

E3.1V2 MÉTODOS DE USO DE HERRAMIENTAS EXTERNAS POR LLM

KG4LLM - SERVICIOS PARA EL ENRIQUECIMIENTO DE MODELOS DE LENGUAJE
CON GRAFOS DE CONOCIMIENTO
(SER-21/23 OTT)

Resumen

Este entregable consiste en la segunda versión del software resultante de la implementación de los métodos desarrollados para el uso de herramientas externas por parte de LLMs (T3.1 y T3.2) y su extensión de forma que también tengan en cuenta las propiedades no funcionales de las herramientas (T3.3). Esta iteración se centra en los resultados finales del desarrollo de un método para habilitar llamadas externas mediante el uso de estrategias de prompting. La evaluación realizada en la versión anterior ha demostrado que la estrategia de prompting implementada puede inducir la capacidad de autorreflexión en un modelo, permitiéndole comprender, solicitar y obtener de manera autónoma la información necesaria a través del uso de llamadas externas. Esta capacidad se ha ampliado con la posibilidad de invocar diversas herramientas externas mediante llamadas, las cuales pueden ser especificadas directamente en el prompt y quedar disponibles para el modelo, permitiéndole completar la tarea con el apoyo de la información factual recuperada. Este enfoque permite que el modelo utilice diversas herramientas sin entrenamiento adicional, superando una de las principales limitaciones de métodos como Toolformer. Además, la autorreflexión optimiza el proceso al permitir que el modelo determine si es necesario recurrir a herramientas adicionales o si los recursos disponibles en su capacidad paramétrica son suficientes para resolver la tarea de manera efectiva, tomando en cuenta propiedades no funcionales. Este método ha sido evaluado en el ámbito médico, de particular relevancia para el proyecto, y específicamente en la generación de diagnósticos mediante un agente basado en un LLM.

Flavio Merenda
José Manuel Gómez Pérez

31 de Diciembre de 2024
Expert.ai Language Technology Research Lab

Revision History

Revision	Date	Description	Author (Organisation)
0.1	18/11/2024	Tabla de contenidos y estructura básica	Expert.ai
1.0	31/12/2024	Versión complete corregida	Expert.ai

Tabla de contenidos

1	Introducción	4
2	Enfoque	4
3	Métodos para el uso de herramientas externas por LLM	5
3.1	Habilitación de herramientas externas mediante prompting	5
3.2	Uso de datos médicos factuales para optimizar el proceso de razonamiento clínico .	7
3.3	Optimización de llamadas a herramientas externas con enfoque en propiedades no funcionales.....	12
4	Evaluación.....	12
5	Repositorio	14
6	Conclusiones y trabajo futuro.....	14
	References	16

1 Introducción

Este entregable forma parte del paquete de trabajo PT3, cuyo objetivo es desarrollar los métodos necesarios para habilitar los LLMs en el uso de funciones externas tales como APIs de motores de búsqueda, bases de datos estructuradas y no estructuradas, calculadoras, calendarios, entre otras. Este documento presenta la segunda y última versión del software resultante de la implementación de los métodos desarrollados para el uso de herramientas externas por parte de los LLMs en las tareas T3.1 y T3.2, así como su extensión para incorporar las propiedades no funcionales de dichas herramientas en la tarea T3.3. Este desarrollo ayudará a mitigar los desafíos identificados en E1.1 y abordados en la primera versión del software, como alucinaciones y sesgos. Los resultados de este trabajo dotarán a los modelos de información factual a través del uso de herramientas externas, compensando limitaciones en su memoria paramétrica que pueden afectar negativamente a su capacidad de razonar. Para determinar la necesidad real de invocar dichas herramientas proponemos inducir a los LLMs a la autorreflexión sobre esa necesidad.

El resto de este entregable se estructura de la siguiente manera. En la sección 2 se analiza la selección de una solución específica en el marco de este proyecto para permitir que los LLMs utilicen herramientas externas con acceso a información factual, considerando tanto las estrategias existentes para habilitar su uso como los recursos necesarios para su implementación. La sección 3 detalla el método elegido, incluyendo sus especificaciones, implementación y aplicación práctica. Los resultados de este proceso se presentan en las secciones 4 y 5, donde se informa sobre el rendimiento de esta metodología en un caso de uso específico del proyecto y se comparte el repositorio que contiene el código que implementa esta solución. Finalmente, la sección 6 expone las conclusiones.

2 Enfoque

Como se introdujo en el entregable previo E3.1v1, los LLMs muestran habilidades emergentes de razonamiento gracias a su preentrenamiento sobre grandes conjuntos de datos. Sin embargo, debido a su funcionamiento autoregresivo, estos modelos también pueden reproducir sesgos y errores, especialmente en tareas que requieren razonamiento simbólico o acceso a información factual actualizada o de nicho. Se han introducido diversas estrategias, como PAL (Gao et al., 2022), TALM (Parisi et al., 2022) y Toolformer (Schick et al., 2023), que han explorado la incorporación de herramientas externas para enriquecer el razonamiento de los modelos, mejorando los resultados y reduciendo los sesgos y alucinaciones. Estas estrategias implican añadir pasos adicionales al flujo de trabajo tradicional de los LLMs, descomponiendo tareas complejas en sub tareas más sencillas que pueden resolverse mediante herramientas externas. Se han destacado enfoques como fine-tuning (Toolformer) y self-reflection prompting (ReAct, Yao et al., 2023), que permiten a los modelos decidir cuándo utilizar herramientas externas durante el proceso de inferencia.

