**Jerónimo Molina Molina**

# Procesamiento del Lenguaje Natural

## 8. Modelos grandes del lenguaje y R.A.G.

En realidad, este tema enlaza con el anterior, en el que, a través del último ejercicio propuesto, ya se entraba en el terreno de los LLM.

En este tema, para finalizar la introducción al procesamiento del lenguaje natural, abordaremos un enfoque de negocio de los modelos grandes del lenguaje, así como apuntaremos las nuevas tendencias y posibilidades "on Edge" de los modelos conversacionales.

Porque… se te ocurren posibilidades de negocio de los modelos generativos del lenguaje, ¿verdad?

Bueno, de momento centrémonos en texto (olvidemos la multimodalidad en input y en output) y pensemos en cómo gestionar conocimiento con independencia de lo estructurado que esté…

## Modelos grandes del lenguaje y el marco de orquestación Langchain

A veces, cuando chateamos con un modelo conversacional generativo, podemos tener la sensación de estar charlando con una persona o con un "ayudante inteligente", hasta podríamos pensar que tiene su propia personalidad, pero, aunque ya lo sepas, permíteme que te diga que nada de eso es cierto. Este es un tema recurrente y que tiene mucho que ver con las IA's fuertes o AGI's -que no se si llegaremos a crear, mi opinión personal es que son y serán ciencia ficción, veremos…-

Y es que, si de algo debemos estar seguros es de que un modelo generativo no es más que una especie de "papagallo" más o menos decente, es decir, nunca jamás debemos perder de vista que es simplemente un modelo matemático que, a partir de datos que tiene almacenados en formato de embedding, es capaz de vectorizar una entrada y compararla con la información de que dispone, para, posteriormente, expresarla en la salida a partir de una construcción realizada por un modelo generativo. Dicho de otro modo, no piensa, es decir:

- No tiene consciencia
- No tiene conciencia
- Simplemente emite respuestas, proporcionadas por un modelo generativo, y basadas en el conocimiento que tiene almacenado.
- No es más que un modelo estadístico que, genera su secuencia de salida (respuesta) infiriendo la probabilidad de que una palabra -token- sea "la siguiente".

Bueno, esto es una explicación muy muy resumida, entre otros detalles, hay que tener siempre en consideración, igualmente, el contexto.

Debe tenerse muy en cuenta que son modelos de propósito general, con una arquitectura tipo Transformers y entrenados con unas cantidades inmensas de información, al estilo "fuerza bruta" y de manera "no supervisada".

Para conocer la historia de los LLM, te recomiendo este más que interesante artículo de Juan Ignacio Bagnato. LLM: ¿Qué son los Grandes Modelos de Lenguaje? | Aprende Machine Learning

Los modelos grandes del lenguaje (LLM) son modelos de deep learning que se preentrenan con grandes cantidades de datos. El transformer subyacente es, como ya sabes, una arquitectura neuronal que consta de un codificador y un decodificador con capacidades de autoatención (Attention is all you need). De este modo, codificador y decodificador extraen significados

(vectorización) de una secuencia de texto y comprenden las relaciones entre las palabras y las frases que contiene.

Los transformer empleados en la generación de LLM's son capaces de entrenarse sin supervisión, aunque se podría afirmar que los transformers llevan a cabo un autoaprendizaje, proceso a través del cual aprenden a entender la gramática, los idiomas y los conocimientos básicos.

A diferencia de las redes neuronales recurrentes (RNN), que procesaban las entradas de forma secuencial, los transformers procesan secuencias enteras en paralelo, lo que permite el uso de GPU's para entrenar LLM basados en transformers.

Estos modelos a gran escala pueden incorporan cantidades ingentes de datos, a menudo de Internet, pero también de fuentes como Common Crawl, que comprende más de 50 000 millones de páginas web, y Wikipedia, que tiene aproximadamente 57 millones de páginas. Una persona no sería capaz de procesar tantísima información en su vida, ni tan siquiera leyendo a una velocidad de 15 palabras por segundo.

Te recomiendo la lectura de este artículo de Amazon, que te servirá para tener una visión general de los modelos grandes del lenguaje y sus aplicaciones: ¿Qué son los modelos de lenguaje de gran tamaño? - Explicación sobre los LLM de IA - AWS (amazon.com)


**Alucinaciones**

Es muy cierto lo que se cuenta por ahí sobre las “alucinaciones” de los modelos del lenguaje. Pero… ¿qué son y por qué pasan?

