Universitat Oberta de Catalunya (UOC)

Máster Universitario en Ciencia de Datos (*Data Science*)

TRABAJO FINAL DE MÁSTER

Área: Natural Language Processing and Visual Analytics

Data Mining, Graphs and Natural Language Processing

IA Generativa para la recuperación de información de convocatorias de ayudas a empresas

---

Autor: José Luis Rodríguez Andreu

Tutor: Diego Calvo Barreno

Profesor: Josep Anton Mir Tutusaus

Barcelona, 21 de marzo de 2025

FICHA DEL TRABAJO FINAL

Título del trabajo:	IA Generativa para la recuperación de información de convocatorias de ayudas a empresas
Nombre del autor:	José Luis Rodríguez Andreu
Nombre del colaborador/a docente:	Diego Calvo Barreno
Nombre del PRA:	Josep Anton Mir Tutusaus
Fecha de entrega (mm/aaaa):	06/2025
Titulación o programa:	Máster Universitario en Ciencia de Datos
Área del Trabajo Final:	Trabajo Fin de Máster
Idioma del trabajo:	Español
Palabras clave	LLM, RAG, AI

Dedicatoria/Cita

Breves palabras de dedicatoria y/o una cita.

Agradecimientos

Si se considera oportuno, mencionar a las personas, empresas o instituciones que hayan contribuido en la realización de este proyecto.

Abstract

In recent years, it has become increasingly difficult to find calls for financial aid from governmental institutions focused on companies and organizations. This growing need makes it essential to have systems that optimize the identification of calls for financial aid.

Currently, the lack of automated tools capable of interpreting and synthesizing available information hinders efficient access to these resources, forcing organizations to conduct manual searches that consume both time and effort.

This work presents the development of an Artificial Intelligence (AI)-based tool for extracting and retrieving information from economic aid calls.

By leveraging advanced Natural Language Processing (NLP) techniques and Generative AI, the solution can analyze, structure, and filter information automatically, providing relevant results based on the specific characteristics of each entity.

The main objective of the tool is to process and transform scattered aid calls into a structured dataset, facilitating their consultation and retrieval. This organized structure will allow companies to quickly and efficiently access the most relevant information, enhancing strategic decision-making.

In this way, the project addresses the challenge of filtering and synthesizing large volumes of unstructured and dispersed data from various platforms, streamlining the search process and improving access to funding opportunities.

Keywords: LLMs, IA, RAG

Resumen

En los últimos años cada vez es mas complicado encontrar convocatorias de ayuda economica por parte de instituciones gubernamentales enfocadas a empresas y entidades. Esta creciente necesidad hace imprescindible contar con sistemas que optimicen la identificación de convocatorias de ayudas económicas.

Actualmente, la ausencia de herramientas automatizadas que interpreten y sinteticen la información disponible dificulta el acceso eficiente a estos recursos, obligando a las organizaciones a realizar búsquedas manuales que consumen tiempo y recursos. Este trabajo presenta el desarrollo de una herramienta basada en Inteligencia Artificial (IA) para la extracción y recuperación de información de convocatorias de ayudas económicas.

Utilizando técnicas avanzadas de Procesamiento de Lenguaje Natural (NLP) e IA Generativa, la solución permite analizar, estructurar y filtrar la información de manera automatizada, proporcionando resultados relevantes en función de las características específicas de cada entidad.

El objetivo principal de la herramienta es procesar y convertir las convocatorias de ayudas dispersas en un conjunto de datos estructurados, lo que facilita su consulta y recuperación. Esta estructura organizada permitirá a las empresas acceder de forma rápida y eficiente a la información más relevante, mejorando la toma de decisiones estratégicas.

De este modo, se aborda el desafío de filtrar y sintetizar grandes volúmenes de datos no estructurados y dispersos de diversas plataformas, simplificando el proceso de búsqueda y optimizando el acceso a oportunidades de financiamiento.

Palabras clave: LLMs, IA, RAG

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

1. Descripción general del problema

En la actualidad, abundan las ofertas de financiación y apoyo económico promovidas por organismos tanto públicos como privados. Sin embargo, las organizaciones empresariales encuentran dificultades para determinar qué oportunidades realmente se ajustan a su perfil específico. El gran caudal de datos y su heterogeneidad crean un panorama confuso, agravado por la carencia de sistemas automatizados que faciliten una búsqueda eficiente.

Entre las dificultades no solo se encuentra el hallazgo de convocatorias apropiadas, sino también entender la documentación requerida, los cronogramas de presentación, y determinar si estas ayudas son realmente aplicables al contexto empresarial particular. Adicionalmente, las compañías deben gestionar información fragmentada y frecuentemente desorganizada distribuida en múltiples fuentes digitales, lo que incrementa la complejidad del proceso de filtrado y selección.

Este proyecto plantea una solución que agilize, simplifique y optimice este proceso de búsqueda de financiación. Mediante la combinación de tecnologías como Inteligencia Artificial Generativa [32], Procesamiento de Lenguaje Natural (NLP) [15] y métodos de extracción de datos web o scraping [22], es posible extraer información precisa, relevante y estructurada sobre la documentación de estas convocatorias.

La importancia de esta solución se encuentra en su potencial para reducir tiempos y aumentar la efectividad al elegir opciones de financiamiento adecuadas. El hecho de implementar una solución como esta influirá positivamente en las tasas de éxito de las solicitudes presentadas y en la consecución de recursos económicos. El propósito de esta solución es democratizar el acceso a oportunidades financieras, fomentando condiciones más favorables para el desarrollo y la continuidad de las empresas.

