Using Keeper with Large Language Models

Martijn Schuemie

2024-06-18

Contents

1 Introduction

This vignette describes how one could use Keeper to generate patient summaries, and have them be reviewed by a large-language model (LLM).

2 Running Keeper

As an example, we'll run Keeper on Eunomia. Eunomia is an OHDSI package that contains a tiny simulated dataset in the Common Data Model (CDM). It is mostly focused on NSAIDs and gastrointestinal (GI) bleeding, so we'll use GI bleed as our example outcome.

2.1 Setting up Eunomia

First we must download and install Eunomia:

```
install.packages("Eunomia")
```

Next, we can obtain connection details to the Eunomia database. (Note: this will download the database from the internet):

```
library(Eunomia)
connectionDetails <- getEunomiaConnectionDetails()
```

We can have the default set of cohorts generated in Eunomia:

```
createCohorts(connectionDetails)
```

```
## Cohorts created in table main.cohort
```

```
##   cohortId       name
## 1        1  Celecoxib    A simplified cohort definition for new users of celecoxib, designed specifi
```

```
## 2        2 Diclofenac    A simplified cohort definition for new users ofdiclofenac, designed specifi
## 3        3    GiBleed A simplified cohort definition for gastrointestinal bleeding, designed specifi
## 4        4     NSAIDs       A simplified cohort definition for new users of NSAIDs, designed specifi
```

2.2 Run Keeper

Next, we run Keeper. We meticulously select concepts for each Keeper category:

```
keeper <- createKeeper(
    connectionDetails = connectionDetails,
    databaseId = "Synpuf",
    cdmDatabaseSchema = "main",
    cohortDatabaseSchema = "main",
    cohortTable = "cohort",
    cohortDefinitionId = 3,
    cohortName = "GI Bleed",
    sampleSize = 100,
    assignNewId = TRUE,
    useAncestor = TRUE,
    doi = c(4202064, 192671, 2108878, 2108900, 2002608),
    symptoms = c(4103703, 443530, 4245614, 28779),
    comorbidities = c(81893, 201606, 313217, 318800, 432585, 4027663, 4180790, 4212540,
                      40481531, 42535737, 46271022),
    drugs = c(904453, 906780, 923645, 929887, 948078, 953076, 961047, 985247, 992956,
              997276, 1102917, 1113648, 1115008, 1118045, 1118084, 1124300, 1126128,
              1136980, 1146810, 1150345, 1153928, 1177480, 1178663, 1185922, 1195492,
              1236607, 1303425, 1313200, 1353766, 1507835, 1522957, 1721543, 1746940,
              1777806, 19044727, 19119253, 36863425),
    diagnosticProcedures = c(4087381, 4143985, 4294382, 42872565, 45888171, 46257627),
    measurements    = c(3000905, 3000963, 3003458, 3012471, 3016251, 3018677, 3020416,
                        3022217, 3023314, 3024929, 3034426),
    alternativeDiagnosis = c(24966, 76725, 195562, 316457, 318800, 4096682),
    treatmentProcedures = c(0),
    complications =  c(132797, 196152, 439777, 4192647)
  )
```

```
##   |
##   |
## Getting cohort table.
##   |
## Getting patient data for keeperOutput.
##   |
```

```
keeper
```

```
## # A tibble: 100 x 18
##    personId   age gender observationPeriod  visitContext presentation comorbidities symptoms priorDi
##       <dbl> <dbl> <chr>  <chr>              <chr>        <chr>        <chr>         <chr>    <chr>
##  1       85    37 Female -13752.0 days - 4~ "Inpatient ~ Gastrointes~ ""            ""       ""
##  2       29    44 Female -15831.0 days - 1~ "Inpatient ~ Gastrointes~ ""            ""       ""
##  3       47    39 Male   -13990.0 days - 1~ "Inpatient ~ Gastrointes~ "Peptic ulce~ ""       ""
##  4       49    35 Female -12560.0 days - 1~ "Inpatient ~ Gastrointes~ ""            ""       ""
##  5       34    39 Female -14487.0 days - 8~ "Inpatient ~ Gastrointes~ ""            ""       ""
##  6       66    33 Female -12283.0 days - 2~ "Inpatient ~ Gastrointes~ "Peptic ulce~ ""       ""
```

```
##  7       57    40 Female -14725.0 days - 3~ ""           Gastrointes~ ""           ""       ""
##  8       41    35 Female -12933.0 days - 1~ ""           Gastrointes~ ""           ""       ""
##  9       72    42 Male   -15114.0 days - 6~ "Inpatient ~ Gastrointes~ ""           ""       ""
## 10       14    38 Male   -14108.0 days - 6~ "Inpatient ~ Gastrointes~ ""           ""       ""
## # i 90 more rows
## # i abbreviated name: 1: priorTreatmentProcedures
## # i 5 more variables: alternativeDiagnosis <chr>, afterDisease <chr>, afterTreatmentProcedures <chr>
```

3 LLM review

We can convert the Keeper output to prompts for a LLM, and parse the output of the LLM to get a classification of whether the patient truly had GI bleeding.

3.1 Generating prompts

We need two prompts: the system prompt is a general description of how the LLM should behave. The (main) prompt contains the patient-specific information. First we create settings, then we generate the system prompt:

```
# Use the default settings:
settings <- createPromptSettings()
systempPrompt <- createSystemPrompt(setting = settings,
                                    diseaseName = "Gastrointestinal bleeding")
writeLines(systempPrompt)
```

```
## Act as a medical doctor reviewing a patient's healthcare data captured during routine clinical care,
## Write a medical narrative that fits the recorded health data followed by a determination of whether
##
## Remember that recording a diagnosis for a disease could occur either because the patient had the dis
##
## In your final summary, indicate "yes" if the most probable scenario is that the patient had Gastroin
## Indicate "no" if it is not the most probable scenario, for example when it is more likely that the p
##
## Use the following format:
##
## Clinical narrative:
##
## Evidence in favor of Gastrointestinal bleeding:
##
## Evidence against Gastrointestinal bleeding:
##
## Summary: (Only "yes" or "no")
```

Then, for each patient we can generate the main prompt:

```
row <- keeper[1, ]
prompt <- createPrompt(setting = settings,
                       diseaseName = "Gastrointestinal bleeding",
                       keeperRow = row)
writeLines(prompt)
```

```
## Demographics and details about the visit: Female, thirty-seven yo; Visit: Inpatient Visit (1.0 days)
##
## Diagnoses recorded on the day of the visit: Gastrointestinal hemorrhage;
##
## Diagnoses recorded prior to the visit: None
```

```
##
## Treatments recorded prior to the visit: Ibuprofen (day -6182.0, for 28.0 days); celecoxib (day -17.0
##
## Diagnostic procedures recorded proximal to the visit: None
##
## Laboratory tests recorded proximal to the visit: None
##
## Alternative diagnoses recorded proximal to the visit: None
##
## Diagnoses recorded after the visit: None
##
## Treatments recorded during or after the visit: Naproxen (day 1436.0, for 35.0 days);
```

3.2 Generate a response

Next, we can use the system prompt and prompt together to query the LLM. Many LLMs are available, including open source ones. Setting up and interacting with the LLM is beyond the scope of this vignette. Here we assume a LLM was used to generate this response:

```
response <- "Summary: yes"
```

3.3 Parse the response

LLMs can be quite verbose, and often we just want a yes or no answer. For this we can use the `parseLlmResponse()` function that uses a set of patterns to decide between 'yes', 'no', or 'I don't know':

```
parseLlmResponse(response)
```

```
## [1] "yes"
```