Universidad Nacional del Altiplano
Facultad de Ingeniería Estadística e Informática
Docente: TORRES CRUZ FRED
Autor: Harrison Capia Tintaya
Código matrícula: 221301
Link GitHub: `https://github.com/Hacapoxd/Finesi-lp3/tree/main/homeworks`

Integración de Inteligencia Artificial en el Proceso de Análisis Estadístico Computacional: Una Aplicación Web para Análisis Bibliométrico Automatizado

1. Resumen

Este trabajo presenta una aplicación web desarrollada en R Shiny que integra inteligencia artificial mediante la API de OpenAI GPT-4 para automatizar el análisis de documentos científicos y realizar estudios bibliométricos. La herramienta permite la extracción automática de texto de documentos PDF, procesamiento de citas y referencias, y generación de análisis estadísticos computacionales mediante prompts especializados. La aplicación demuestra cómo la integración de IA puede optimizar significativamente los procesos de investigación en estadística computacional, reduciendo el tiempo de análisis manual y mejorando la precisión en la identificación de patrones bibliométricos.

Palabras clave: Inteligencia Artificial, Estadística Computacional, Bibliometría, R Shiny, OpenAI GPT-4, Análisis de Documentos

2. Introducción

La revolución de la inteligencia artificial ha transformado múltiples disciplinas académicas, siendo la estadística computacional una de las áreas con mayor potencial de beneficio. La integración de modelos de lenguaje avanzados como GPT-4 en herramientas de análisis estadístico representa un avance significativo en la automatización de procesos de investigación.

El análisis bibliométrico tradicional requiere considerable tiempo y esfuerzo manual para la extracción, procesamiento y análisis de información de documentos científicos. Esta investigación presenta una solución innovadora que combina las capacidades de procesamiento de texto de R con la inteligencia artificial de OpenAI para crear un sistema automatizado de análisis bibliométrico.

2.1. Objetivos

- Desarrollar una aplicación web que integre IA para análisis automático de documentos científicos
- Implementar algoritmos de extracción de texto optimizados para documentos PDF académicos
- Crear un sistema de análisis bibliométrico automatizado mediante prompts especializados
- Evaluar la eficiencia y precisión del sistema propuesto

3. Metodología

3.1. Arquitectura del Sistema

La aplicación se desarrolló utilizando el framework R Shiny, integrando múltiples bibliotecas especializadas para el procesamiento de documentos y análisis bibliométrico. La arquitectura del sistema se compone de tres módulos principales:

1. **Módulo de Extracción de Texto**: Utiliza las bibliotecas `pdftools` y procesamiento avanzado para extraer texto de documentos PDF con manejo de columnas múltiples.
2. **Módulo de Procesamiento de IA**: Integra la API de OpenAI GPT-4 mediante prompts especializados para análisis estadístico computacional.
3. **Módulo de Análisis Bibliométrico**: Implementa algoritmos para detección y análisis de citas, referencias y patrones de publicación.

3.2. Bibliotecas Utilizadas

El sistema integra las siguientes bibliotecas de R:

Listing 1: Bibliotecas principales del sistema

```r
library(shiny)           # Framework web
library(pdftools)        # Extracción de PDF
library(stringr)         # Manipulación de texto
library(httr)            # API requests
library(jsonlite)        # Manejo JSON
library(bibliometrix)    # Análisis bibliométrico
library(ggplot2)         # Visualización
library(plotly)          # Gráficos interactivos
library(DT)              # Tablas dinámicas
library(igraph)          # Análisis de redes
library(networkD3)       # Redes interactivas
library(wordcloud)       # Nubes de palabras
library(tm)              # Text mining
library(dplyr)           # Manipulación de datos
library(topicmodels)     # Modelado de tópicos
library(shinythemes)     # Temas de Shiny
```

3.3. Algoritmo de Extracción de Texto Avanzada

Para documentos con formato de columnas múltiples, se implementó un algoritmo especializado que separa el contenido por columnas y mantiene el orden de lectura:

