Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
Primer Semestre de 2025

IIC 2440 – Procesamiento de Datos Masivos

Tarea 2

Instrucciones. Esta tarea debe resolverse en grupos de a dos personas.

El formato de entrega consta de los siguientes archivos

- Un archvio comprimido donde este todo el código usado para el trabajo (pero no los datos ni el código de los sistemas).
- Un informe donde se reporten los resultados.

Fechas. La fecha de entrega de la tarea es el 4 de Julio, a las 20:00 hrs.

Introducción. En la página del curso encontraras una vectorización (usando un modelo tipo transformer) del material que usaste para la tarea 1. La meta de esta tarea es montar distintas opciones para realizar RAG sobre estos datos. **Importante:** Es obligación para el curso utilizar el sistema MillenniumDB. Este sistema está en desarrollo continuo, por lo que puedes encontrar errores de funcionamiento (bugs). Si encuentras algún error, lo logras replicar y lo pones en el foro del curso, recibirás media décima de bono en la tarea.

Datos Vas a volver a trabajar con los datos del congreso, de la tarea 1. Tienes a tu disposición un archivo llamado $\mathrm{to}_q m_f ull.pyqueconviertelosdatosdelatarea1aformato.qm, queesloqueesperalabasededatos$.

Sigue las instrucciones para importar y crear una base de datos con esta información.

Luego de eso, investiga la estructura del grafo usando la interfaz de consulta de la base de datos.

Son 5 tipos de entidades: `:Parlamentario`, que corresponde a parlamentarios, `Partido`, correspondiente a partidos políticos de Chile, `Unidad`, correspondiente a pactos políticos de Chile, `Participacion`, correspondiente a los discursos que han dado los parlamentarios en el congreso, y `Embedding`, que contiene un vector asociado a un split de la una participación (cuando los textos son muy largos, se generan chunks y todos estos apuntan a su nodo participacion).

Para ver todos los nodos de un cierto tipo (en este caso parlamentarios), escribe:

```
MATCH (?n:Parlamentario)
RETURN ?n
LIMIT 10
```

Índice para LSH Para poder realizar consultas con LSH, debes primero decirle a la base de datos que construya el índice. En nuestro caso, se hace con la consulta:

```
CREATE HNSW INDEX "example" WITH {
  property: "value",
  dimension: 768,
  maxEdges: 8,
  maxCandidates: 16,
  metric: "cosineDistance"
}
```

El valor de property correspnde a qué propiedad se indexa, en este caso es la propiedad "value", que solo esta en los nodos tipo ."Embedding". El valor de la dimensión es el tamaño de los vectores, y los otros tres parámetros son internos al índice, puedes ver que el LSH se hace con distancia de coseno.

Llave OpenAI La llave de OpenAI será enviada por correo, al igual que la vez pasada.

Preguntas para el RAG Considera las siguientes preguntas:

1. Qué se ha dicho en el congreso sobre el litio?
2. Qué dicen los congresistas hombres sobre el aborto?
3. Qué opinión tienen los congresistas de la izquierda sobre la tenencia responsable de mascotas?
4. Qué opinión tienen los congresistas del frente amplio con respecto a la permisología?
5. Cómo cambia la opinión del congreso sobre las pensiones en los distintos años?
6. Una pregunta adicional que construyan como grupo.

Problema a resolver. La idea es poder facilitar a un modelo de lenguaje a que responda estas preguntas usando RAG.

En concreto, debes implementar dos modalidades de RAG (descritas más abajo), evaluar la respuesta a estas preguntas, y escirbir un reporte describiendo las diferencias en las respuestas, así como explicar por qué ocurren o no ocurren diferencias.

RAG Clásico Para el RAG Clásico, puedes cargar los vectores a MillenniumDB, usando un índice para realizar LSH, e implementa RAG como lo vimos en clases. Ten en cuenta que la estructura del grafo no almacena el texto de la participación directamente en el nodo al cuál se le asigna el vector, por lo que deberás navegar el grafo para traer los strings una vez que consigas los vectores más similares a la pregunta.

RAG con Knowledge Graph El segundo esquema a implementar es un RAG con Knowledge Graph. Este esquema puede ordenarse de la siguiente forma.

1. El usuario le entrega una consulta al LLM.
2. Esta consulta es llevada a un vector.
3. Usando el índice de vectores de MillenniumDB, se encuentran los 10 nodos más similares a esa consulta, y luego se ubica el nodo intervención correspondiente a ese vector, tal y como antes.
4. Para esos nodos, se ejecuta además una consulta que entrega toda la información del o la parlamentaria que haya hecho esta intervención, así como toda la información de donde y cuándo se hizo la intervención.
5. Toda esta información se entrega junto con el string mismo de la intervención (en un formato amigable para un LLM) y se pasa como contexto.
6. Tal y como en un RAG clásico, se le pide al LLM responder la consulta original utilizando el contexto que preparaste antes.