**Tratamiento de la normativa de la ARN con herramientas de inteligencia artificial**

**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

Aplicar herramientas de IA, en particular las de NLP, a un dominio particular técnico como la normativa nuclear requiere extender el alcance de modelos entrenados con un corpus de texto general. Este trabajo trató la normativa de ARN modelándola en un grafo dirigido de citas y explorando un método de hacerla más accesible con RAG apoyado en un LLM de acceso libre.

El procesamiento de lenguaje natural (Natural Language Processing, NLP) es un fecundo subcampo de la inteligencia artificial (Artificial Intelligence, AI). El entrenamiento de grandes modelos de lenguaje (Large Language Models, LLM), e.g. ChatGPT o Gemini, con grandes corpus de terminología general les hace inespecíficos para el tratamiento de texto técnico, cómo el del acotado dominio de la actividad nuclear [1].

En este trabajo mostramos dos aplicaciones que mostraron resultados en el tratamiento de las 64 normas y 10 guías regulatorias publicadas por la Autoridad Regulatoria Nuclear (ARN). En conjunto, estas reglan la actividad nuclear en el país.

Para construir el corpus de texto, es decir, el conjunto de datos requeridos para el NLP, se realizó un "raspado web" (web scrapping) del sitio de ARN que la publica en formato PDF. Se utilizó la biblioteca Selenium del lenguaje Python para descargarles secuencialmente y con la biblioteca PyPDF2 se obtuvo su texto plano.

En primer término se buscó construir un grafo dirigido en el que cada documento oficia de nodo y las citas, las aristas. Con esto se buscó estudiar el agrupamiento (clustering) según el objeto del documento, e.g. "Seguridad", "Protección radiológica", "Criterio de diseño", etc., entre otros previamente clasificados en forma manual. La detección automática de citas a otras normas usando expresiones regulares (regex) permitió armar un diccionario de citas cruzadas. Utilizando la biblioteca networkx se pudieron realizar representaciones gráficas que permitieron cuantificar tal grado de agrupamiento.

La detección automática de un error topológico en el grafo de citas evidenció el potencial del tratamiento por IA de la normativa. La norma AR 4.8.1 cita a AR 3.6.1, que no está publicada. Tras advertir de ésto a responsables del área correspondiente en ARN, se aclaró que esta última norma fue subsumida en la 10.6.1, pero AR 4.8.1 sigue sin actualizarse.

Tras este análisis se construyó un asesor, basado en un LLM, que responda en lenguaje natural consultas sobre la información contenida en la normativa. Los antecesores de los LLM actuales de hace pocos años contaban con, a lo sumo, 3E6 parámetros, lo que hacía que adaptarles a un dominio particular requiriera un ajuste fino supervisado (supervised fine-tuning) en el corpus de texto de tal dominio. Desde el advenimiento de LLM con más de 2E9 parámetros, es muy demandante en términos de recursos tal reentrenamiento. El consenso actual es que si el corpus general de entrenamiento no resulta en un desempeño adecuado para un dominio específico, es conveniente construir un sistema de recuperación de texto por generación aumentada (Retrieval Augmented Generation, RAG) que suma al LLM la capacidad de interactuar con una base de datos [2].

Este es el caso de la normativa nuclear, cuyo nivel de especificidad, le excluye de estar contenida en los corpus de entrenamiento utilizados por los generadores de los LLM con más de 3E9 parámetros: GPT-4o (OpenAI), Claude (Anthropic), Gemini (Google) o Llama 3.1 (Meta) entre otros. Éste último es el seleccionado para este trabajo, pues a la fecha es el que muestra el mejor desempeño en ensayos públicos con corpus en lenguaje castellano [3]. Para este trabajo se aumentó el alcance de este LLM alojando en una base de datos la representación vectorial (embeddings) que realiza el LLM de los recortes de texto (tokens). El entorno (framework) LangChain para lenguaje Python gestiona la base Chroma, así como el recorte en tokens del corpus. Luego basta con ejecutar el recuperador (retriever) con una plantilla que intime al modelo a personificar a un experto en normativa nuclear antecediendo las consultas (prompts) que el usuario efectúa al RAG.

**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] 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/>.

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