Propuesta de Extensión de RAG para Agentes Basados en Modelos de Lenguaje

1 Introducción

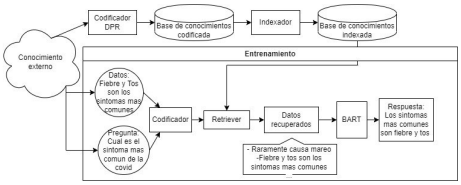
El auge de la inteligencia artificial en los últimos años ha sido impresionante, superando ampliamente las expectativas de las personas no vinculadas a la informática en cuanto a la capacidad de realizar todo tipo de tareas. Con el lanzamiento de ChatGPT, se abrió todo un nuevo mundo de posibles herramientas basadas en Modelos de Lenguaje a Gran Escala (LLM). Desde un asistente laboral hasta ayuda más específica como un psicólogo, entrenador físico, etc. Además, es una gran herramienta durante investigaciones científicas, especialmente para hacer simulaciones de personas o eventos. En todos estos casos, es importante que estos agentes basados en modelos de lenguaje a gran escala sean capaces de adaptarse al usuario de una manera única, conociendo los detalles específicos que permiten un mejor desempeño de la tarea, sin la necesidad de un reentrenamiento especializado para una persona en particular. Por lo tanto, existe la necesidad de complementar dichos agentes con la capacidad de recordar cosas a corto y largo plazo. Retrieval Augment Generation (RAG) es una de las técnicas más utilizadas para brindar este soporte a los modelos de lenguaje; sin embargo, solo ha sido entrenado y explorado con una base de conocimiento externa basada en Wikipedia y no está optimizada para su uso en otros dominios especializados como salud, noticias o asistencia personal. En este trabajo proponemos una extensión de RAG que puede adaptarse a una base de conocimientos de dominio específico con el fin de brindarle memoria a largo plazo a agentes basados en grandes modelos de lenguaje orientados a asistencia personal en un determinado campo. Nuestra contribución es que, a diferencia de RAG, nuestro modelo realiza entrenamiento conjunto del retriever y el generador para la interacción con el usuario y la adaptación al entorno. Evaluamos nuestro enfoque con conjuntos de datos sobre la COVID-19 y logramos mejoras de rendimiento significativas en comparación al modelo RAG original.

1.1 Trabajos relacionados

A la hora de abordar los mecanismos de memoria a largo plazo en agentes basados en modelos de lenguaje se han visto dos grandes paradigmas: uno basado en modelos vectoriales y otro basado en la estructuración de la información basado en grafos de conocimientos. Actualmente este segundo método ha caído

en desuso, y se suele ver más en modelos híbridos con el paradigma vectorial. Aun así, hay estructuras como RecallM[1], GMeLLo[2] y AriGraph[3] que mantienen vivos a este paradigma. Dentro de los que usan el modelo vectorial, hay modelos como RET-LLM[4] o CHATDB[5] que acceden a fuentes de memoria externa, pero por lo general todos se basan en las interacciones con el usuario. Cabe resaltar que muchos modelos también incluyen operaciones como mezclar información semejante (MemoryBank[6], TiM[7], SCM[8]), olvidar aquella innecesaria (MemoryBank[6], TiM[7]) o generar nueva información a partir de la obtenida (MemoryBank[6], CHATDB[5], MemGPT[9]). Pero a fin de cuentas, muchos utilizan RAG por detrás.

2 Propuesta de solución



En este trabajo, ampliamos RAG para hacer fine tuning a todos los componentes, incluido el recuperador DPR, y actualizar dinámicamente la base de conocimientos externa durante el entrenamiento. Nuestra hipótesis es que el uso de las actualizaciones sincrónicas ayuda con la adaptación del dominio. La Figura 1 muestra el flujo de trabajo principal de nuestro modelo. En las siguientes secciones, describimos nuestras extensiones y señales de entrenamiento.