2. Explicación de la motivación personal

La motivación tras este proyecto nace del interés personal sobre el uso de las nuevas tecnologías, especialmente la Inteligencia Artificial, en aspectos de la vida, tanto personales como profesionales, donde pueden suponer un cambio importante en la forma de realizar ciertas tareas: Agilizando procesos, reduciendo la dificultad en algunos casos y, en resumen, facilitar y hacer más accesible ciertos aspectos de la vida personal y profesional que pueden resultar tediosos.

El uso de tecnologías como el Procesamiento de Lenguaje Natural y la Inteligencia Artificial Generativa están cambiando desde hace unos años nuestra tecnología a un ritmo nunca visto. Prácticamente cada pocos meses aparecen nuevas tecnologías basadas en este campo de conocimiento, pasando por nuevos Grandes Modelos del Lenguaje como GPT-4o o DeepSeek, nuevas herramientas de desarrollo como Langchain, LLamaIndex, u Ollama, e incluso aplicaciones basadas en IA como Cursor, NotebookLM, o diferentes aplicaciones que te permiten generar texto, imágenes o música sin ser un experto.

Todas estas tecnologías están cambiando la forma en la que vemos el mundo, y aunque es necesario cierto control y regulación para no acabar en unos años en una sociedad distópica digna de la ciencia ficción, sí que considero que tenemos que aprovechar el potencial de estas tecnologías para seguir el camino hacia una sociedad más justa, equitativa, y donde tenga más peso la calidad de nuestras vidas y los derechos sociales, que las obligaciones económicas y laborales que marcan nuestro día a día.

En este caso concreto del proyecto, el uso de estas técnicas permite democratizar y hacer más accesible este tipo de ayudas. Crear una empresa y mantenerla a flote no es fácil, y en muchos casos sólo unas pocas sobreviven más de unos pocos años tras su creación, generalmente porque parten de unas capacidades económicas por detrás que no disponen el resto. Este tipo de ayudas económicas permiten a empresas con menos recursos de partida salir adelante, y herramientas como estas facilitan la búsqueda y su participación en éstas.

3. Definición de los objetivos

3.1. Objetivo Principal

El objetivo principal de este proyecto es el desarrollo de una solución automática basada en Inteligencia Artificial Generativa, que permita la indexación de información a partir de unas fuentes de datos concretas, en este caso convocatorias de ayudas a empresas, y realice tareas de extracción y estructuración de la información. De esta forma, se generará un barrido de todas las posibles convocatorias y se generará una base de datos con información relevante para su consulta y explotación.

3.2. Objetivos Específicos

- **Implementación de una herramienta de extracción de información**:
  En primer lugar se diseñará una herramienta que sea capaz de identificar las diferentes convocatorias de ayudas a partir de las fuentes disponibles y extraer la información necesaria:

  - Codigo fuente de la página web de convocatorias.
  - Ficha técnica de las convocatorias.
  - Documentos asociados.

  Esta herramienta será una combinación de soluciones basadas en Inteligencia Artificial Generativa y Web Scraping.

- **Sistema NLP de extracción y procesado de información**:
  Una vez extraída la información de la convocatoria, se emplearán diferentes técnicas de Procesamiento de Lenguaje Natural para diferentes tareas de procesado de texto y extracción de información. Este sistema empleará Grandes Modelos del Lenguaje (LLMs) en combinación con diferentes frameworks de orquestación, como Langchain o LLamaIndex. Se emplearán diferentes propuestas de LLMs para evaluar su eficacia en la extracción de información.

- **Herramienta de consulta de información**:
  Finalmente, se implementará una solución de consulta y recuperación de información sobre las diferentes convocatorias, partiendo tanto de los datos estructurados generados en el paso anterior como de las fuentes originales de datos. Esta solución se basará en un RAG multiagente, empleando técnicas avanzadas en cuanto a Question Answering y procesamiento de texto.

4. Descripción de la metodología empleada en el desarrollo del proyecto

En este proyecto, se ha optado por implementar la metodología Agile debido a su enfoque iterativo y flexible, lo que nos permitirá adaptarnos rápidamente a cambios en los requisitos y mejorar continuamente el resultado a través de entregas incrementales. Agile fomenta la colaboración, la comunicación y la retroalimentación continua, asegurando que el desarrollo se mantenga alineado con las necesidades del proyecto. Además, esta metodología promueve la eficiencia y la optimización del tiempo, reduciendo riesgos y mejorando la calidad del resultado final.

- **Estrategia de investigación**
  La estrategia de investigación sigue el enfoque propuesto por Oates en su libro *Researching Information Systems and Computing* [24], combinando técnicas de análisis de datos cualitativos y cuantitativos para asegurar una comprensión holística del dominio. Durante el proyecto, se aplicarán estrategias presentadas en el enfoque anterior para la obtención de datos de las convocatorias. Las principales tecnologías empleadas en el desarrollo serán Python como lenguaje de programación, frameworks como Langchain, LLamaIndex o HuggingFace, y diferentes librerías de NLP y Web Scraping, así como diferentes Grandes Modelos del Lenguaje, algunos explotados desde su propia API, y otros desde orquestadores locales, como Ollama o LMStudio.

- **Fases del desarrollo**
  El proyecto se dividirá en diferentes fases, cada una de las cuales se centrará en una tarea específica.

