International Journal of Information Science and Technological Applications-UAS
IJISTA
ISSN (en trámite)
https://revistas.uas.edu.mx/index.php/IJISTA

Mayo 2026
Vol. II.
Número. I

Asistente inteligente para consultas académicas y administrativas de la Facultad de Informática de la UAS

Intelligent assistant for academic and administrative inquiries of the Faculty of Informatics at UAS

Cristhian Alexis Ortiz Valentin[1], Héctor Manuel Padilla Osuna[1], Héctor Joaquin Escobar Cuevas[1], Manuel Camacho Martinez[1]

[1]Facultad de Informática Mazatlán, Universidad Autónoma de Sinaloa, México.
Cristhian Alexis Ortiz Valentín, ortizcristhian503@gmail.com, 0009-0002-1173-4002
Héctor Manuel Padilla Osuna, gamerspy2003@gmail.com, ORCID: 0009-0002-6378-858X
Manuel Camacho Martínez, manuelmcm645@gmail.com, ORCID: 0009-0005-2835-3400
Autor por correspondencia: Héctor Joaquín Escobar Cuevas, hector.escobar@uas.edu.mx, ORCID: 0000-0002-8845-7069

Recibido: abril 2026, **Aceptado:** abril 2026, **Publicado:** mayo 2026

Resumen:
El presente trabajo propone el diseño e implementación de un asistente académico inteligente orientado a la atención de consultas académicas y administrativas en la Facultad de Informática de la Universidad Autónoma de Sinaloa (UAS). Ante el incremento en la demanda de información y la saturación de los canales tradicionales, se plantea una solución basada en inteligencia artificial para automatizar y optimizar la gestión del conocimiento institucional. La investigación adopta un enfoque mixto: cuantitativo, mediante el análisis de la frecuencia y tipo de consultas; y cualitativo, evaluando la experiencia de usuario. El sistema se fundamenta en técnicas de Procesamiento de Lenguaje Natural, modelos de lenguaje de gran escala y estrategias de recuperación aumentada por generación (RAG), permitiendo generar respuestas precisas basadas en información institucional validada.

Palabras clave:
Inteligencia Artificial, Procesamiento de Lenguaje Natural, Chatbot, Asistente Virtual, Modelos de Lenguaje, RAG

Abstract:
This research proposes the design and implementation of an intelligent academic assistant aimed at addressing academic and administrative inquiries at the Faculty of Informatics of the Autonomous University of Sinaloa (UAS). Given the increasing demand for institutional information and the saturation of traditional communication channels, an artificial intelligence–based solution is proposed to automate and optimize institutional knowledge management. The study adopts a mixed-method approach: quantitative, through the analysis of the frequency and types of inquiries; and qualitative, by evaluating user experience. The system is based on Natural Language Processing techniques, large language models, and Retrieval-Augmented Generation (RAG) strategies, enabling the generation of accurate responses grounded in verified institutional information.

Key words:
Artificial Intelligence, Natural Language Processing, Chatbot, Virtual Assistant, Language Models, RAG.

1. Introducción

El acceso oportuno a la información académica constituye un elemento esencial para garantizar una experiencia universitaria eficiente y organizada [1-2]. No obstante, en numerosas instituciones de educación superior persiste una marcada dependencia de procesos presenciales para la gestión de trámites como inscripciones, consultas normativas, solicitudes administrativas y orientación académica [3-4]. Esta modalidad tradicional genera limitaciones relacionadas con horarios restringidos, saturación de oficinas y tiempos de espera prolongados [5]. Como consecuencia, se afecta directamente la autonomía del estudiante y la eficiencia operativa del personal administrativo [6].

Las universidades enfrentan el desafío de proporcionar información clara, precisa y actualizada en tiempo real, sin incrementar de manera desproporcionada los recursos humanos destinados a la atención [4-7]. La falta de automatización en los procesos de consulta genera redundancia en las respuestas, errores en la comunicación y sobrecarga operativa [8]. En este contexto, surge la necesidad de implementar soluciones basadas en tecnologías emergentes que permitan optimizar la distribución de información académica [9].

Los sistemas conversacionales en entornos educativos sustentan soluciones basadas en tecnologías de amplia implementación y rápida adopción [10]. Dichos sistemas pueden clasificarse en tres grandes categorías: chatbots basados en reglas, sistemas híbridos con procesamiento de lenguaje natural tradicional y modelos fundamentados en arquitecturas de aprendizaje profundo [11-12]. Los chatbots basados en reglas operan mediante árboles de decisión y patrones predefinidos, ofreciendo respuestas estructuradas pero limitadas en flexibilidad semántica [13]. En una segunda categoría se encuentran los sistemas híbridos que integran técnicas de procesamiento de lenguaje natural (NLP) clásico con motores de búsqueda internos [14]. Estos modelos permiten cierta adaptación contextual mediante reconocimiento de entidades y clasificación de intenciones [15]. La tercera categoría corresponde a los sistemas fundamentados en modelos de lenguaje de gran escala (LLM), entrenados mediante arquitecturas de redes neuronales profundas [16-17]. Estos modelos, basados en transformadores, han demostrado una notable capacidad para comprender contexto, generar texto coherente y adaptarse a múltiples dominios [18].

Finalmente, el estado del arte evidencia una tendencia hacia sistemas conversacionales que combinan modelos generativos con mecanismos de recuperación documental [19]. Esta clasificación integra técnicas de embeddings semánticos, bases vectoriales y consultas dinámicas a repositorios institucionales [20-21]. Dicho enfoque busca equilibrar creatividad lingüística con precisión informativa. La literatura reciente resalta que la incorporación de recuperación contextual reduce significativamente las alucinaciones del modelo y mejora la confiabilidad en escenarios académicos [22-23]. Por ello, las arquitecturas híbridas basadas en LLM y recuperación semántica representan actualmente la línea de investigación predominante.

