Capítulo 3: Desarrollo del Sistema

FALTA PONER EL KEYBERT

**5. Extracción de keywords con KeyBERT**

* Para cada fragmento limpio llamamos a KeyBERT (modelo español entrenado en dominio biomédico) y extraemos las 5 palabras o bigramas más representativos, con MMR para diversidad.

from keybert import KeyBERT

kw\_model = KeyBERT(model=keybert\_models['es'])

keywords = [kw for kw,\_ in kw\_model.extract\_keywords(

cleaned, top\_n=5, use\_mmr=True, diversity=0.2

)]

meta['palabras\_clave'] = list(set(meta.get('palabras\_clave', [])) | set(keywords))

FALTA DIVISION DE LOS CHUNKS

**7. División en chunks**

f"tokens avg={np.mean(tokens):.1f} (min={min(tokens)},max={max(tokens)})"

)

**8. Enriquecimiento de metadatos por chunk**

Para cada chunk generamos un diccionario metadatas con:

* document\_id (p.ej. "pdf\_nauseas\_vomitos\_adultos\_es", "web\_cancer-colorrectal")
* chunk\_index y chunk\_total
* chunk\_position (“3/13”)
* length\_chars, length\_tokens
* chunk\_terms (keywords que aparecen en ese fragmento)
* Heredamos campos originales (categoria, idioma, publico\_objetivo, etc.)

enriched = {

\*\*meta,

"chunk\_id": f"{meta['filename']}\_chunk{idx}",

"chunk\_position": f"{idx+1}/{total\_chunks}",

"length\_chars": len(chunk),

"length\_tokens": len(chunk.split()),

"chunk\_terms": ", ".join(matched\_terms[:5]),

* }

**3.1 Análisis del Problema y Decisiones Arquitectónicas**

**3.1.1 Requisitos del Sistema Médico**

El diseño de un sistema RAG especializado en información médica implica afrontar retos críticos de precisión semántica, multilingüe y escalabilidad. A continuación se resumen los requisitos clave:

* **Multilingüismo contextual:** consultas en español y euskera. Se emplea traducción neuronal manteniendo la precisión semántica médica.
* **Precisión terminológica:** interpretación correcta de términos clínicos e híbridos (coloquiales/profesionales).
* **Filtrado jerárquico:** por categoría y subcategoría médica.
* **Escalabilidad horizontal:** admisión de nuevos documentos sin degradación de rendimiento.
* **Robustez lingüística:** manejo de errores ortográficos y ambigüedades sintácticas.

Las decisiones de diseño del sistema se basaron en las mejores prácticas actuales de la industria, fundamentadas en formación especializada (ver Anexo A) y referencias técnicas específicas (ver Anexo B)

**3.2 Fuentes de Datos y Corpus Médico**

**3.2.0 Creación del Ground Truth**

Para evaluar el sistema de recuperación, se creó un dataset manual denominado dataset\_test.json, que actúa como ground truth. Este conjunto está compuesto por pares pregunta-respuesta categorizados, cubriendo las **9 categorías médicas principales** definidas en el corpus (como vacunas, alimentación, salud sexual, etc.). Este dataset con preguntas en tono coloquial como el lenguaje técnico contiene el corpus 44 documentos y 322 consultas

Cada ejemplo del dataset incluye:

* query: consulta del usuario
* categoria: categoría médica asociada
* relevantes: lista de document\_id o chunk\_id considerados relevantes

Esto permite evaluar la precisión de los distintos métodos de recuperación (TF-IDF, BGE, Fine-Tuning, CrossEncoder) bajo condiciones controladas y realistas.

*{*

*"queries": [*

*{*

*"text\_es": "¿Qué pruebas diagnósticas existen para detectar el cáncer de mama?",*

*"text\_eu": "Zer diagnostiko-proba daude bularreko minbizia detektatzeko?",*

*"relevant\_docs": [*

*"web\_cancer-de-mama",*

*"pdf\_cancer\_mama\_es",*

*"pdf\_cancer\_mama\_despues\_es",*

*"web\_programa-cancer-de-mama"*

*],*

*"categoria": "cancer",*

*"subcategoria": "cancer\_mama",*

*"dificultad": "dificil"*

*}*

*],*

*"corpus": {*

*pdf\_cancer\_mama\_es": "Presentación\nEl cáncer de mama es una enfermedad tumoral frecuente en la mujer\nque, además de poner en peligro su vida, incide en su imagen y pue-\nde tener repercusiones sicológicas… "*

*}*

*}*

**3.2.1 Selección de Fuentes**

El corpus se construyó a partir de dos tipos principales de fuentes:

* **PDFs:** documentos descargados de Osakidetza, asociaciones profesionales y laboratorios.
* **Web:** URLs del portal oficial de Osakidetza extraídas mediante scraping.

