

Aplicación Meta-Análisis Automatizado con R, Shiny y GPT-4

Salome Margaret Quispe Hilasaca

1. Objetivo

La aplicación desarrollada permite analizar automáticamente artículos científicos en formato PDF. Utiliza Shiny (R) como interfaz gráfica y la API de OpenAI para resumir secciones específicas del texto. El objetivo es automatizar el proceso de revisión preliminar de literatura para investigaciones tipo meta-análisis cualitativo.

2. Librerías Utilizadas

- shiny: crea la aplicación web.
- pdftools: extrae texto de archivos PDF.
- httr y jsonlite: permiten conectarse y comunicarse con la API de OpenAI.
- DT: muestra los resultados en una tabla interactiva.

3. Función de análisis con OpenAI

`analizar_con_gpt()` toma una sección de texto y la envía como entrada a GPT-4. Luego procesa la respuesta para extraer un resumen.

```
analizar_con_gpt <- function(seccion, texto) {  
  prompt <- paste0("Resume la secci n: ", seccion, texto)  
  ...  
  res <- POST("https://api.openai.com/v1/chat/completions", ...)  
  parsed <- content(res, as = "parsed")  
  return(parsed$choices[[1]]$message$content)  
}
```

4. Interfaz de Usuario (UI)

La interfaz usa `fluidPage()` para dividir en dos columnas:



Figure 1: Interfaz

- Izquierda: carga de PDFs y botón de análisis.
- Derecha: visualización de resultados.

5. Servidor y procesamiento de texto

Al presionar “Analizar Artículos”:

1. Se leen los PDFs con `pdf_text()`.
2. Se dividen en secciones: Título, Abstract, Introducción, etc.
3. Cada sección se envía a la función `analizar_con_gpt()`.
4. Se arma una tabla con los resúmenes.

```
observeEvent(input$analizar, {
  texto <- pdf_text(input$pdfs$datapath[i])
  secciones <- list(Abstract = ..., Resultados = ...)
  analisis <- sapply(secciones, analizar_con_gpt)
})
```

6. Visualización

La tabla final se muestra con `datatable()` del paquete DT en el panel derecho.

7. Estética

Se añadió CSS personalizado para mejorar la visualización:

- Títulos estilizados.
- Paneles con bordes redondeados.
- Iconos (, ,) para mejor UX.

8. Resultado Esperado

Cada documento PDF es analizado y se genera una tabla con las secciones resumidas como muestra:



Meta-Análisis Automático de PDFs

Subir artículos en PDF

✓ Análisis completado con éxito!

Resultados del Análisis

Documento	Seccion
Titulo	Archeeology, chronology, and sedimentological context of the youngest Midd...
Abstract	This section presents new archaeologic-ical, sedimentological, and chronologi...
Introducción	Due to the scarcity of well-dated archaeological horizons, diachronic litnic traditions are not well undorsto...
Resultados	The site was dated using the optically stimulated luminescence (OSL) metho-around 80,000 years ago.
Discusión	The lithic assomblage consisted main-ly of elongated Levallois flakes and bidirectional blades.
Conciusiones	The findings suggest distinct stone tool production trajectories at the <i>Rub'</i> al Kriali desert's termination.
Conclusiones	The study indicates varied human settlement patterns in ties this period.

Aplicación Meta-Análisis Automatizado con R, Shiny y GPT-4 Salome Margaret Quispe Hilasaca

9. Código Completo de la Aplicación

```
library(shiny)
library(pdftools)
library(httr)
library(jsonlite)
```

```

library(DT)

api_key <- "sk-proj-..." # clave protegida

analizar_con_gpt <- function(seccion, texto) {
  if (nchar(texto) < 100) return("Contenido insuficiente.")
  prompt <- paste0("Resume la siguiente secci n de un art culo
cient fico, indicando: tema, lugar, t cnicas, cronolog a,
hallazgos importantes.\n\nSECCI N: ", seccion, "\n\nTEXT0:\n",
  substr(texto, 1, 3000))

  tryCatch({
    res <- POST(
      url = "https://api.openai.com/v1/chat/completions",
      add_headers(Authorization = paste("Bearer", api_key)),
      content_type_json(),
      encode = "json",
      body = list(
        model = "gpt-4",
        messages = list(list(role = "user", content = prompt)),
        temperature = 0.3
      )
    )
    parsed <- content(res, as = "parsed")
    if (!is.null(parsed$choices)) {
      return(parsed$choices[[1]]$message$content)
    } else if (!is.null(parsed$error$message)) {
      return(paste("Error API:", parsed$error$message))
    } else {
      return("Respuesta vac a o no estructurada.")
    }
  }, error = function(e) {
    return(paste("Error de conexi n:", e$message))
  })
}

ui <- fluidPage(
  tags$head(tags$style(HTML(".title-custom { font-size: 28px; font-
weight: bold; margin-top: 20px; margin-bottom: 20px; }
.table-style td, .table-style th { text-align: center; vertical-
align: middle; }
.well-custom { background-color: #f7f7f7; padding: 15px; border-
radius: 10px; border: 1px solid #ccc; }"))),

  titlePanel(div(icon("book-open", lib = "font-awesome"), span("
Meta-An lisis Autom tico de PDFs", class = "title-custom"))),

  fluidRow(

```

```

column(4,
  wellPanel(
    fileInput("pdfs", "Subir art culos en PDF",
      multiple = TRUE, accept = ".pdf"),
    actionButton("analizar", "Analizar Art culos",
      class = "btn btn-primary btn-block"),
    br(),
    textOutput("estado")
  )
),
column(8,
  wellPanel(
    h4("Resultados del An lisis"),
    DTOutput("tabla_resultados")
  )
)
)
)

server <- function(input, output, session) {
  resultados <- reactiveVal(data.frame())

  observeEvent(input$analizar, {
    req(input$pdfs)
    output$estado <- renderText("Analizando documentos...")

    lista_final <- list()
    for (i in seq_along(input$pdfs$datapath)) {
      texto <- paste(pdf_text(input$pdfs$datapath[i]), collapse = "\n")
      secciones <- list(
        T tulo = substr(texto, 1, 300),
        Abstract = sub("(?i).*Abstract", "", texto),
        Introducci n = sub("(?i).*Introduction", "", texto),
        Metodolog a = sub("(?i).*Material and methods|Methods", "", texto),
        Resultados = sub("(?i).*Results", "", texto),
        Discusi n = sub("(?i).*Discussion", "", texto),
        Conclusiones = sub("(?i).*Conclusion", "", texto)
      )
      analisis <- sapply(names(secciones), function(seccion) {
        analizar_con_gpt(seccion, secciones[[seccion]])
      }, USE.NAMES = TRUE)
      tabla <- data.frame(
        Documento = rep(input$pdfs$name[i], length(analisis)),
        Seccion = names(analisis),
        Analisis = unname(analisis),
        stringsAsFactors = FALSE

```

```

    )
    lista_final[[i]] <- tabla
  }

  resultado_total <- do.call(rbind, lista_final)
  resultados(resultado_total)
  output$estado <- renderText("      Anlisis      completado con
      xito !")
})

output$tabla_resultados <- renderDT({
  req(resultados())
  datatable(resultados(), options = list(pageLength = 10,
    autoWidth = TRUE), class = "table table-bordered table-style"
  )
})
}

shinyApp(ui = ui, server = server)

```

10. Conclusión

Esta app automatiza una parte importante de un meta-análisis cualitativo: la revisión de literatura. Reduce tiempos, mejora la lectura rápida de artículos extensos y puede ser extendida para análisis estadísticos futuros.