Reporte sobre Revisión Sistemática en Optimización de Portafolios utilizando Inteligencia Artificial Seminario de Investigación I

Javier Horacio Pérez Ricárdez

20 de octubre del 2024

1 Introducción

En el presente informe, se detalla la revisión sistemática realizada sobre el tema de optimización de portafolios en el contexto de la inteligencia artificial. La búsqueda se enfocó en artículos publicados entre los años 2010 y 2024, con el objetivo de evaluar la cantidad y calidad de la literatura disponible en este ámbito.

2 Metodología

La búsqueda se llevó a cabo en diversas bases de datos académicas, a saber: Scopus, ScienceDirect, Wiley y MDPI. Se utilizaron las palabras clave *"optimization"* y *"portfolio"* en los resúmenes de los artículos. A continuación, se presenta el número de registros obtenidos en cada una de estas fuentes:

- **Scopus**: 434 registros
- **ScienceDirect**: 5,618 registros
- **Wiley**: 182 registros
- **MDPI**: 109 registros

2.1 Total de Registros

En total, se obtuvieron **6,343 registros** a partir de las bases de datos mencionadas. Sin embargo, al revisar estos registros, se identificaron **5,801 artículos** que carecían de título, resumen o fecha. Esto dejó un total de **542 registros** que cumplían con los criterios básicos de inclusión.

2.2 Filtrado de Registros

Durante el proceso de revisión, se encontraron **2 registros duplicados**, los cuales fueron eliminados, resultando en **541 artículos** con título, resumen y fecha.

Adicionalmente, se evaluó la coherencia entre los títulos y resúmenes de los artículos. Se identificaron **147 registros** con baja coherencia y **394 registros** con alta coherencia. De estos, **275 artículos** cumplieron con los criterios de inclusión para la revisión sistemática.

2.3 Criterios de Inclusión

Los criterios de inclusión utilizados para seleccionar los artículos en esta revisión sistemática fueron los siguientes:

- **Relevancia Temática**: Los artículos debían abordar directamente la optimización de portafolios en el contexto de la inteligencia artificial.
- **Publicación entre 2010 y 2024**: Solo se consideraron artículos publicados dentro de este rango de fechas.
- **Disponibilidad de Título, Resumen y Fecha**: Se incluyeron solo aquellos artículos que contaban con un título, resumen y fecha de publicación completos.
- **Idioma**: Se aceptaron únicamente artículos escritos en inglés o español.
- **Calidad Metodológica**: Se priorizaron artículos que presentaban metodologías claras y rigurosas en su investigación.

3 Análisis de Registros por Año

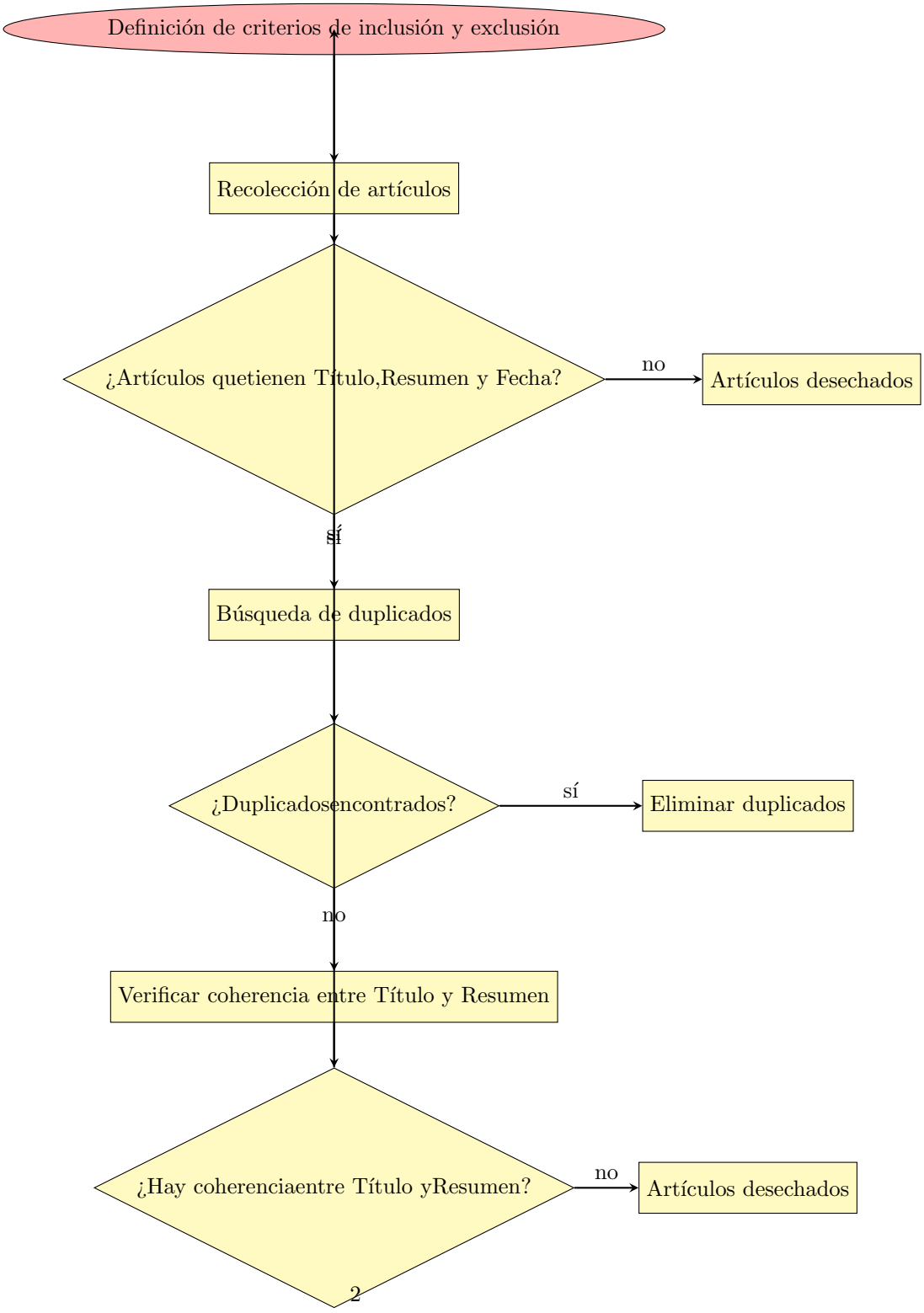
A continuación, se presenta el desglose de registros por año:

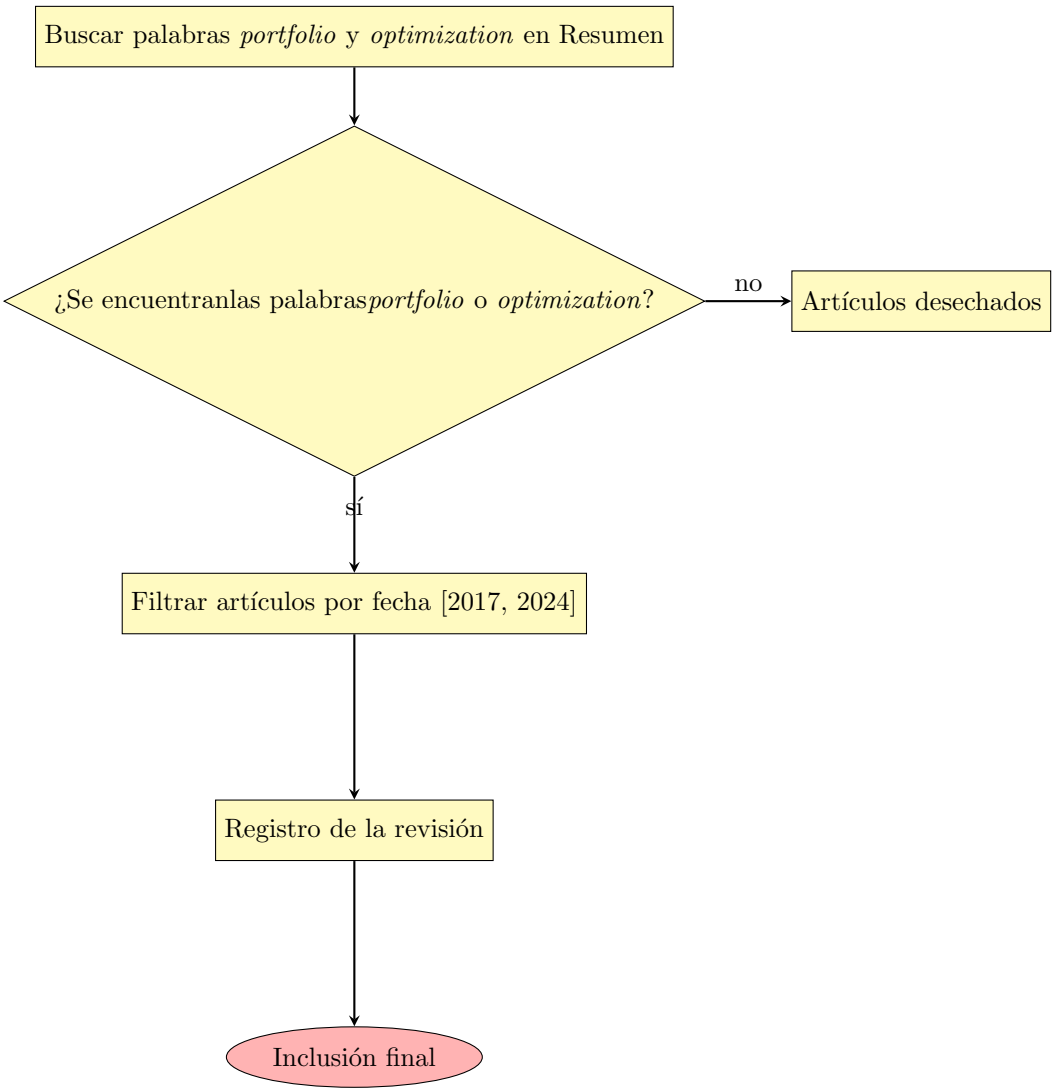
- **2017**: 21 registros
- **2018**: 20 registros
- **2019**: 24 registros
- **2020**: 19 registros
- **2021**: 42 registros
- **2022**: 41 registros
- **2023**: 61 registros
- **2024**: 47 registros