Cada documento fue etiquetado con metadatos estructurados: idioma, categoría, subcategoría, fuente, público objetivo y palabras clave.

**3.2.2 Extracción de PDFs**

La función extract\_pdf\_text() emplea PyMuPDF para recorrer todas las páginas del documento. Posteriormente se aplica clean\_text() que elimina hifenaciones, normaliza saltos de línea y colapsa espacios.

**3.2.3 Extracción Web Jerárquica**

La función extract\_web\_text() reconoce diferentes estructuras de la web de Osakidetza:

* **Pestañas horizontales** (nav-tabs)
* **Pestañas verticales** (nav-pills)
* **Acordeones** (panel-collapse)
* **Contenido plano** (fallback con limpieza de <script>, <nav>, etc.)

En todos los casos el texto resultante es limpiado y unificado antes del chunking.

**3.2.4 Traducción del Euskera**

El modelo Helsinki-NLP/opus-mt-eu-es se empleó para traducir contenido euskérico al español, manteniendo equivalencia semántica. Esto garantiza consistencia en el espacio vectorial de embeddings.

**3.2.5 Metadatos y Almacenamiento**

Se generaron dos ficheros JSON:

* pdf\_metadata.json
* web\_metadata.json

Contienen información estructurada por documento (fuente, categoría, idioma, público, palabras clave). Estos metadatos se integran en los chunks durante la indexación.

**3.3 Evaluación y Selección de Modelos de Embedding**

**3.3.1 Comparativa Inicial**

Se evaluaron los siguientes modelos con el dataset, dataset\_test.json. Los resultados obtenidos :

| ***Modelo*** | ***Precision@3*** | ***Recall@3*** | ***F1@3*** | ***MRR*** |
| --- | --- | --- | --- | --- |
| ***Qwen/Qwen3-Embedding-8B*** | *0.3364* | *0.7237* | *0.4387* | *0.7116* |
| ***BAAI/bge-m3*** | *0.3354* | *0.7256* | *0.4375* | *0.7003* |
| ***Fine-tuned: bge\_m3\_epochs/epoch4\_MRR0.9717*** | *0.3520* | *0.7551* | *0.4571* | *0.6806* |
| ***Fine-tuned: bio\_roberta\_epochs/*** ***epoch4\_MRR0.9726*** | *0.3012* | *0.6501* | *0.3926* | *0.6028* |
| *sentence-transformers/all-MiniLM-L6-v2* | *0.2723* | *0.5901* | *0.3541* | *0.5870* |
| *IIC/RigoBERTa-Clinical* | *0.1853* | *0.4170* | *0.2458* | *0.4061* |
| *PlanTL-GOB-ES/bsc-bio-ehr-es* | *0.1718* | *0.3759* | *0.2269* | *0.3941* |

Las métricas empleadas fueron: Precision@K, Recall@K, F1@K, MRR, nDCG.

Al analizar los resultados, observamos que **las diferencias entre los dos modelos líderes son mínimas. Lo que ha llevado a elegir bge\_m3 ha sido :**

* **BGE-M3**: 568M parámetros, eficiente en CPU **Qwen3**: 8B parámetros, requiere GPU para los experimentos

## Capítulo 3.3 – Fine-tuning del Modelo Biomédico

### 3.3.1 Motivación y Justificación Técnica

Durante la fase de experimentación inicial se empleó el modelo BAAI/bge-m3 como referencia, dada su robustez multilingüe y su rendimiento competitivo en tareas de recuperación. Sin embargo, debido a la naturaleza altamente especializada del lenguaje médico, se plantearon dos hipótesis contrastantes:

* H1 (Especialización): Un modelo biomédico preentrenado, ajustado con consultas médicas específicas, superará tanto a modelos generalistas como a su versión base.
* H2 (Generalización): Un modelo generalista robusto, con fine-tuning dirigido, puede superar a modelos especializados, especialmente con datasets limitados.

### 3.3.2 Estrategia Experimental y Configuración

Se diseñó un experimento dual para contrastar las hipótesis:

* Experimento A: Fine-tuning sobre el modelo biomédico PlanTL-GOB-ES/roberta-base-biomedical-es
* Experimento B: Fine-tuning sobre el modelo generalista BAAI/bge-m3

Ambos entrenamientos utilizaron la misma configuración para que los resultados fuesen comparables

|  |  |  |
| --- | --- | --- |
| **Parámetro** | **Valor** | **Justificación** |
| Learning Rate | 2e-5 |  |
| Batch Size | 16 |  |
| Épocas máximas | 10 | Early Stopping activado (paciencia = 3) |
| Func. de Pérdida | MultipleNegativesRankingLoss |  |
| Métrica de validación | MRR | Evalúa la posición del primer resultado relevante |

3.3.3 Dataset de entrenamiento

Se elaboró un dataset manual con 2.033 pares pregunta-respuesta en castellano, incluyendo lenguaje técnico (40%), coloquial (50%) y mixto (10%). Las consultas cubrían 8 categorías clínicas principales.