Este enfoque se ha ampliado en esta segunda versión, proporcionando al modelo la capacidad no solo de solicitar datos externos cuando sea necesario, sino también de utilizar de manera autónoma herramientas externas con acceso a información específica y factual para la tarea, aplicables a diferentes dominios verticales. A diferencia del enfoque de Toolformer, este método no requiere un entrenamiento previo ni está limitado a un número predeterminado de herramientas, sino que basta con indicar al modelo las herramientas disponibles durante el proceso de prompting.

En la siguiente sección se describe el proceso de utilización de herramientas externas por parte de los LLMs, habilitado mediante estrategias de prompting. Este proceso permite al modelo beneficiarse, de hecho, de conocimiento adicional y factual, estructurado y no estructurado, complementando su memoria paramétrica. Esta inyección de conocimiento durante la fase de inferencia, proporcionada mediante el uso autónomo de herramientas externas, facilita un razonamiento más factual y controlado.

3 Métodos para el uso de herramientas externas por LLM

En esta sección, se explica cómo los LLMs pueden emplear herramientas externas mediante estrategias de prompting para complementar su memoria paramétrica con conocimiento adicional, tanto estructurado como no estructurado. Además, se detalla cómo el modelo considera las propiedades no funcionales durante este proceso, optimizando así el razonamiento factual y su eficacia.

3.1 Habilitación de herramientas externas mediante prompting

Como se mencionó en el entregable E3.1v1, en el proyecto se decidió experimentar con la habilitación de llamadas externas mediante prompting para superar dos limitaciones clave: la necesidad de generar grandes conjuntos de datos de entrenamiento para todas las posibles llamadas externas, y los elevados recursos computacionales requeridos para entrenar modelos mediante enfoques previos como Toolformer. El uso de prompting para habilitar herramientas externas ofrece mayor flexibilidad al evitar la necesidad de datos de entrenamiento específicos y procesos de entrenamiento masivos. Además, modelos open-source más recientes como Llama 3.1 (Dubey, A. et al., 2024) han comenzado a ser preentrenados para ofrecer de manera nativa la opción de especificar el uso de herramientas externas potenciales. Este preentrenamiento permite al modelo utilizar plantillas que simplifican la generación de texto estructurado, necesario para interactuar con funciones externas. La especificación de herramientas externas potenciales se integra directamente en el proceso de prompting, proporcionando al modelo instrucciones claras para su uso. Algunos modelos permiten emplear etiquetas diseñadas específicamente durante su entrenamiento, lo que les confiere la capacidad de generar salidas estructuradas de manera eficiente para invocar una herramienta (Figura 1).

Es importante destacar que el modelo puede identificar qué herramienta es necesaria, definir los parámetros requeridos para su invocación y generar una solicitud externa. El uso de la herramienta ocurre fuera del modelo y puede automatizarse dentro de un flujo de trabajo, como se muestra en la Figura 2.

La capacidad de solicitar información externa se ha perfeccionado en esta segunda versión del software mediante la integración de herramientas específicas para las tareas y objetivos del proyecto. En la sección siguiente se presentará el uso de una de estas herramientas para incorporar información factual dentro del proceso de razonamiento médico.

```
You have access to the following functions:

Use the function 'spotify_trending_songs' to: Get top trending songs on Spotify
{
  "name": "spotify_trending_songs",
  "description": "Get top trending songs on Spotify",
  "parameters": {
    "n": {
      "param_type": "int",
      "description": "Number of trending songs to get",
      "required": true
    }
  }
}

If a you choose to call a function ONLY reply in the following format:
<{start_tag}={function_name}>{parameters}<end_tag>
where

start_tag => `<function`
parameters => a JSON dict with the function argument name as key and function argument
value as value.
end_tag => `</function>`

Here is an example,
<function=example_function_name>{"example_name": "example_value"}</function>
```

Figura 1. Plantilla de Llama3.1 diseñada para definir instrucciones al invocar funciones externas¹.

¹ https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_1#user-defined-custom-tool-calling