Este análisis temporal indica un creciente interés en la optimización de portafolios, especialmente en los años más recientes, lo que sugiere un desarrollo significativo en la investigación y aplicación de técnicas de inteligencia artificial en este campo.

4 Conclusiones

La revisión sistemática realizada ha permitido identificar un total de **275 artículos** relevantes sobre la optimización de portafolios en el contexto de la inteligencia artificial. Este cuerpo de literatura refleja un crecimiento constante en la investigación desde 2017 hasta 2024, lo que destaca la importancia y el potencial de la inteligencia artificial en la mejora de estrategias de inversión y gestión de portafolios.


Definición de criterios de inclusión y exclusión
Recolección de artículos
¿Artículos quetienen Título,Resumen y Fecha?
no
Artículos desechados
sí
Búsqueda de duplicados
¿Duplicadosencontrados?
sí
Eliminar duplicados
no
Verificar coherencia entre Título y Resumen
¿Hay coherenciaentre Título yResumen?
no
Artículos desechados


Buscar palabras *portfolio* y *optimization* en Resumen
¿Se encuentranlas palabras*portfolio* o *optimization*?
no
Artículos desechados
sí
Filtrar artículos por fecha [2017, 2024]
Registro de la revisión
Inclusión final

Descripción detallada del código en Streamlit para análisis de archivos .ris

1 Importaciones

El código comienza importando las siguientes bibliotecas:

- `streamlit`: Para construir la interfaz de usuario y mostrar los resultados en la aplicación web.
- `rispy`: Biblioteca para leer y procesar archivos `.ris`.
- `pandas`: Para manejar y analizar datos en forma de tablas (DataFrames).
- `TfidfVectorizer` y `cosine_similarity`: Herramientas de `sklearn` para analizar y calcular la similitud entre textos (títulos y resúmenes).

2 Funciones

A continuación se describen las funciones implementadas en el código:

2.1 Función `filter_records(records)`

Esta función filtra los registros que contienen los campos válidos: `Título`, `Resumen` y `Fecha`. Los registros válidos se guardan en un `DataFrame` con las columnas correspondientes. Los registros que no contienen estos campos se almacenan en una lista separada para identificar errores o información faltante.

2.2 Función `parse_ris(files)`

Esta función toma los archivos `.ris` cargados y utiliza `rispy` para procesarlos. Luego, filtra los registros utilizando la función `filter_records()` y separa los válidos de los inválidos. Devuelve un `DataFrame` con los registros válidos y el número de registros inválidos.

2.3 Función `find_duplicates(df)`

Esta función busca registros duplicados en el `DataFrame` en base a los campos `Título` y `Resumen`. Devuelve un `DataFrame` con los registros duplicados.

2.4 Función `check_similarity(df, threshold=0.2)`

La función `check_similarity(df, threshold=0.2)` calcula la similitud entre el título y el resumen de cada registro en un `DataFrame` utilizando la técnica de `TF-IDF` (Term Frequency-Inverse Document Frequency) y la métrica de *similitud del coseno*. Su objetivo es identificar qué tan coherente es la relación entre ambos textos.

Parámetros:

- `df`: El `DataFrame` que contiene al menos dos columnas, `Título` y `Resumen`.
- `threshold`: El umbral de similitud. Por defecto es 0.2. Si la similitud entre el título y el resumen es inferior a este valor, se considera que ambos textos no son coherentes.