  - **Fase de investigación**: Revisión y análisis des las fuentes de datos disponibles, que en este casos son las diferentes webs proporcionadas de convocatorias de ayudas. A partir de los datos disponibles en éstas, se podrán definir los requisitos y funcionalidades que tiene que tener el sistema en cuanto a ectracción y procesado de información.
  - **Desarrollo de la herramienta de identificación y extracción de convocatorias**: En esta fase se desarrollará la herramienta de extracción de datos de convocatorias. Esta herramienta empleará una combinación de web scraping e IA Generativa para acceder a los sitios web de las ayudas, identificar las diferentes convocatorias y extraer las fuentes de datos en formato textual.

- **Desarrollo del sistema de procesamiento**: Una vez extraída la información, se desarrollará un sistema de procesamiento basado en IA Generativa que permita la identificación y extracción de información relevante de las convocatorias. Por un lado, extraerá datos en formato de texto web, que será procesado para cer accesible mediante Grandes Modelos del Lenguaje. Por otro lado, también procesará los ficheros PDF generalmente asociados a estas convocatorias. como resultado, se generará una base de datos con información estructurada, y una base de datos vectorial que almacenará los embeddings de los documentos.
- **Desarrollo del sistema de consulta de información**: A partir de las bases de datos generadas en el paso anterior, se construirá un sistema basado en RAG que permita acceder a esas fuentes y realizar consultas sobre los datos. Esas consultas permitirán extraer información estructurada en el formato solicitado.

5. Planificación o plan de investigación del proyecto

El plan de desarrollo de este proyecto va marcado por las diferentes etapas que establece la metodología de la UOC. En cada una de esos bloques de trabajo se abordarán las diferentes etapas indicadas del proyecto:

- **19/02/2025 al 09/03/2025**: Definición del TFM: Enunciado y entrega (M1). Definición de los requisitos del proyecto, análisis de fuentes de datos y tecnologías disponibles.
- **10/03/2025 al 30/03/2025**: Estado del Arte: Enunciado y entrega de la actividad (M2). En este bloque de trabajo se redactará el capítulo del Estado del Arte, en base a un trabajo de investigación donde se recopilarán las herramientas y tecnologías con potencial de ser empleadas en el proyecto. Este capítulo principalmente recopilará los últimos avances en Inteligencia Artificial Generativa, Grandes Modelos del Lenguaje, y frameworks asociados. Paralelamente comenzará el desarrollo de la herramienta de identificación y extracción de convocatorias.
- **31/03/2025 al 04/05/2025**: Implementación: Enunciado y entrega de la actividad (M3). Implementación de los diferentes bloques ya definidos: Herramienta de identificación de convocatorias, Sistema de procesamiento y la solución de consulta de información.
- **05/05/2025 al 18/05/2025**: Redacción de la memoria: Entrega preliminar (M4).
- **19/05/2025 al 25/05/2025**: Redacción de la memoria: Entrega final (M4).
- **26/05/2025 al 03/06/2025**: Presentación audiovisual del trabajo (M4).
- **04/06/2025 al 06/06/2025**: Entrega de la documentación al tribunal (M5).
- **07/06/2025 al 27/06/2025**: Defensa pública del trabajo (M5).

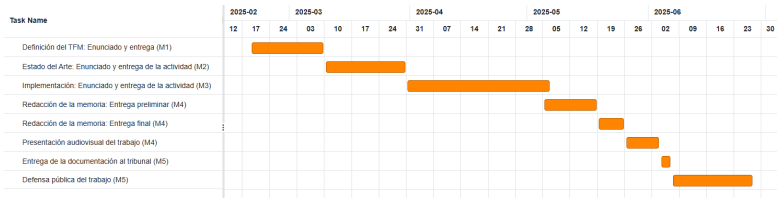

Figura 1.1: Timeline de tareas

Capítulo 2

Estado del Arte

1. Introducción

El objetivo de este capítulo es realizar un análisis de los diferentes avances, desarrollos y tecnologías disponibles en el ámbito de la solución planteada.

Este análisis tiene como objetivo identificar enfoques y metodologías en distintas áreas, como la extracción de información a partir de fuentes web, el análisis y procesado de texto y el uso de técnicas de Inteligencia Artificial aplicadas al Procesamiento de Lenguaje Natural.

De esta forma se puede establecer un contexto para el problema a resolver y justificar la elección de las tecnologías y metodologías a utilizar en el desarrollo de la solución propuesta.

2. Problemática a resolver

La búsqueda de ayudas y subvenciones es una tarea que la mayoría de las empresas, sobre todo las que tienen menos recursos, realizan en su día a día. Para ello, existen diferentes plataformas de ayudas a empresas, algunas nacionales y otras de carácter internacional. Sin embargo, esta tarea puede resultar complicada y tediosa, ya que implica una búsqueda constante de nuevas posibilidades de financiacióna través de distintas fuentes. Además, la información sobre estas convocatorias suele estar distribuidas en diferentes fuentes, desde las propias plataformas a documentación oficial del estado. Esto supone que a la hora de realizar una búsqueda de posibles convocatorias de financiación, se acabe con un conjunto de fuentes con diferentes estructuras y formatos.

En la mayoría de los casos, las convocatorias suelen tener asociados diferentes documentos, en su mayoría en formato PDF, los cuales pueden ser extensos, y además usan un lenguaje técnico, típico de este tipo de documentos, que dificulta su comprensión. Esto al final supone una complicación por parte de las empresas a la hora de acceder a información clave de las

convocatorias de forma mas rápida, como requisitos, plazos, presupuesto o condiciones de participación. Estos problemas de accesibilidad y estandarización de las convocatorias de ayudas suponen una barrera de acceso importante, que reduce las oportunidades de acceso a financiación para algunas empresas, y suponen una inversión en tiempo y esfuerzo en la tarea de búsqueda y filtrado por parte de éstas.