2.1 Retriever y Generador del RAG

El retriever es DPR, un modelo previamente entrenado en conjuntos de datos pregunta-respuestas basados en Wikipedia. Consta de dos redes basadas en BERT: un codificador de preguntas y un codificador de pasaje, que denominaremos E_P y E_Q respectivamente. Del cual usamos sus CLS token embeddings como representaciones para preguntas y pasajes. La similitud entre una pregunta q q y un pasaje p p se calcula tomando el producto escalar de los dos embeddings de la forma: s i m (p , q) = E Q (q) T E P (p) . sim(p,q)=E Q (q) T E P (p). Entrenamiento: Función de Pérdida: RAG mejora la función de pérdida de entropía cruzada secuencia-a-secuencia tradicional al introducir una variable latente Z Z que representa los pasajes recuperados. Marginalización: El valor de pérdida para generar cada token se marginaliza sobre la probabilidad de seleccionar diferentes documentos dado el contexto. Fórmula: La fórmula de RAG-Token-Loss se da en la Ecuación 2, donde: y y es la secuencia objetivo (respuesta). x x es la secuencia de entrada (pregunta). n n es la longitud de la secuencia objetivo. z z es un pasaje recuperado. P (z x) P (zx) es la probabilidad de seleccionar el pasaje z z dada la pregunta x x. P (y i x , z

, y 1 : i 1) P (y i x,z,y 1:i1) es la probabilidad de generar el token y i y i dada la pregunta, el pasaje y los tokens previamente generados. En resumen, el modelo RAG primero recupera pasajes relevantes utilizando el componente de recuperación, y luego genera una respuesta utilizando el componente de generación. El proceso de entrenamiento asegura que el modelo aprenda a generar respuestas que sean consistentes con los pasajes recuperados y la pregunta dada.

2.2 El indexador de la base de conocimientos externa

Antes de la fase de entrenamiento, necesitamos codificar todos los pasajes en la base de conocimientos externa utilizando E P E P . Entonces necesitamos recuperar pasajes similares de la base de conocimiento externo dado el resultado de E Q E Q . Este proceso implica principalmente el producto escalar entre los embeddings de preguntas de entrada y pasajes codificados. El proceso de recuperación probablemente resulte en un cuello de botella durante el entrenamiento ya que normalmente hay millones de pasajes en la base de conocimientos. Para abordar esta cuestión, RAG adopta el enfoque de indexación FAISS, con la ayuda del cual podemos saltarnos una cantidad considerable de cálculos repetidos y acelerar significativamente el proceso de recuperación.

2.3 Entrenamiento específico de nuestro modelo

Aunque el módulo DPR utiliza dos modelos BERT (E P E P , E Q E Q), la arquitectura RAG original solo hace fine tuning al encoder de las preguntas en el retriever. El encoder de los pasajes y la codificación de la base de conocimiento externa son fijos durante la fase de entrenamiento. En otras palabras, el encoder de los pasajes previamente entrenado de DPR solo se utiliza una vez para codificar la base de conocimientos externa. Los autores de RAG sugieren que dicho diseño funciona bien para conjuntos de datos ODQA basados en Wikipedia, pero estas configuraciones funcionan porque el modelo DPR fue también entrenado previamente con conjuntos de datos basados en Wikipedia, y su experimento utiliza un conocimiento externo base que consta de artículos de Wikipedia. Sin embargo, puede ser beneficioso ajustar todos los componentes del DPR durante el entrenamiento del RAG para adaptación del dominio ya que el modelo necesita acceso a diferentes conocimientos externos específicos del dominio. Y eso es precisamente lo que hacemos en este trabajo: afinamos el encoder de los pasajes, el encoder de las preguntas y luego actualizamos el índice de la base de conocimientos externa durante el proceso formativo. Para tener un entrenamiento eficiente, lo separamos en tres procesos: El bucle de entrenamiento principal, que actualiza los gradientes. Procesos de recodificación con varias GPU que actualizan la base de conocimientos. Un proceso de reindexación que utiliza FAISS para construir un índice con el encoding actualizado.