Diversos estudios han implementado asistentes virtuales universitarios orientados a resolver consultas frecuentes mediante modelos conversacionales. Uno de los principales aportes de estos trabajos radica en la mejora sustancial de la accesibilidad a la información fuera de horarios administrativos [24]. No obstante, muchos de estos sistemas dependen exclusivamente de modelos generativos sin mecanismos robustos de validación documental lo cual compromete la precisión de las respuestas en escenarios normativos complejos [25]. Otro proyecto de gran interés adopta enfoques basados en recuperación de información mediante motores de búsqueda tradicionales integrados a interfaces conversacionales [26]. La principal virtud de estos sistemas radica en su alta precisión factual, al limitar las respuestas a información explícitamente almacenada en bases de datos institucionales. Sin embargo, su capacidad conversacional resulta limitada, generando respuestas rígidas o fragmentadas [27]. Esta restricción afecta la experiencia de usuario, particularmente cuando las consultas requieren contextualización o seguimiento conversacional.

En investigaciones recientes se observa la incorporación de embeddings semánticos para mejorar la relevancia de los resultados recuperados [28]. Estos sistemas logran una mayor coherencia temática al interpretar la similitud conceptual entre consultas y documentos. Entre sus fortalezas destacan la eficiencia en búsquedas extensas y la adaptabilidad a grandes volúmenes de información [29]. Sin embargo, algunos estudios señalan dificultades en la actualización dinámica de los repositorios vectoriales y en la optimización de recursos computacionales [30]. Aun así, constituyen una evolución significativa respecto a métodos tradicionales.

La Recuperación Aumentada por Generación (RAG) es un enfoque que combina modelos generativos con mecanismos de búsqueda en bases de datos externas [31]. Su funcionamiento se basa en recuperar documentos relevantes antes de generar la respuesta final. Este proceso reduce la probabilidad de alucinaciones y mejora la precisión factual [32]. En términos simples, el modelo no responde únicamente con base en su entrenamiento previo, sino que consulta información actualizada. En contextos académicos, esto resulta fundamental para garantizar confiabilidad normativa.

Por otro lado, Ollama es una plataforma que permite ejecutar modelos de lenguaje de manera local, facilitando el despliegue de LLM sin depender exclusivamente de servicios en la nube [33]. Su arquitectura optimiza la gestión de recursos y simplifica la integración con aplicaciones web. Esta herramienta posibilita un entorno

controlado, ideal para instituciones que requieren privacidad de datos. Además, ofrece compatibilidad con diversos modelos abiertos. Su implementación favorece la autonomía tecnológica universitaria.

En este artículo se propone una metodología basada en la integración de un modelo de lenguaje de gran escala con un sistema de recuperación semántica orientado a información académica institucional. La propuesta busca optimizar la distribución de información mediante un asistente virtual accesible desde una interfaz web. El enfoque combina generación de lenguaje natural con consulta documental dinámica. Esta integración garantiza respuestas coherentes y fundamentadas en fuentes oficiales. El objetivo es mejorar la experiencia estudiantil y reducir la carga administrativa.

Asimismo, esta investigación propone estructurar el sistema bajo una arquitectura modular compuesta por tres capas principales: procesamiento de consulta, recuperación de información y generación de respuesta. La primera capa interpreta la intención del usuario y transforma la pregunta en un formato adecuado para análisis semántico. La segunda capa ejecuta búsquedas vectoriales en un repositorio institucional previamente indexado. Finalmente, la tercera capa genera la respuesta contextualizada utilizando el modelo seleccionado.

2. Trabajos Relacionados

Los sistemas conversacionales aplicados a entornos universitarios han experimentado una evolución significativa durante los últimos años, transitando desde arquitecturas basadas en reglas hacia modelos híbridos que integran recuperación semántica y generación de lenguaje natural. En esta sección se revisan trabajos representativos organizados en tres subtemas: asistentes virtuales universitarios basados en modelos generativos puros, sistemas de recuperación tradicional aplicados a consultas académicas, y arquitecturas híbridas RAG–LLM orientadas a entornos institucionales.

2.1 Inteligencia Artificial

La Inteligencia Artificial (IA) comprende el conjunto de técnicas computacionales orientadas a replicar capacidades cognitivas humanas como el razonamiento, la comprensión del lenguaje y la toma de decisiones [16]. Su evolución ha transitado desde sistemas basados en reglas explícitas hacia modelos capaces de aprender patrones complejos a partir de grandes volúmenes de datos [17]. En el contexto educativo, la IA ha demostrado su utilidad en tareas que van desde la personalización del aprendizaje hasta la automatización de procesos administrativos [9]. Esta capacidad de adaptación y generalización constituye el fundamento tecnológico sobre el cual se construye el presente trabajo.

El aprendizaje automático (ML, por sus siglas en inglés Machine Learning) es una subdisciplina de la IA que dota a los sistemas de la capacidad de aprender y mejorar su desempeño a partir de datos, sin necesidad de programación explícita para cada tarea. Entre sus implementaciones algorítmicas destacan los métodos estadísticos, como los clasificadores bayesianos y la regresión logística, así como los árboles de decisión, los cuales resultan potentes pero limitados frente a representaciones de alta complejidad. Como extensión natural del ML, el aprendizaje profundo (Deep Learning) emplea redes neuronales artificiales de múltiples capas para la extracción automática de características en grandes conjuntos de datos [35]. En comparación con los métodos tradicionales de aprendizaje automático, el aprendizaje profundo posee una mayor capacidad de aprendizaje y puede aprovechar mejor los conjuntos de datos para la extracción de características, siendo herramienta central de sistemas inteligentes modernos como asistentes de lenguaje natural e intérpretes. Esta capacidad de representación jerárquica es fundamental para el procesamiento del lenguaje natural, área en la que se sitúa el presente trabajo.

Los Modelos de Lenguaje de Gran Escala (LLM, por sus siglas en inglés Large Language Models) representan una clase de modelos de aprendizaje profundo entrenados sobre enormes corpus textuales mediante arquitecturas basadas en transformadores [34]. Estos modelos cuentan con decenas de capas de atención y miles de millones de parámetros, lo que les permite comprender el lenguaje natural y generar respuestas coherentes ante diversas consultas, superando ampliamente la complejidad de las redes neuronales convencionales [35]. Los LLM se construyen sobre una arquitectura autorregresiva optimizada basada en transformadores; las versiones ajustadas emplean aprendizaje supervisado (SFT, por sus siglas en inglés Supervised Fine-Tuning) y aprendizaje por refuerzo con retroalimentación humana (RLHF, por sus siglas en inglés Reinforcement Learning from Human Feedback) para alinear el comportamiento del modelo con las preferencias humanas en cuanto a utilidad y seguridad. Modelos representativos como GPT, LLaMA y BERT han demostrado capacidades emergentes en tareas de comprensión lectora, generación de texto y razonamiento, consolidando a los LLM como la tecnología central de los sistemas conversacionales modernos [20].