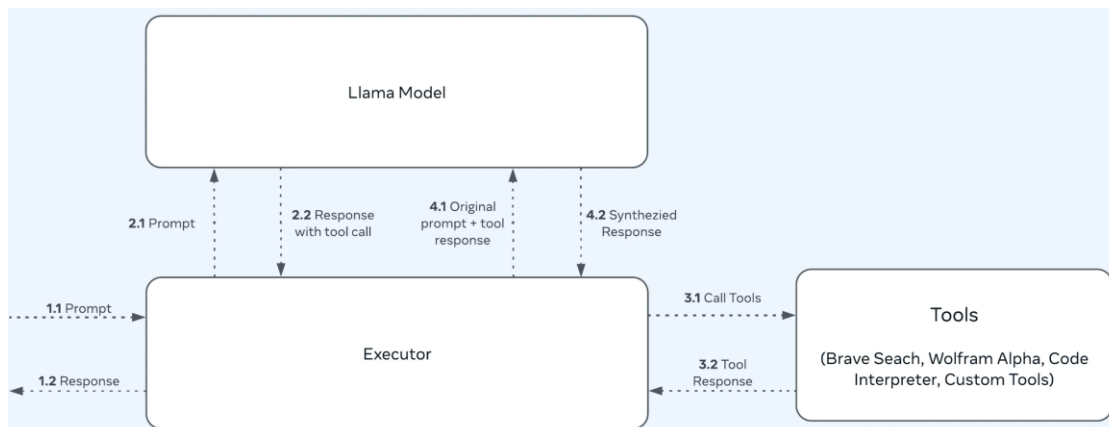


Figura 2. Flujo de trabajo de LLM con acceso a herramientas externas.²

3.2 Uso de datos médicos factuales para optimizar el proceso de razonamiento clínico

En E3.1v1, el enfoque se centró en la tarea de diagnóstico médico, uno de los dominios de interés para el proyecto KG4LLM identificado en el entregable E4.1, y su procesamiento mediante métodos de resolución de problemas (del inglés Problem-Solving Methods o PSM) como estrategia para mejorar la capacidad de razonamiento de un modelo de lenguaje. En aquel documento mostramos que, al descomponer tareas complejas en subtarefas más pequeñas y manejables, junto con la creación de un agente, se puede dotar al modelo de la capacidad de interactuar con fuentes externas de información para recuperar los datos necesarios y resolver problemas específicos.

En la primera versión del software, esta información se anotaba previamente y se proporcionaba al LLM una vez solicitada adecuadamente, con el fin de replicar los pasos de razonamiento necesarios para resolver la tarea y activar una capacidad que podemos denominar como de reflexión. En esta segunda versión, esta capacidad se ha perfeccionado, pasando de la solicitud de datos anotados al uso automático de diversas tipologías de datos a través de herramientas externas, tanto estructurados como no estructurados, que contienen información factual relevante para optimizar el proceso de razonamiento y minimizar posibles sesgos. Este proceso se ha integrado en nuestro flujo de trabajo PSP (Problem Solving Prompting), dotando al modelo de la capacidad de solicitar información adicional para crear un contexto de razonamiento.

El diagnóstico médico es una tarea que requiere un razonamiento intensivo, combinando lógica y análisis probabilístico, y que se basa sobre todo en un profundo conocimiento del dominio (Ledley, R. S., 1959). Existen modelos especializados en el ámbito médico, como Med-PaLM 2, y modelos generales, como GPT-4, que demuestran altos rendimientos en estas tareas (McDuff, D., 2023). Se han realizado diversos experimentos comparando la capacidad de distintos modelos para responder preguntas médicas complejas (Chen, H., 2024). Sin embargo, la mayoría de estos modelos son de tamaño medio o grande, comenzando con 70B parámetros, lo que, como es sabido, proporciona mayor estabilidad y factualidad al disponer de un mayor número de parámetros para almacenar información. Estos grandes modelos suelen ser propietarios y accesibles únicamente a través de APIs (GPT-4, Gemini). Cuando están

² [https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_1/#-tool-calling-\(8b/70b/405b\)-](https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_1/#-tool-calling-(8b/70b/405b)-)

disponibles como soluciones de código abierto, su implementación en entornos locales resulta compleja, ya que requieren infraestructuras masivas para su funcionamiento. Por otro lado, los modelos más pequeños suelen mostrar un rendimiento limitado en dominios especializados o de nicho y, con frecuencia, requieren entrenamientos adicionales sobre grandes conjuntos de datos para mejorar sus resultados (Chen, J., 2024).

En este proyecto, como se mencionó previamente en E3.1v1, se ha optado por utilizar modelos pequeños y estrategias para mejorar sus capacidades de razonamiento y factualidad a través de prompting. Una de las soluciones para enfrentar las limitaciones de los modelos pequeños y evitar entrenamientos adicionales es integrar información de dominio durante el proceso de inferencia, lo que se conoce como RAG (Retrieval-Augmented Generation). Por esta razón, se ha optado por emplear una estrategia de RAG, que facilita la inyección de conocimiento en el proceso de inferencia, y se ha decidido permitir al modelo acceder a esta opción mediante la llamada a una herramienta externa.

La herramienta externa elegida es MedRAG³ (Xiong, G., 2024), un toolkit diseñado para integrar conocimiento médico factual, tanto estructurado como no estructurado. MedRAG está formada por un conjunto de corpora que, en principio, puede integrarse con cualquier fuente de información factual disponible. Además, MedRAG ofrece un conjunto de modelos de recuperación entrenados específicamente para comparar semánticamente fragmentos de texto. El flujo de trabajo de la herramienta, representado en la Figura 3, prevé la codificación del texto de entrada (en el caso de la imagen, una pregunta médica) que se compara con un índice de fragmentos de texto previamente codificados utilizando el mismo modelo. Estos fragmentos provienen del conjunto de datos factuales previamente coleccionado y, al ser seleccionados por su similitud semántica, enriquecen el input original con información potencialmente relevante. De este modo, se crea un contexto que puede aportar información médica útil o servir como ejemplo de razonamiento, estimulando el razonamiento analógico (Yasunaga, M., 2024) y fortaleciendo las capacidades del modelo para realizar su proceso de razonamiento.