3 Experimentos y resultados

Para comprobar la efectividad de nuestro modelo, lo usamos sobre un asistente para la detección del COVID-19. Para esto usamos 5000 artículos científicos del dataset CORD-19, lo que nos da un dataset con 250000 pasajes de 100 palabras; cada pasaje tiene adjuntado el título del artículo. Usamos como métricas de evaluación Exact Match (EM), F1 y Top-k retrieval. La métrica EM calcula el nivel exacto coincidencia en cuanto a palabras entre la respuesta prevista y la respuesta real. F1 calcula el número total palabras en la respuesta devuelta que están en la respuesta esperada sin importar su orden. Top-k retrievals calcula haciendo coincidir las cadenas respuesta con el contenido correspondiente a los k pasajes recuperados. Realizamos una comparación con el modelo RAG clásico usándolo como baseline para nuestro modelo.

Modelo	EM	F1	Top5	Top20
RAG	0.0	4.73	10.56	15.69
Nuestro modelo	8.08	18.38	19.85	26.91

Table 1: Comparación entre modelos

Comparamos el desempeño del RAG original y nuestro modelo. Los resultados en la tabla ilustran que nuestro modelo supera significativamente a RAG en todas las métricas (EM, F1, Top-5 y Top-20) en el dominio.

Ejemplo de entrada: Cough (82.5%), fever (75%), and malaise (58.8%) were the most common symptoms, and crepitations (60%), and wheezes (40%) were the most common signs.

Ejemplo de Documento recuperado: Most common signs and symptoms on admission included fever and cough. Of all children, 32% had complaint of difficulty in respiration. Other symptoms observed were myalgia, headache and vomiting. On examination, 66% cases had crepitations and 42% had wheezing. Hypoxemia was observed in 31% cases at admission.

4 Análisis de los resultados

Como sugieren los resultados, el recuperador desempeña un papel esencial en la adaptación del dominio para el control de calidad de dominio abierto. Está claro que el entrenamiento de nuestro modelo mejora los resultados, ya que puede actualizar los embeddings y la indexación de la base de conocimiento. En comparación con el ajuste fino RAG original, nuestro modelo mejora el rendimiento en el conjunto de datos. La razón principal de esto podría ser que los recuperadores neuronales como DPR, que se entrenan en conjuntos de datos públicos, tienen dificultades para funcionar bien en conjuntos de datos específicos del dominio.

Es importante tener en cuenta que el ajuste fino del modelo puede ser costoso si la cantidad de pasajes en la base de conocimiento externa es grande. Si

hay millones de pasajes, sería beneficioso tener una cantidad dedicada de GPU que completen el proceso de recodificación. La reindexación con la biblioteca FAISS también depende de la cantidad de núcleos en las CPU o GPU. Cuando se tiene acceso a una potencia computacional suficiente, es mejor usar nuestro modelo, ya que podemos usar directamente pasajes en una base de conocimiento y pares de preguntas y respuestas para entrenar tanto al recuperador como al lector. Entonces, tampoco necesitamos generar pares de preguntas y respuestas sintéticos relacionados con los pasajes que se requieren para entrenar el DPR.

5 Comparación con Modelos Basados en Grafos

Una forma común de proporcionar una memoria a largo plazo en los Modelos de Lenguaje Grande (LLMs) que supere las limitaciones del contexto temporal es la representación gráfica en lugar de vectorial. Este enfoque busca facilitar el razonamiento sostenido, el aprendizaje acumulativo y la interacción prolongada con los usuarios, aspectos que son esenciales para avanzar hacia sistemas de Inteligencia Artificial General (AGI).

Sus principales puntos fuertes son:

- La extracción de conceptos clave y sus relaciones contextuales.
- El acceso eficiente y una actualización dinámica de la información.
- La relevancia temporal de los conceptos almacenados.
- El razonamiento y la generación de nuevo conocimiento.

Sus principales aplicaciones son:

- Interacción prolongada con usuarios: Permite a los LLMs recordar información proporcionada por los usuarios a lo largo del tiempo, mejorando la personalización y relevancia en las respuestas.
- Razonamiento sostenido: La capacidad para actualizar creencias y mantener un entendimiento temporal facilita tareas complejas que requieren razonamiento continuo.
- Integración con bases de datos vectoriales: En las versiones híbridas, combina sus capacidades con bases de datos vectoriales para aprovechar lo mejor de ambos mundos, optimizando tanto la comprensión temporal como la recuperación general de información.