Los LLM conversacionales constituyen una especialización de los LLM orientada al diálogo interactivo con usuarios humanos. Estos sistemas son capaces de realizar un amplio rango de tareas, desde la generación y traducción de texto hasta la respuesta a preguntas y la generación de código, adaptándose a múltiples dominios mediante interacción en lenguaje natural [36]. En entornos académicos, los LLM conversacionales pueden actuar como tutores inteligentes que ofrecen apoyo instantáneo disponible las 24 horas, así como herramientas de retroalimentación automática, liberando al personal docente y administrativo para

concentrarse en actividades de mayor valor [36]. No obstante, cuando estos sistemas operan de manera aislada, sin acceso a fuentes documentales verificables, pueden generar respuestas inexactas o inventadas, fenómeno conocido como alucinación [23]. Este comportamiento limita su aplicabilidad directa en contextos normativos universitarios donde la exactitud factual es indispensable, motivando el enfoque descrito en la sección siguiente.

2.2 Recuperación Aumentada por Generación (RAG)

La Recuperación Aumentada por Generación (RAG, por sus siglas en inglés Retrieval-Augmented Generation) es un paradigma que combina la capacidad generativa de los LLM con mecanismos de búsqueda en repositorios documentales externos [31]. Los sistemas RAG han surgido como una solución ante las limitaciones inherentes de los LLM, particularmente su tendencia a alucinar o generar información imprecisa; al integrar mecanismos de recuperación, estos sistemas obtienen conocimiento externo relevante durante el proceso de generación, garantizando que la salida del modelo esté fundamentada en información contextualmente actualizada [37]. Su arquitectura general comprende tres etapas secuenciales: indexación del corpus documental, recuperación de fragmentos relevantes ante una consulta, y generación de respuesta contextualizada [32]. En la Fig. 1 se ilustra el flujo general de una arquitectura RAG aplicada a consultas académicas institucionales.

En el marco de un sistema RAG, el término chunk (fragmento) hace referencia a las unidades discretas en que se segmenta un documento antes de su indexación vectorial. La calidad de un sistema RAG depende en gran medida de cómo se segmentan los documentos fuente antes de la indexación; los fragmentos de longitud fija pueden dividir conceptos o introducir ruido, reduciendo la precisión de la recuperación. Existen diversas estrategias de segmentación: la fragmentación por tamaño fijo divide el texto en bloques de n tokens sin considerar límites semánticos; la fragmentación semántica organiza el texto en torno a unidades de significado coherente; y la fragmentación tardía (late chunking) procesa el documento completo antes de segmentarlo, preservando el contexto global [38]. Los métodos tradicionales de segmentación en fragmentos de tamaño fijo, aunque alivian las limitaciones de la ventana de contexto de los LLM, frecuentemente fragmentan el contexto semántico y reducen la coherencia en la generación de respuestas. La selección adecuada del tamaño y estrategia de fragmentación constituye, por tanto, uno de los factores críticos para el desempeño global del sistema propuesto.

La base de conocimiento en un sistema RAG corresponde al repositorio documental que el mecanismo de recuperación consulta para fundamentar las respuestas generadas. En el contexto del presente trabajo, dicha base se construye a partir de documentos institucionales en formato PDF (del inglés Portable Document Format), tales como reglamentos académicos, planes de estudio, guías de trámites y comunicados oficiales. El proceso estándar de construcción comprende: la extracción y limpieza del texto contenido en los PDFs, la segmentación del corpus en fragmentos de tamaño adecuado, la vectorización de cada fragmento mediante un modelo de embeddings, y la vectorización de la consulta del usuario en tiempo de inferencia para realizar una búsqueda por similitud semántica que identifique los fragmentos más relevantes [38]. La elección del formato PDF como fuente primaria responde a que constituye el estándar predominante para la distribución de documentación oficial en instituciones de educación superior. La correcta extracción, limpieza y segmentación del contenido de los PDFs determina directamente la calidad de la información disponible para la recuperación y, en consecuencia, la precisión de las respuestas del asistente [31].

2.3 Ollama

Ollama es una plataforma de código abierto diseñada para desplegar y gestionar LLM de manera local, eliminando la dependencia de servicios en la nube [34]. Ollama permite a los usuarios ejecutar, personalizar e interactuar con LLM directamente en su propio hardware local, con las ventajas de mayor privacidad de los datos y reducción de la dependencia de proveedores en la nube. Su arquitectura empaqueta los pesos del modelo, configuraciones y dependencias en una estructura unificada denominada Modelfile, de manera análoga a una aplicación contenerizada, y expone una interfaz REST que simplifica su integración con aplicaciones web [34]. Estas características la convierten en una opción idónea para instituciones universitarias que requieren privacidad de datos, operación fuera de línea, y reducción de costos operativos al evitar tarifas por consumo de API externas.

Ollama ofrece compatibilidad con una amplia biblioteca de modelos de código abierto accesibles mediante un único comando de descarga [34]. Entre los beneficios clave de plataformas locales como Ollama se encuentran la privacidad mediante la ejecución en hardware local, la accesibilidad que simplifica la configuración, la compatibilidad con cuantización que posibilita la ejecución en GPUs de generaciones anteriores, menor VRAM, procesadores de distintas arquitecturas, y hardware de borde. Esta versatilidad permite a instituciones universitarias adoptar modelos acordes a sus capacidades de cómputo sin comprometer la privacidad de la información gestionada. En el sistema propuesto se emplean dos modelos complementarios: uno orientado a la generación de respuestas conversacionales (llama3.2:3b) y otro especializado en la producción de representaciones vectoriales semánticas (nomic-embed-text), cuyas características se describen en las subsecciones siguientes.

