E4.2 LLM ADAPTADOS A DOMINIO

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

Resumen

Este entregable contiene los modelos adaptados y su aplicación a dominio, documentación adicional e información sobre el despliegue de los modelos adaptados en INESData hasta la fecha.

Raúl Ortega González
José Manuel Gómez Pérez

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

Revision History

Revision	Date	Description	Author (Organisation)
0.1	02/12/2024	Tabla de contenidos y estructura básica	Raúl Ortega
1.0	21/12/2024	Primera versión completa	Raúl Ortega
1.1	30/12/2024	Versión final revisada	José Manuel Gómez Pérez

Contents

1 Introducción

Este entregable recoge los recursos generados a lo largo del proyecto, que serán publicados en el Espacio de Datos de INESDATA una vez esté habilitado. A lo largo de las subsiguientes secciones se mostrarán aquellos modelos adaptados a dominio después de su entrenamiento siguiendo los métodos descritos en los entregables E2.1v2 (Berrio y Gómez-Pérez, 2024) y E4.3 (García y Gómez-Pérez., 2024), además de los conjuntos de datos con los que se ha llevado a cabo los diferentes ajustes allí descritos. De igual forma, este entregable contextualiza la elección de dichos modelos y métodos en el marco general del proyecto y ofrece información respecto a su integración en INESData. También se recogen los métodos de prompting presentados en el entregable E3.1v2 (Merenda y Gómez-Pérez, 2024) y su posible integración como recurso en el espacio de datos.

Así pues, el resto del documento se estructura de la manera siguiente. La sección 2 introduce los conjuntos de datos de entrenamiento y evaluación, donde mostraremos con qué datos hemos entrenado los modelos base del proyecto para mejorar su factualidad y su adaptación a dominio, así como los que hemos utilizado para adaptar modelos a la generación de ontologías en el dominio de las misiones espaciales. En la sección 3 recordamos los conjuntos de datos de verificación sobre los que hemos llevado la evaluación basada en referencias para los dominios de Seguros y Salud. En la sección 4, mostramos los modelos que hemos generado a partir de estos ajustes, además de la elección de los modelos base sobre los que se ha llevado a cabo. En la sección 5, mencionamos los recursos en forma de prompt con los que contamos para mejorar el desempeño en inferencia de los modelos para dominios específicos. Por último, en la sección 6, presentamos el estado actual de la integración de dichos modelos y recursos en el espacio de datos de INESData, así como los siguientes pasos del proyecto en relación con el soporte multilingüe de los mismos.

2 Conjuntos de datos de entrenamiento y evaluación

En este apartado recogemos los conjuntos de datos de entrenamiento para la adaptación a dominio y la factualidad de LLMs (E2.1v2, Berrio y Gómez-Pérez, 2024), tanto para el dominio de seguros como para el de salud, además de los conjuntos de datos utilizados para el entrenamiento de modelos con arquitectura Pythia para la generación de archivos RDF en formato turtle en el dominio espacial (E4.3, García y Gómez-Pérez, 2024). En total, **35 conjuntos de datos.**

2.1 Seguros

En este apartado mostramos los conjuntos de datos que hemos generado para el entrenamiento de un modelo base usando SFT y DPO, teniendo en cuenta cuatro estimadores: Llama+GPT4, GPT4+GPT4, Model Confidence y FactScore. Como mencionamos en el entregable E2.1v2, generamos conjuntos de datos de preferencia con el siguiente número de respuestas por prompt: 5, 10, 20, 30 y 40. La nomenclatura elegida para estos conjuntos de datos sigue este formato:

{DOMINIO}_{SFT/DPO}_{MODELO/ESTIMADOR}_{NUM_MUESTRAS}_{SUBCONJUNTO}.jsonl.

Para el caso de SFT, se mostrará el modelo con el que se ha generado, mientras que para DPO se hará referencia al estimador con el que se ha generado el dataset de preferencias. En ambos casos también se mostrará el número de muestras generadas para el entrenamiento y el subconjunto que esté representando, ya sea el de entrenamiento (train) como el de validación (val).