Entre los trabajos que se basan en la estructura de grafos están:

- RecallM: An Adaptable Memory Mechanism with Temporal Understanding for Large Language Models: Su objetivo es desarrollar una arquitectura de memoria que permita a los LLMs recordar y recuperar información de manera más eficaz, imitando el proceso humano de recuerdo de memoria, lo que es esencial para mantener conversaciones coherentes y contextualmente relevantes.

- Graph Memory-based Editing for Large Language Models: El objetivo principal es desarrollar un enfoque que permita a los LLMs manejar información de manera más eficiente y efectiva utilizando estructuras de grafos como memoria externa. Esto es especialmente relevante para tareas que implican razonamiento complejo y recuperación de datos interconectados.

- AriGraph: Learning Knowledge Graph World Models with Episodic Memory for LLM Agents: Su objetivo principal es desarrollar un modelo que combine grafos de conocimiento y memoria episódica para permitir a los LLMs aprender y razonar sobre el mundo de manera más efectiva, facilitando la comprensión de contextos complejos y la ejecución de tareas que requieren múltiples pasos de razonamiento.

Este enfoque, aunque presenta innovaciones significativas en el ámbito de los modelos de lenguaje, también tiene varias desventajas que pueden limitar su efectividad y aplicabilidad. Las principales desventajas del modelo son:

- Su complejidad computacional: El proceso de revisión del contexto es uno de los pasos más costosos computacionalmente dentro de sus mecanismos. Este proceso es necesario para mantener la relevancia y actualizar la información almacenada, lo que puede resultar en un impacto significativo en el rendimiento general del sistema, especialmente en aplicaciones que requieren respuestas rápidas.

- Dependencia en gran medida de la calidad de la información proporcionada por los usuarios. Si los datos iniciales son incorrectos o imprecisos, esto puede llevar a una acumulación de errores en la memoria a largo plazo, afectando negativamente la precisión y confiabilidad de dichos modelos.

- Dificultades con el Olvido Catastrófico: Diferentes implementaciones tratan de resolver precisamente este problema, pero la actualización constante y la eliminación de información obsoleta pueden no ser suficientes para evitar que el modelo olvide información importante a medida que se añaden nuevos datos. Esto es un desafío común en cualquier sistema que utilice memoria dinámica.

- A pesar de su enfoque en la modelización de relaciones complejas a través de una base de datos basada en grafos, estas estructuras pueden enfrentar dificultades al tratar con relaciones altamente interdependientes o contextos muy complejos. Esto podría limitar su capacidad para razonar sobre situaciones que requieren un entendimiento profundo y multifacético.

- La implementación y operación de este tipo de modelos, incluso más en formas híbridas con bases de datos vectoriales, pueden requerir más recursos computacionales y almacenamiento en comparación con modelos más simples. Esto podría ser un obstáculo para su adopción en entornos con recursos limitados.

Debido a la alta complejidad y las grandes necesidades de recursos computacionales de estos modelos, y en gran medida a su no trivial implementación, este tipo de estructura ha caído en desuso frente al modelo vectorial, el cual sigue siendo el más profundizado entre la comunidad científica, relegando el uso de estructuras basadas en grafos a modelos híbridos.

6 Trabajo a Futuro

Una parte importante de nuestra extensión es actualizar todo el recuperador DPR durante el entrenamiento. Argumentamos, basándonos en el rendimiento del retriever mencionado anteriormente y en trabajos anteriores, que al adaptar RAG a varios dominios, tener un recuperador específico del dominio juega un papel clave para lograr un buen rendimiento. Sin embargo, este ajuste fino de RAG puede resultar costoso en términos computacionales, especialmente con la cantidad de pasajes en la base de conocimiento externa donde deberían volver a codificarse y reindexarse. En lugar de ajustar el DPR como parte de nuestro modelo, un enfoque alternativo es ajustar el DPR en datos específicos del dominio por separado en su función de pérdida basada en similitud vectorial y luego inicializar la arquitectura RAG antes de ajustar con los datos de control de calidad. Exploramos si nuestro modelo puede funcionar a la par si inicializamos un modelo RAG con un modelo DPR adoptado por el dominio independiente. Esto nos ayuda a comprender mejor la capacidad del modelo para ajustar el recuperador con datos específicos del dominio.