Ejemplos:

* Técnico: "¿Cuáles son los criterios diagnósticos para diabetes mellitus tipo 2?"
* Coloquial: "¿Tener el azúcar alto significa que soy diabético?"

La división fue 80% para entrenamiento y 20% para validación (almacenado en dataset\_finetune.json)

3.3.4 Implementación técnica del Fine-Tuning

El entrenamiento se realizó con la librería SentenceTransformers. La arquitectura implementada fue (ver 2-train\_with\_earlystoping.py):

*word\_embedding\_model = models.Transformer("PlanTL-GOB-ES/roberta-base-biomedical-es", max\_seq\_length=512)*

*pooling\_model = models.Pooling(word\_embedding\_model.get\_word\_embedding\_dimension(), pooling\_mode\_mean\_tokens=True)*

*model = SentenceTransformer(modules=[word\_embedding\_model, pooling\_model])*

La función de pérdida seleccionada fue MultipleNegativesRankingLoss, ideal para tareas de matching semántico donde no hay negativos explícitos. Esta función penaliza implícitamente al resto del batch como negativos.

3.3.5 Evaluación y Resultados

La evaluación se realizó tras cada epoch usando métricas clásicas de IR (implementadas en 2-train\_with\_earlystoping.py):

Resultados en época 4 (mejor validación):

**Modelo biomédico**:

MRR: 0.9726

Recall@3: 1.0000

Precision@3: 0.3358

F1@3: 0.5024

**Modelo generalista (bge-m3):**

MRR: 0.9746

Recall@3: 1.0000

Precision@3: 0.3333

F1@3: 0.5000

3.3.6 Análisis Semántico Cualitativo

Tras el fine-tuning con el modelo PlanTL-GOB-ES/bsc-bio-ehr-es y su versión ajustada (bio\_roberta\_epochs/epoch4\_MRR0.9726), se realizó un análisis de similitud semántica entre pares de términos médicos técnicos y expresiones coloquiales.utilizadas por pacientes, dado que el dataset de entrenamiento incluía específicamente este tipo de variaciones lingüísticas.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Par Evaluado** | **Modelo Base** | **Fine-tuneado** | **Diferencia** | **Variación (%)** |
| diabetes ↔ azúcar alto | 0.868 | 0.648 | -0.220 | -25.4% |
| cáncer colorrectal ↔ tumor maligno | 0.930 | 0.685 | -0.245 | -26.3% |
| cáncer ↔ tumor | 0.968 | 0.863 | -0.105 | -10.9% |
| cáncer de colon ↔ colonoscopia | 0.779 | 0.679 | -0.100 | -12.8% |
| recidiva del cáncer de mama ↔ reaparición del cáncer | 0.897 | 0.670 | -0.228 | -25.4% |
| cáncer de mama ↔ mamografía | 0.774 | 0.656 | -0.119 | -15.3% |
| depresión ↔ tristeza | 0.994 | 0.906 | -0.088 | -8.9% |
| HbA1c ↔ promedio de glucosa en sangre | 0.696 | 0.622 | -0.074 | -10.6% |
| cáncer pulmón ↔ tos persistente | 0.859 | 0.513 | -0.346 | -40.3% |
| hipoglucemia ↔ bajada de azúcar | 0.744 | 0.549 | -0.196 | -26.3% |
| ictus ↔ interrupción repentina del flujo sanguíneo en el cerebro | 0.454 | 0.361 | -0.093 | -20.6% |

**Promedio general:**  
Modelo base: 0.607 | Fine-tuneado: 0.628 | Diferencia: +0.021 (+3.5%)

Tras el fine-tuning con el modelo generalista BAAI/bge-m3 y su versión ajustada (bge\_m3\_epochs/epoch4\_MRR0.9717), se realizó un análisis de similitud semántica entre pares de términos médicos técnicos y expresiones coloquiales.

| **Par Evaluado** | **Modelo Base** | **Fine-tuneado** | **Diferencia** | **Variación (%)** |
| --- | --- | --- | --- | --- |
| diabetes ↔ azúcar alto | 0.646 | 0.636 | -0.010 | -1.5% |
| cáncer colorrectal ↔ tumor maligno | 0.603 | 0.633 | +0.030 | +5.0% |
| cáncer ↔ tumor | 0.803 | 0.876 | +0.073 | +9.1% |
| cáncer de colon ↔ colonoscopia | 0.693 | 0.657 | -0.036 | -5.2% |
| recidiva del cáncer de mama ↔ reaparición del cáncer | 0.746 | 0.818 | +0.071 | +9.5% |
| cáncer de mama ↔ mamografía | 0.628 | 0.651 | +0.023 | +3.7% |
| depresión ↔ tristeza | 0.617 | 0.724 | +0.106 | +17.2% |
| HbA1c ↔ promedio de glucosa en sangre | 0.437 | 0.393 | -0.044 | -10.0% |
| cáncer pulmón ↔ tos persistente | 0.422 | 0.210 | -0.213 | -50.4% |
| hipoglucemia ↔ bajada de azúcar | 0.670 | 0.776 | +0.106 | +15.8% |
| ictus ↔ interrupción repentina del flujo sanguíneo en el cerebro | 0.503 | 0.437 | -0.066 | -13.2% |

