Procesamiento de Lenguaje Natural
Proyecto Final: Primer Entrega

Rayo Mosquera, Jhon Stewar
j.rayom@uniandes.edu.co

De La Rosa Peredo, Carlos Raul
c.delarosap@uniandes.edu.co

Mario Garrido Córdoba
m.garrido10@uniandes.edu.co

Noviembre 2024

1 Introducción

El proyecto **Recuperación de Información Regulatoria y Generación de Respuestas (RIRAG)** tiene como objetivo el desarrollo de soluciones avanzadas para la recuperación y comprensión de información en contextos regulatorios, donde la precisión y relevancia de los datos son esenciales para la toma de decisiones. En entornos regulatorios, como el financiero o el legal, los profesionales requieren acceder a documentos normativos extensos y complejos, lo que genera la necesidad de herramientas de recuperación de información que no solo localicen pasajes relevantes, sino que también interpreten el contexto de las consultas.

Este informe documenta la primera entrega del proyecto, específicamente centrada en el *Subtask 1: Recuperación de Pasajes*. En esta fase, nuestro objetivo es mejorar el rendimiento de la recuperación de pasajes en comparación con el baseline basado en el algoritmo BM25 [1] proporcionado en [2] para la identificación de pasajes relevantes en consultas específicas. BM25 es una técnica ampliamente utilizada en recuperación de información que emplea la frecuencia de términos y la longitud de los documentos para ordenar los resultados. No obstante, su enfoque en coincidencias exactas limita su capacidad para identificar relaciones semánticas complejas, como sinónimos y términos relacionados, lo cual resulta esencial en el análisis de documentos legales.

Para abordar estas limitaciones, desarrollamos un enfoque que combina modelos sintácticos y semánticos, integrando técnicas tradicionales de recuperación con métodos basados en embeddings semánticos. Esta aproximación híbrida no solo busca superar la precisión del baseline, sino también aportar una solución robusta que pueda capturar la naturaleza multifacética de las consultas regulatorias. En particular, nuestro sistema integra un modelo de recuperación basado en BM25 y un modelo semántico ajustado, con el fin de optimizar tanto la precisión como la comprensión del contexto.

Este documento detalla los métodos y experimentos realizados para la optimización de la recuperación de pasajes, así como la metodología de preprocesamiento y entrenamiento empleada. Además, se presentan los resultados obtenidos y una discusión sobre las implicaciones de nuestro enfoque en tareas futuras de generación de respuestas, estableciendo así una base sólida para el Subtask 2.

2 Metodología

2.1 Descripción del Conjunto de Datos

El **conjunto de datos ObliQA**[2] es una colección de documentos regulatorios provenientes de *Abu Dhabi Global Markets (ADGM)*. Incluye 40 documentos legales y un conjunto de preguntas asociadas, distribuidas en conjuntos de entrenamiento (22,295), validación (2,786) y prueba (2,888). Cada pregunta está relacionada con uno o más pasajes relevantes dentro de los documentos, proporcionando una base sólida para tareas de recuperación y generación de respuestas.

2.2 Preprocesamiento del Texto

Llevamos a cabo las siguientes acciones:

- **Expansión de Contracciones:** Utilizamos la librería `contractions` para expandir términos como *don't* a *do not*, evitando ambigüedades.
- **Normalización y Limpieza:** Convertimos el texto a minúsculas y eliminamos caracteres no alfanuméricos y signos de puntuación mediante expresiones regulares con la librería `re`.
- **Eliminación de Espacios Redundantes:** Eliminamos espacios en blanco adicionales y caracteres de control para mantener la consistencia en el texto.
- **Preservación de Formato Legal:** Evitamos la normalización Unicode para conservar caracteres especiales que puedan ser relevantes en contextos legales.
- **Eliminación de Stopwords:** Combinamos las stopwords en inglés de `nltk`[3] y `scikit-learn`[4] para crear una lista adaptada al dominio, eliminando palabras comunes que no aportan significado.
- **Stemming:** Aplicamos el **algoritmo Snowball Stemmer**[5] para reducir las palabras a sus raíces, unificando diferentes formas de una misma palabra y reduciendo la dimensionalidad.
- **Tokenización:** Implementamos una función personalizada que genera **unigramas y bigramas** para capturar tanto términos individuales como combinaciones de palabras significativas.

2.3 Modelos Implementados

2.3.1 Modelo Sintáctico

BM25 Utilizamos la implementación de `rank_bm25`[6] para construir un modelo BM25 sobre el texto procesado. Este modelo considera la frecuencia de términos y la longitud del documento para asignar puntuaciones de relevancia. Configuramos los parámetros $k1 = 1.5$ y $b = 0.75$, valores comúnmente utilizados que ofrecen un equilibrio entre frecuencia y longitud.