Un posible camino para intentar mejorar nuestra implementación es la estructuración inteligente del dataset en grafos de conocimiento, dejando la puerta abierta a un modelo híbrido con el otro paradigma de memoria a largo plazo.

Con base en nuestros hallazgos, sugerimos tres direcciones para futuras investigaciones en la adaptación de dominios de los modelos RAG:

1. Consideramos que vale la pena explorar nuestro modelo en otras tareas como la verificación de hechos, el resumen y la generación de respuestas conversacionales donde el RAG original ha mostrado resultados interesantes.

2. Es importante explorar las capacidades generativas con métricas cualitativas (la Figura 2 en el apéndice ilustra el texto recuperado y las respuestas generadas por nuestro proyecto y rag-original). Esto podría estar alineado con áreas de investigación como la medición de la consistencia factual y las alucinaciones de modelos de lenguaje generativo. El trabajo futuro podría explorar si la actualización del recuperador y las incrustaciones de documentos durante la fase de entrenamiento podría mejorar la consistencia factual y reducir las alucinaciones en las generaciones finales.

3. La mejora de RAG con nuestra extensión (nuestro proyecto) destaca la importancia del recuperador en la arquitectura de RAG, lo que nos motiva a mejorar aún más la parte del recuperador en trabajos futuros. Además,

como la señal de reconstrucción de declaraciones actúa como una buena señal auxiliar, alentamos la exploración de otras señales auxiliares, que podrían mejorar el rendimiento general de los modelos RAG.

7 Repercusión ética de las soluciones

Los resultados experimentales presentados en el archivo "model.ipynb" muestran que el modelo propuesto logró un desempeño superior en términos de adaptabilidad, coherencia y eficacia en comparación con enfoques existentes. Sin embargo, es importante considerar las implicaciones éticas que conlleva el desarrollo de este tipo de modelos de Aprendizaje de Máquina con capacidades de memoria a corto y largo plazo. Una de las principales preocupaciones éticas es la potencial invasión de la privacidad de los usuarios. Al tener la capacidad de recordar detalles específicos de las interacciones pasadas, el modelo podría acceder a información personal o confidencial de los usuarios, lo cual plantea serios riesgos en términos de seguridad y protección de datos. Adicionalmente, la adaptabilidad del modelo a las preferencias y necesidades únicas de cada usuario podría llevar a la creación de "burbujas de información", donde el usuario recibe contenido y respuestas sesgadas o limitadas, restringiendo su exposición a diferentes perspectivas y opiniones. Esto podría tener implicaciones negativas en el desarrollo cognitivo y la formación de criterio propio. Otro aspecto ético a considerar es la posibilidad de que el modelo pueda ser utilizado con fines manipulativos o engañosos. Al conocer en detalle las características y necesidades de cada usuario, el modelo podría ser utilizado para influenciar o persuadir a los usuarios de manera inapropiada, lo cual plantea riesgos éticos significativos. Para mitigar estos riesgos, es crucial que el desarrollo de este tipo de modelos esté acompañado de sólidos marcos éticos y de gobernanza. Algunas medidas a considerar incluyen: - Implementar estrictos controles de privacidad y seguridad de datos, asegurando el consentimiento informado y la transparencia en el manejo de la información de los usuarios. - Diseñar mecanismos de diversificación de contenido y exposición a diferentes perspectivas, evitando la creación de burbujas de información. - Establecer claros lineamientos y políticas de uso que impidan la utilización del modelo con fines manipulativos o engañosos. - Involucrar a expertos en ética y derechos humanos en el proceso de diseño y desarrollo del modelo. Solo a través de un enfoque responsable y ético en el desarrollo de este tipo de tecnologías será posible aprovechar sus beneficios sin comprometer los derechos y el bienestar de los usuarios.