Para el entrenamiento SFT contamos con dos tipos de muestras. Las primeras generadas con Llama:

- **insurance_sft_llama_2k_train.jsonl/insurance_sft _llama_2k_val.jsonl.**
- **insurance_sft_llama_5k_train.jsonl/insurance_sft _llama_5k_val.jsonl.**
- **insurance_sft_llama_10k_train.jsonl/insurance_sft_llama_10k_val.jsonl.**
- **insurance_sft_llama_15k_train.jsonl/insurance_sft_llama_15k_val.jsonl.**
- **insurance_sft_llama_19k_train.jsonl/insurance_sft_llama_19k_val.jsonl.**

Y las que están generadas con GPT4o:

- **insurance_sft_gpt4o_2k_train.jsonl/insurance_gpt4o _llama_2k_val.jsonl.**
- **insurance_sft_gpt4o_5k_train.jsonl/insurance_gpt4o_llama_5k_val.jsonl.**
- **insurance_sft_gpt4o_10k_train.jsonl/insurance_gpt4o_llama_10k_val.jsonl.**
- **insurance_sft_gpt4o_15k_train.jsonl/insurance_gpt4o_llama_15k_val.jsonl.**
- **insurance_sft_gpt4o_19k_train.jsonl/insurance_gpt4o_llama_19k_val.jsonl.**

Para el entrenamiento DPO con el estimador de Llama+GPT4 generamos los siguientes conjuntos de datos:

- **insurance_dpo_lg_3k_train.jsonl/insurance_dpo_lg_3k_val.jsonl.**
- **insurance_dpo_lg_14k_train.jsonl/insurance_dpo_lg_14k_val.jsonl.**
- **insurance_dpo_lg_60k_train.jsonl/insurance_dpo_lg_60k_val.jsonl.**
- **insurance_dpo_lg_137k_train.jsonl/insurance_dpo_lg_137k_val.jsonl.**
- **insurance_dpo_lg_245k_train.jsonl/insurance_dpo_lg_245k_val.jsonl.**

Con el estimador de GPT4+GPT4 generamos los siguientes conjuntos de datos:

- **insurance_dpo_gg_5k_train.jsonl/insurance_dpo_gg_5k_val.jsonl.**
- **insurance_dpo_gg_21k_train.jsonl/insurance_dpo_gg_21k_val.jsonl.**
- **insurance_dpo_gg_89k_train.jsonl/insurance_dpo_gg_89k_val.jsonl.**
- **insurance_dpo_gg_208k_train.jsonl/insurance_dpo_gg_208k_val.jsonl.**
- **insurance_dpo_gg_366k_train.jsonl/insurance_dpo_gg_366k_val.jsonl.**

Con el estimador de Model Confidence generamos los siguientes conjuntos de datos:

- **insurance_dpo_mc_5k_train.jsonl/insurance_dpo_mc_5k_val.jsonl.**
- **insurance_dpo_mc_23k_train.jsonl/insurance_dpo_mc_23k_val.jsonl.**
- **insurance_dpo_mc_98k_train.jsonl/insurance_dpo_mc_98k_val.jsonl.**
- **insurance_dpo_mc_226k_train.jsonl/insurance_dpo_mc_226k_val.jsonl.**
- **insurance_dpo_mc_404k_train.jsonl/insurance_dpo_mc_404k_val.jsonl.**

Por último, con el estimador de Fact Score generamos los siguientes conjuntos de datos:

- **insurance_dpo_fs_5k_train.jsonl/insurance_dpo_fs_5k_val.jsonl.**
- **insurance_dpo_fs_23k_train.jsonl/insurance_dpo_fs_23k_val.jsonl.**
- **insurance_dpo_fs_81k_train.jsonl/insurance_dpo_fs_81k_val.jsonl.**

Además de los conjuntos de datos de entrenamiento en SFT y DPO, para llevar a cabo una evaluación de la factualidad de un modelo para un dominio en específico es necesario contar con el conjunto de preguntas basado en las entidades semilla que describimos en el entregable E2.1v2. Por ello, generamos una serie de conjuntos de datos (**insurance_questions_dataset_{train/val/test}.json**) que contenga estas entidades y sus preguntas asociadas, generadas mediante GPT3.5. Gracias a este conjunto de datos, es posible llevar a cabo entrenamientos y evaluaciones alternativas con otros modelos.