**Promedio general:**  
Modelo base: 0.607 | Fine-tuneado: 0.628 | Diferencia: +0.021 (+3.5%)

### 3.3.7 Evaluación Comparativa sobre Dataset Independiente (dataset\_test.json)

Finalizado el entrenamiento, los modelos se evaluaron sobre un conjunto externo independiente (dataset\_test.json) en igualdad de condiciones con múltiples modelos del estado del arte. Las métricas evaluadas incluyen Precision@K, Recall@K, F1@K y MRR, para K ∈ {1, 3, 5, 10}.

Este script (3-evaluar\_modelos.py) permite comparar el rendimiento de diferentes modelos usando métricas estándar de recuperación de información. El conjunto de evaluación incluye 322 preguntas formuladas en lenguaje técnico y coloquial, abarcando 8 categorías clínicas principales y 19 subcategorías. Esto permite un análisis granular por dominio médico específico.

Cada consulta incluye:

json

{

"text\_es": "¿Se puede prevenir la diabetes tipo 2?",

"text\_eu": "",

"relevant\_docs": [

"pdf\_diabetes\_es",

"web\_tipos-de-diabetes-mellitus"

],

"categoria": "endocrinologia",

"subcategoria": "diabetes",

"dificultad": "dificil"

}

El formato del dataset incluye campos clave como:

*{*

*"text\_es": "¿Se puede prevenir la diabetes tipo 2?",*

*"text\_eu": "",*

*"relevant\_docs": [*

*"pdf\_diabetes\_es",*

*"web\_tipos-de-diabetes-mellitus",*

*"web\_complicaciones-de-la-diabetes",*

*"pdf\_diabetes\_cuidarse\_es"*

*],*

*"categoria": "endocrinologia",*

*"subcategoria": "diabetes",*

*"dificultad": "dificil"*

*}*

Se calcularon las métricas estándar:

Precision@K: proporción de documentos relevantes en el top-K.

Recall@K: proporción de documentos relevantes recuperados.

F1@K: media armónica entre precisión y recall.

MRR: recíproco de la posición del primer resultado relevante.

Modelos evaluados sobre dataset\_test.json:

| **Modelo** | **Tipo** | **Especialización** | **Dimensión** |
| --- | --- | --- | --- |
| BAAI/bge-m3 | Generalista | Multilingüe, retrieval | 1024 |
| Qwen/Qwen3-Embedding-8B | Generalista | Última generación | 8192 |
| PlanTL-GOB-ES/bsc-bio-ehr-es | Biomédico | Registros clínicos ES | 768 |
| IIC/RigoBERTa-Clinical | Biomédico | Clínico multilingüe | 768 |
| *PlanTL-GOB-ES/roberta-base-biomedical-es* |  |  |  |
| sentence-transformers/all-MiniLM-L6-v2 | Generalista | Compacto | 384 |
| *jinaai/jina-embeddings-v2-base-es* |  |  |  |
| *bio\_roberta\_epochs/epoch4\_MRR0.9726* | **Fine-tuned** | **Consultas médicas ES** | **768** |
| *bge\_m3\_epochs/epoch4\_MRR0.9717* | **Fine-tune** | **Consultas médicas ES** | **1024** |

***Tabla Comparativa de Rendimiento (K=3)***

| ***Modelo*** | ***Precision@3*** | ***Recall@3*** | ***F1@3*** | ***MRR*** |
| --- | --- | --- | --- | --- |
| ***Qwen/Qwen3-Embedding-8B*** | *0.3364* | *0.7237* | *0.4387* | *0.7116* |
| ***BAAI/bge-m3*** | *0.3354* | *0.7256* | *0.4375* | *0.7003* |
| ***Fine-tuned: bge\_m3\_epochs/epoch4\_MRR0.9717*** | *0.3520* | *0.7551* | *0.4571* | *0.6806* |
| ***Fine-tuned: bio\_roberta\_epochs/*** ***epoch4\_MRR0.9726*** | *0.3012* | *0.6501* | *0.3926* | *0.6028* |
| *sentence-transformers/all-MiniLM-L6-v2* | *0.2723* | *0.5901* | *0.3541* | *0.5870* |
| *IIC/RigoBERTa-Clinical* | *0.1853* | *0.4170* | *0.2458* | *0.4061* |
| *PlanTL-GOB-ES/bsc-bio-ehr-es* | *0.1718* | *0.3759* | *0.2269* | *0.3941* |

