasis,
      soft_tissue_metastasis,
      other_organ_metastasis,
      no_tumor,
      response_to_trt,
      response_to_trt_pet
    ),
    names_to = "item",
    values_to = "value"
  ) |>
  select(-c(oncology_radiology_extraction_complete, justification, prompt,
sample_complete)) |>
  ungroup()

ground_truth_long <- long_data |>
    filter(redcap_event_name == "Ground Truth") |>
    select(ier_bk, item, truth = value) |>
    group_by(ier_bk) |>
    mutate(
      truth = case_when(
        # When response to trt not applicable, then set no_tumor to NA
        item == "no_tumor" &
          any(truth[item == "response_to_trt" | item == "response_to_trt_pet"] == "Not
applicable") ~ "Not applicable",
```

```
        TRUE ~ truth
      )
    )

analysis_data <- long_data |>
    filter(redcap_event_name != "Ground Truth") |>
    group_by(ier_bk, redcap_event_name) |>
    mutate(
      value = case_when(
        # I only want to change the value where item == "no_tumor"
        item == "no_tumor" &
          any(value[item == "response_to_trt" | item == "response_to_trt_pet"] == "Not
applicable") ~ "Not applicable",
        TRUE ~ value
      )
    ) |>
    ungroup() |>
    left_join(ground_truth_long, by = c("ier_bk", "item")) |>
    # where NA for ground truth and value, then question was not applicable -> remove
    filter(!is.na(value) & !is.na(truth)) |>
    mutate(
        correct = if_else(value == truth, 1, 0),
        task  = case_when(
          str_detect(item, "metastasis") ~ "Metastasis",
          str_detect(item, "primary_tumor") ~ "Diagnosis",
          str_detect(item, "response_to_trt") ~ "Response to Treatment",
          str_detect(item, "no_tumor") ~ "Radiologically Tumor Free",
          TRUE ~ NA
        ),
        report_length = nchar(imaging_report)
    )

analysis_data_human <- analysis_data |>
  filter(!str_detect(redcap_event_name, "LLM"))

if(any(is.na(analysis_data_human$value))) {
    message("There are NA values in the long_data, which may affect analysis.")
}

all_ids <- na.omit(unique(analysis_data_human$extractor_id))
md <- c("DH", "AIM", "BT", "AMS")
stud <- setdiff(all_ids, md)

analysis_data_human <- analysis_data_human |>
  mutate(
    human_group = case_when(
      extractor_id %in% md ~ "MD",
      extractor_id %in% stud ~ "Stud",
      TRUE ~ "Unknown"
    ),
    body_region_grouped = fct_lump_n(body_region, n = 6, other_level = "Other"),
    #add the length of each report
  )
```

```
write_parquet(ground_truth_long, sink = here( "data", "oncology",
"ground_truth_long.parquet"))
write_parquet(analysis_data, sink = here("data","oncology","analysis_data.parquet"))
write_parquet(analysis_data_human, sink =
here("data","oncology","analysis_data_human.parquet"))
```