2.3.2 Modelo Semántico

Seleccionamos el modelo `BAAI/bge-small-en-v1.5`[7], un transformer preentrenado capaz de generar embeddings semánticos de alta calidad. Realizamos *fine-tuning* utilizando el conjunto de datos procesado, enfocándonos en maximizar la similitud entre pares de preguntas y pasajes relevantes. Configuramos el modelo con las siguientes especificaciones:

- **Capa de Embeddings:** Utilizamos `SentenceTransformer`[8] con `torch_dtype=float16` para optimizar la memoria y velocidad.

- **Pooling y Normalización:** Incluimos capas de `Pooling` (utilizando el token [CLS]) y `Normalize` para mejorar la representación vectorial de los textos.
- **Incremento en la dimensionalidad**: Se aumentó la dimensión de los embeddings de 384 a 512 para propiciar una captura más fina de la semántica del dominio particular.

Para ajustar el modelo al dominio semántico de interés se utilizó PyTorch[9] a través de la librería SentenceTransformers, la cual provee todo un ecosistema de funcionalidades para entrenar y evaluar modelos de embeddings:

- **Configuración de Entrenamiento:**
  - **Épocas:** 10.
  - **Batch size:** 64 con `gradient_accumulation_steps=4` para simular un batch de 256.
  - **Learning rate:** 2×10^{-5}.
  - **Función de Pérdida:** Empleamos `MultipleNegativesRankingLoss`[10] para entrenar el modelo en tareas de recuperación de información considerando unicamente pares positivos.
- **Estrategias de Optimización:** Utilizamos `warmup_ratio=0.1` y desactivamos `gradient_checkpointing` para mejorar la eficiencia.
- **Evaluación Continua:** Empleamos `InformationRetrievalEvaluator` para monitorear el rendimiento en cada época, enfocándonos en métricas propias al problema de recuperación de información sobre el conjunto de validación.

En la tabla 1 se muestra el desempeño del modelo base y el modelo ajustado en los conjuntos de prueba y validación, evidenciando una mejora notable en el rendimiento tras el proceso de entrenamiento, además de una clara generalización sobre ambos conjuntos.

Modelo	**Dataset**	**Recall@10**	**MAP@10**
Modelo Base	validación	0.7135	0.5462
Modelo Base	prueba	0.7017	0.5357
Modelo Ajustado	validación	0.8158	0.6315
Modelo Ajustado	prueba	**0.8111**	**0.6261**

Tabla 1: Rendimiento comparativo entre el modelo base y el ajustado.

El modelo resultante ha sido publicado en Hugging Face Hub, permitiendo su uso abierto por la comunidad interesada en desarrollar o implementar tecnología para la recuperación y análisis de información en textos regulatorios en inglés. Adicionalmente, el repositorio con toda la metodología está disponible en GitHub.

2.3.3 Sistema Híbrido de Recuperación

Para combinar las ventajas de los modelos sintácticos y semánticos, desarrollamos un sistema híbrido que integra ambos enfoques mediante un promedio ponderado de sus puntuaciones:

$$\text{Puntuación Híbrida} = \alpha \times \text{Puntuación Semántica} + (1 - \alpha) \times \text{Puntuación Sintáctica}$$

Donde $\alpha = 0.65$ asigna mayor peso al componente semántico, este valor fue obtenido experimentalmente. Normalizamos las puntuaciones de ambos modelos entre 0 y 1 para asegurar una combinación equilibrada.

3 Experimentos y Resultados

3.1 Configuración Experimental

Realizamos los experimentos utilizando los conjuntos de datos de entrenamiento, validación y prueba proporcionados por ObliQA. Los modelos fueron entrenados y evaluados en una GPU NVIDIA A40 para acelerar el procesamiento.

3.2 Evaluación y Métricas

En la tabla 2 se muestran las métricas **Recall@10**, **MAP@10**, **Recall@20** y **MAP@20** para evaluar el rendimiento de los modelos. Estas métricas son estándar en tareas de recuperación de información y proporcionan una visión integral de la capacidad de los modelos para recuperar y ordenar pasajes relevantes. La evaluación se realizó con la herramienta `trec_eval`[11], siguiendo los estándares de la comunidad en IR (Information Retrieval).

3.3 Rendimiento Comparativo

Los resultados presentados en la tabla 2 muestran que todas las implementaciones propuestas superan el rendimiento del baseline, incluyendo el modelo basado únicamente en BM25. Esto evidencia que el preprocesamiento adicional y la optimización de los parámetros específicos contribuyen significativamente a mejorar la efectividad de la recuperación de información en el contexto regulatorio.

Modelo	**Recall@10**	**MAP@10**	**Recall@20**	**MAP@20**
BM25 (Baseline)	0.7611	0.6237	0.8022	0.6274
Modelo Sintáctico	0.7791	0.6415	0.8204	0.6453
Modelo Semántico	0.8103	0.6286	0.8622	0.6334
Sistema Híbrido	**0.8333**	**0.7016**	**0.8704**	**0.7053**

Tabla 2: Rendimiento comparativo de los modelos en el conjunto de prueba.