La tabla presenta una comparación detallada del rendimiento de varios modelos de embeddings en términos de Precision@3, Recall@3, F1@3 y MRR. Los modelos generalistas 'Qwen/Qwen3-Embedding-8B' y 'BAAI/bge-m3' muestran un rendimiento equilibrado y alto en todas las métricas, lo que los hace candidatos prometedores para el sistema RAG médico. El modelo fine-tuneado 'bge\_m3\_epochs/epoch4\_MRR0.9717' destaca por su alto Recall@3 y F1@3, indicando una buena capacidad para recuperar documentos relevantes, aunque su MRR es ligeramente inferior. Esto sugiere que, aunque el modelo fine-tuneado es efectivo para recuperar documentos relevantes, puede no ser tan eficaz en posicionar el documento más relevante en la parte superior de los resultados.

Los modelos especializados en biomedicina, como 'IIC/RigoBERTa-Clinical' y 'PlanTL-GOB-ES/bsc-bio-ehr-es', muestran un rendimiento significativamente inferior en todas las métricas, lo que indica que pueden no ser adecuados para esta tarea específica. Esto podría deberse a que estos modelos están optimizados para tareas diferentes o a que no capturan tan bien las relaciones semánticas presentes en el dataset utilizado.

Se decide utiliza entre los modelos generalistas, el modelo BAAI/bge-m3 y el modelo y  el modelo fine-tuneado basado en BAAI/bge-m3 (bge\_m3\_epochs/epoch4\_MRR0.9717).

La elección del modelo BAAI/bge-m3 para realizar los experimentos se justificada por:

Esto permite comparar directamente el modelo base con su versión optimizada.

* **Menor Demanda de recursos computacionales:** BAAI/bge-m3 tiene 568M parámetros, lo que lo hace más eficiente en términos de recursos computacionales en comparación con modelos más grandes como Qwen/Qwen3-Embedding-8B, que tiene 8B parámetros. Esto permite ejecutar el modelo eficientemente en una CPU sin la necesidad de una GPU de alta gama.
* Aunque Qwen/Qwen3-Embedding-8B puede ofrecer un rendimiento ligeramente superior en algunas métricas, la diferencia no es tan grande como para justificar el gasto extra en recursos.
* **Compatibilidad con Fine-Tuning**: La existencia de un modelo fine-tuneado basado en BAAI/bge-m3 (bge\_m3\_epochs/epoch4\_MRR0.9717) permite una comparación directa y coherente entre el modelo base y su versión optimizada, facilitando el análisis de los beneficios del fine-tuning en el contexto médico.

### 3.3.8 Consideraciones Metodológicas y Mejoras Futuras

Se identifican varias líneas de mejora:

* **Ampliación del dataset**: Incluir más ejemplos en subcategorías clínicas poco representadas, y aplicar técnicas de data augmentation semántico.
* **Entrenamiento con tripletas**: Implementar esquemas de aprendizaje con (query, positiva, negativa "difícil") para mejorar la capacidad discriminativa del modelo.
* **Optimización de hiperparámetros: Learning rate scheduling**
* **Congelación parcial de capas**: Experimentar con congelar selectivamente capas del modelo base para preservar conocimiento general y acelerar el
  + **BGE-M3 como candidato principal**: Su capacidad de mejora en fine-tuning médico lo convierte en opción sólida
  + DESCUBRIMIENTO: Los modelos generalistas con arquitecturas robustas pueden adaptarse mejor al fine-tuning médico que modelos pre-especializados.

**3.4 Pipeline de Indexación**

**3.4.1  Preparación de Textos**

En la construcción de un sistema RAG (Retrieval-Augmented Generation), una etapa crítica es la preparación de los textos que serán utilizados para la recuperación. Esta preparación incluye la división del texto en fragmentos (chunks) manejables y el enriquecimiento de estos fragmentos con metadatos relevantes.

Para lograr esto, se empleó el **RecursiveCharacterTextSplitter** de LangChain con los siguientes parámetros:

* **chunk\_size=300 chars**: Tamaño de cada fragmento de texto.
* **chunk\_overlap=50 chars**: Solapamiento entre fragmentos para mantener el contexto.
* Separadores: **["\n\n","\n",". ","; ","• "," "]**

Cada fragmento (chunk) resultante se almacena en una colección junto con una serie de metadatos que facilitan su posterior recuperación y procesamiento. Estos metadatos incluyen información sobre la categoría médica, palabras clave, posición del chunk dentro del documento y otros atributos relevantes que permiten una búsqueda y filtrado más preciso.