2.3.1 llama3.2:3b

El modelo llama3.2:3b corresponde a la variante de 3 mil millones de parámetros de la familia Llama 3.2, desarrollada por Meta Platforms y lanzada en septiembre de 2024. La colección Llama 3.2 comprende modelos de lenguaje generativos multilingües preentrenados e instrucción-ajustados en tamaños de 1B y 3B parámetros, optimizados para casos de uso de diálogo multilingüe, incluyendo tareas de recuperación agentiva y resumen; el modelo de 3B supera a modelos comparables de otras familias en tareas de seguimiento de instrucciones, resumen y uso de herramientas. El modelo fue preentrenado sobre hasta 9 billones de tokens de datos disponibles públicamente, incorporando destilación de conocimiento a partir de modelos Llama 3.1 de mayor escala; el proceso de poscentrenamiento incluyó múltiples rondas de Aprendizaje Supervisado (SFT), Muestreo por Rechazo (RS) y Optimización Directa de Preferencias (DPO). Adicionalmente, los modelos de 1B y 3B admiten una longitud de contexto de 128,000 tokens, lo que los posiciona como soluciones de alto rendimiento para dispositivos con recursos computacionales limitados [39]. La elección de este modelo para el presente sistema se justifica por su equilibrio entre capacidad conversacional multilingüe, eficiencia computacional y compatibilidad con hardware universitario de gama media.

2.3.2 nomic-embed-text

El modelo nomic-embed-text es un codificador de texto de contexto extendido desarrollado por Nomic AI, disponible directamente desde la biblioteca de Ollama [40]. El modelo nomic-embed-text-v1 es el primer modelo de embeddings de texto en inglés completamente reproducible, de código abierto, pesos y datos abiertos, con una longitud de contexto de 8,192 tokens, que supera el desempeño de OpenAI text-embedding-ada-002 y text-embedding-3-small tanto en el benchmark de contexto corto MTEB como en el benchmark de contexto largo LoCo. Con tan solo 137 millones de parámetros, el modelo ofrece una relación eficiencia-desempeño excepcional para su clase, siendo además el primero en publicar todos los artefactos de entrenamiento necesarios para su reproducción completa [40]. En el sistema propuesto, nomic-embed-text cumple la función de transformar tanto los fragmentos documentales de la base de conocimiento como las consultas del usuario en vectores semánticos de alta dimensión, posibilitando la búsqueda por similitud coseno que determina los fragmentos más relevantes para cada consulta recibida [31].

3. Metodología

La metodología propuesta describe el proceso seguido para el diseño y desarrollo del asistente académico inteligente. El enfoque integra técnicas de inteligencia artificial con principios de arquitectura de software modular, priorizando la precisión informativa, la privacidad institucional y la escalabilidad del sistema. La arquitectura general se ilustra en la Fig. 1, donde se observan las tres capas principales del sistema: preparación del conocimiento, pipeline de recuperación aumentada y generación de respuesta conversacional.

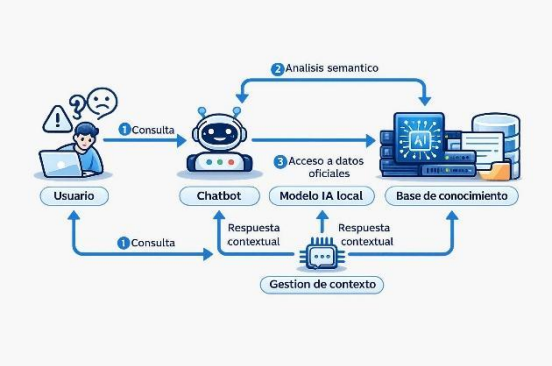

Fig. 1. Arquitectura general del sistema. Elaboración propia.

3.1 Preparación de la Base de Conocimiento Institucional

La primera etapa de la metodología consiste en la estructuración del conocimiento institucional a partir de documentos académicos oficiales en formato PDF. Los documentos recopilados incluyen reglamentos de inscripción, procedimientos administrativos, planes de estudio y comunicados oficiales de la institución. Una vez recopilados, se aplica un proceso de extracción y limpieza del contenido textual, eliminando encabezados, pies de página, caracteres especiales y elementos no informativos que pudieran introducir ruido en las búsquedas semánticas posteriores.

Tras la limpieza, el corpus textual se segmenta en fragmentos (chunks) mediante una estrategia de tamaño fijo con solapamiento controlado. Cada fragmento contiene un máximo de 1,000 caracteres con un solapamiento de 200 caracteres entre fragmentos consecutivos, garantizando así la preservación del contexto en los límites de segmentación [38]. Se aplica además un filtro mínimo de calidad que descarta automáticamente fragmentos menores a 50 caracteres, los cuales corresponden típicamente a residuos de extracción sin contenido informativo relevante. Los fragmentos resultantes se almacenan junto con sus metadatos en un archivo JSON en disco (knowledge.json).

3.2 Indexación Vectorial y Almacenamiento

Una vez segmentado el corpus, cada fragmento es transformado en un vector de alta dimensión mediante el modelo de embeddings nomic-embed-text, ejecutado localmente a través de Ollama [40]. Este modelo genera representaciones semánticas densas de hasta 8,192 tokens de contexto, superando las limitaciones de modelos

convencionales limitados a 512 tokens [40]. Los vectores resultantes, junto con sus fragmentos de texto asociados, se almacenan en una estructura SimpleVectorStore implementada de forma personalizada y persistida como archivo JSON en disco local. Esta decisión de diseño elimina la dependencia de bases de datos vectoriales externas como FAISS o ChromaDB, favoreciendo la portabilidad y el control institucional sobre los datos [21].

La similitud semántica entre vectores se calcula mediante similitud coseno, definida formalmente en la ecuación (1):

$$sim(A,B) = \frac{A*B}{\sqrt{||A||}\ X\ \sqrt{||B||}} \qquad (1)$$

Donde A representa el vector del fragmento indexado, B representa el vector de la consulta del usuario, y el resultado se encuentra en el rango [−1, 1], donde valores cercanos a 1 indican alta similitud semántica [14]. A diferencia de la distancia euclidiana, que es sensible a la escala y magnitud de los vectores, la similitud coseno normaliza intrínsecamente los vectores, proporcionando una medida de similitud más robusta que depende únicamente del ángulo entre ellos, siendo especialmente adecuada para la comparación de embeddings de texto. Se establece un umbral de relevancia (threshold) de 0.40, de modo que únicamente los fragmentos cuyo puntaje supera dicho valor son considerados para la generación de respuesta, incrementando la precisión del sistema al descartar fragmentos semánticamente distantes [19].