En los propios modelos, pueden influir muchos factores para que se produzcan las denominadas “alucinaciones”:

- Calidad de la información: No olvidemos que, las mas de las veces, es información “poco verificada oficialmente”.
- Debido al método de generación: Son siempre estadísticos, lo que ya, de por sí, incorpora de forma natural el margen de error, pero hay diferentes técnicas, tales como “Búsqueda en haz”, “Reinforcement learning”, MLE o Muestreo.

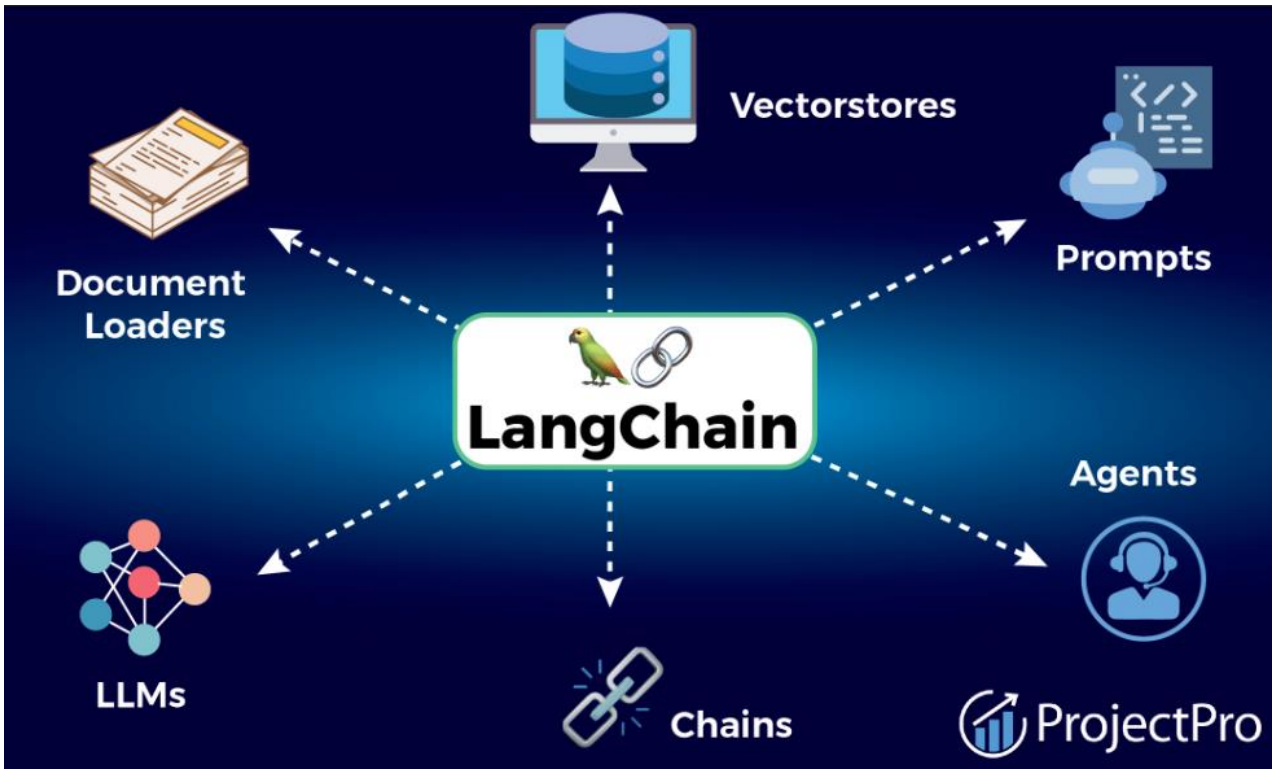
- El input o Prompt de entrada: dependiendo de la especifidad y calidad de la entrada, la respuesta podría variar mucho, o incluso inducir a confusión al modelo.

**Generar modelos LLM de negocio**

Tal y como se explica en el artículo de Bagnato, una alternativa para la generación de modelos responsables y con conocimiento específico, la primera vía es el "Fine Tunning", que pasa, inexorablemente por el retraining conforme crece la base de información.

Frente a esta situación, que genera diferentes problemas, han surgido vías alternativas. Una en concreto, que viene teniendo muchísimo éxito es el uso de una capa, por encima de los modelos del lenguaje, que permite integrar conocimiento, así como implementar diferentes artificios lógicos que implican a los modelos y permiten el diseño de casos de uso concretos. Me estoy refiriendo, en concreto, a una librería que se llama Langchain.