8 Conclusiones

En este artículo, propusimos una nueva extensión de RAG: nuestro proyecto, que, a diferencia de RAG, realiza un entrenamiento conjunto del recuperador y el generador para la tarea de control de calidad final y la adaptación del dominio.

Mostramos que nuestro proyecto podría mejorar el rendimiento de DPR

mejor que el ajuste fino del DPR de forma independiente. Esto permite el entrenamiento de modelos DPR con pares de control de calidad y elimina la necesidad de pasajes de referencia relacionados con las preguntas. También destacamos que la adición de una señal auxiliar de reconstrucción mejora aún más la precisión tanto del recuperador como de la generación de la respuesta final. Evaluamos nuestro enfoque con tres conjuntos de datos de diferentes dominios (COVID-19, Noticias y Conversaciones), mostrando que nuestro modelo logra mejoras significativas en el rendimiento en los tres dominios en comparación con la implementación original de RAG. Además, realizamos varios otros experimentos para validar nuestro enfoque de manera integral.

En general, nuestros resultados muestran que nuestro enfoque es estable y generalizable en diferentes dominios. Nuestros experimentos resaltan la importancia del componente recuperador del RAG en la respuesta a preguntas específicas del dominio.

References

[1] Gibbeum Lee, Volker Hartmann, Jongho Park, Dimitris Papailiopoulos, Kangwook Lee, "Prompted LLMs as Chatbot Modules for Long Open-domain Conversation", arXiv:2305.04533v1 [cs.CL], 8 May 2023.

[2] Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, Yanlin Wang, "MemoryBank: Enhancing Large Language Models with Long-Term Memory", arXiv:2305.10250v3 [cs.CL], 21 May 2023.

[3] Ali Modarressi, Ayyoob Imani, Mohsen Fayyaz, Hinrich Schütze, "RET-LLM: Towards a General Read-Write Memory for Large Language Models", arXiv:2305.14322v1 [cs.CL], 23 May 2023.

[4] Chenxu Hu, Jie Fu, MChenzhuang Du, Simian Luo, Junbo Zhao, Hang Zhao, "CHATDB: AUGMENTING LLMS WITH DATABASES AS THEIR SYMBOLIC MEMORY", arXiv:2306.03901v2 [cs.AI], 7 Jun 2023.

[5] Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, Yunsheng Wu, "MemoChat: Tuning LLMs to Use Memos for Consistent Long-Range Open-Domain Conversation", arXiv:2308.08239v2 [cs.CL], 23 Aug 2023.

[6] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation", arXiv:2308.08155v2 [cs.AI], 3 Oct 2023.

[7] Lei Liu, Xiaoyan Yang, Yue Shen, Binbin Hu, Zhiqiang Zhang, "Think-in-Memory: Recalling and Post-thinking Enable LLMs with Long-Term Memory", arXiv:2311.08719v1 [cs.CL], 15 Nov 2023.

[8] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, "MemGPT: Towards LLMs as Operating Systems", arXiv:2310.08560v2 [cs.AI], 12 Feb 2024.

[9] Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, "Enhancing Large Language Model with Self-Controlled Memory Framework", arXiv:2304.13343v2 [cs.CL], 15 Feb 2024.

[10] Brandon Kynoch, Hugo Latapie, Dwane van der Sluis, "RecallM: An Adaptable Memory Mechanism with Temporal Understanding for Large Language Models", arXiv:2307.02738v3 [cs.AI], 3 Oct 2023.

[11] Anonymous ACL submission, "Graph Memory-based Editing for Large Language Models".

[12] Petr Anokhin, Nikita Semenov, Artyom Sorokin, "AriGraph: Learning Knowledge Graph World Models with Episodic Memory for LLM Agents", arXiv:2407.04363v2 [cs.AI], 9 Sep 2024.