eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

INFORMATIKA FAKULTATEA
FACULTAD DE INFORMÁTICA

Trabajo de Fin de Grado

Grado en Ingeniería Informática

Ingeniería de Computadores

Hau izenburua da/Título del trabajo

Egilearen izen-abizenak / Nombre y apellidos de la o el autor

Dirección
Zuzendaria 1 / Director(a) 1
Zuzendaria 2 / Director(a) 2

28 de marzo de 2025

Esker onak / Agradecimientos

Eskerrak eman nahi izanez gero, hemen idatzi testua. En caso de querer añadir agradecimientos, escribir aquí el texto.

Atal hau nahi ez baduzu, *main.tex* fitxategian komentatu lerro hori. En caso de no querer este apartado, comentalo en el fichero *main.tex*.

Laburpena / Resumen

Idatzi hemen laburpena. Escribe aquí el resumen.

Índice de contenidos

Índice de figuras

Índice de tablas

Lista de Algoritmos

CAPÍTULO 1

Introducción

LLMs han avanzado mucho, las apps que usan gen IA están surgiendo blah blah blah.

Los agentes ya hace mucho, pero ahora se están popularizando agentes LLM. Para esto están surgiendo frameworks blah blah blah.

Además se suele trabajar con ajustes de modelos, sistemas RAG para sacarle más provecho a las aplicaciones.

En este contexto LKS Next quiere investigar sobre las diferentes aquitecturas de agentes y sus aplicaciones. Para ello se dispone del escenario de onboarding.

a

CAPÍTULO 2

Antecedentes

Resumen de diferentes partes:

- Ajuste de agentes ->LoRa? ->comentar lo de los agentes instruct. - El protocolo MCP - El estado del arte en arquitecturas de agentes LLM y sistemas RAG - - El estado del arte en agentes integrados a proyectos software + contexto de onboarding quizás en la introducción. - Redes neuronales?

2. Agentes LLM (Fundamentos y funcionamiento básico)

¿Qué son los agentes LLM? 2.1. Modelos LLM 2.2. Interacción con herramientas 2.3. Abstracciones en frameworks

2.1. Agentes LLM

Los agentes de Inteligencia Artificial son programas informáticos que implementan modelos computacionales avanzados para ejecutar diversas funciones específicas del contexto en el que se aplican. Tras siete décadas y media de investigación, los esfuerzos en el campo se han focalizado en agentes basados en Grandes Modelos de Lenguaje (LLM).

Estos agentes, en esencia, constituyen sistemas informáticos que utilizan dichos modelos como núcleo operativo para la resolución de problemas. Para ello, les suministran los datos de entrada pertinentes, interpretan sus respuestas y transforman estas salidas en acciones concretas que permiten alcanzar los resultados deseados.

2.1.1. Modelos LLM

Los modelos de lenguaje (LM) constituyen sistemas computacionales especializados en el procesamiento del lenguaje natural. Entre estos, destacan los grandes modelos de lenguaje, diseñados para acometer tareas como la clasificación y generación de contenido lingüístico. Su fundamento técnico reside en redes neuronales que implementan la arquitectura Transformer

[Attention is All You Need]. Para comprender el funcionamiento de dicha arquitectura, resulta imprescindible asimilar previamente conceptos como la tokenización y las representaciones vectoriales del lenguaje.

Tokens Los tokens constituyen la unidad mínima de texto que el modelo puede procesar. Dado que operan sobre estructuras matemáticas, requieren transformar el lenguaje natural en representaciones matriciales. Para lograr esta conversión, el texto se segmenta en dichas unidades mínimas, que pueden corresponder a caracteres individuales, fragmentos de texto o palabras completas. El conjunto íntegro de estas unidades reconocibles por el modelo configura su vocabulario.

Representaciones vectoriales Constituyen vectores numéricos de dimensionalidad fija que codifican la semántica inherente a cada token. Estos vectores pueden comprender desde 768 dimensiones en arquitecturas como BERT-base hasta superar las 16.000 dimensiones en los modelos más avanzados del estado del arte, capturando conceptos semánticos en profundidad. Por ejemplo, una dimensión específica podría especializarse en representar conceptos abstractos. En este contexto, la representación vectorial del token "animal" presentaría un valor más elevado en dicha dimensión que la correspondiente al término "gato", reflejando su mayor grado de abstracción conceptual.

2.1.1.1. Arquitectura Transformer

Esta arquitectura emplea el mecanismo de "atención" para enriquecer la comprensión textual del modelo. El proceso transforma inicialmente la representación de las palabras de entrada incorporando información contextual mediante múltiples capas de atención, que consisten en operaciones matriciales que transfieren información semántica entre palabras.

Los agentes LLM utilizan predominantemente modelos decodificadores autorregresivos (GPT, Claude-Sonnet, Llama), optimizados para la generación secuencial de texto. Su característica distintiva es la aplicación de atención únicamente sobre tokens precedentes, permitiendo que el modelo compute la influencia de tokens anteriores sobre el actual sin modificar representaciones previas.

La figura 2.1 ilustra un ejemplo simplificado del cómputo de atención para la frase "El gato duerme". Al generar la siguiente palabra, el modelo presta atención diferenciada a elementos previos, considerando más relevante que el gato esté durmiendo que su color. Así, calcula la probabilidad del siguiente token considerando todo el vocabulario disponible. Es importante destacar que, debido a la naturaleza autorregresiva, la influencia de un token posterior como "negro" no se aplica a tokens anteriores como "gato".