A continuación se muestra un ejemplo de cómo se almacena la nformación:

* **ID de ejemplo**: **5c963b18-3b00-4222-82bc-46011fcbd620**
* **Metadatos de ejemplo**:

{

'categoria': 'enfermedades\_comunes',

'chunk\_id': 'web\_informacion\_chunk0',

'chunk\_keywords': 'tipos diferentes, los más, virus entre, tipos los, enfermedad infecciosa',

'chunk\_position': '1/10',

'chunk\_terms': 'tipos diferentes, los más, enfermedad infecciosa',

'document\_id': 'web\_informacion',

'fuente': 'Web',

'idioma': 'es',

'length\_chars': 158,

'length\_tokens': 28,

'next\_chunk': 'web\_informacion\_chunk1',

'palabras\_clave': 'fiebre alta, dolor cabeza, dolor muscular, congestión nasal, dolor garganta',

'publico\_objetivo': 'general',

'subcategoria': 'Gripe',

'url': 'https://www.euskadi.eus/informacion/que-es-la-gripe/web01-a3gripe/es/'

}

La inclusión de metadatos detallados como la categoría, subcategoría, palabras clave y posición del chunk facilita una recuperación más precisa y contextualizada de la información médica, lo cual es esencial para la efectividad del sistema RAG en este dominio.

Los metadatos enriquecidos no solo facilitan la recuperación precisa de la información, sino que también permiten un análisis más detallado y un filtrado más efectivo durante la fase de búsqueda. Por ejemplo, los campos como **categoria** y **subcategoria** permiten restringir las búsquedas a dominios médicos específicos, mientras que **palabras\_clave** y **chunk\_keywords** ayudan a identificar el contenido relevante dentro de cada fragmento.

Además, la inclusión de información sobre la posición del chunk dentro del documento (**chunk\_position**) y su relación con otros chunks (**next\_chunk**) permite una reconstrucción contextual más efectiva cuando se recuperan múltiples fragmentos relacionados. Esto es particularmente importante en el dominio médico, donde la información relevante puede estar distribuida a lo largo de varios párrafos o secciones de un documento.

**3.4.2 Extracción de Palabras Clave con KeyBERT**

Se utilizó BSC-LT/roberta-base-biomedical-es como backend de KeyBERT. Para cada fragmento se extrajeron las 5 keywords más representativas con diversidad mediante MMR:

keywords = kw\_model.extract\_keywords(cleaned, top\_n=5, use\_mmr=True, diversity=0.2)

Estas palabras se almacenaron en los metadatos para enriquecer las futuras consultas.

**3.4.3 Función de Embeddings Personalizada**

La clase FuncionEmbeddingsPersonalizada adapta el pooling al tipo de modelo:

* CLS token para BGE y RoBERTa
* Mean pooling para BERT con atención

Todos los vectores se L2-normalizan para permitir recuperación por similitud coseno.

**3.4.4 Extracción Unificada de Contenido**

Se implementaron dos funciones modulares:

* extract\_pdf\_text(): recorre páginas con PyMuPDF y limpia el texto.
* extract\_web\_text(): extrae desde HTML estructuras jerárquicas o planas.

Ambas convergen en un texto limpio, apto para segmentación e indexación.

**3.4.5 Evaluación de NER (Descartado)**

Se testearon varios modelos de NER biomédico en español. El rendimiento fue insatisfactorio: errores en fragmentación, etiquetas irrelevantes y cobertura deficiente. Se descartó esta función para la versión final.

**3.5 Creación de Colecciones Vectoriales**

Para evaluar el rendimiento de diferentes enfoques de recuperación de información, se prepararon dos colecciones de documentos indexadas con distintos modelos de embeddings:

Colección de documentos - Modelo generalista: La primera colección, denominada "documentos\_osakidetza", fue indexada utilizando el modelo generalista BAAI/bge-m3. Esta colección se generó mediante el script 1-init\_indexing.py, el cual aplica un proceso de segmentación, enriquecimiento de metadatos y normalización lingüística multilingüe.

Cada documento en esta colección fue fragmentado en chunks con un solapamiento controlado (chunk\_overlap) para asegurar la continuidad del contexto entre fragmentos. Los metadatos asociados a cada chunk incluyen información detallada como:

* Identificadores únicos (document\_id, chunk\_id)
* Clasificación médica (categoria, subcategoria)
* Atributos lingüísticos (idioma)
* Información sobre la fuente y público objetivo
* Posición del chunk dentro del documento (chunk\_position)
* Palabras clave relevantes (palabras\_clave)

El proceso de indexación generó estadísticas post-proceso que incluyen el número total de chunks, longitud media de los fragmentos, y el idioma dominante en la colección. Además, se generó una visualización UMAP de los embeddings, la cual sirve como evidencia del agrupamiento semántico logrado por el modelo.

Colección de documentos - Modelo fine-tuneado: La segunda colección, llamada "documentos\_finetuneado\_bge\_m3", fue creada utilizando un modelo de embeddings específico para el dominio médico: bge\_m3\_epochs/epoch4\_MRR0.9717. Este modelo es una versión fine-tuneada del BAAI/bge-m3, optimizado para tareas de recuperación de información médica.