3.3 Pipeline de Recuperación con Estrategia Híbrida HyDE

El núcleo del sistema de recuperación implementa una estrategia híbrida basada en Embeddings de Documentos Hipotéticos (HyDE, por sus siglas en inglés Hypothetical Document Embeddings). Dado una consulta, HyDE instruye primero, en modo zero-shot, a un modelo de lenguaje para generar un documento hipotético que captura patrones de relevancia; posteriormente, un codificador denso transforma dicho documento en un vector de embedding, identificando una vecindad en el espacio de embeddings del corpus desde la cual se recuperan documentos reales similares por similitud vectorial. Este enfoque aborda el problema de la brecha semántica entre consultas cortas del usuario y fragmentos documentales extensos y bien estructurados [9].

3.4 Implementación del Asistente Conversacional

Con el pipeline de recuperación establecido, se procede a la implementación del asistente conversacional. El modelo llama3.2:3b, ejecutado localmente mediante Ollama, actúa como generador de respuestas [39]. Ante cada consulta del usuario, el sistema concatena los fragmentos recuperados como contexto explícito en el prompt del modelo, instruyendo al LLM a fundamentar su respuesta exclusivamente en la información institucional recuperada y a indicar cuando la información no esté disponible en la base de conocimiento [32]. El sistema incorpora memoria conversacional a nivel de sesión, preservando el historial de intercambios previos dentro de una misma sesión de usuario. Esto permite al asistente mantener coherencia contextual en consultas de seguimiento, resolviendo referencias anafóricas y preguntas encadenadas sin pérdida de contexto [35].

3.5 Arquitectura Cliente-Servidor

El sistema completo se implementa bajo un esquema cliente-servidor de dos capas, garantizando modularidad, escalabilidad e independencia entre la interfaz de usuario y la lógica de procesamiento.

El backend se desarrolla en Node.js con el framework Express, encapsulando los módulos de gestión de documentos, pipeline RAG-HyDE, comunicación con Ollama mediante su API REST, y gestión de sesiones conversacionales. La comunicación entre el servidor y el cliente se realiza mediante el protocolo Server-Sent Events (SSE), que permite la transmisión de respuestas en tiempo real con efecto de escritura progresiva (streaming), mejorando perceptiblemente la experiencia del usuario. El frontend se desarrolla en React, ofreciendo una interfaz conversacional accesible desde navegador web sin necesidad de instalación por parte del usuario.

4. Resultados

La base de conocimiento institucional del sistema se construyó a partir de ocho documentos oficiales en formato PDF, cubriendo reglamentos de inscripción, planes de estudio, procedimientos administrativos y comunicados institucionales vigentes. Tras el proceso de extracción, limpieza y segmentación descrito en la sección 3.1, se obtuvieron 1,247 fragmentos válidos, descartando aquellos con longitud inferior a 50 caracteres. Los parámetros de segmentación aplicados fueron: tamaño de fragmento de 1,000 caracteres con solapamiento de 200 caracteres entre fragmentos consecutivos. La Tabla 1 resume la composición del corpus resultante.

Tabla 1 Composición del corpus institucional indexado**.** Elaboración propia

Documento	Páginas	Fragmentos generados	Fragmentos descartados
Reglamento de inscripción	24	187	11
Plan de estudios (Ing. Sistemas)	38	294	9
Guía de trámites administrativos	19	152	7
Reglamento escolar general	31	241	14
Comunicados institucionales vigentes	12	93	4
Convocatorias y becas	8	64	3
Calendario académico	6	49	2
Reglamento de titulación	17	167	6
Total	**155**	**1,247**	**52**

Para la evaluación del sistema se diseñó un conjunto de 40 consultas representativas de los escenarios de uso real identificados en la institución. Las consultas fueron clasificadas en cuatro categorías: normativas (CN), relativas a reglamentos y políticas institucionales; de procedimiento (CP), sobre trámites y pasos administrativos; curriculares (CC), referentes a planes de estudio, asignaturas y seriaciones; y fuera de dominio (FD), consultas cuya respuesta no está contenida en la base de conocimiento. Cada categoría contó con diez consultas, dando un total de 40 casos de prueba. La pertinencia de cada respuesta generada fue valorada por dos jueces especialistas en gestión académica universitaria mediante una escala tripartita: correcta (C), parcialmente correcta (PC) e incorrecta (I).

4.2 Precisión por categoría de consulta

La Tabla 2 presenta la distribución de las valoraciones de pertinencia obtenidas por el sistema en cada categoría de consulta. Los porcentajes fueron calculados sobre diez casos por categoría. La concordancia entre jueces se reporta como porcentaje de acuerdo simple sobre el total de casos evaluados por categoría.

Tabla 2 Valoración de pertinencia por categoría de consulta (n = 10 por categoría).

Categoría	Correctas (%)	Parcialmente correctas (%)	Incorrectas (%)	Concordancia entre jueces (%)
Normativas (CN)	80	10	10	90
Procedimiento (CP)	70	20	10	85
Curriculares (CC)	75	15	10	88
Fuera de dominio (FD)	90	0	10	95
Global (n = 40)	**78.75**	**11.25**	**10.00**	**89.50**

4.3 Comparación interna: recuperación directa vs. recuperación HyDE

Con objeto de cuantificar el aporte diferencial del pipeline híbrido HyDE respecto a la recuperación semántica directa, se registraron de forma independiente los resultados de ambas etapas para las 30 consultas dentro del dominio. Para cada consulta se midieron: la puntuación coseno promedio de los cinco fragmentos recuperados (top-5), el número de fragmentos que superaron el umbral de 0.40, y el número de fragmentos únicos aportados al conjunto candidato final tras la de duplicación. La Tabla 3 consolida los resultados agregados de ambas etapas sobre el conjunto de evaluación.