2.2 Salud

Los conjuntos de datos de este apartado fueron generados a partir de un conjunto de entidades seleccionadas dentro del vocabulario presente en Pubmed y UMLS. Dichas entidades fueron las de mayor presencia en artículos relacionados con el COVID-19 o SARS-CoV-2. A partir de dichas entidades, se generaron una serie de párrafos mediante el modelo base (Llama-2), que a su vez sirvieron como base para la formación de los conjuntos de datos para el entrenamiento de SFT y DPO con los distintos estimadores que aplicamos en este dominio: Llama+GPT4, Model Confidence y FactScore. En este caso, para el dataset de preferencias necesario para el entrenamiento de DPO se utilizó únicamente una muestra de 5 respuestas por prompt. A continuación, listamos todos estos conjuntos de datos para el entrenamiento en SFT y DPO:

- **covid_sft_train.jsonl/covid_sft_val.jsonl.** Conjuntos de datos necesarios para el entrenamiento de SFT.
- **covid_dpo_lg_train.jsonl/covid_dpo_lg_val.jsonl.** Conjuntos de datos necesarios para el entrenamiento de DPO con el estimador Llama+GPT4.
- **covid_dpo_mc_train.jsonl/covid_dpo_mc_val.jsonl.** Conjuntos de datos necesarios para el entrenamiento de DPO con el estimador de Model Confidence.
- **covid_dpo_fs_train.jsonl/covid_dpo_fs_val.jsonl.** Conjuntos de datos necesarios para el entrenamiento de DPO con el estimador de FactScore.

Al igual que en el dominio de seguros, publicaremos los conjuntos de datos de entidades y preguntas asociadas (**covid_questions_dataset_{train/val/test}.json**) para poder entrenar y evaluar diferentes modelos.

2.3 Espacio

En cuanto al dominio del espacio, siguiendo lo descrito en el entregable E4.3, se generó un conjunto de datos a partir del texto de misiones tomadas de EOportal para el ajuste de modelos Pythia a la hora de generar RDF en formato turtle para dichas misiones. Este conjunto de datos fue dividido en datos de entrenamiento (853 misiones) y datos de evaluación (172 misiones).

3 Conjuntos de datos de verificación

Tal y como mencionamos en el entregable E5.1 (Gómez-Pérez et al., 2024), la evaluación de la factualidad basada en referencias requiere de un conjunto de datos de verificación específico del dominio, para evaluar si una afirmación está soportada o refutada por el mismo.

En el caso del dominio de los seguros seguimos el procedimiento desarrollado por Min et al. (2023), que utiliza un volcado de la Wikipedia en inglés y las entidades semilla para encontrar artículos relacionados con los párrafos y afirmaciones generados a partir de ellas. Por su parte, en el dominio de Salud hemos seleccionado un subconjunto de abstracts provenientes de Pubmed para la generación de un índice de verificación de afirmaciones biomédicas.

4 Modelos

4.1 Selección de modelos base

Como ya mencionamos en el entregable E1.1 (Gómez-Pérez et al., 2024), hay que tener en cuenta diferentes factores a la hora de seleccionar un LLM sobre el que llevar a cabo una adaptación al dominio o una mejora de su factualidad: su tamaño, grado de apertura (código, datos y pesos del modelo) o a qué lenguajes da soporte. En ese mismo entregable, mencionamos diferentes familias de LLMs: **LLaMA** (Touvron et al., 2023) y su adaptación al euskera con Latxa, **Falcon** (Almazrouei et al., 2023) y su adaptación al castellano y catalán con Aguila, y **Chinchilla** (Hoffmann et al., 2022) y su adaptación al catalán y castellano con FLOR.

Por otro lado, durante el transcurso del proyecto se han presentado nuevos modelos de pesos abiertos y/o con soporte para las lenguas oficiales de España. Por ejemplo, **Carballo** (Gamallo et al., 2024) es un modelo de 2.1 billones de parámetros basado en FLOR y ajustado usando datos en gallego gracias al corpus CorpusNos. **OLMo** (Groeneveld et al., 2024) presenta otra familia de modelos de 1 y 7 billones de parámetros que han sido entrenados con el conjunto de datos en inglés de libre acceso dOLMa. Por último, mencionamos la familia de modelos de **Pythia** (Biderman et al., 2023), que destaca por la gran diversidad de versiones en cuanto a número de parámetros entrenables[1] con los que cuenta su arquitectura, si bien sólo están disponibles para el inglés.