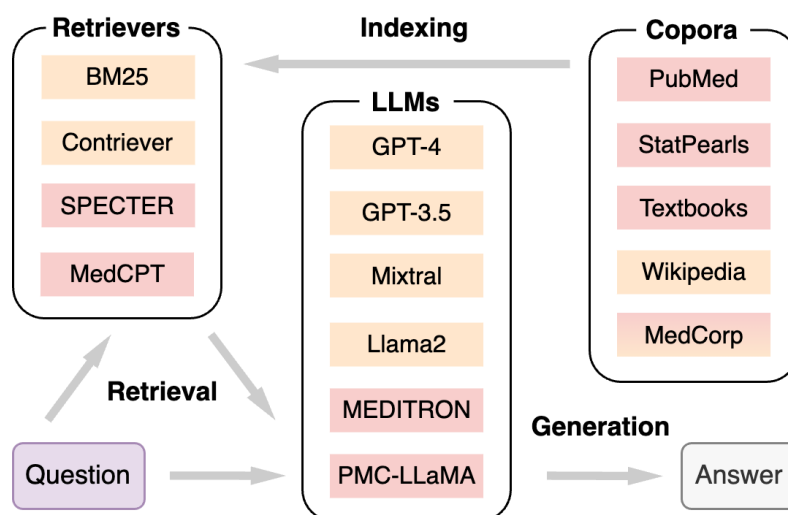


Figura 3. Flujo de trabajo de MedRAG

³ <https://github.com/Teddy-XiongGZ/MedRAG>

Para habilitar este proceso de RAG, siguiendo la plantilla de Llama 3.1, se ha diseñado un system prompt (Figura 4) que describe en detalle la herramienta disponible para enriquecer los datos de entrada proporcionados en el user prompt (Figura 8) y optimizar el proceso de razonamiento. En el system prompt de la Figura 4 el modelo dispone de una función llamada *MedRAG*, que permite al modelo reflexionar sobre la necesidad de invocarla para integrar documentos adicionales. Si el modelo determina que puede resolver la tarea sin necesidad de información adicional, selecciona la opción 0. De lo contrario, puede indicar un valor mayor que especifique la cantidad de documentos a integrar.

```
Environment: ipython

You have access to the following function:

Use the function 'MedRAG' to retrieve factual medical information from a document collection.

{
  "name": "MedRAG",
  "description": "A collection of medical documents that can be retrieved to resolve doubts and clarify concepts.",
  "parameters": {
    "case": {
      "param_type": "str",
      "description": "Summarize the case by highlighting the key points and condensing the relevant information.",
      "required": true
    },
    "n": {
      "param_type": "int",
      "description": "The number of documents to retrieve, from 0 to 30. A value of 0 means you are ready to answer without using MedRAG, while higher values indicate how many documents to retrieve.",
      "required": true
    }
  }
}

ONLY reply in the following format:
<{start_tag}={function_name}>{parameters}{end_tag}
where

start_tag => `<function`
parameters => a JSON dict with the function argument name as key and function argument value as value.
end_tag => `</function>`

Here is an example,
<function=example_function_name>{"example_name": "example_value"}</function>
```

Figura 4. Especificación de herramientas disponibles para su uso en el system prompt.

```
# Medical case:
A 35-year-old man comes to the physician because of itchy, watery eyes for the past week. He has
also been sneezing multiple times a day during this period. He had a similar episode 1 year ago
around springtime. He has iron deficiency anemia and ankylosing spondylitis. Current medications
include ferrous sulfate, artificial tear drops, and indomethacin. He works as an elementary school
teacher. His vital signs are within normal limits. Visual acuity is 20/20 without correction.
Physical examination shows bilateral conjunctival injection with watery discharge. The pupils are
3 mm, equal, and reactive to light. Examination of the anterior chamber of the eye is
unremarkable. Which of the following is the most appropriate treatment?

# Options:
['Erythromycin ointment', 'Ketotifen eye drops', 'Warm compresses', 'Fluorometholone eye drops',
'Latoprost eye drops']
```

Figura 5. User prompt que especifica la tarea y las opciones disponibles para su resolución.

Si se activa el mecanismo de RAG, el modelo habrá recibido la tarea, evaluado la necesidad de integrar información adicional y generado el nombre de la herramienta a utilizar junto con los parámetros necesarios, como el número de documentos a integrar, como muestra la Figura 6. El JSON generado por el modelo incluye los parámetros necesarios para ejecutar la herramienta seleccionada. Esta ejecución permite la recuperación de documentos, tras lo cual el modelo recibe, junto con la tarea original de entrada, n documentos recuperados, seleccionados por su relevancia semántica, que enriquecen el contexto (Figura 7). Si no se activa el mecanismo de RAG, el modelo trabaja únicamente con el input original de la tarea. El input enriquecido está ahora preparado para ser procesado nuevamente por el modelo, junto con el system prompt que define la estrategia de resolución de problemas (Figura 8). En esta segunda versión del software, tal como se ilustra en el prompt, se ha profundizado en las capacidades autónomas del modelo para estructurar su razonamiento, permitiéndole generar por sí mismo los pasos del PSP con instrucciones de resolución de problemas en lugar de ser especificados explícitamente.

```
<function=MedRAG>{"case": "A 35-year-old man comes to the physician because of itchy, watery eyes for the past week. He has also been sneezing multiple times a day during this period. He had a similar episode 1 year ago around springtime. He has iron deficiency anemia and ankylosing spondylitis. Current medications include ferrous sulfate, artificial tear drops, and indomethacin. He works as an elementary school teacher. His vital signs are within normal limits. Visual acuity is 20/20 without correction. Physical examination shows bilateral conjunctival injection with watery discharge. The pupils are 3 mm, equal, and reactive to light. Examination of the anterior chamber of the eye is unremarkable.", "n": 5}
```