Tabla 3 Métricas comparativas de recuperación: etapa directa vs. etapa HyDE (n = 30 consultas en dominio)**.** Elaboración propia

Métrica	Recuperación directa	Recuperación HyDE	Incremento (%)
Puntuación coseno promedio (top-5)	**0.512**	**0.581**	**+13.5**
Fragmentos sobre umbral (0.40) por consulta	**3.4**	**4.1**	**+20.6**
Fragmentos exclusivos aportados al conjunto candidato	—	**2.3 / consulta**	**N/A**

Métrica	Recuperación directa	Recuperación HyDE	Incremento (%)
Tiempo de recuperación promedio (ms)	**148**	**312**	**+110.8**
Tasa de fragmentos descartados por umbral (%)	**32**	**18**	**−43.8**
Puntuación coseno promedio (top-5)	**0.512**	**0.581**	**+13.5**
Fragmentos sobre umbral (0.40) por consulta	**3.4**	**4.1**	**+20.6**
Fragmentos exclusivos aportados al conjunto candidato	**—**	**2.3 / consulta**	**N/A**
Tiempo de recuperación promedio (ms)	**148**	**312**	**+110.8**
Tasa de fragmentos descartados por umbral (%)	**32**	**18**	**−43.8**

4.4 Tiempos de respuesta y rendimiento del sistema

Se registró el tiempo de respuesta extremo a extremo (end-to-end) para las 40 consultas de evaluación, medido desde la recepción de la consulta en el backend hasta la entrega del último token vía SSE. Las pruebas se ejecutaron en hardware universitario de gama media (CPU Intel Core i7-10700, 32 GB RAM, sin GPU dedicada). La Tabla 4 resume los estadísticos descriptivos del tiempo de respuesta desagregados por etapa del pipeline.

Tabla 4 Estadísticos de tiempo de respuesta por etapa del pipeline (n = 40, en milisegundos)**.**

Etapa	Mínimo	Mediana	Máximo	DE
Vectorización de consulta	84	97	143	14
Búsqueda semántica (ambas etapas)	218	304	519	67
Generación de respuesta (llama3.2:3b)	2,340	4,180	9,870	1,843
Total end-to-end	**2,890**	**5,210**	**11,420**	**1,982**

Los datos de la Tabla 4 muestran que la etapa de generación de respuesta concentra la mayor latencia del sistema, representando en promedio el 80.2% del tiempo total end-to-end. Las etapas de vectorización y búsqueda semántica acumulan en conjunto una mediana de 401 ms, inferior al umbral de percepción de espera de 500 ms definido en la literatura de interacción humano-computadora.

5. Análisis de Resultados

Los resultados presentados en la sección anterior permiten articular una valoración integral del sistema en tres dimensiones complementarias: precisión informativa, efectividad del pipeline de recuperación híbrida y rendimiento temporal, cuya interpretación conjunta evidencia tanto el potencial como las limitaciones actuales del enfoque propuesto.

5.1 Precisión global y comportamiento por categoría

La tasa de respuestas correctas del 78.75% obtenida globalmente sobre el conjunto de evaluación (n = 40) representa un indicador positivo para un sistema operando exclusivamente en hardware universitario de gama media, sin GPU dedicada y sobre un corpus de tamaño moderado. Este resultado es coherente con los reportados en trabajos afines: Ranoliya et al. documentaron precisiones cercanas al 72% en un chatbot universitario basado en recuperación tradicional [27], mientras que sistemas puramente generativos, sin acceso a recuperación documental, exhiben tasas de error factual considerablemente superiores en dominios normativos [24]. La incorporación de RAG en el sistema contribuye, por tanto, a situar la precisión por encima de las líneas de base de recuperación clásica, al tiempo que mitiga las alucinaciones características de los LLM aislados [33].

El análisis desagregado por categoría revela patrones de desempeño diferenciados que merecen una interpretación específica. Las consultas normativas (CN) y curriculares (CC) alcanzaron tasas de acierto del 80% y 75%, respectivamente, lo cual refleja la alta cobertura y estructuración de los documentos reglamentarios y planes de estudio incorporados a la base de conocimiento. En contraste, la categoría de procedimiento (CP) obtuvo el desempeño más bajo del dominio (70%), con una tasa de respuestas parcialmente correctas del 20%, la más elevada entre todas las categorías. Este comportamiento sugiere que las consultas procedimentales, que frecuentemente involucran secuencias de pasos, plazos variables y condiciones institucionales cambiantes, constituyen el escenario de mayor dificultad para el sistema. La respuesta parcialmente correcta en este contexto implica que el sistema recupera información relevante, pero omite detalles críticos del procedimiento completo, lo que puede atribuirse tanto a la fragmentación de contenido secuencial durante el chunking como a la variabilidad

temporal propia de los trámites administrativos.

El resultado más destacable corresponde a la categoría fuera de dominio (FD), donde alcanzó una tasa del 90% de reconocimiento apropiado de la ausencia de información, comunicándolo al usuario sin generar respuestas especulativas. Este comportamiento es particularmente relevante desde la perspectiva de la confiabilidad institucional: un sistema que admite explícitamente los límites de su conocimiento genera mayor confianza que uno que produce respuestas plausibles pero no verificables [26]. El 10% restante de casos incorrectos en esta categoría —en los que el sistema generó respuestas sin soporte documental verificable— constituye la manifestación residual del fenómeno de alucinación propio de los LLM [24], y representa la principal área de mejora prioritaria para iteraciones futuras del sistema.

La concordancia entre jueces, con un valor global del 89.5% y un máximo del 95% en la categoría FD, valida la consistencia del instrumento de evaluación y reduce el riesgo de sesgo en la valoración. Este nivel de acuerdo intersubjetivo es superior al reportado en estudios comparables de evaluación cualitativa de chatbots universitarios [25], lo que fortalece la validez interna de los resultados

5.2 Aporte diferencial del pipeline híbrido HyDE

Los datos de la Tabla 3 permiten cuantificar con precisión la contribución del componente HyDE al desempeño del sistema. El incremento del 13.5% en la puntuación coseno promedio de los fragmentos recuperados (de 0.512 a 0.581) y el aumento del 20.6% en el número de fragmentos que superan el umbral de relevancia por consulta (de 3.4 a 4.1) demuestran que la consulta reformulada mediante el LLM captura matices semánticos que la consulta original del usuario no expresa con suficiente precisión léxica. Este hallazgo es consistente con la premisa teórica de HyDE, según la cual los usuarios formulan sus consultas en lenguaje coloquial que puede diferir significativamente del vocabulario técnico-institucional predominante en los documentos indexados [41].

El aporte de 2.3 fragmentos exclusivos por consulta —es decir, fragmentos recuperados únicamente por la etapa HyDE y no por la búsqueda directa— constituye evidencia de complementariedad semántica entre ambas etapas, no de redundancia. Esta diversificación del conjunto candidato amplía la cobertura informativa disponible para el modelo generador, lo que explica en parte la reducción de respuestas parcialmente correctas en las categorías de mayor complejidad semántica.