Esta colección fue generada mediante el script 1\_init\_indexing\_finetuneado.py, siguiendo un proceso similar al de la colección generalista, pero utilizando el modelo especializado. La especificación del modelo fine-tuneado se realiza a través del campo name\_finetuning en el archivo de configuración config.yaml.

La comparación entre estas dos colecciones permite evaluar el impacto del fine-tuning en el rendimiento del sistema de recuperación. Mientras que la colección generalista ofrece un buen punto de partida, la colección con el modelo fine-tuneado se espera que muestre una mejor comprensión de los términos médicos y relaciones semánticas específicas del dominio.

Ambas colecciones están implementadas utilizando ChromaDB como sistema de almacenamiento y recuperación vectorial, lo que permite una búsqueda eficiente de documentos basada en similitud semántica. La elección de ChromaDB como backend de almacenamiento se debe a su facilidad de uso, rendimiento y capacidad para manejar colecciones de documentos con metadatos complejos.

El proceso completo de indexación, desde la preparación de los textos hasta su almacenamiento en ChromaDB, está automatizado y parametrizado a través de archivos de configuración (config.yaml), lo que garantiza la reproducibilidad de los experimentos y facilita la adaptación a diferentes conjuntos de datos o modelos.

En resumen, este apartado describe cómo se prepararon y almacenaron los documentos para su posterior recuperación y evaluación. La existencia de dos colecciones, una con un modelo generalista y otra con un modelo especializado en el dominio médico, permite una comparación directa del impacto del fine-tuning en el rendimiento del sistema RAG.

**3.5.1 Configuración del Sistema**

collection:

name: "documentos\_osakidetza"

name\_finetuneado: "documentos\_finetuneado\_bge\_m3"

model:

name\_embedding: "BAAI/bge-m3"

name\_finetuning: "bge\_m3\_epochs/epoch4\_MRR0.9717"

**3.5.2 Indexación en ChromaDB**

Se crearon dos colecciones:

* documentos\_osakidetza (modelo base BGE-M3)
* documentos\_finetuneado\_bge\_m3 (modelo fine-tuneado)

Cada chunk indexado contiene metadatos enriquecidos.

**3.5.3 Estadísticas Post-Indexación**

| **Métrica** | **Valor** |
| --- | --- |
| Nº total de chunks | 8.257 |
| Longitud media (caracteres) | 220.92 |
| Longitud mín / máx | 19 / 300 |
| Percentil 10 / 90 (caracteres) | 121 / 296 |
| Tokens promedio | 35.2 |
| Tokens mín / máx | 5 / 99 |

Estas estadísticas muestran un corpus equilibrado, apto para tareas de recuperación granular y precisa.

**3.5.4 Validación con UMAP**

Para comprobar la coherencia semántica de los embeddings generados, se aplicó la técnica de reducción de dimensionalidad **UMAP**, proyectando los vectores de alta dimensión (1024D) a un espacio bidimensional (2D).

Se utilizó el script *visualizar\_umap.py*, el cual extrae embeddings y metadatos desde la colección ChromaDB, aplica UMAP y genera visualizaciones categorizadas. Este script permite representar gráficamente los chunks según metadatos como categoria, fuente o idioma.

Las Figuras 3.1 y 3.2 muestran los resultados de esta proyección sobre las dos colecciones indexadas:

Mapa, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

**Figura 3.1 Distribución semántica por categoría médica (modelo base BGE)**

Mapa

El contenido generado por IA puede ser incorrecto.

**Figura 3.2:** **Distribución semántica por categoría médica (modelo fine-tuneado**  bge\_m3\_epochs/epoch4\_MRR0.9717

En ambas se observa que las categorías médicas forman clústeres diferenciables, confirmando que los embeddings preservan estructura semántica.

En la Figura 3.1 (modelo base BGE), los clústeres presentan solapamientos, especialmente entre categorías como salud mental, respiratorias y cardiovascular. Aunque el agrupamiento general es razonable, se observan puntos dispersos y zonas donde diferentes categorías se mezclan.

En la Figura 3.2 (modelo fine-tuneado), los clústeres están mejor definidos, con fronteras más claras entre categorías. La distribución es más coherente y compacta, especialmente en dominios como cáncer, enfermedades cardiovasculares o ITS. Esta mejora visual confirma el ajuste del modelo a las estructuras semánticas del corpus tras el fine-tuning.

**3.5.5 Ejemplo de Chunk Indexado**

A continuación se muestran ejemplos reales de cómo se almacenan fragmentos (chunks) en ambas colecciones indexadas (documentos\_finetuneado\_bge\_m3 y documentos\_osakidetza).