El desarrollo planteado en este proyecto pretende ser una solución a esta problemática, proporcionando una herramienta que sea capaz de identificar y extraer la documentación de las convocatorias, y aplicar técnicas de Inteligencia Artificial para extraer la información clave y dotarla de una estructura mas estandarizada, así como permitir la consulta de esta información de forma sencilla a partir de un agente conversacional.

3. Soluciones disponibles

Dejando a parte de momento las soluciones basadas en Inteligencia Artificial, las cuales se comentarán en secciones posteriores, existen diferentes metodologías para la búsqueda de información sobre estas convocatorias:

3.1. Plataformas de convocatorias

Existen diferentes plataformas que recopilan información sobre convocatorias de ayudas y subvenciones, a las cuales las empresas pueden acceder para explorar las diferentes opciones de convocatorias, y valorar si se ajustan a su situación. Estos portales de convocatorias suelen estar disponibles en plataformas tanto gubernamentales como privadas, que recopilan y organizan la información sobre diferentes ayudas disponibles. Además, estas herramientas suelen permitir aplicar filtros en las búsquedas por diferentes características, como el área geográfica, el perfil de la empresa solicitante, o el tipo de ayuda.

- **Portales de ayudas gubernamentales**:
  Las diferentes instituciones públicas suelen ofrecer portales informativos donde publican este tipo d econvocatorias, ya sean a nivel local, regional o nacional. Estos portales permiten visualizar estas voncovatorias, pero a un nivel básico en cuanto a experiencia de usuario, y aunque la totalidad de la información siempre está disponible, ya sea en el propio portal o mediante enlaces a diferentes fuentes documentales, el análisis y búsqueda de información clave es tediosa y lenta.

  - CDTI: Centro para el Desarrollo Tecnológico y la Innovación[2].
  - Grupo SPRI[4].

- SODERCAN: Sociedad para el desarrollo ragional de Cantabria[7].
- Portal de ayudas del Ministerio para la Transofrmación Digital y de la Función Pública[6].
- Andalucía Trade: Incentivos para Desarrollo Induatrial y Proyectos de I+D+i Empresarial[1].

- **Plataformas privadas de información sobre subvenciones**:
  Algunas empresas recopilan información sobre ayudas a empresas, las estructuran en bases de datos y ofrecen el acceso a esta información como servicio, garantizando en éste la calidad de la información y una actualización constante del listado de ayudas disponibles. El inconveniente de estas plataformas es que, pese a ofrecer servicios de búsqueda que suelen tener interfaces mas amigables e información mas directa, suelen ser herramientas de pago, y el acceso completo a la información puede supoer un coste económico adicional. Algunos ejemplos de estas plataformas son Fandit [3] u OpenGrants [5].

3.2. Web Scraping

Una solución alternativa a la búsqueda manual en portales de ayudas es el uso de herramientas de Web Scraping. El Web Scraping [19] es una técnica utilizada para extraer información de sitios web de manera automatizada. Consiste en el uso de programas o scripts que navegan por páginas web, recopilan datos estructurados y los almacenan en un formato más accesible, como bases de datos o archivos locales JSON o CSV, por ejemplo. Esta práctica es ampliamente utilizada en diversos sectores para la recopilación y análisis de información a gran escala.

Generalmente el proceso de Web Scraping se desarrolla en varias etapas:

- Solicitud HTTP: La herramienta de scraping envía una solitcitud HTTP a una página web para obtener su contenido.
- Extracción de datos: Se analiza el código fuente de la página, y según la configuración establecida en el scraper, se extraen los datos requeridos mediante comandos de parseo propios de la herramienta, expresiones regulares, o bots de navegación automatizada.
- Almacenamiento de la información: Una vez obtenidos los datos, estos se pueden formatear y almacenar según convenga en el caso de uso.

Existen diferentes metodologías de Web Scraping:

- **Análisis HTML**: En múltiples sitios web, se generan automáticamente grandes volúmenes de páginas a partir de fuentes de datos estructuradas, como bases de datos, mediante

scripts o plantillas que organizan la información en formatos homogéneos. En minería de datos, un wrapper es un programa que identifica plantillas en una fuente de datos, extrae su contenido y lo transforma en una estructura relacional. La inducción de wrappers asume que las páginas de entrada siguen un patrón identificable, usualmente a través de formatos de URL comunes. Además, lenguajes de consulta para datos semiestructurados, como XQuery y HTQL, permiten analizar, extraer y modificar información en sitios web HTML [9].

- **Análisis DOM**: Los programas pueden acceder a contenido dinámico generado por scripts del lado del cliente mediante la integración de un navegador web, como Internet Explorer o Mozilla browser control [11]. Estas aplicaciones analizan las páginas web y las estructuran en un árbol del Document Object Model (DOM), lo que permite extraer secciones específicas del contenido. El modelo DOM organiza una página web en una estructura arbórea, permitiendo su interpretación y almacenamiento a partir de una dirección web especificada, como ocurre en los motores de búsqueda. Este enfoque ofrece gran flexibilidad y agilidad, ya que permite rastrear elementos presentes en la página sin depender de que el equipo de desarrollo web los exponga explícitamente en la capa de datos.