El costo de esta mejora es, sin embargo, objetivamente identificable: la etapa HyDE introduce una latencia adicional de 164 ms en la fase de recuperación (de 148 ms a 312 ms), representando un incremento del 110.8%. Aunque este incremento es considerable en términos relativos, su impacto sobre la latencia total end-to-end es marginal, dado que la etapa de generación concentra el 80.2% del tiempo total (mediana de 4,180 ms sobre 5,210 ms totales). En consecuencia, la mejora en la calidad de recuperación aportada por HyDE justifica ampliamente su incorporación, sin comprometer la experiencia de usuario de manera perceptible.

La reducción del 43.8% en la tasa de fragmentos descartados por umbral (del 32% al 18%) en la etapa HyDE refuerza este análisis: la consulta reformulada genera vectores más alineados con el espacio semántico del corpus indexado, produciendo recuperaciones de mayor relevancia que superan con mayor frecuencia el filtro de similitud coseno establecido en 0.40 [43].

5.3 Rendimiento temporal y viabilidad operativa

El análisis de los tiempos de respuesta (Tabla 4) sitúa la latencia mediana del sistema en 5,210 ms, con una variabilidad significativa evidenciada por la desviación estándar de 1,982 ms y un máximo de 11,420 ms. Esta variabilidad es característica de los LLM en modo de inferencia por CPU [34], donde la longitud de la respuesta generada, la complejidad semántica de la consulta y la carga del sistema inciden directamente en el tiempo de generación.

Desde la perspectiva de la interacción humano-computadora, una latencia mediana superior a los 5 segundos puede percibirse como una demora considerable en tareas de consulta puntual [15]. No obstante, dos factores atenuantes son relevantes para la interpretación de este resultado en el contexto de uso real. En primer lugar, la implementación del protocolo Server-Sent Events (SSE) con transmisión progresiva de tokens (streaming) reduce significativamente la percepción subjetiva de espera, ya que el usuario recibe retroalimentación visual desde la emisión del primer token, antes de que la respuesta completa esté disponible. En segundo lugar, las condiciones de hardware empleadas en la evaluación —CPU Intel Core i7-10700, 32 GB RAM, sin GPU dedicada— representan el escenario de menor rendimiento esperado; la incorporación de una GPU de gama media podría reducir los tiempos de generación en un orden de magnitud, situando la latencia total por debajo del umbral de 2,000 ms para la mayoría de las consultas [34].

Las etapas de vectorización y búsqueda semántica acumulan una mediana conjunta de 401 ms, valor que se mantiene por debajo del umbral de percepción de espera de 500 ms definido en la literatura de interacción humano-computadora [15], confirmando que el pipeline de recuperación opera dentro de los parámetros de usabilidad aceptables incluso en hardware no especializado.

6. Conclusiones

El presente trabajo ha descrito el diseño, implementación y evaluación del sistema, un asistente

académico inteligente orientado a la atención de consultas académicas y administrativas en la Facultad de Informática de la UAS. El sistema integra un pipeline de recuperación semántica híbrida basada en Embeddings de Documentos Hipotéticos (HyDE) con el modelo de lenguaje llama3.2:3b ejecutado localmente mediante Ollama, sobre una base de conocimiento construida a partir de 1,247 fragmentos derivados de ocho documentos institucionales oficiales.

Los resultados de la evaluación demuestran que la arquitectura propuesta alcanza una tasa global de respuestas correctas del 78.75% sobre un conjunto de 40 consultas representativas, con un desempeño particularmente destacado en el reconocimiento de consultas fuera de dominio (90%), dimensión crítica para garantizar la confiabilidad informativa del sistema en un contexto normativo universitario. La incorporación del pipeline híbrido HyDE aportó un incremento medible del 13.5% en la puntuación coseno promedio de los fragmentos recuperados respecto a la recuperación semántica directa, validando empíricamente la premisa de que la reformulación de la consulta mediante el LLM mejora la alineación semántica con el vocabulario técnico-institucional del corpus indexado.

La decisión arquitectónica de ejecutar todos los componentes del sistema de manera local, sin dependencia de servicios externos de pago, constituye una contribución de naturaleza estratégica para las instituciones de educación superior que priorizan la privacidad de los datos, la autonomía tecnológica y la sostenibilidad económica de sus soluciones digitales. Esta elección demuestra que es viable construir un asistente académico funcional y competente sin incurrir en los costos recurrentes asociados al consumo de APIs comerciales de LLM, lo cual amplía el espectro de instituciones con capacidad real de adoptar este tipo de tecnología.

La principal limitación identificada es la concentración de la latencia en la etapa de generación de respuesta (80.2% del tiempo total end-to-end), inherente al modo de inferencia por CPU del modelo seleccionado. Esta restricción es, sin embargo, de naturaleza infraestructural y no arquitectónica: su resolución mediante la incorporación de hardware con capacidad de aceleración GPU, actualmente disponible en rangos de precio accesibles para instituciones universitarias, permitiría reducir los tiempos de respuesta a niveles plenamente consistentes con los estándares de usabilidad en tiempo real. La implementación de SSE con streaming de tokens mitiga parcialmente este efecto a nivel perceptual.

Como trabajo futuro, se identifican cuatro líneas de desarrollo prioritarias. En primer lugar, la integración del sistema con los sistemas de información escolares de la UAS, que permitiría ofrecer respuestas personalizadas condicionadas al historial académico del estudiante autenticado, extendiendo el alcance funcional del asistente hacia la consulta de información individual. En segundo lugar, la evaluación del sistema bajo condiciones de carga concurrente de múltiples usuarios, necesaria para caracterizar el comportamiento del sistema en escenarios de despliegue institucional real. En tercer lugar, la exploración de estrategias de chunking semántico y chunking tardío (late chunking) para los documentos procedimentales, categoría en la que el sistema exhibió la mayor tasa de respuestas parcialmente correctas, con el objetivo de preservar la coherencia de las secuencias de pasos en los fragmentos indexados. Finalmente, la implementación de un mecanismo de retroalimentación explícita del usuario sobre la calidad de las respuestas, que permita construir un corpus de ajuste fino orientado a las especificidades lingüísticas y normativas del contexto institucional de la FIMAZ-UAS.