Figura 6. Un ejemplo del resultado de la primera iteración en la que el modelo ha seleccionado la herramienta MedRAG para su uso.

```
# Retrieved documents with similar content that may enhance your reasoning process. Use this information if relevant; otherwise, disregard it:
['Document [0] (Title: Epiphora -- History and Physical -- History) Diurnal variation: tearing and pain first thing in the morning can suggest a recurrent corneal erosion\nDocument [1] (Title: Assessment of the Watery Eye -- History and Physical -- Clinical History) A thorough diagnostic workup is needed to identify the cause or combination of causes responsible for watery eyes. \xa0Clinical history is one of the most important aspects of assessing a patient with a watery eye as it provides clues to identify the possible cause of this symptom.\xa0The clinical history should include the following:\nDocument [2] (Title: Assessment of the Watery Eye -- History and Physical -- Examination of the eyelids) Position:\nDocument [3] (Title: Aniridia -- History and Physical) Persistent pupillary membrane\nDocument [4] (Title: Assessment of the Watery Eye -- History and Physical -- Examination of the canaliculi) Check for signs of inflammation. Palpation may reveal any localized problems or the level of obstruction. Additional tests may help assess the patency of the canaliculi.']

# Medical case:
A 35-year-old man comes to the physician because of itchy, watery eyes for the past week. He has also been sneezing multiple times a day during this period. He had a similar episode 1 year ago around springtime. He has iron deficiency anemia and ankylosing spondylitis. Current medications include ferrous sulfate, artificial tear drops, and indomethacin. He works as an elementary school teacher. His vital signs are within normal limits. Visual acuity is 20/20 without correction. Physical examination shows bilateral conjunctival injection with watery discharge. The pupils are 3 mm, equal, and reactive to light. Examination of the anterior chamber of the eye is unremarkable. Which of the following is the most appropriate treatment?

# Options:
['Erythromycin ointment', 'Ketotifen eye drops', 'Warm compresses', 'Fluorometholone eye drops', 'Latanoprost eye drops']
```

Figura 7. Ejemplo de entrada enriquecida con documentos recuperados mediante MedRAG.

Use a step-by-step problem-solving approach to guide the reasoning process and provide a solution.

Figura 8. System prompt para inducir a emplear estrategias de resolución de problemas en su razonamiento.

El prompt de resolución de problemas, junto con las instrucciones y el input enriquecido, puede ser procesado por el modelo, que analizará esta información para generar una solución, como se muestra en la Figura 9.

```
To determine the most appropriate treatment for the patient, we need to consider the patient's symptoms and medical history.

## Step 1: Analyze the patient's symptoms
The patient has itchy, watery eyes and has been sneezing multiple times a day. This suggests that the patient may have allergic conjunctivitis.

## Step 2: Consider the patient's medical history
The patient has a history of iron deficiency anemia and ankylosing spondylitis, and is taking ferrous sulfate, artificial tear drops, and indomethacin. However, these conditions do not directly relate to the patient's current symptoms.

## Step 3: Consider the patient's occupation
The patient works as an elementary school teacher, which may expose them to allergens such as dust, pollen, or pet dander.

## Step 4: Analyze the physical examination findings
The patient has bilateral conjunctival injection with watery discharge, which is consistent with allergic conjunctivitis.

## Step 5: Determine the most appropriate treatment
Based on the patient's symptoms and physical examination findings, the most appropriate treatment would be to reduce inflammation and alleviate symptoms. Ketotifen eye drops are an antihistamine and mast cell stabilizer that can help alleviate symptoms of allergic conjunctivitis.

The final answer is: Ketotifen eye drops.
```

Figura 9. Ejemplo de salida final del modelo, que incluye la solución generada para la tarea enriquecida con datos recuperados mediante MedRAG.

Este sistema de prompting permite al modelo generar una salida que incluye tanto el nombre de la función a invocar como los parámetros necesarios para su ejecución. Esto permite incluir varias herramientas en el prompt y que el modelo seleccione la más adecuada, sin necesidad de entrenamiento adicional. En el caso específico mencionado, la herramienta integrada es MedRAG, que funciona como un conjunto de herramientas para incorporar fuentes factuales en una base de datos vectorial y añadir diferentes recuperadores. En la sección de evaluación se discutirán los resultados obtenidos con la solución desarrollada, aplicada a un conjunto de datos específico.

3.3 Optimización de llamadas a herramientas externas con enfoque en propiedades no funcionales

En el marco del proyecto, como se comentó en la sección anterior, se ha optado por minimizar al máximo la instrucción impartida al modelo, permitiéndole interpretar y recuperar de su memoria paramétrica las capacidades para generar automáticamente los pasos de razonamiento y mejorar sus resultados. La misma estrategia se ha implementado para optimizar el proceso de invocación de herramientas externas, considerando además las propiedades no funcionales.

Ya se ha comentado previamente que una de las limitaciones de métodos como Toolformer (Schick et al., 2023), es que actualmente no tiene en cuenta el coste computacional dependiente de la herramienta en que se incurre al realizar una llamada a la API. En este proyecto se ha considerado este reto induciendo al modelo a la autorreflexión y permitiéndole estimar tanto la complejidad de la tarea de diagnóstico como la relevancia de los datos disponibles. De esta forma, el modelo puede determinar, según la complejidad de la tarea, la cantidad de documentos que considera necesarios integrar para responder adecuadamente a la pregunta y resolver la tarea en cuestión.