- **HTML DOM (Hyper Text Markup Language Document Object Model)**: Es un estándar para la obtención, manipulación y modificación de elementos HTML [13]. Define objetos y propiedades para cada componente HTML, así como métodos para acceder a ellos, optimizando la eficiencia del DOM. JavaScript, como lenguaje principal, permite acceder y manipular todos los elementos de un documento HTML a través del DOM. En este modelo, cada elemento HTML se trata como un objeto, cuya interfaz de programación está compuesta por métodos y propiedades específicas.

- **Expresiones regulares (RegEx)**: Las expresiones regulares son fórmulas que definen patrones específicos para identificar conjuntos de caracteres en diversas cadenas de texto [23]. Se componen de caracteres ordinarios y metacaracteres, los cuales modifican la interpretación del patrón. Aunque su sintaxis puede parecer compleja, las expresiones regulares son una herramienta esencial para el análisis y procesamiento de datos en cadenas de texto, por lo que es fundamental comprenderlas al menos a nivel básico.

- **XPath**: XPath es el componente principal del estándar XSLT (Stylesheet Language Transformation) y se utiliza para navegar y seleccionar elementos y atributos dentro de documentos XML [10]. Además, puede aplicarse en documentos HTML. XPath funciona como un lenguaje de selección de nodos en estructuras XML, siendo la expresión

más utilizada la ruta de ubicación (location path). Esta ruta emplea al menos un paso de ubicación para identificar un conjunto de nodos dentro de un documento. La forma más simple es la selección del nodo raíz del documento, representada por el símbolo /", que también es el indicador del directorio raíz en sistemas de archivos Unix.

- **Reconocimiento de anotaciones semánticas**: Las páginas extraídas pueden incluir metadatos, marcas semánticas y anotaciones que permiten identificar datos específicos [19]. Por ejemplo, esta técnica puede considerarse un caso particular del análisis DOM si las anotaciones están integradas en las páginas, como ocurre con Microformat. En otro caso, las anotaciones se almacenan y gestionan de manera independiente de las páginas web, organizándose en una capa semántica, de modo que los scrapers pueden obtener el esquema e instrucciones desde esta capa antes de realizar el raspado de las páginas.

En el ámbito del Web Scraping, específicamente con el lenguaje de Programación Python, podemos encontrar las siguientes librerías:

- **BeautyfulSoup**: BeautifulSoup es una biblioteca de Python diseñada para el análisis, extracción y manipulación de datos en documentos HTML y XML. Su funcionamiento se basa en la creación de un árbol de análisis sintáctico (parse tree), que estructura el contenido de la página web de manera jerárquica, permitiendo navegar por los nodos, buscar elementos específicos y modificar el contenido. BeautifulSoup admite múltiples analizadores (parsers), como lxml, html.parser y html5lib, cada uno con diferentes niveles de velocidad y compatibilidad. Su sintaxis flexible permite localizar elementos a través de etiquetas, atributos y selectores CSS, facilitando la extracción de datos estructurados de páginas web. Además, cuenta con métodos para limpiar el contenido, eliminar etiquetas HTML y exportar la información en diversos formatos [20].

- **Scrapy**: Scrapy es un framework de Python diseñado para la extracción estructurada de datos mediante web scraping y crawling. Su arquitectura modular permite gestionar solicitudes HTTP, procesar respuestas y almacenar datos de manera eficiente. Scrapy opera a través de un flujo de trabajo basado en spiders, que son clases definidas por el usuario encargadas de especificar la lógica de extracción [11].

  El motor de Scrapy (Scrapy Engine) coordina los componentes principales:

  - Scheduler, que organiza las solicitudes pendientes.
  - Downloader, que ejecuta las peticiones HTTP y recibe las respuestas.
  - Spiders, que analizan y extraen información relevante.

- Item Pipeline, que transforma, valida y almacena los datos obtenidos en formatos como JSON, CSV o bases de datos SQL y NoSQL.

Además, Scrapy admite el uso de middlewares, tanto en el Downloader como en el Spider, para modificar solicitudes y respuestas, gestionar sesiones y evitar bloqueos mediante técnicas como rotación de proxies y user agents. Su diseño asincrónico optimiza el rendimiento, permitiendo la extracción masiva de datos con alta eficiencia.

- **Selenium**: Selenium es un framework de automatización de navegadores de código abierto utilizado para la ejecución de pruebas y la extracción de datos mediante web scraping [21]. Su funcionamiento se basa en la interacción con páginas web a través de un WebDriver, que actúa como un controlador para manipular elementos de la interfaz de usuario, simular clics, completar formularios y desplazarse por el contenido dinámico generado mediante JavaScript. El ecosistema de Selenium está compuesto por varios módulos:

  - Selenium WebDriver, que permite la automatización de navegadores como Chrome, Firefox y Edge mediante controladores específicos.
  - Selenium Grid, que posibilita la ejecución distribuida de pruebas y scraping en múltiples máquinas.
  - Selenium IDE, una extensión que facilita la grabación y reproducción de secuencias de prueba en navegadores.

  Para realizar web scraping con Selenium, se inicia una sesión de navegador con el WebDriver, se navega a la URL objetivo, y se localizan los elementos deseados mediante selectores XPath o CSS. A diferencia de frameworks como Scrapy o BeautifulSoup, Selenium es ideal para interactuar con sitios que requieren ejecución de JavaScript o carga dinámica de contenido, aunque su rendimiento puede ser inferior debido a la sobrecarga computacional del manejo de un navegador real.