**Langchain**




Langchain es una capa de programación -librería- que permite explotar LLM's -incluso más de uno a la vez-, gestionando el flujo conversacional, integrándolos con fuentes de información y permitiendo la articulación de diferentes técnicas y artefactos, conceptos entre los que cabe destacar los siguientes:

- Cargadores de documentos: Permiten el procesamiento de diferentes datos (estructurados o no) en LangChain
- Cadenas: Son el principio básico en que se basan las acciones complejas a implementar, y que permiten la ejecución de tareas sencillas, enlazadas unas con otras, para que en secuencia, alcancen un objetivo específico.

- Agentes: Cadenas específicas, con acceso a herramientas externas, en las que se puede brindar al LLM tiene mayor control, para obtener respuestas más precisas y actualizadas.
- RAG: Retrieval Augmented Generation, en términos generales es una técnica que permite la optimización de la salida de un LLM.

Volviendo al tema de las alucinaciones, dentro del contexto Langchain, pensemos en un modulador de la "creatividad" de los modelos generativos, un control, ajustable de 0 a 100 ([0,1]) que sirva para otorgarle más o menos creatividad al modelo. Esto, que se le llama "temperatura" de un modelo, afecta no solo al proceso de búsqueda de contenidos relacionados, sino a la forma en que se genera la respuesta.

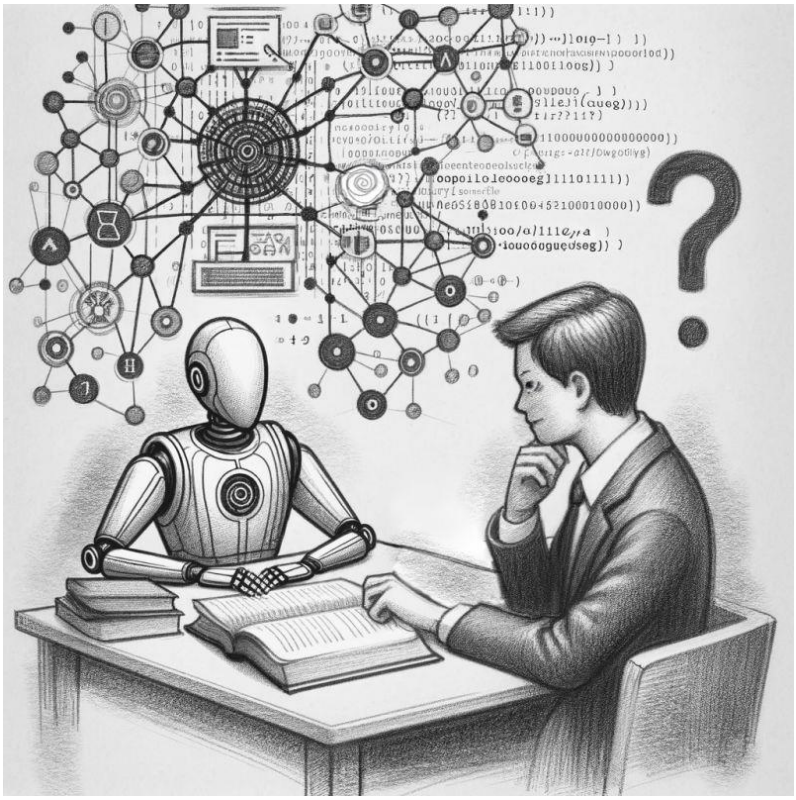
Esto permite, sin duda, gestionar la fiabilidad de las respuestas, dependiendo siempre del caso de uso en el que se trabaje, lo que, bien empleado, da mucho juego, pues no es lo mismo una herramienta de generación de contenido creativo que un gestor del conocimiento del ámbito científico.

Para profundizar en Langchain hay multitud de recursos, te recomiendo los siguientes:

- https://www.evoacademy.cl/curso-introduccion-a-langchain-tutorial-fundamentos-langchain-models-prompt-templates-chain/
- https://www.youtube.com/watch?v=RoR4XJw8wIc
- https://www.langchain.com/
- https://www.paradigmadigital.com/dev/que-es-langchain-como-crear-aplicaciones-python-libreria/
- https://www.linkedin.com/pulse/langchain-qu%C3%A9-espor-lo-necesitamos-c%C3%B3mo-funciona-caso-alvarez/?originalSubdomain=es

**RAG**

Consiste en complementar a los modelos generativos (sean cuales sean) con datos privados o propietarios.