**Colección: documentos\_finetuneado\_bge\_m3**

**Texto del chunk:**  
OSASUN SAILA DEPARTAMENTO DE SALUD GRIPE Se contagia de persona a persona por contacto con las gotitas que se expulsan por la boca o la nariz al toser o estornudar, y que pueden quedarse en las manos. Se puede contagiar a otras personas hasta 7 días después del inicio de la enfermedad

**ID:** 6f86fb48-7825-4525-8547-5170fae4c57e

**Metadatos:**

{

"categoria": "enfermedades\_comunes",

"chunk\_id": "pdf\_consejos\_gripe\_general\_es\_chunk0",

"chunk\_keywords": "osasun saila, se expulsan, osasun, saila departamento, expulsan por",

"chunk\_position": "1/8",

"chunk\_terms": "osasun saila, se expulsan, osasun, saila departamento, expulsan por",

"document\_id": "pdf\_consejos\_gripe\_general\_es",

"filename": "consejos\_gripe\_general\_es.pdf",

"fuente": "PDF",

"idioma": "es",

"length\_chars": 285,

"length\_tokens": 52,

"next\_chunk": "pdf\_consejos\_gripe\_general\_es\_chunk1",

"palabras\_clave": "gripe, fiebre alta, virus, escalofrios",

"publico\_objetivo": "general",

"subcategoria": "Gripe"

}

**Colección: documentos\_osakidetza** (modelo base BGE-M3)

**Texto del chunk:**  
OSASUN SAILA DEPARTAMENTO DE SALUD GRIPE Se contagia de persona a persona por contacto con las gotitas que se expulsan por la boca o la nariz al toser o estornudar, y que pueden quedarse en las manos. Se puede contagiar a otras personas hasta 7 días después del inicio de la enfermedad

**ID:** ce6a0dd9-3bd2-4a60-8da3-1b3e57ae4e35

**Metadatos:**

{

"categoria": "enfermedades\_comunes",

"chunk\_id": "pdf\_consejos\_gripe\_general\_es\_chunk0",

"chunk\_keywords": "osasun saila, se expulsan, osasun, saila departamento, expulsan por",

"chunk\_position": "1/8",

"chunk\_terms": "osasun saila, se expulsan, osasun, saila departamento, expulsan por",

"document\_id": "pdf\_consejos\_gripe\_general\_es",

"filename": "consejos\_gripe\_general\_es.pdf",

"fuente": "PDF",

"idioma": "es",

"length\_chars": 285,

"length\_tokens": 52,

"next\_chunk": "pdf\_consejos\_gripe\_general\_es\_chunk1",

"palabras\_clave": "gripe, fiebre alta, virus, escalofrios",

"publico\_objetivo": "general",

"subcategoria": "Gripe"

}

Este tipo de estructura semánticamente enriquecida se repite para cada chunk del corpus, facilitando búsquedas avanzadas y filtrado jerárquico.

Este desarrollo asegura que cada etapa del sistema RAG (extracción, limpieza, segmentación, indexación y evaluación) está justificada técnica y metodológicamente, permitiendo una recuperación semántica robusta en el dominio médico.

**3.6 Scripts y Configuración**

La siguiente tabla enumera los scripts más relevantes utilizados durante la implementación del sistema, organizados por su funcionalidad y ubicación dentro del repositorio.

| **Script / Archivo** | **Ubicación** | **Funcionalidad principal** |
| --- | --- | --- |
| 1-init\_indexing.py | scripts/ | Ejecuta el pipeline completo de limpieza, embeddings, metadatos e indexación en ChromaDB |
| visualizar\_umap.py | scripts/ | Visualiza la distribución semántica de los embeddings proyectados con UMAP |
| config.yaml | raíz del proyecto | Define parámetros de configuración: modelos, colecciones, rutas |
| load\_model.py | embeddings/ | Carga del modelo de embeddings base o fine-tuneado |
| dataset\_test.json | evaluation/ | Ground truth con consultas categorizadas para evaluar los métodos de recuperación |
| dataset\_finetune.json | finetuning/ | Dataset para entrenar el modelo BGE mediante fine-tuning |
| 2-train\_with\_earlystoping.py | finetuning/ | Script de entrenamiento con early stopping (modelo generalista) |
| 2-train\_with\_earlystoping\_bge.py | finetuning/ | Script específico de fine-tuning del modelo BAAI/bge-m3 |
| comparar\_similitud.py | finetuning/ | Comparación de similitud entre embeddings base |
| comparar\_similitud\_bge.py | finetuning/ | Comparación entre embeddings base y fine-tuneados |
| cargarmodelo.py | retrieval/ | Carga dinámica del modelo y de la colección en modo consulta |
| chroma\_utils.py | retrieval/ | Funciones auxiliares para recuperación, construcción de queries, etc. |

Esta estructura modular favorece la trazabilidad, reutilización y mantenimiento del código fuente en el desarrollo del sistema.