3.3. Procesamiento de Lenguaje Natural

El Procesamiento de Lenguaje Natural (NLP) es una disciplina de la inteligencia artificial y la lingüística computacional que permite a las máquinas interpretar, comprender, generar y manipular el lenguaje humano de manera estructurada [16]. Su aplicación abarca desde la traducción automática y el análisis de sentimientos hasta la generación de texto y los asistentes virtuales.

El Procesamiento de Lenguaje Natural emplea diversas técnicas computacionales para la interpretación y manipulación del lenguaje humano. Estas técnicas pueden dividirse en métodos

estadísticos, basados en reglas y de aprendizaje profundo, siendo ampliamente utilizadas en tareas como la clasificación de textos, el análisis de sentimientos y la traducción automática.

- **Preprocesamiento de Texto**

  - **Tokenización**: La tokenización es un proceso esencial en el Procesamiento de Lenguaje Natural que divide un texto en unidades denominadas tokens. Existen varios tipos [30], siendo los principales: tokenización por palabras, que separa el texto en palabras individuales y es efectiva en idiomas con delimitadores claros como el inglés; tokenización por subpalabras, utilizada en modelos como BERT y GPT, que emplea técnicas como Byte Pair Encoding (BPE) para manejar vocabularios extensos y palabras fuera de vocabulario (OOV); tokenización por oraciones, que segmenta el texto en unidades sintácticas más grandes basándose en puntuación y reglas lingüísticas; y tokenización por caracteres, útil en modelos de aprendizaje profundo y generación de texto. La selección del método adecuado depende del idioma y la tarea específica, siendo crucial en aplicaciones como traducción automática, análisis de sentimientos y recuperación de información.
  - **Lematización y Stemming**: Son técnicas empleadas para normalizar palabras reduciéndolas a su raíz morfológica [17]. El stemming elimina afijos mediante reglas predefinidas, sin considerar el contexto, lo que lo hace rápido pero propenso a errores (ejemplo: running"→ run", pero "better"→ "bet"). En contraste, la lematización utiliza análisis morfológico y diccionarios lingüísticos para obtener la forma base correcta (ejemplo: "better"→ "good"), ofreciendo mayor precisión. Mientras que el stemming es más eficiente en grandes volúmenes de datos, la lematización es preferida en tareas como análisis de sentimientos, recuperación de información y traducción automática, donde la precisión semántica es crucial.
  - **Eliminación de stopwords**: Las stopwords son palabras de alta frecuencia en un idioma que generalmente no aportan significado relevante en tareas de Procesamiento de Lenguaje Natural, como artículos, preposiciones y pronombres (ejemplo: .$^{\text{el}}$", "de", z", "pero"). Se eliminan para reducir la dimensionalidad del texto y mejorar la eficiencia de los modelos de análisis de texto [29]. Bibliotecas como NLTK, SpaCy y Scikit-learn incluyen listas de stopwords predefinidas, aunque pueden personalizarse según la aplicación. Si bien su eliminación es útil en tareas como búsqueda de información y clasificación de textos, en ciertos casos, como en análisis de sentimientos o generación de texto, pueden ser necesarias para preservar el contexto semántico.

- **Análisis Morfosintáctico y Semántico**

- **Etiquetado gramatical (POS Tagging)**: Es una técnica del Procesamiento de Lenguaje Natural que asigna a cada palabra de un texto su categoría gramatical correspondiente (sustantivo, verbo, adjetivo, etc.) en función de su contexto [18]. Se basa en reglas lingüísticas, modelos estadísticos o redes neuronales para mejorar la precisión del análisis. Herramientas como NLTK, SpaCy y Stanford NLP utilizan algoritmos como Hidden Markov Models (HMM) o Redes Neuronales Recurrentes (RNNs) para realizar esta tarea. El POS Tagging es clave en aplicaciones como análisis de sentimientos, desambiguación semántica y traducción automática, ya que ayuda a comprender la estructura y significado del lenguaje.
- **Parsing sintáctico**: El parsing sintáctico o análisis sintáctico consiste en la construcción de la estructura jerárquica de una oración para comprender su sintaxis, identificando la relación jerárquica entre las palabras mediante árboles sintácticos o dependencias gramaticales [36]. Existen dos enfoques principales: el parsing basado en constituyentes, que descompone la oración en frases (sintagmas nominales, verbales, etc.), y el parsing basado en dependencias, que representa las relaciones entre palabras mediante un grafo dirigido. Herramientas como NLTK, SpaCy y Stanford Parser emplean algoritmos como CYK, Earley o modelos de redes neuronales para esta tarea. El parsing sintáctico es esencial en aplicaciones como traducción automática, generación de texto y comprensión del lenguaje natural, donde la estructura de la oración influye en su interpretación.
- **Reconocimiento de Entidades Nombradas (NER)**: Es una técnica que tiene como objetivo identificar y clasificar entidades dentro de un texto, como nombres de personas, lugares, organizaciones, fechas, entre otros [27]. Utiliza enfoques basados en reglas lingüísticas, modelos estadísticos o aprendizaje automático para detectar estas entidades en el contexto del texto. Herramientas como SpaCy, NLTK y Stanford NER emplean modelos entrenados en grandes corpus de datos para reconocer entidades y asignarles una etiqueta adecuada. El NER es crucial en aplicaciones como extracción de información, análisis de noticias, y búsqueda semántica, ya que facilita la identificación y categorización de información relevante dentro de grandes volúmenes de datos.