Teniendo en cuenta el análisis realizado en el entregable E1.1 y lo reseñado en este, para el trabajo realizado hasta ahora hemos decidido centrar nuestro estudio de la adaptación de LLMs a dominio y de la mejora de la factualidad en dos de estas familias de modelos: **LLaMA** y **Pythia**. LLaMA nos proporciona una arquitectura de pesos abiertos con un uso extendido en la comunidad, con soporte multilingüe y una amplia gama de tamaños. Por otro lado, Pythia nos amplía esta diversidad de tamaños, dándonos la oportunidad de estudiar la influencia del número de parámetros en esta adaptación a dominio de los modelos.

[1] Pythia cuenta con versiones de 14, 70, 160 y 410 millones de parámetros, así como con versiones de 1, 1.4, 2.8, 6.9 y 12 billones de parámetros.

4.2 Seguros

Para el dominio de Seguros partimos de dos arquitecturas base: Llama y Pythia. Siguiendo estas dos arquitecturas, y utilizando los conjuntos de datos mencionados con anterioridad, generamos los **59 modelos** que veremos a continuación. La nomenclatura elegida para estos modelos sigue este formato:

{DOMINIO}_{SFT/DPO}_{MODELO_GENERACION/ESTIMADOR}_{NUM_MUESTRAS}_{MODELO_BASE}

Al igual que en la nomenclatura de los conjuntos de datos mencionada en el anterior apartado, para SFT se hará referencia al modelo usado para la generación (Llama o GPT4o) y para DPO al estimador utilizado (LG/GG/FS/MC). Por último, para ambos casos, se referenciará el número de muestras y el modelo base utilizado para el entrenamiento.

A partir del entrenamiento SFT, con Llama2 como modelo base, generamos:

- **insurance_ sft_llama_2k_llama2.**
- **insurance_ sft_llama_5k_llama2.**
- **insurance_sft_llama_10k_llama2.**
- **insurance_sft_llama_15k_llama2.**
- **insurance_sft_llama_19k_llama2.**
- **insurance_sft_gpt4o_2k_llama2.**
- **insurance_sft_gpt4o_5k_llama2.**
- **insurance_sft_gpt4o_10k_llama2.**
- **insurance_sft_gpt4o_15k_llama2.**
- **insurance_sft_gpt4o_19k_llama2.**

Con SFT, a partir de modelos Pythia y el entrenamiento con el conjunto de datos 10 muestras por prompt generado por Llama tenemos:

- **insurance_sft_llama_5k_pythia70m**
- **insurance_sft_llama_5k_pythia160m**
- **insurance_sft_llama_5k_pythia410m**
- **insurance_sft_llama_5k_pythia1B**
- **insurance_sft_llama_5k_pythia1.4B**
- **insurance_sft_llama_5k_pythia2.8B**
- **insurance_sft_llama_5k_pythia6.9B**
- **insurance_sft_llama_5k_pythia12B**

Del entrenamiento DPO con el estimador Llama+GPT4 obtenemos con Llama2 como modelo base:

- **insurance_dpo_lg_3k_llama2.**
- **insurance_dpo_lg_14k_llama2.**
- **insurance_dpo_lg_60k_llama2.**
- **insurance_dpo_lg_137k_llama2.**
- **insurance_dpo_lg_245k_llama2.**

Del entrenamiento DPO con el estimador Llama+GPT4 obtenemos a partir de la suite de modelos de Pythia:

- **insurance_dpo_lg_14k_pythia70m**
- **insurance_dpo_lg_14k_pythia160m**
- **insurance_dpo_lg_14k_pythia410m**
- **insurance_dpo_lg_14k_pythia1B**
- **insurance_dpo_lg_14k_pythia1.4B**
- **insurance_dpo_lg_14k_pythia2.8B**
- **insurance_dpo_lg_14k_pythia6.9B**
- **insurance_dpo_lg_14k_pythia12B**

Del entrenamiento DPO con el estimador GPT4+GPT4 obtenemos los siguientes modelos con Llama2 como modelo base:

- **insurance_dpo_gg_5k_llama2.**
- **insurance_dpo_gg_21k_llama2.**
- **insurance_dpo_gg_89k_llama2.**
- **insurance_dpo_gg_208k_llama2.**
- **insurance_dpo_gg_366k_llama2.**

Del entrenamiento DPO con el estimador de Model Confidence y Llama2 como modelo base:

- **insurance_dpo_mc_5k_llama2.**
- **insurance_dpo_mc_23k_llama2.**
- **insurance_dpo_mc_98k_llama2.**
- **insurance_dpo_mc_226k_llama2.**
- **insurance_dpo_mc_404k_llama2.**

Por último, del entrenamiento DPO con el estimador de FactScore y Llama2 como modelo base:

- **insurance_dpo_fs_5k_llama2.**
- **insurance_dpo_fs_23k_llama2.**
- **insurance_dpo_fs_81k_llama2.**

Además de todos estos modelos, también hemos ajustado un modelo **Llama2-7B en InsuranceQA** (Feng et al., 2015), un conjunto de datos que permite evaluar la factualidad de modelos en el dominio de seguros.