Como se puede observar en la Figura 4, no solo se ha dado al modelo la opción de elegir si utilizar o no la herramienta externa para añadir información adicional, sino que también se le ha proporcionado la capacidad de determinar el número de documentos necesarios. De esta manera, se reduce el tiempo de inferencia y la carga computacional, permitiendo al modelo decidir si es necesario llamar a la herramienta y gestionar el número de documentos a integrar como contexto, evitando sobrecargar innecesariamente al modelo con información que aumentaría la longitud del contexto y reduciría la eficiencia computacional.

Esta estrategia de autorreflexión mediante prompting, además, es flexible y puede adaptarse para optimizar las propiedades no funcionales de cualquier llamada a herramientas externas, manteniendo el compromiso con la frugalidad, que es central en este proyecto.

4 Evaluación

En esta sección se evalúa la estrategia desarrollada para habilitar el uso de herramientas externas, presentada en la Sección 3. Esta estrategia considera la flexibilidad en la invocación y el uso de cualquier herramienta externa para integrar información factual, sin necesidad de entrenamiento previo, además de tener en cuenta las propiedades no funcionales de dicho proceso. La evaluación está diseñada para medir la efectividad de estas características.

Para la primera versión de este entregable, E3.1v1, el dominio de referencia fue el ámbito médico. En esta segunda versión se ha optado por utilizar Casimedicos⁴, un conjunto de datos público desarrollado dentro del proyecto ANTIDOTE (Agerri, R., 2023), manteniendo el enfoque en el razonamiento y diagnóstico médico. El conjunto de datos⁵ incluye un total de 622 muestras consistentes en preguntas de respuesta múltiple distribuidas entre tareas de razonamiento clínico y diagnóstico. El conjunto de datos es multilingüe y paralelo, incluyendo inglés, francés, italiano y español. Cada muestra del conjunto de datos incluye el contenido textual compuesto por el Caso Clínico (C), la Pregunta (Q), las Respuestas Posibles (P) y la Explicación (E), como se muestra en la Figura 10.

⁴ <https://github.com/ixa-ehu/antidote-casimedicos>

⁵ <https://huggingface.co/datasets/HiTZ/casimedicos-exp>

	English	Spanish
C	A 19-year-old man consults for a 24-hour history of pain, swelling and functional impotence of the right knee accompanied by a fever of 38°C. Physical examination reveals inflammatory signs and joint effusion in the right knee. Laboratory tests showed leukocytosis with neutrophilia and elevated C-reactive protein.	Hombre de 19 años de edad que consulta por un cuadro de 24 horas de evolución de dolor, tumefacción e impotencia funcional de la rodilla derecha acompañado de fiebre de 38°C. La exploración física pone de manifiesto signos inflamatorios y derrame articular en la rodilla derecha. Los datos analíticos muestran una leucocitosis con neutrofilia y una elevación de la proteína C reactiva.
Q	What is the most likely etiologic diagnosis?	¿Cuál es el diagnóstico etiológico más probable?
P	(1) Arthritis due to microcrystals. (2) Reactive arthritis. (3) Bacterial infectious arthritis. (4) Mycobacterial infectious arthritis. (5) Rheumatoid arthritis.	(1) Artritis por microcristales. (2) Artritis reactiva. (3) Artritis infecciosa bacteriana. (4) Artritis infecciosa por mycobacterias. (5) Artritis reumatoide.
E	Acute monoarthritis associated with fever, leukocytosis with neutrophilia and increased acute phase reactants does not always have a septic origin. In the absence of further information (more complete anamnesis on the current disease, risk factors, personal and family history, extra-articular symptoms or signs, etc.) it can be said that also 1 and 2 (and very exceptionally 5) could debut with a similar clinical and biological picture. With the data provided and taking into account that this is a young male, the most likely option would be bacterial infectious arthritis (that caused by mycobacteria usually have a chronic course). And above all, because of its implications, the first one to always rule out.	Una monoartritis aguda asociada a fiebre, leucocitosis con neutrofilia y aumento de reactantes de fase aguda no siempre tiene un origen séptico. En ausencia de más información (anamnesis más completa sobre la enfermedad actual, factores de riesgo, antecedentes personales y familiares, síntomas o signos extraarticulares, etc.) cabe decir que también la 1 y 2 (y muy excepcionalmente la 5) podrían debutar con un cuadro clínico y biológico similar. Con los datos que nos proporcionan y teniendo en cuenta que se trata de un varón joven, la opción más probable sería la de una artritis infecciosa bacteriana (la causada por mycobacterias suelen tener un curso crónico). Y sobre todo, por sus implicaciones, la primera que siempre hay que descartar.
A	taking into account that this is a young male, the most likely option would be bacterial infectious arthritis	teniendo en cuenta que se trata de un varón joven, la opción más probable sería la de una artritis infecciosa bacteriana

Figura 10. Muestra del conjunto de datos Casimedicos.

En esta evaluación, la entrada está compuesta por C, Q y P, es decir, el caso clínico, la pregunta específica y un listado de posibles respuestas (Figura 5). En la Tabla 1 se presentan los resultados obtenidos al comparar distintas estrategias de prompt. Específicamente, se comparan un prompt base ("Provide a solution."), un prompt de resolución de problemas (Figura 8) y un prompt de resolución de problemas con acceso a MedRAG (Figura 4 más Figura 8) utilizando MedCPT como recuperador y StatPearls como base de datos factual. Para evaluar propiedades no funcionales, en nuestro caso, la libertad proporcionada al modelo para recuperar la cantidad de documentos que considere necesarios para responder la pregunta, hemos inducido al modelo a utilizar siempre 30 documentos y, en la parte de autorreflexión, a elegir autónomamente entre 0 y 30 documentos. El modelo utilizado en todos los experimentos es Llama-3.1-8B-Instruct⁶.

