

# Resume Screening LLM

Objetivo	1
Arquitectura	1
Configuración del Colab	3
Resultados	5

## Objetivo

El objetivo de este documento es mostrar como se ha creado un libro en Colab para hacer Resume Screening de hojas de vida de acuerdo a un rol en particular.  
Es un proceso básico y de inicio, abierto a mejoras que se consideren.

Los desafíos que enfrentan los reclutadores durante la selección de currículums: el alto volumen de currículums recibidos (hasta el 88 % de ellos no están calificados) aumenta enormemente el tiempo de contratación. Los reclutadores enfrentan una mayor presión para demostrar la calidad de la contratación, pero carecen de herramientas para vincular la selección de currículums con las métricas posteriores a la contratación.

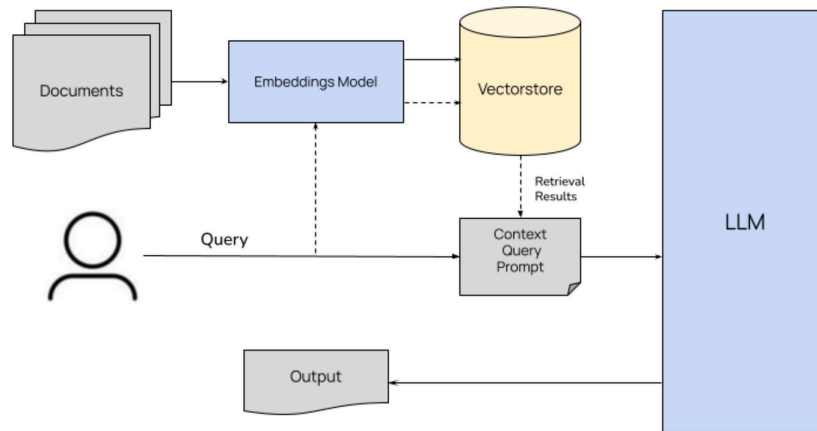
Resumen de la selección de currículums:  
Basado en un rol que se requiere buscar, el proceso determina cuales candidatos aplican de acuerdo al registro de su educación, experiencia y otra información registrada en su currículum.

Gracias a las innovaciones tecnológicas en la selección de currículums, se usa selección inteligente de currículums mediante IA para aprovechar las capacidades de los LLM y técnicas RAG.

Este proceso aprovecha el poder de Langchain y OpenAI para automatizar el proceso de análisis de currículums. Esta herramienta permite a los reclutadores procesar de manera eficiente grandes volúmenes de currículums, extraer automáticamente características importantes y generar una puntuación de coincidencia basada en una descripción de trabajo predefinida.

## Arquitectura

A continuación se muestra a alto nivel la arquitectura del sistema.



El proceso implementa la técnica RAG. La generación aumentada por recuperación (RAG) es una técnica de IA para mejorar la calidad de las respuestas generadas por LLM al basar el modelo en fuentes externas de conocimiento para complementar la representación interna de la información del LLM.

Las fuentes externas serán los CVs de los candidatos.

El proceso usa las siguientes librerías

Proceso de AI and LLM

langchain

langchain\_openai

langchain\_community

Lectura de pdf

pypdf

unstructured

PyPDF2

Manejo de bases de datos de Vectores

chromadb

Langchain-chroma

Embeddings y Bases de datos Vectores

Los embeddings son una técnica de procesamiento de lenguaje natural que convierte el lenguaje humano en vectores matemáticos. Estos vectores son una representación del significado subyacente de las palabras, lo que permite que las computadoras procesen el lenguaje de manera más efectiva.

Una de las formas más comunes de almacenar y buscar datos no estructurados es embeberlos y almacenar en vectores de como embeddings y, luego, en el momento de la consulta, en la consulta no estructurada y recuperar los vectores de embeddings que sean "más similares" a la consulta planteada en forma de

embeddings. Una base de datos de vectores se encarga de almacenar los datos incrustados y realizar la búsqueda de vectores.

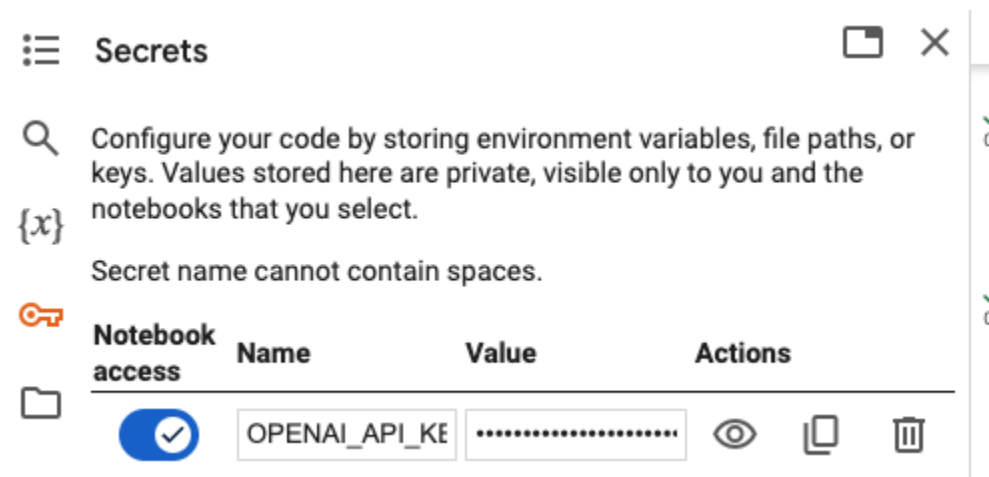
## LangChain

LangChain está diseñado para ayudar a los desarrolladores a crear aplicaciones de extremo a extremo utilizando modelos de lenguaje. Ofrece una variedad de herramientas, componentes e interfaces que simplifican el proceso de creación de aplicaciones basadas en grandes modelos de lenguaje y modelos de chat. LangChain optimiza la gestión de interacciones con los LLM, encadenando varios componentes e integrando recursos adicionales, como API y bases de datos.