▪ **Representación de Texto**

- **Bag of Words (BoW)**: Es una técnica de representación de texto que convierte éste en una matriz de características, donde cada documento se representa como un conjunto de palabras sin tener en cuenta el orden o la gramática [25]. En el modelo BoW, cada palabra única en el corpus se convierte en una característica

(o columna) y cada documento se representa como un vector en el que el valor de cada entrada corresponde a la frecuencia de la palabra en ese documento. Aunque es simple y eficiente, BoW presenta varios inconvenientes, como la pérdida de contexto y el orden de las palabras, la alta dimensionalidad, la falta de captura de relaciones semánticas y la presencia de ruido en los datos, lo que puede afectar la precisión y la interpretación del modelo. Esta técnica, sin embargo, sigue siendo útil en tareas como clasificación de texto, análisis de sentimientos y recuperación de información, pero sus limitaciones han llevado al desarrollo de enfoques más avanzados.

- **TF-IDF (Term Frequency - Inverse Document Frequency)**: Método que pondera términos relevantes dentro de un corpus [28]. Es una técnica de representación de texto que asigna un peso a cada término de un documento, basándose en dos componentes: la frecuencia de término (TF), que mide cuántas veces aparece un término en un documento, y la frecuencia inversa de documento (IDF), que mide la importancia de un término dentro de un conjunto de documentos. El cálculo de TF es sencillo y se basa en la cantidad de veces que un término aparece en un documento en comparación con el total de términos del documento, mientras que IDF ajusta el peso del término según su frecuencia en todos los documentos, penalizando las palabras comunes que aparecen en muchos documentos. El resultado es un valor que refleja la relevancia de un término en un documento en particular dentro de un corpus. TF-IDF es ampliamente utilizado en tareas como clasificación de texto, búsqueda de información y análisis de contenido, ya que ayuda a identificar términos significativos que son relevantes en el contexto de un conjunto de documentos. Sin embargo, también presenta algunas limitaciones, como la incapacidad para capturar relaciones semánticas entre palabras y su dependencia de la estructura del corpus.

- **Word Embeddings**: Técnicas avanzadas de representación de palabras que convierte las palabras en vectores numéricos de alta dimensión [8]. A diferencia de las representaciones tradicionales como Bag of Words o TF-IDF, que tratan las palabras de manera independiente, los word embeddings capturan las relaciones semánticas y contextuales entre palabras, representando palabras similares en espacios vectoriales cercanos. Modelos como Word2Vec, GloVe, FastText o los posteriores modelos de Embeddings asociados a Grandes Modelos del Lenguaje aprenden estas representaciones mediante redes neuronales entrenadas sobre grandes corpus de texto, aprovechando el contexto de las palabras en las oraciones para generar sus vectores. Los word embeddings permiten capturar propiedades lingüísticas, como sinónimos, analogías y jerarquías semánticas, mejorando el rendimiento en tareas como traducción automática, análisis de sentimientos, clasificación de texto y respuestas

automáticas. Aunque potentes, los embeddings también presentan desafíos, como la dificultad para representar términos poco frecuentes o palabras con múltiples significados (polisemia).

- **Modelos de Aprendizaje Automático y Profundo**

  - **Modelos basados en aprendizaje automático clásico**: Utilización de algoritmos tanto supervisados como no supervisados, como regresiones logísticas, árboles de decisión, SVM, k-means o Random Forest, entre otros, para tareas de clasificación o clusterización de texto [14]. Estos modelos estan diseñados para trabajar con datos numéricos, por lo que es necesario aplicar técnicas de representación numérica de texto como las ya anteriormente comentadas.

  - **Redes Neuronales Recurrentes (RNN) y LSTM**: Modelos diseñados para procesar secuencias de texto y capturar dependencias contextuales. Son arquitecturas de Deep Learning diseñadas para procesar datos secuenciales, lo que las hace especialmente útiles en Procesamiento de Lenguaje Natural. A diferencia de los modelos tradicionales de Machine Learning, las RNN pueden capturar dependencias temporales en el texto, ya que su estructura permite que la información de estados previos influya en la interpretación de los siguientes [31]. Sin embargo, las RNN convencionales presentan problemas con secuencias largas debido a la desaparición o explosión del gradiente. Para solucionar esto, surgieron las LSTM, que incorporan puertas de memoria que regulan el flujo de información, permitiendo recordar dependencias a largo plazo de manera más eficiente [33]. Estas redes han sido ampliamente utilizadas en tareas como traducción automática, generación de texto, análisis de sentimientos y reconocimiento de voz. Aunque las LSTM han mejorado el manejo de secuencias largas, han sido en gran parte reemplazadas por arquitecturas más avanzadas como Transformers, que manejan contexto de manera más eficiente mediante mecanismos de atención.

  - **Arquitecturas de Deep Learning basadas en Transformers**: Estas nuevas arquitecturas revolucionaron el Procesamiento de Lenguaje Natural al superar las limitaciones de las arquitecturas de Redes Neuronales anteriores, gracias a su capacidad para procesar secuencias en paralelo y capturar relaciones a largo plazo mediante el mecanismo de atención. Introducidos en el paper .ªttention Is All You Need"[34], los Transformers utilizan mecanismos de self-attention para asignar pesos a cada palabra en función de su relevancia dentro del contexto, lo que permite una comprensión más profunda del significado del texto. Modelos como BERT (Bidirectional Encoder Representations from Transformers [12]), GPT (Generative Pre-trained Transformer

[35]) y T5 (Text-to-Text Transfer Transformer [26]) lograron en su momento avances significativos en tareas como traducción automática, generación de texto, análisis de sentimientos y respuesta a preguntas. Estas arquitecturas destacan por su capacidad de preentrenamiento en grandes corpus de datos y posterior ajuste fino en tareas específicas, lo que ha permitido obtener resultados de vanguardia en múltiples aplicaciones de PLN. Tomando como base estas arquitecturas, han aparecido en los últimos años los Grandes Modelos del Lenguaje (LLMs), dando origen al desarrollo de la llamada Inteligencia Artificial Generativa.