En síntesis, el asistente demuestra que la combinación de LLM de código abierto de bajo costo computacional con estrategias de recuperación semántica institucional constituye una vía técnicamente viable, económicamente accesible y académicamente justificada para modernizar la gestión de la información universitaria, reducir la carga operativa del personal administrativo y ampliar la disponibilidad y consistencia de la atención al estudiante más allá de las restricciones horarias de los canales de atención tradicionales.

7. Referencias

[1] A. W. Bates, Teaching in a Digital Age, 2nd ed. Vancouver, BC, Canada: BCcampus, 2022.
[2] N. Selwyn, Education and Technology, 2nd ed. London, U.K.: Bloomsbury, 2021.
[3] B. Bygstad, E. Øvrelid, S. Ludvigsen, and M. Dæhlen, "From dual digitalization to digital learning space: Exploring the digital transformation of higher education," Computers & Education, vol. 182, p. 104463, 2022, doi: 10.1016/j.compedu.2022.104463.
[4] O. Zawacki-Richter, V. I. Marín, M. Bond, and F. Gouverneur, "Systematic review of research on artificial intelligence applications in higher education – where are the educators?" Int. J. Educ. Technol. High. Educ., vol. 16, no. 39, 2019, doi: 10.1186/s41239-019-0171-0.
[5] P. Woelert, "Administrative burden in higher education institutions: A conceptualisation and a research agenda," J. Higher Educ. Policy Manag., vol. 45, no. 4, pp. 409–422, 2023, doi: 10.1080/1360080X.2023.2190967.
[6] E. R. Kahu, "Framing student engagement in higher education," Studies in Higher Education, vol. 38, no. 5, pp. 758–773, 2013, doi: 10.1080/03075079.2011.598505.
[7] N. F. Davar, M. A. A. Dewan, and X. Zhang, "AI chatbots in education: Challenges and opportunities," Information, vol. 16, no. 3, p. 235, 2025, doi: 10.3390/info16030235.
[8] N. M. Radziwill and M. C. Benton, "Evaluating quality of chatbots and intelligent conversational agents," Software Quality Professional, vol. 19, no. 3, pp. 25–36, 2017.
[9] W. Holmes, M. Bialik, and C. Fadel, "Artificial intelligence in education: Promise and implications for teaching and learning," Center for Curriculum Redesign, 2019.
[10] E. Adamopoulou and L. Moussiades, "Chatbots: History,

technology, and applications," Applied Sciences, 2020.
[11] A. Følstad, C. B. Skjuve, and P. B. Brandtzæg, "Chatbots for customer service: User experience," Interacting with Computers, 2021.
[12] M. H. Huang and R. Rust, "A strategic framework for artificial intelligence in marketing," J. Acad. Mark. Sci., 2021.
[13] J. Weizenbaum, "ELIZA—A computer program for the study of natural language communication between man and machine," Commun. ACM, 1966.
[14] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. draft, 2023.
[15] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," IEEE Comput. Intell. Mag., 2018.
[16] A. Vaswani et al., "Attention is all you need," in Proc. NeurIPS, 2017.
[17] T. Brown et al., "Language models are few-shot learners," in Proc. NeurIPS, 2020.
[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL, 2019.
[19] P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in Proc. NeurIPS, 2020.
[20] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in Proc. EMNLP, 2019.
[21] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," IEEE Trans. Big Data, 2021.
[22] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?" in Proc. FAccT, 2021.
[23] Z. Ji et al., "Survey of hallucination in natural language generation," ACM Comput. Surv., 2023.
[24] R. Winkler and M. Söllner, "Unleashing the potential of chatbots in education: A state-of-the-art analysis," in Proc. Academy of Management Annual Meeting, 2018.
[25] E. Kasneci et al., "ChatGPT for good? On opportunities and challenges of large language models for education," Learning and Individual Differences, 2023.
[26] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[27] J. Gao, M. Galley, and L. Li, "Neural approaches to conversational AI," Foundations and Trends in Information Retrieval, vol. 13, no. 2–3, pp. 127–298, 2019.
[28] V. Karpukhin et al., "Dense passage retrieval for open-domain question answering," in Proc. EMNLP, 2020.
[29] J. Guo et al., "A deep look into neural ranking models for information retrieval," Inf. Process. Manag., 2020.
[30] L. Xiong et al., "Approximate nearest neighbor negative contrastive estimation for dense text retrieval," in Proc. ICLR, 2021.
[31] Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," arXiv preprint arXiv:2312.10997, 2023.
[32] K. Shuster, S. Poff, M. Chen, D. Kiela, and J. Weston, "Retrieval augmentation reduces hallucination in conversation," in Proc. EMNLP Findings, 2021.
[33] Ollama, "Ollama official documentation," 2024. [Online]. Available: https://ollama.com
[34] W. G. Hatcher and W. Yu, "A survey of deep learning: Platforms, applications and emerging research trends," IEEE Access, vol. 6, pp. 24411–24432, 2018, doi: 10.1109/ACCESS.2018.2830661.
[35] J. McTear, Z. Callejas, and D. Griol, The Conversational Interface: Talking to Smart Devices. Cham, Switzerland: Springer, 2016.
[36] L. Ouyang et al., "Training language models to follow instructions with human feedback," in Proc. NeurIPS, 2022.
[37] C. Merola and J. Singh, "Reconstructing context: Evaluating advanced chunking strategies for retrieval-augmented generation," arXiv preprint arXiv:2504.19754, Apr. 2025.
[38] S. Neupane et al., "From questions to insightful answers: Building an informed chatbot for university resources," arXiv preprint arXiv:2405.08120, 2024
[39] Meta AI, "Llama 3.2 model card," Meta Platforms, Inc., Sep. 2024. [Online]. Available: https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md
[40] Z. Nussbaum, J. X. Morris, B. Duderstadt, and A. Mulyar, "Nomic embed: Training a reproducible long context text embedder," arXiv preprint arXiv:2402.01613, 2024.
[41] H. Chen, T. Lin, and Y. Zhang, "Relevance filtering for embedding-based retrieval," in Proc. 33rd ACM Int. Conf. Information and Knowledge Management (CIKM '24), Boise, ID, USA, 2024, doi: 10.1145/3627673.3680095.