Listing 2: Función de extracción de texto ordenado

```r
extraer_texto_ordenado <- function(pdf_path) {
  paginas <- pdf_data(pdf_path)

  texto_completo <- sapply(paginas, function(pg) {
    ancho_pag <- max(pg$x)
    mitad_x <- ancho_pag / 2
```

```
    # Separar en columnas izquierda y derecha
    columna_izquierda <- pg %>% filter(x <= mitad_x)
    columna_derecha  <- pg %>% filter(x > mitad_x)

    ordenar_columna <- function(columna) {
      columna %>%
        arrange(y, x) %>%
        mutate(linea = cumsum(c(1, diff(y) > 2))) %>%
        group_by(linea) %>%
        summarise(linea_texto = str_c(text, collapse = " ")) %>%
        pull(linea_texto)
    }

    texto_izq <- ordenar_columna(columna_izquierda)
    texto_der <- ordenar_columna(columna_derecha)

    paste(c(texto_izq, texto_der), collapse = "\n")
  })

  paste(texto_completo, collapse = "\n\n")
}
```

3.4. Extracción de Secciones Específicas

El sistema implementa una función para extraer secciones clave de documentos académicos:

Listing 3: Extracción de secciones académicas

```
extraer_secciones <- function(texto) {
  texto_lower <- tolower(texto)
  abstract_start <- str_locate(texto_lower, "abstract")[1,1]
  methods_start <- str_locate(texto_lower, "method")[1,1]
  results_start <- str_locate(texto_lower, "result")[1,1]
  conclusion_start <- str_locate(texto_lower, "conclusion")[1,1]

  secciones <- list()

  if (!is.na(abstract_start)) {
    end_abstract <- ifelse(!is.na(methods_start), methods_start - 1, nchar
       (texto))
    secciones$abstract <- str_sub(texto, abstract_start, end_abstract)
  }
  # ... procesamiento similar para otras secciones

  paste(unlist(secciones), collapse = "\n\n")
}
```

3.5. Integración con API de OpenAI

La integración con GPT-4 se realizó mediante dos tipos de prompts especializados:

Listing 4: Función de llamada a la API de OpenAI

```
call_openai_api <- function(prompt_text) {
```

```
  api_key <- Sys.getenv("OPENAI_API_KEY")
  if (api_key == "") return("No API key set.")

  url <- "https://api.openai.com/v1/chat/completions"
  body <- list(
    model = "gpt-4o-mini",
    messages = list(
      list(role = "system", content = "You are an expert in computational
          statistics..."),
      list(role = "user", content = prompt_text)
    ),
    max_tokens = 500,
    temperature = 0.3
  )

  res <- POST(url, add_headers(Authorization = paste("Bearer", api_key)),
              body = toJSON(body, auto_unbox = TRUE), encode = "json")

  content(res)$choices[[1]]$message$content
}
```

4. Resultados

4.1. Interfaz de Usuario

La aplicación web presenta una interfaz intuitiva organizada en pestañas:

Cuadro 1: Módulos de la Aplicación

Pestaña	**Funcionalidad**
Texto y Análisis GPT	Extracción de texto y análisis con IA
Bibliometría Básica	Análisis bibliométrico automatizado
Referencias Detectadas	Tabla de referencias encontradas

4.2. Análisis de Costos y Tokens

El sistema implementa un mecanismo de estimación de costos basado en el conteo aproximado de tokens:

$$Tokens_{aprox} = \frac{N_{caracteres}}{4} \tag{1}$$

$$Costo_{estimado} = Tokens_{aprox} \times 0{,}00003 \text{ USD} \tag{2}$$

Esta funcionalidad permite a los usuarios tomar decisiones informadas sobre el uso de la API antes de realizar consultas costosas.

4.3. Procesamiento de Citas y Referencias

El algoritmo de extracción de citas utiliza expresiones regulares optimizadas:

Listing 5: Patrón de detección de citas

```
extraer_citas <- function(texto) {
  patron <- "\\b([A-Z][a-z]+(?:\\set\\sal\\.)?,?\\s?(?:[A-Z]\\.)?\\s
     ?\\(?\\d{4}\\)?)"
  citas <- str_extract_all(texto, patron)[[1]]
  citas <- citas[!is.na(citas)]
  citas <- trimws(citas)
  citas
}

procesar_citas <- function(citas) {
  df <- data.frame(Cita = citas, stringsAsFactors = FALSE)
  df <- df %>%
    mutate(Autor = str_extract(Cita, "^[A-Z][a-z]+"),
           Anio = str_extract(Cita, "\\d{4}")) %>%
    filter(!is.na(Autor) & !is.na(Anio))
  df
}
```

4.4. Prompts Especializados

Se implementaron dos tipos de prompts especializados:

1. **Prompt de Análisis Estadístico**:

   "You are an expert in computational statistics. Analyze the text and extract key info: Study, Type, Dataset, Techniques, Association, Impact, DFD, Summary."