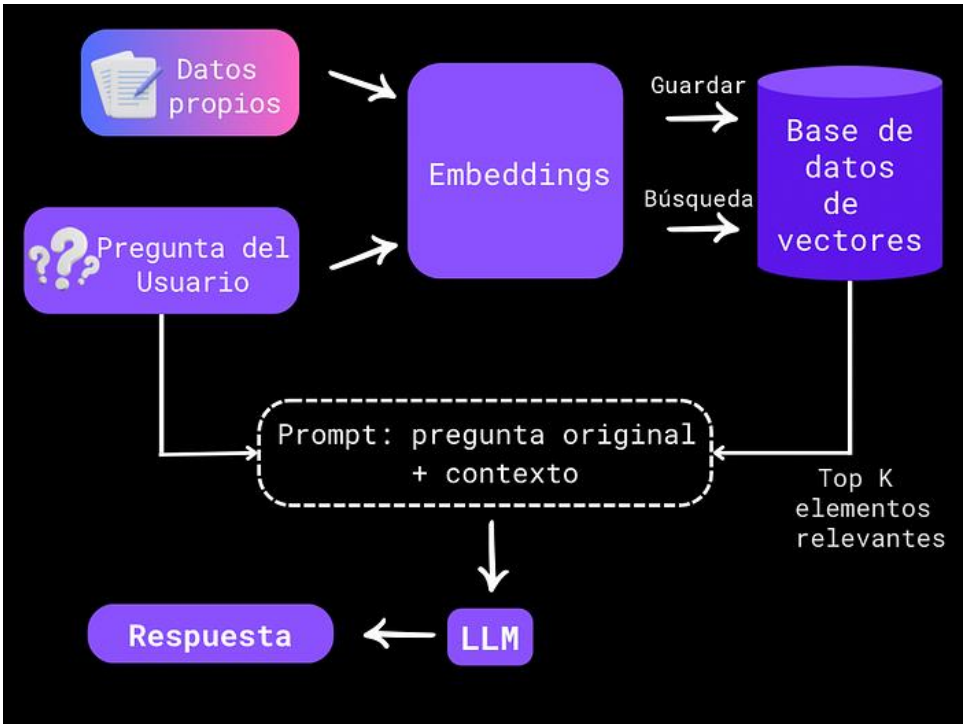


La base conceptual de RAG es muy sencilla. Consiste en recuperar trozos de nuestros propios documentos, que son relevantes como respuesta a una pregunta, y que sirven como contexto de la respuesta, de modo que se aprovechen dos enfoques diferentes:

- Recuperación
- Generación

-de ahí las siglas RAG-.

Para ello, resulta imprescindible procesar previamente la información privada, vectorizarla y almacenarla en este formato, por lo que las bases de datos vectoriales juegan aquí un papel importantísimo.




De este modo, cuando se consulta al LLM, se puede recuperar antes la información más relevante almacenada en nuestro vectorstore y adjuntarla como base de conocimiento contextual al LLM, para enriquecer su respuesta.

**Prompting**

El prompting es la forma en que se pregunta a los modelos grandes del lenguaje, una serie o conjunto de técnicas orientadas a la formulación eficiente de preguntas con el objetivo de obtener respuestas de mayor valor.

| PROMPT ENGINEERING | | |
|---|---|---|
| Disciplina orientada a potenciar el uso eficiente de modelos grandes del lenguaje | Centrada en el diseño y la prueba de frases, preguntas o inputs de los modelos | Que tiene por objetivo obtener mejores despuestas de los grandes modelos del lenguaje |

Así, el prompt engineering consiste en la creación de afirmaciones o preguntas que permitan guiar a los modelos de IA para que generen las respuestas esperadas, y es que, disponer de prompts bien diseñados, permite explotar al máximo el potencial de estos modelos de inteligencia artificial,

Pero, además, dentro del contexto de LangChain, cabe destacar los Prompts, como inputs con estructuras específicas que aportan contexto al modelo. LangChain brinda acceso a diferentes plantillas de prompt, que permiten enriquecer sustancialmente las aplicaciones conversacionales.

Para profundizar en prompt engineering, un área de conocimiento que va a experimentar un crecimiento sostenido muy interesante en los años próximos, te recomiendo el siguiente artículo: https://www.hostinger.es/tutoriales/prompt-engineering

**Bases de datos vectoriales y con soporte de grafos**

Antes hemos mencionado los almacenes vectoriales de información. Son muchas las bases de datos vectoriales, y tienen la inmensa ventaja de almacenar la información de contexto de las piezas de información (chunks) que componen un documento.