Una característica clave de LangChain es su compatibilidad con indicaciones, que abarca la gestión de indicaciones, la optimización de indicaciones y una interfaz genérica para todos los LLM. El marco también proporciona utilidades comunes para trabajar con LLM.

## Configuración del Colab

Para usar el libro, es necesario asignar una KEY de OpenAI que se debe registrar como secreto del libro



Como LLM se usa el modelo de OpenAI gpt-4o-mini-2024-07-18

Se deben cargar los archivos de manera local en el Colab los archivos PDF deseados, en el caso del ejercicio, se ha creado una carpeta llamada CVs con los PDF de prueba

Se debe definir un arreglo de las rutas de los PDF con el objetivo de realizar el proceso de lectura de la información y luego aplicar proceso de Chunking

```
pdf_folder = 'CVs'
```

```
lista_archivos = [  
    f'/content/{pdf_folder}/CV1.pdf',  
    f'/content/{pdf_folder}/CV2.pdf',  
    f'/content/{pdf_folder}/CV3.pdf',  
    f'/content/{pdf_folder}/CV4.pdf',  
    f'/content/{pdf_folder}/CV5.pdf',  
    f'/content/{pdf_folder}/CV6.pdf',  
    f'/content/{pdf_folder}/CV7.pdf',  
    f'/content/{pdf_folder}/CV8.pdf',  
    f'/content/{pdf_folder}/CV9.pdf'  
]
```

## Chunking

El Chunking es una técnica que divide grandes fragmentos de texto en unidades más pequeñas conocidas como chunks.

## Importancia del Chunking

El Chunking es crucial para manejar una recopilación extensa de datos en aplicaciones LLM. Al hacer chunking, podemos alimentar el LLM con la información más relevante, lo que puede ayudarnos a evitar alucinaciones porque la información recuperada a través de RAG puede respaldarse para la generación de LLM. El chunking eficaz de la información puede mejorar la relevancia de la información que se carga en la ventana LLM durante la generación y esto, a su vez, hace que el LLM genere respuestas más precisas.

## Estrategias de Chunking

Nuestro objetivo es evaluar varias estrategias de chunking en función de las métricas de evaluación relevantes. Al comprender las fortalezas y debilidades de cada una, podemos determinar el enfoque más adecuado para escenarios específicos.

Para este caso hemos aplicado "RecursiveCharacterTextSplitter", que implica dividir iterativamente el texto en fragmentos más pequeños utilizando separadores predefinidos. Este proceso tiene como objetivo lograr uniformidad en el tamaño de los segmentos, aunque los tamaños exactos pueden variar.

El Prompt del proceso se define de la siguiente manera:

```
query = "Experience in Sales"  
prompt = dedent("""You are an expert recruiter.
```

Based on the context return output string with the following JSON structure, the following structure:

```
[
  {
    "name": CV name
    "email": CV email
    "contact_phone": CV phone
    "years_of_experience": CV Years of experience
    "skills": CV skills
    "ai_skills": CV evaluate if candidate has experience in any branches of Artificial
Intelligence (AI, ML, GenAI). Yes/No.
    "score": Evaluate between 1 and 10 the score related to requeriments in question
  }
]
context = Give information about CV.
question = "" +query+"".
If no matches are found, return the empty list.
""")
```

Donde:

**query** = Indica el rol o la frase que se desea buscar en los CVs que se han registrado.

**prompt** = Es la variable que crea el PROMPT para el LLM, debe crearse con un contexto de como se desea retornar la respuesta, en este caso en formato JSON (devuelve un string que debe ser interpretado como un objeto JSON).

NOTA: El prompt y las definiciones deben realizarse en ingles, ya que los CVs de prueba estan en este lenguaje. En caso de implementar otro, es posible que no se obtengan los resultados esperados, o deba realizar un proceso de traducción antes.

## Resultados

Cuando se realiza la pregunta por un rol con experiencia en ventas (sales), nos devuelve el siguiente resultado:

```
{
  "name": "Powell Finwood",
  "email": "hello@reallygreatsite.com",
  "contact_phone": "+123-456-7890",
  "years_of_experience": 5,
  "skills": [
    "Account Management",
    "Customer Relationship Management Software",
    "Strategic and Social Selling"
  ],
  "ai_skills": "No",
  "score": 8
}
```

```
{
  "name": "Alice Clark",
  "email": "Email me on Indeed",
  "contact_phone": "Not provided",
  "years_of_experience": "20+ years",
  "skills": [
    "data handling",
    "design",
    "development",
    "data analysis",
    "star/snowflake schema data modelling",
    "database designing",
    "scalability",
    "back-up and recovery",
    "writing and optimizing SQL code",
    "Stored Procedures",
    "creating functions",
    "views",
    "triggers",
    "indexes",
    "Microsoft Azure cloud services",
    "Document DB",
    "SQL Azure",
    "Stream Analytics",
    "Event hub",
    "Power BI",
    "Web Job",
    "Web App",
    "Azure data lake analytics (U-SQL)",
    "big data processing",
    "Azure data factory"
  ],
  "ai_skills": "Yes",
  "score": 8
}
```