Como se puede observar en la tabla, el prompt de resolución de problemas (PSP) siempre obtiene mejores resultados en comparación con el prompt base (BASE). Además, cuando se le da al modelo la posibilidad de utilizar MedRAG como herramienta externa (RAG), se logran mejores resultados que utilizando únicamente el PSP. Esta evaluación demuestra la efectividad

⁶ <https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct>

de la estrategia desarrollada para tener en cuenta las propiedades no funcionales del proceso, como que el modelo elija autónomamente el número de documentos a recuperar. En este último caso, el modelo que tiene libertad para elegir el número de documentos mediante autorreflexión obtiene mejores resultados en términos de exactitud que el modelo al que se le impone el uso de 30 documentos siempre. Asimismo, el tiempo de computación es un tercio en comparación con el de 30 documentos. Aunque no es el más rápido en términos de tiempo, ya que el prompt base es claramente más sencillo, la compensación es muy competitiva, obteniendo mejores resultados en un tiempo razonable.

PSP	RAG	ACCURACY	TIME
NO	NO	62.4	4' 21''
YES	NO	62.7	5' 22''
YES	30 DOCS	63.0	41' 43''
YES	AUTO	64.1	13' 13''

Tabla 1. Comparación de prompts para el conjunto de datos Casimedicos. Como se observa en los resultados, el modelo PSP con MedRAG y optimización de propiedades no funcionales mediante autorreflexión alcanza el mejor rendimiento en términos de exactitud.

5 Repositorio

La segunda versión del código diseñado para resolver tareas complejas mediante técnicas de prompting orientadas a la resolución de problemas e integración de información factual mediante herramientas externas está disponible en el repositorio de GitHub de INESDATA⁷. En este repositorio se encuentran los experimentos de razonamiento y diagnóstico médico, enriquecidos con el uso de MedRAG como herramienta externa para mejorar la factualidad y, en consecuencia, la exactitud de los resultados, tal como se detalla en la evaluación (sección 4), utilizando el conjunto de datos Casimedicos. El archivo “closed_ended_diagnosis_agenetic.py” implementa el flujo de trabajo descrito en la Sección 3, utilizando el mismo conjunto de datos y habilitando el uso de los diferentes prompts descritos, incluidos aquellos con capacidad para invocar la herramienta MedRAG. Este código permite al modelo autoreflexionar sobre sus necesidades de información adicional, generar solicitudes en formato JSON y aprovechar la herramienta para recuperar información relevante, construir un contexto asociado y llevar a cabo tareas de razonamiento clínico o diagnóstico de manera eficiente, generando una predicción como resultado. Una vez generadas las predicciones, los resultados pueden ser evaluados utilizando el archivo “evaluating_agenetic.py”.

6 Conclusiones y trabajo futuro

En este entregable se ha propuesto una metodología que permite a un LLM utilizar herramientas externas como fuentes de información factual para mejorar sus resultados. Siguiendo la misma línea que en la primera versión del software, se ha optado por una estrategia experimental que

⁷ <https://github.com/oeg-upm/inesdata-problem-solving>

no requiere procesos de entrenamiento, atendiendo a las necesidades multilingües del proyecto y a la dificultad de generar datasets específicos para diferentes casos de uso en diversos dominios. Esta elección permite superar los desafíos asociados a la creación de datasets y a la organización de entrenamientos masivos, como se ha observado en trabajos previos.

La capacidad de dividir el flujo de trabajo tradicional end-to-end en tareas más sencillas consistentes en pasos de razonamiento estructurados habilita al LLM para interactuar externamente dentro de un flujo de trabajo más complejo. En el marco del proyecto, se ha logrado integrar métodos de resolución de problemas (PSM) con estrategias de recuperación de información factual mediante RAG, utilizando herramientas externas. El enfoque inicial se ha aplicado al dominio médico, específicamente en tareas de razonamiento clínico y diagnóstico, evaluando esta estrategia sobre un conjunto de datos público.

Los resultados de la evaluación han demostrado que esta metodología permite al modelo reflexionar de manera efectiva sobre los problemas, identificar y obtener de forma autónoma la información necesaria para completar el proceso de razonamiento, y generar predicciones con la solución más probable. Todo esto se ha logrado exclusivamente a través de técnicas de prompting, incorporando además consideraciones sobre propiedades no funcionales, como la capacidad del modelo para reducir de manera autónoma la cantidad de datos que necesita recuperar para responder a preguntas específicas.

Finalmente, cabe destacar la flexibilidad de esta estrategia, que puede adaptarse a prácticamente cualquier dominio y aprovechar cualquier herramienta disponible. La única condición es diseñar estrategias de prompting que permitan al modelo utilizarlas eficazmente, dotándolo de capacidad de autorreflexión para evaluar su necesidad real de recursos, lo que contribuye a reducir la carga de trabajo y los costos computacionales.

La creación de agentes se perfila como una de las tecnologías más prometedoras en el ámbito de los LLMs, con constantes avances en esta línea de investigación. Un posible enfoque para trabajos futuros consiste en el modelado de este proceso completamente dentro de una arquitectura basada en agentes, lo que permitiría optimizar aún más tanto la precisión como la autonomía de los LLMs.