Estas técnicas han permitido avances en aplicaciones como asistentes virtuales, generación de texto automatizada y motores de búsqueda, optimizando la interacción entre humanos y sistemas computacionales.

4. Inteligencia Artificial Generativa

Bibliografía

[1] Andalucía trade, https://www.andaluciatrade.es/financiacion-empresarial/incentivos-para-las-empresas/.

[2] Centro para el desarrollo tecnológico y la innovación, https://www.cdti.es/.

[3] Fandit, https://fandit.es/.

[4] Grupo spri, https://www.spri.eus/es.

[5] Opengrants/opengrants.io.

[6] Portal de ayudas del ministerio para la transofrmación digital y de la función pública, https://portalayudas.digital.gob.es/paginas/convocatorias-ayudas.aspx.

[7] Sociedad para el desarrollo ragional de cantabria, https://ayudas.sodercan.es/ayudas.

[8] Felipe Almeida and Geraldo Xexéo. Word embeddings: A survey, 2023.

[9] M. El Asikri, S. Krit, H. Chaib, M. Kabrane, H. Ouadani, K. Karimi, K. Bendaouad, and H. Elbousty. Mining the web for learning ontologies: State of art and critical review. pages 1–7, 2017.

[10] M. El Asikri, S. Krit, H. Chaib, M. Kabrane, H. Ouadani, K. Karimi, K. Bendaouad, and H. Elbousty. Mining the web for learning ontologies: State of art and critical review. pages 1–7, 2017.

[11] M Asikri[1], S Krit, Hassan Chaib, and Krit Salah-ddine. Using web scraping in a knowledge environment to build ontologies using python and scrapy. *European Journal of Translational and Clinical Medicine*, 7:433–442, 10 2020.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

[13] Rohmat Gunawan, Alam Rahmatulloh, Irfan Darmawan, and Firman Firdaus. Comparison of web scraping techniques : Regular expression, html dom and xpath. pages 283–287, 2019/03.

[14] Emmanouil Ikonomakis, Sotiris Kotsiantis, and V. Tampakas. Text classification using machine learning techniques. *WSEAS transactions on computers*, 4:966–974, 08 2005.

[15] Diksha Khurana, Aditya Koli, Kiran Khatter, and Sukhdev Singh. Natural language processing: state of the art, current trends and challenges. *Multimedia Tools and Applications*, 82(3):3713–3744, July 2022.

[16] Koli Khurana, K Khatter, et al. Natural language processing: State of the art, current trends and challenges. 2023. Available online.

[17] Divya Khyani, Siddhartha B S, N. Niveditha, Divya M., and Dr Y M. An interpretation of lemmatization and stemming in natural language processing. *Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology*, 22:350–357, 01 2021.

[18] Deepika Kumawat and Vinesh Jain. Pos tagging approaches: A comparison. *International Journal of Computer Applications*, 118:32–38, 05 2015.

[19] Chaimaa Lotfi, Swetha Srinivasan, Myriam Ertz, and Imen Latrous. *Web Scraping Techniques and Applications: A Literature Review*, pages 381–394. 01 2021.

[20] Stephanie Lunn, Jia Zhu, and Monique Ross. Utilizing web scraping and natural language processing to better inform pedagogical practice. pages 1–9, 2020.

[21] K Usha Manjari, Syed Rousha, Dasi Sumanth, and J Sirisha Devi. Extractive text summarization from web pages using selenium and tf-idf algorithm. pages 648–652, 2020.

[22] Laia Subirats Maté and Mireia Calvo González. Web scraping. *Editorial UOC.*, 2019.

[23] R. (2018) Mitchell. Web scraping with python: Collecting more data from the modern web. o'reilly media, inc., 2018.

[24] Briony June Oates. Researching information systems and computing. 2005.

[25] Wisam Qader, Musa M. Ameen, and Bilal Ahmed. An overview of bag of words;importance, implementation, applications, and challenges. pages 200–204, 06 2019.

[26] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[27] Arya Roy. Recent trends in named entity recognition (ner), 2021.

[28] Claude Sammut and Geoffrey I. Webb, editors. *TF–IDF*, pages 986–987. Springer US, Boston, MA, 2010.

[29] Serhad Sarica and Jianxi Luo. Stopwords in technical language processing. *PLOS ONE*, 16(8):e0254937, August 2021.

[30] Craig W. Schmidt, Varshini Reddy, Haoran Zhang, Alec Alameddine, Omri Uzan, Yuval Pinter, and Chris Tanner. Tokenization is more than compression, 2024.

[31] Robin M. Schmidt. Recurrent neural networks (rnns): A gentle introduction and overview, 2019.

[32] Sandeep Singh Sengar, Affan Bin Hasan, Sanjay Kumar, and Fiona Carroll. Generative artificial intelligence: A systematic review and applications, 2024.

[33] Ralf C. Staudemeyer and Eric Rothstein Morris. Understanding lstm – a tutorial into long short-term memory recurrent neural networks, 2019.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[35] Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions, 2023.

[36] Meishan Zhang. A survey of syntactic-semantic parsing based on constituent and dependency structures, 2020.