El modelo híbrido demostró ser la opción más efectiva, combinando la precisión de coincidencia del modelo sintáctico BM25 con la habilidad del modelo semántico para interpretar términos relacionados, logrando una mejor cobertura en la recuperación de pasajes relevantes. El repositorio con toda la metodología y resultados está disponible en GitHub.

4 Discusión

Los experimentos realizados confirman que la combinación de modelos sintácticos y semánticos representa una mejora notable en la recuperación de información dentro del dominio regulatorio. La implementación de BM25 resulta eficaz para recuperar pasajes con coincidencias exactas, lo que es particularmente útil cuando las consultas contienen términos específicos del lenguaje legal. Sin embargo, este modelo carece de la capacidad para reconocer sinónimos o términos relacionados que pueden ser fundamentales para comprender el contexto completo de una consulta regulatoria.

En contraste, el modelo semántico proporciona una ventaja significativa al capturar relaciones más profundas entre términos, logrando identificar similitudes contextuales entre palabras y frases que no coinciden de forma literal. Esta capacidad es crucial en dominios donde el lenguaje tiende a ser intrincado y con una alta dependencia de la interpretación. No obstante, esta aproximación semántica puede comprometer la precisión en coincidencias exactas, una característica que es frecuentemente necesaria en la recuperación

de información legal.

El sistema híbrido que desarrollamos busca equilibrar estas limitaciones combinando los puntos fuertes de ambos modelos. La elección de un valor de ponderación $\alpha = 0.65$ para el componente semántico subraya la importancia de captar la semántica dentro de este dominio, sin dejar de lado la precisión sintáctica. Este valor, obtenido de manera experimental, optimiza el rendimiento al asignar mayor peso al componente semántico, logrando así un balance efectivo que se refleja en un mejor desempeño en las métricas **Recall@10** y **MAP@10**.

Adicionalmente, observamos que el modelo semántico ajustado mantiene las capacidades generales del modelo base, lo cual es ventajoso para manejar consultas fuera del dominio específico del conjunto de datos, lo cual aumenta su flexibilidad y valor práctico para aplicaciones en entornos donde las consultas pueden variar en complejidad y terminología.

En conjunto, los resultados refuerzan la utilidad de un sistema híbrido para la recuperación de información en dominios especializados, donde la combinación de técnicas sintácticas y semánticas no solo incrementa la precisión en la identificación de pasajes relevantes, sino que también permite una cobertura más amplia de los matices legales y regulatorios.

5 Conclusiones

Hemos desarrollado un sistema de recuperación de pasajes que supera al baseline BM25 mediante la integración de modelos sintácticos y semánticos. Nuestro enfoque híbrido demuestra ser efectivo en capturar tanto coincidencias exactas como relaciones semánticas, lo cual es crucial en el contexto regulatorio. La integración de un modelo semántico ajustado permite mejorar la precisión en la recuperación de pasajes relevantes, incluso en consultas complejas y contextos específicos que requieren comprender la relación entre términos legales.

Los resultados obtenidos validan nuestra hipótesis y resaltan la importancia de combinar técnicas tradicionales y avanzadas en procesamiento de lenguaje natural. Este enfoque híbrido no solo mejora el rendimiento en métricas estándar como **Recall@10** y **MAP@10**, sino que también ofrece un método más robusto y adaptable para entornos regulatorios en los que la precisión es fundamental.

Además, este trabajo sienta una base sólida para abordar el Subtask 2, enfocado en la generación de respuestas. La capacidad del sistema para recuperar pasajes precisos y contextualmente relevantes permitirá una generación de respuestas más coherente y enfocada en las necesidades del usuario. Esta característica es esencial en el ámbito de la conformidad regulatoria, ya que permite responder preguntas complejas con información específica y fundamentada en documentos legales.

Finalmente, la publicación del modelo y el código en plataformas como Hugging Face Hub y GitHub no solo facilita la replicación de nuestra metodología y resultados, sino que también contribuye al desarrollo de la comunidad en el área de la recuperación de información y generación de respuestas.

Referencias

[1] Stephen E Robertson, Steve Walker, MM Beaulieu, Mike Gatford, and Alison Payne. Okapi at trec-4. *Nist Special Publication Sp*, pages 73–96, 1996.

[2] Tuba Gokhan, Kexin Wang, Iryna Gurevych, and Ted Briscoe. Regnlp in action: Facilitating compliance through automated information retrieval and answer generation, 2024.

[3] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc., June 2009.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[5] Martin F. Porter. Snowball: A language for stemming algorithms. 2001.

[6] Dorian Brown. Rank-BM25: A Collection of BM25 Algorithms in Python, 2020.

[7] Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. C-pack: Packaged resources to advance general chinese embedding, 2023.

[8] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[10] Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun hsuan Sung, Laszlo Lukacs, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. Efficient natural language response suggestion for smart reply, 2017.

[11] E. Voorhees, D. K. Harman, National Institute of Standards, and Technology (U.S.). Trec: experiment and evaluation in information retrieval, 2005.