References

- Agerri, R., Alonso, I., Atutxa, A., Berrondo, A., Estarrona, A., Garcia-Ferrero, I., ... & Yeginbergenova, A. (2023). **Hitz@ antidote: Argumentation-driven explainable artificial intelligence for digital medicine**. arXiv preprint arXiv:2306.06029.
- Yasunaga, M., Chen, X., Li, Y., Pasupat, P., Leskovec, J., Liang, P., ... & Zhou, D. (2023). **Large language models as analogical reasoners**. arXiv preprint arXiv:2310.01714.
- Xiong, G., Jin, Q., Lu, Z., & Zhang, A. (2024). **Benchmarking retrieval-augmented generation for medicine**. arXiv preprint arXiv:2402.13178.
- Chen, J., Gui, C., Gao, A., Ji, K., Wang, X., Wan, X., & Wang, B. (2024). **CoD, Towards an Interpretable Medical Agent using Chain of Diagnosis**. arXiv preprint arXiv:2407.13301.
- Chen, H., Fang, Z., Singla, Y., & Dredze, M. (2024). **Benchmarking Large Language Models on Answering and Explaining Challenging Medical Questions**. arXiv preprint arXiv:2402.18060.
- McDuff, D., Schaekermann, M., Tu, T., Palepu, A., Wang, A., Garrison, J., ... & Natarajan, V. (2023). **Towards accurate differential diagnosis with large language models**. arXiv preprint arXiv:2312.00164.
- Ledley, R. S., & Lusted, L. B. (1959). **Reasoning foundations of medical diagnosis: symbolic logic, probability, and value theory aid our understanding of how physicians reason**. *Science*, 130(3366), 9-21.
- Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., ... & Ganapathy, R. (2024). **The llama 3 herd of models**. arXiv preprint arXiv:2407.21783.
- Fansi Tchango, A., Goel, R., Martel, J., Wen, Z., Marceau Caron, G., & Ghosn, J. (2022). **Towards trustworthy automatic diagnosis systems by emulating doctors' reasoning with deep reinforcement learning**. *Advances in Neural Information Processing Systems*, 35, 24502-24515.
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2022). **React: Synergizing reasoning and acting in language models**. arXiv preprint arXiv:2210.03629
- Asai, A., Wu, Z., Wang, Y., Sil, A., & Hajishirzi, H. (2023). **Self-rag: Learning to retrieve, generate, and critique through self-reflection**. arXiv preprint arXiv:2310.11511.
- Benjamins, V. R. (1993). **Problem Solving Methods for Diagnosis**. PhD thesis, University of Amsterdam, Amsterdam, The Netherlands.
- Benjamins, V. R., & Fensel, D. (1998). **Problem-solving methods**. *International Journal of Human-Computer Studies*, 49(4), 305-313.
- Chern, I. et al. (2023). **Factool: Factuality detection in generative ai – a tool augmented framework for**
- Fensel, D.A. (2000). **Problem-Solving Methods: Understanding, Description, Development, and Reuse**. *Lecture Notes in Computer Science 1791*, Springer 2000, ISBN 3-540-67816-6
- Gallegos, I. O., Rossi, R. A., Barrow, J., Tanjim, M. M., Kim, S., Dernoncourt, F., ... & Ahmed, N. K. (2023). **Bias and fairness in large language models: A survey**. arXiv preprint arXiv:2309.00770.
- Gao, S., et al. (2024). **Efficient Tool Use with Chain-of-Abstraction Reasoning**. arXiv:2401.17464
- Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., ... & Wang, H. (2023). **Retrieval-augmented generation for large language models: A survey**. arXiv preprint arXiv:2312.10997.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022). **Large language models are zero-shot reasoners**. *Advances in neural information processing systems*, 35, 22199-22213.
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). **Retrieval-augmented generation for knowledge-intensive nlp tasks**. *Advances in Neural Information Processing Systems*, 33, 9459-9474.
- Li, K., Patel, O., Vi'egas, F., Pfister, H., & Wattenberg, M. (2023). **Inference-Time Intervention: Eliciting Truthful Answers from a Language Model**. ArXiv, abs/2306.03341.
- Parisi, A. & Zhao, Y. & Fiedel, N. (2022). **TALM: Tool Augmented Language Models**. 10.48550/arXiv.2205.12255.

Schick, T., et al. (2023). **Toolformer: Language Models Can Teach Themselves to Use Tools**. ArXiv, abs/2302.04761.

Schreiber, G. et al. (1994). **CommonKADS: A comprehensive methodology for KBS development**. IEEE Expert. 9. 28-37. 10.1109/64.363263.

Schreiber, G. (2000). **Knowledge engineering and management: the CommonKADS methodology**. MIT press.

Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R.K., & Lim, E. (2023). **Plan-and-Solve Prompting: Improving Zero-Shot Chain-of-Thought Reasoning by Large Language Models**. Annual Meeting of the Association for Computational Linguistics.

Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K. W., & Lim, E. P. (2023). **Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models**. arXiv preprint arXiv:2305.04091.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., ... & Zhou, D. (2022). **Chain-of-thought prompting elicits reasoning in large language models**. Advances in Neural Information Processing Systems, 35, 24824-24837

Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023). **Tree of thoughts: Deliberate problem solving with large language models**, may 2023. *arXiv preprint arXiv:2305.10601*, 14.