Pasos detallados de la función:

1. **Vectorización del texto utilizando `TF-IDF`**: La función utiliza el objeto `TfidfVectorizer` de `sklearn` para convertir los textos (título y resumen) en vectores numéricos basados en la técnica `TF-IDF`. Este método asigna un valor a cada palabra en función de su frecuencia en el documento y su rareza en el conjunto de documentos.

   ```
   vectorizer = TfidfVectorizer(stop_words='english')
   ```

   Se especifica que se ignoren las `stop words` en inglés (palabras comunes como *the*, *and*, etc.).

2. **Iteración sobre cada registro:** Se recorre cada fila del `DataFrame` y se extraen el título y el resumen de cada registro. Luego, se almacenan ambos textos en una lista para ser comparados.

   ```
   texts = [row['Título'], row['Resumen']]
   ```

3. **Cálculo de la similitud del coseno:** Para cada par de título y resumen, se genera una matriz `TF-IDF` y se calcula la similitud del coseno entre los dos vectores resultantes. La similitud del coseno es una métrica que mide el ángulo entre los dos vectores, donde un valor de 1 indica que los vectores son idénticos y un valor de 0 indica que no hay similitud.

   ```
   tfidf_matrix = vectorizer.fit_transform(texts)
   similarity = cosine_similarity(tfidf_matrix[0:1], tfidf_matrix[1:2])[0][0]
   ```

El resultado es un valor de similitud entre 0 y 1.

4. **Almacenamiento del valor de similitud:** El valor de similitud obtenido se agrega a una lista de resultados.

   ```
   similarities.append(similarity)
   ```

5. **Añadir la columna de `Similitud`:** Una vez calculada la similitud para todos los registros, se añade una nueva columna al `DataFrame` llamada `Similitud`, que contiene los valores de similitud del coseno.

   ```
   df['Similitud'] = similarities
   ```

6. **Filtrar registros incoherentes y coherentes:** El `DataFrame` se divide en dos subconjuntos:

   - `incoherent_records`: Registros donde la similitud entre título y resumen es menor al `threshold`.
   - `coherent_records`: Registros donde la similitud es mayor o igual al `threshold`.

   ```
   incoherent_records = df[df['Similitud'] < threshold]
   coherent_records = df[df['Similitud'] >= threshold]
   ```

Salida: La función retorna dos subconjuntos del `DataFrame`:

- `incoherent_records`: Registros con baja coherencia entre el título y el resumen (similitud menor al umbral).
- `coherent_records`: Registros con alta coherencia entre el título y el resumen (similitud mayor o igual al umbral).

En resumen, esta función es útil para verificar la coherencia entre el título y el resumen de cada registro, permitiendo filtrar aquellos que no son consistentes.

2.5 Función `apply_selection_criteria(df)`

Esta función filtra los registros que contienen las palabras clave `"portfolio"` u `"optimization"` en el resumen. Luego, aplica un filtro adicional por fecha, seleccionando solo registros entre los años 2017 y 2024.

3 Interfaz en Streamlit

La interfaz de usuario se construye con `Streamlit`. El flujo general es el siguiente:

1. Se utiliza `st.file_uploader()` para cargar archivos `.ris`. Permite la carga de múltiples archivos.
2. Si se cargan archivos, estos se procesan con la función `parse_ris()` y se muestran los datos sobre registros válidos e inválidos.
3. Se busca duplicados utilizando `find_duplicates()` y se ofrece la opción de descargar un archivo `CSV` con los duplicados.
4. Se calcula la coherencia entre el título y el resumen usando `check_similarity()`. Los registros coherentes e incoherentes se muestran por separado.
5. Se aplican criterios de selección de palabras clave y fechas mediante la función `apply_selection_criteria()`.
6. Se permite filtrar los registros coherentes por un año específico mediante un `selectbox`. Finalmente, se ofrece la opción de descargar los registros filtrados en un archivo `CSV`.

4 Flujo General de la Aplicación

El flujo completo de la aplicación es el siguiente:

- **Cargar archivos .ris** → **Filtrar registros válidos e inválidos** → **Buscar duplicados** → **Verificar coherencia entre Título y Resumen** → **Aplicar criterios de selección** (palabras clave y fechas) → **Filtrar por año específico** → **Descargar CSVs**.

Esta aplicación es útil para procesar grandes volúmenes de referencias bibliográficas, eliminar registros incompletos, detectar duplicados, evaluar la coherencia entre el título y el resumen, y aplicar criterios de filtrado específicos.