Si a esta forma de representar y almacenar los chunks le sumamos la posibilidad de representar las relaciones entre estos trozos informacionales mediante grafos, las posibilidades que surgen en el proceso de recuperación de la información son inmensas.

Así, los nodos de cada grafo, son piezas de contenido (embeddings), mientras que las aristas expresan la relaciónn entre nodos, pudiendo ser dirigidas o no.

Por tratarse de una materia que, por un lado da para un temario completo, y por otro lado ser un asunto complementario a NLP, no se va a profundizar más en ello, si bien, se propone el siguiente enlace para ampliar conocimientos al respecto: https://www.cs.us.es/~fsancho/Blog/posts/Bases_Datos_Grafo.md.html

**SLM's**

El auge de la computación "on Edge" y de la AIoT (Artificial Intelligence of things) ha estimulado el resurgir de modelos del lenguaje de menor tamaño.

Desde unite.ai, lo resumen de forma transparente:

> **Ventajas de los modelos de lenguaje más pequeños**
>
> El atractivo de los modelos de lenguaje más pequeños radica en su eficiencia y versatilidad. Ofrecen tiempos de capacitación e inferencia más rápidos, reducen las huellas de carbono y de agua y son más adecuados para su implementación en dispositivos con recursos limitados, como los teléfonos móviles. Esta adaptabilidad es cada vez más crucial en una industria que prioriza la accesibilidad y el rendimiento de la IA en una amplia gama de dispositivos.

Puedes consultar el artículo completo en: https://www.unite.ai/es/creciente-impacto-de-los-modelos-de-lenguaje-peque%C3%B1o/

Y es que, muchas veces el tamaño no es tan importante. Prueba de ello es el modelo Phi-2 de Microsoft, que ofrece un rendimiento que supera con creces al de modelos de mayor tamaño: https://www.europapress.es/portaltic/sector/noticia-microsoft-lanza-phi-modelo-lenguaje-pequeno-rendimiento-superior-modelos-25-veces-mas-grandes-20231213150934.html

Todo parece indicar que generar modelos específicos permite una "hiperespecialización" de los mismos muy interesante y que este es un camino con posibilidades muy interesantes a futuro.

## Ejercicios propuestos

**Ejercicio Langchain 1**

Estudia en profundidad este artículo de médium:

https://medium.com/@onkarmishra/using-langchain-for-question-answering-on-own-data-3af0a82789ed

Implementa un sistema que permita mantener conversaciones sobre un tema concreto, empleando para ello Langchain.

**Ejercicio Langchain 2**

El ejercicio anterior no permite mantener un contexto conversacional. En este ejercicio, debes estudiar el siguiente artículo:

https://betterprogramming.pub/build-a-chatbot-on-your-csv-data-with-langchain-and-openai-ed121f85f0cd

modifica el ejercicio 1 para que se mantenga un contexto histórico de la conversación.

## Bibliografía para completar el aprendizaje

1. ¿Qué son los modelos de lenguaje de gran tamaño? - Explicación sobre los LLM de IA - AWS (amazon.com)
2. LLM: ¿Qué son los Grandes Modelos de Lenguaje? | Aprende Machine Learning
3. https://www.evoacademy.cl/curso-introduccion-a-langchain-tutorial-fundamentos-langchain-models-prompt-templates-chain/
4. https://www.youtube.com/watch?v=RoR4XJw8wIc
5. https://www.langchain.com/
6. https://www.paradigmadigital.com/dev/que-es-langchain-como-crear-aplicaciones-python-libreria/
7. https://www.linkedin.com/pulse/langchain-qu%C3%A9-espor-lo-necesitamos-c%C3%B3mo-funciona-caso-alvarez/?originalSubdomain=es
8. 
9. https://www.hostinger.es/tutoriales/prompt-engineering
10. https://www.unite.ai/es/creciente-impacto-de-los-modelos-de-lenguaje-peque%C3%B1o/
11. https://www.europapress.es/portaltic/sector/noticia-microsoft-lanza-phi-modelo-lenguaje-pequeno-rendimiento-superior-modelos-25-veces-mas-grandes-20231213150934.html
12. https://medium.com/@onkarmishra/using-langchain-for-question-answering-on-own-data-3af0a82789ed
13. https://betterprogramming.pub/build-a-chatbot-on-your-csv-data-with-langchain-and-openai-ed121f85f0cd