**Aplicación de herramientas de inteligencia artificial en el análisis de las normas regulatorias dictadas por la Autoridad Regulatoria Nuclear**

**A. Gómez\***(1), **V. Bettachini**(2), **H. Costa**(2), **M. Bringas**(2)e **I. Nesterov**(2)

*(1) Subgerencia Normativa Regulatoria, Autoridad Regulatoria Nuclear (ARN), Av. del Libertador 8250, Buenos Aires, Argentina*

*(2) Maestría en explotación de datos y descubrimiento del conocimiento, FCEyN-FI, UBA, Ciudad Universitaria, Buenos Aires, Argentina.*

\*Correo Electrónico (autor de contacto): [agomez@arn.gob.ar](mailto:autor@mmm.gob.ar)

**Tópico:** Normativa

Implementar herramientas de inteligencia artificial (IA), especialmente aquellas fundamentadas en el procesamiento del lenguaje natural (NLP), en un ámbito técnico específico cono las normas regulatorias nucleares, requiere expandir el alcance de los modelos entrenados con corpus de texto general. Este trabajo se organizó la normativa regulatoria vigente dictada por la Autoridad Regulatoria Nuclear (ARN) como un grafo dirigido de citas y exploró un método para hacerla más accesible mediante la técnica de generación aumentada por recuperación (RAG), apoyada en grandes modelos de lenguaje (LLM) de acceso libre.

El NLP es un subcampo clave de la inteligencia artificial (IA). Una de sus herramientas más extendidas, los LLM como ChatGPT o Gemini, se entrenan con grandes corpus de datos, como el conjunto de Wikipedia. En tales corpus la terminología técnica es escasa, por lo que para dominios especializados, como la actividad nuclear, los LLM suelen ser insuficientemente precisos [1].

En el marco de las tareas de la Subgerencia de Normativa Regulatoria de la ARN se realizaron dos desarrollos basados en NLP que mostraron resultados en el tratamiento de las 64 normas y 10 guías regulatorias que rigen la actividad nuclear en la República Argentina [2]. Para construir el corpus de texto, es decir, el conjunto de datos requerido para el NLP, se realizó un web scrapping del sitio de ARN que publica los documentos en formato PDF. Se utilizó la biblioteca Selenium del lenguaje Python para descargarlos secuencialmente y la biblioteca PyPDF2 para extraer su texto plano.

El primer desarrollo tuvo el objetivo de estudiar el clustering de los documentos según su temática, por ejemplo, 'Seguridad', 'Protección radiológica', 'Criterios de diseño', etc. Para esto se construyó un grafo dirigido en el que cada documento actúa como vértice y las citas como aristas. Mediante expresiones regulares se realizó la detección automática de citas a otras normas, lo que permitió crear un diccionario de citas cruzadas. Utilizando la biblioteca networkx, se generaron representaciones gráficas que facilitaron la cuantificación del grado de agrupamiento. La detección automática de un error topológico en el grafo de citas destacó el potencial del uso de la IA en el tratamiento de la documentación normativa, indicando una falta de actualización de la norma AR 4.8.1 que cita a la norma AR 3.6.1, la cual fue subsumida anteriormente en la norma AR 10.6.1.

El segundo desarrollo buscó construir un asistente basado en un LLM para responder consultas sobre información contenida en la normativa. Con los precursores de los LLM actuales, que contaban con a lo sumo 3 millones de parámetros, un sistema capaz de dar respuestas certeras sobre un dominio particular requería un ajuste fino supervisado en su correspondiente corpus particular. Tal reentrenamiento del LLM se ha vuelto muy demandante en términos de recursos informáticos y tiempo para los LLM con más de 3 mil millones de parámetros, como GPT-4o (OpenAI), Claude (Anthropic), Gemini (Google) o Llama 3.1 (Meta), entre otros. Para aplicar alguno de estos LLM al dominio específico de la normativa nuclear, es más recomendable construir un sistema de recuperación de texto RAG, que añade al LLM la capacidad de interactuar con una base de datos [3].

El LLM elegido para la construcción de nuestro RAG es el Llama 3.1, pues hasta la fecha, muestra el mejor desempeño en ensayos públicos con corpus en español [4]. Ante una consulta en lenguaje natural, dicho LLM oficiará de recuperador de la información alojada en una base de datos que contiene la representación vectorial (embeddings) del texto de las normativas segmentado en las unidades en que el LLM le divide, denominadas tokens. El entorno LangChain para lenguaje Python gestiona tal base, denominada Chroma, así como el recorte en tokens del corpus. Para oficiar de recuperador (retriever) el LLM es inicializado con una plantilla que le instruye a personificar a un experto en la normativa nuclear nacional. Tras esto, un usuario del RAG está habilitado a realizarle consultas (prompts) relacionadas con la temática nuclear y obtener una respuesta precisa ajustada a la normativa oficial.

**Referencias**

[1] Dima, Alden, Sarah Lukens, Melinda Hodkiewicz, Thurston Sexton, y Michael P. Brundage. 2021. “Adapting Natural Language Processing for Technical Text”. *Applied AI Letters* 2 (3): e33.<https://doi.org/10.1002/ail2.33>.

[2] Autoridad Regulatoria Nuclear. h[ttps://www.argentina.gob.ar/arn/instalaciones-practicas-y-personal-regulado/marco-regulatorio/normas](https://www.argentina.gob.ar/arn/instalaciones-practicas-y-personal-regulado/marco-regulatorio/normas)

[3] Aditya Jain, Amir Maleki, y Nathalie Saade. 2024. “To Fine-Tune or Not to Fine-Tune”. *Meta AI* (blog). 7 de agosto de 2024.<https://ai.meta.com/blog/when-to-fine-tune-llms-vs-other-techniques/>.

[4] “La Leaderboard - a Hugging Face Space by la-leaderboard”, Hugging Face. <https://huggingface.co/spaces/la-leaderboard/la-leaderboard>