4.3 Salud

Para el dominio de Salud contamos con cuatro modelos, todos ellos entrenados usando LoRA y con un Llama2 como modelo de partida:

- **covid_sft_llama2.** Modelo Llama2 resultante del entrenamiento SFT.
- **covid_dpo_lg_llama2.** Modelo Llama2 resultante del entrenamiento DPO con el estimador Llama-GPT4.
- **covid_dpo_mc_llama2.** Modelo Llama2 resultante del entrenamiento DPO con el estimador de Model Confidence.

- **covid_dpo_fs_llama2.** Modelo Llama2 resultante del entrenamiento DPO con el estimador de FactScore.

4.4 Espacio

En cuanto al dominio del espacio, se ajustaron varios modelos de la arquitectura de Pythia con el conjunto de datos de misiones del EOportal. Después de este proceso, contamos con diez modelos Pythia capaces de generar RDF en formato turtle a partir de texto proveniente de misiones espaciales:

- **pythia-14m**
- **pythia-31m**
- **pythia-70m**
- **pythia-160m**
- **pythia-410m**
- **pythia-1B**
- **pythia-1.4B**
- **pythia-2.8B**
- **pythia-6.9B**
- **pythia-12B**

5 Prompts

Decir que estamos usando un enfoque basado en prompting y no en finetuning como se pensaba antes del inicio de la ejecución del proyecto (e3.1v2).

Decir con cuántos contamos.

En el entregable E3.1v2 de este proyecto (Merenda et al., 2024) se describe el desarrollo de una serie de estrategias en tiempo de inferencia, mediante un proceso de prompting, que permiten inyectar conocimiento haciendo uso de herramientas externas de forma autónoma por parte del modelo. Esto solventa los problemas mencionados en dicho entregable, como la necesidad de grandes datos de entrenamiento para las posibles llamadas externas o el coste computacional de otros enfoques.

Los prompts mostrados en dicho entregable, forman parte del dominio médico y están orientados a su uso en un flujo de trabajo PSP (Process Solving Prompting), que permite mejorar las capacidades de razonamiento de los modelos mediante la adicción de información de contexto, que sería recopilada mediante el uso de herramientas externas. Para extender su uso en este y otros ámbitos, compartiremos las plantillas utilizadas para formar estos prompts, una vez se habilite el espacio de datos de INESDATA.

6 Conclusiones y próximos pasos

A lo largo de este documento hemos ido recopilando los diferentes recursos generados del estudio de la mejora de la factualidad y la adaptación a tres dominios piloto: salud, seguros y espacio (ver Figura 1). Tanto los 59 modelos, como los 35 conjuntos de datos presentados,

estarán disponibles en el espacio de datos de INESDATA una vez esté habilitado. También lo estarán las plantillas utilizadas para las estrategias de prompting para el uso de herramientas externas.