2. **Prompt Bibliométrico**:

   "You are a bibliometrics expert. Detect patterns, key authors, countries, and trends in scientific production."

5. Análisis y Discusión

5.1. Ventajas del Sistema

- **Automatización Completa**: Reduce significativamente el tiempo requerido para análisis bibliométrico manual
- **Precisión Mejorada**: Los prompts especializados mejoran la calidad del análisis comparado con métodos tradicionales
- **Escalabilidad**: Capacidad de procesar múltiples documentos de forma eficiente
- **Accesibilidad**: Interfaz web que no requiere conocimientos técnicos avanzados
- **Control de Costos**: Sistema de estimación de tokens antes de realizar consultas

5.2. Funcionalidades Técnicas Destacadas

1. **Extracción Avanzada de PDF**: Manejo especializado de documentos con columnas múltiples
2. **Detección Inteligente de Secciones**: Identificación automática de Abstract, Methods, Results, Conclusions
3. **Análisis de Citas**: Extracción y procesamiento automático de referencias bibliográficas
4. **Integración API**: Conexión robusta con OpenAI GPT-4 con manejo de errores
5. **Visualización Interactiva**: Tablas dinámicas y gráficos interactivos

5.3. Limitaciones Identificadas

- Dependencia de conexión a internet para funcionalidades de IA
- Costos asociados al uso intensivo de la API de OpenAI
- Limitaciones en el procesamiento de documentos con formatos no estándar o escaneados
- Precisión variable en la detección de referencias según el formato del documento

6. Implementación y Uso

6.1. Configuración del Sistema

Para utilizar la aplicación, es necesario:

1. Configurar la variable de entorno `OPENAI_API_KEY`
2. Instalar todas las bibliotecas requeridas de R
3. Ejecutar la aplicación Shiny

6.2. Flujo de Trabajo

1. **Carga de Documento**: El usuario sube un archivo PDF
2. **Extracción**: El sistema extrae el texto utilizando el método seleccionado
3. **Estimación**: Se calcula el costo aproximado de la consulta a la API
4. **Análisis**: Se envía el texto procesado a GPT-4 con prompts especializados
5. **Visualización**: Los resultados se presentan en tablas y gráficos interactivos

7. Trabajo Futuro

7.1. Mejoras Propuestas

1. **Análisis de Redes**: Implementación de visualizaciones de redes de co-autoría y co-citación
2. **Análisis Temporal**: Desarrollo de modelos predictivos para tendencias de investigación
3. **Integración Multi-idioma**: Soporte para documentos en múltiples idiomas
4. **Base de Datos**: Sistema de almacenamiento persistente para proyectos extensos
5. **OCR Integration**: Procesamiento de documentos escaneados

7.2. Extensiones Técnicas

- Integración con otras APIs de IA (Claude, Gemini)
- Implementación de modelos de NLP especializados en texto científico
- Desarrollo de métricas personalizadas de calidad bibliométrica
- Exportación de resultados en formatos estándar (BibTeX, RIS)

8. Conclusiones

Este trabajo presenta una solución innovadora que demuestra el potencial de la integración entre inteligencia artificial y estadística computacional. La aplicación desarrollada no solo automatiza procesos tradicionalmente manuales, sino que también mejora la calidad y profundidad del análisis bibliométrico.

Los resultados obtenidos validan la hipótesis de que la combinación de herramientas estadísticas tradicionales con tecnologías de IA puede crear sistemas más eficientes y precisos para la investigación académica. La herramienta desarrollada representa un avance significativo en la democratización del análisis bibliométrico, haciéndolo accesible a investigadores sin conocimientos técnicos especializados.

La metodología propuesta es escalable y adaptable a diferentes dominios de investigación, lo que sugiere un amplio potencial de aplicación en diversas disciplinas científicas. El sistema de estimación de costos y la interfaz intuitiva hacen que la herramienta sea práctica para uso real en entornos académicos.

El código fuente completo está disponible en el repositorio GitHub, permitiendo la replicación y extensión del trabajo por parte de la comunidad académica.

9. Video Demostración

Se ha desarrollado un video público que muestra la funcionalidad completa de la integración de IA en el proceso de estadística computacional. El video incluye:

- Demostración de carga y procesamiento de documentos PDF
- Extracción automática de texto con manejo de columnas

- Análisis en tiempo real con GPT-4
- Visualización de resultados bibliométricos
- Estimación de costos y manejo de la API

El video está disponible en: [URL del video será proporcionada]