En implementaciones prácticas, el mecanismo de atención que opera entre un token origen y un token destino se computa mediante operaciones matriciales que involucran tanto las representaciones vectoriales de ambos tokens como las matrices de atención, componentes de los parámetros aprendidos del modelo. Durante el proceso de entrenamiento, estas matrices de atención adquieren la capacidad de capturar fenómenos lingüísticos; por ejemplo, una

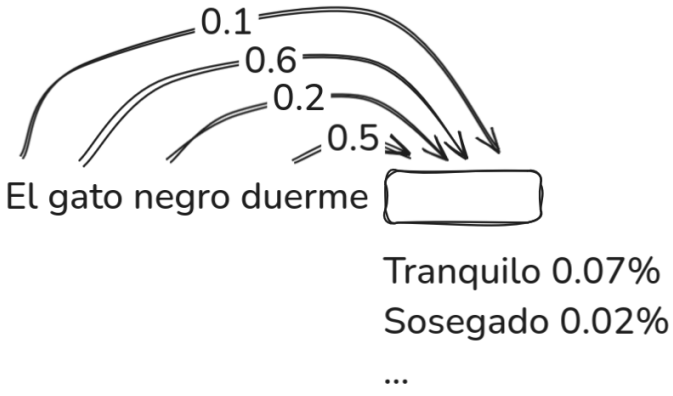

Figura 2.1: Ejemplo simplificado de cómputo de atención para la frase "El gato duerme"

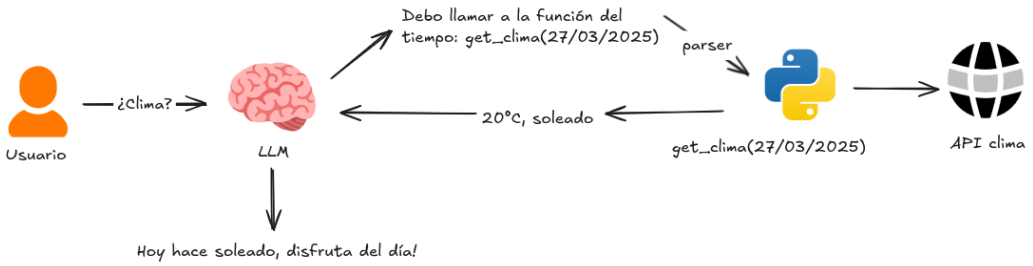

Figura 2.2: Ejemplo de interacción de un modelo LLM con una herramienta externa.

matriz específica podría especializarse en cuantificar la connotación negativa que un token origen proyecta sobre un token destino.

2.1.2. Interacción con herramientas externas

Los agentes LLM poseen la capacidad de interactuar con diversas herramientas como búsquedas web, bases de datos o interfaces de usuario. Fundamentalmente, un LLM solo genera tokens de texto, por lo que la integración de herramientas se implementa mediante palabras clave que el modelo puede producir. Para ello, en el texto de entrada se especifica el esquema de la función a utilizar y, si decide emplearla, el modelo generará el texto correspondiente. Posteriormente, se procesa la respuesta para extraer llamadas a funciones si las hubiese.

La interacción con herramientas es típicamente alternante. El modelo evalúa si requiere una función y, en caso afirmativo, la salida de esta se utilizará como entrada para el modelo. La figura 2.2 ilustra el esquema de un agente con acceso a una API del clima. Como el modelo carece de información climática en tiempo real, se le indica en el prompt la posibilidad de invocar esta función. Al incluir la llamada en su texto de salida, se ejecuta la función y su respuesta se transmite al modelo para generar el resultado final.

2.1.3. Abstraciones en frameworks

Aunque los agentes LLM son una tecnología relativamente emergente, ya existen frameworks que facilitan su implementación. Estos frameworks, como LangChain o LangGraph, proporcionan abstracciones de alto nivel para la creación de agentes LLM, permitiendo a los desarrolladores reutilizar funcionalidades comunes en la mayoría de sistemas de este tipo.

Las funcionalidades más comunes que estos frameworks ofrecen son:

- **Gestión de modelos:** La ejecución de modelos require una comprensión profunda del modelo utilizado. Cada modelo utiliza su propio tokenizador, y está diseñado para utilizar un esquema de entrada y salida concreto. Los frameworks simplifican esta tarea, implementando una interfaz que permite a los desarrolladores utilizar diferentes modelos sin tener que preocuparse por los detalles de implementación.
- **Interacción conversacional:** La interacción con los modelos se realiza mediante un esquema conversacional, donde el modelo recibe un texto de entrada y genera una respuesta. Se van concatenando las respuestas y las entradas para mantenerel contexto de la conversación, de forma que la segunda pregunta contenga todos los mensajes anteriores.
- **Uso de herramientas externas:** Los frameworks permiten la integración de herramientas al modelo directamente desde código, facilitando la interacción con sistemas externos.
- **Interacción entre agentes:** Los agentes pueden interactuar entre sí, permitiendo la creación de sistemas más complejos. Por ejemplo, un agente puede invocar a otro para realizar una tarea específica. Muchos frameworks definen sistemas que definen cómo se comunican los agentes entre sí.

Eranskina / apéndice

Eranskinak

Bibliografía