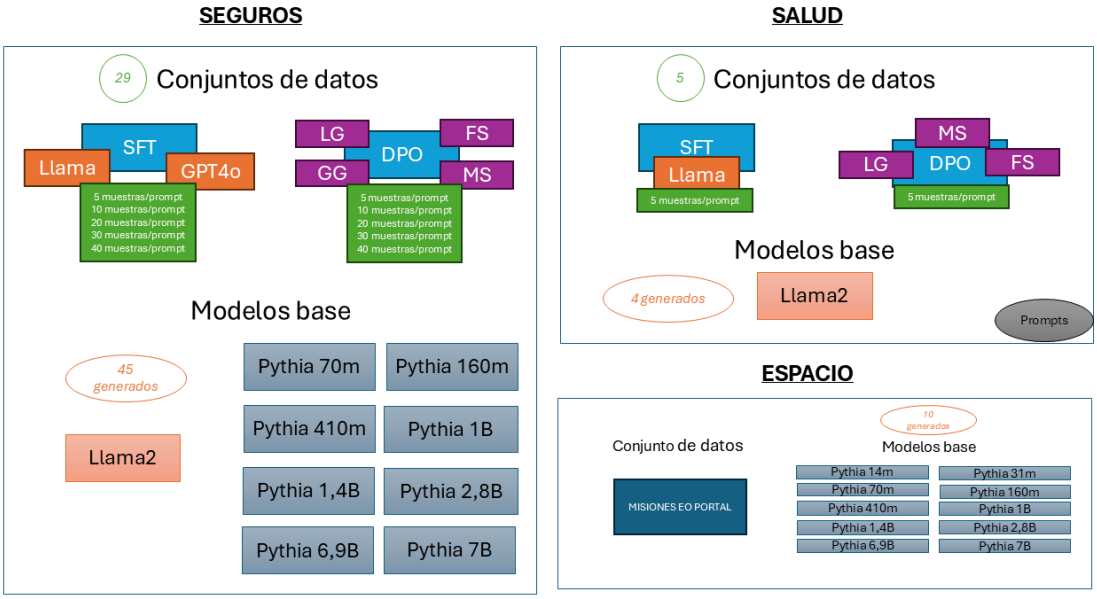

Figura 1. Resumen de los recursos generados por dominio.

Los siguientes pasos de este estudio pasan por el soporte multilingüe de los modelos y datasets correspondientes, en concreto en las lenguas del Estado español: castellano, catalán, euskera y gallego. Durante el tiempo restante del proyecto nos enfocaremos en la mejora de este aspecto, y generaremos una serie de nuevos recursos que también estarán disponibles en el mismo espacio de datos.

References

Almazrouei E., Alobeidli H., Alshamsi A., Cappelli A., Cojocaru R., Debbah M., Goffinet É., Hesslow D., Launay J., Malartic Q., Mazzotta D., Noune B., Pannier B., & Penedo G. (2023). **The Falcon Series of Open Language Models**. arXiv preprint arXiv:2311.16867

Biderman S., Schoelkopf H., Anthony Q., Bradley H., O'Brien K., Hallahan E., Khan M.A., Purohit S., Prashanth U.S., Raff E., Skowron A., Sutawika L., van der Wal O. (2023). **Pythia: A Suite for Analyzing Large Language Models Across Training and Scaling.** In *Proceedings of the 40th International Conference on Machine Learning (Pages 2397 – 2430). Honolulu*, Hawaii, USA.

Feng M., Xiang B., Glass M.R., Wang L. and Zhou B. **Applying deep learning to answer selection: A study and an open task.** (2015) *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Scottsdale, AZ, USA, 2015, pp. 813-820.

Gamallo P. et al. (2024). **Open Generative Large Language Models for Galician.** arXiv preprint arXiv:2406.13893v1

Gómez-Pérez, JM., Ortega, R., García A., Merenda F., Berrio C., (2024). **E1.1 Documento del estado de la cuestión, guías para la inyección de conocimiento en LLM y métricas**. KG4LLM Technical Report.

Berrio C., Gómez-Pérez, JM., (2024). **E2.1v2 Métodos de inyección de conocimiento en LLM**. KG4LLM Technical Report.

García A., Gómez-Pérez, JM., (2024). **E4.3v1 Extracción de grafos de conocimiento a partir de texto mediante LLM.** KG4LLM Technical Report.

Gómez-Pérez, JM., Berrio C., Ortega, R. (2024). **E5.1 Marco de Evaluación y Recursos Asociados.** KG4LLM Technical Report.

Groeneveld D., Beltagy I., Walsh E., Bhagia A., Kinney R., Tafjord O., Jha A., Ivison H., Magnusson I., Wang Y., Arora S., Atkinson D., Authur R., Chandu K., Cohan A., Dumas J., Elazar Y., Gu Y., Hessel J., et al. (2024). **OLMo: Accelerating the Science of Language Models.** In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Hoffmann J. et al. (2022). **Training Compute-Optimal Large Language Models.** arXiv preprint arXiv:2203.15556

Merenda, F. Gómez-Pérez, JM., (2024). **E3.1v2 Métodos de uso externo de herramientas externas por LLM.** KG4LLM Technical Report.

Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W.T., Koh, P., Iyyer, M., Zettlemoyer, L., and Hajishirzi, H. (2023). **Factscore: Fine-grained atomic evaluation of factual precision in long form text generation**.

Touvron, H. et al. (2023). **Llama 2: Open Foundation and Fine-Tuned Chat Models**. ArXiv, abs/2307.09288.