

[PORTADA]

1. Resumen ejecutivo

2. Descripción del reto

 2.1 Descripción de Sabentis

 2.2 Planificación del reto

3. Estudio de los resultados conocidos

4. Análisis de datos

 4.1 Visión general de los datos

 4.2 Preprocesamiento de datos específicos

 4.3 Análisis exploratorio de datos

5. Selección de soluciones técnicas

6. Selección del modelo de incrustación

 6.1 Configuración de la comparación

 6.2 Métricas de comparación

 6.3 Comparación de modelos

 6.3.1 BERT

 6.3.2 word2vec

 6.3.3 TF_IDF

 6.3.4 OpenAI

7. Despliegue

8. Resumen y trabajo futuro

1. Resumen Ejecutivo

1.1 Justificación del proyecto

El desarrollo de un chatbot avanzado para Sabentis surge como una necesidad imperante para optimizar la gestión de la seguridad y salud en el trabajo (SST), un aspecto crítico en cualquier entorno laboral. Sabentis proporciona un conjunto integral de 43 módulos diseñados específicamente para la gestión de SST, los cuales están categorizados en cuatro áreas principales:

1. Módulos transversales
2. Gestión organizativa y planificación
3. Seguridad, salud y bienestar
4. Comunicación, capacitación y cumplimiento

Estos módulos están respaldados por una extensa documentación que incluye más de 100 manuales explicativos. Para mejorar la gestión de estos recursos y facilitar la interacción de los usuarios con el sistema, es fundamental integrar un chatbot en el software de Sabentis. Este chatbot está diseñado para ofrecer acceso rápido y eficiente a la información contenida en los manuales, permitiendo a los usuarios recibir asistencia en tiempo real y gestionar de manera eficiente las tareas relacionadas con la SST.

La implementación de esta tecnología en la plataforma de Sabentis no solo optimiza la eficiencia operativa y proporciona soporte continuo a los usuarios, especialmente en situaciones de emergencia o riesgo, sino que también se alinea con las tendencias contemporáneas en inteligencia artificial y automatización. Este proyecto otorga a Sabentis una ventaja competitiva significativa en la transformación digital del sector SST, reforzando su compromiso con la innovación y la excelencia operativa. La adopción de un chatbot avanzado no solo moderniza la gestión de SST, sino que también posiciona a Sabentis como un líder en la integración de tecnologías avanzadas para mejorar la seguridad y el bienestar en el entorno laboral.

Objetivos

El objetivo principal es el desarrollo e implementación de un chatbot avanzado para la gestión de la seguridad y salud en el trabajo en el software de Sabentis.

1. **Mejorar la eficiencia operativa:** Facilitar el acceso rápido y preciso a la información sobre SST, permitiendo a los usuarios la tomar decisiones informadas de manera ágil.
2. **Automatizar respuestas y gestiones:** Implementar un sistema de respuestas automáticas a consultas frecuentes, liberando al personal para enfocarse en tareas más complejas y estratégicas.
3. **Reducir riesgos laborales:** Permitir una comunicación más fluida y efectiva entre los usuarios y el sistema de gestión de SST, contribuyendo a una identificación y gestión más rápida de los riesgos.
4. **Incrementar la satisfacción del cliente:** Proporcionar respuestas rápidas y personalizadas, mejorando la experiencia del usuario y la percepción del servicio ofrecido por Sabentis.

5. **Impacto a nivel de negocio:** Aumentar la competitividad de Sabentis mediante la adopción de tecnologías avanzadas que mejoren la eficiencia y la capacidad de respuesta del servicio, generando ahorros significativos en costos operativos y mejorando el rendimiento global del negocio.

Problema de Estudio

Sabentis cuenta con 43 módulos diseñados específicamente para la gestión de SST, respaldados por una extensa documentación que incluye más de 100 manuales explicativos. Los usuarios no pueden acceder a dicha información de forma rápida y ordenada. La gestión de consultas frecuentes y la necesidad de soporte continuo en el ámbito de SST presentan un desafío significativo. La capacidad de ofrecer respuestas precisas y personalizadas en tiempo real es crucial para la eficacia operativa y la seguridad en el trabajo. Actualmente, la dependencia de respuestas manuales y la falta de automatización pueden llevar a retrasos y errores en la gestión de la información, afectando negativamente la seguridad y la eficiencia operativa.

Material y Metodología

El equipo del proyecto, compuesto por Brandon Maldonado Alonso, Victor Aranda Belmonte y Verónica Sánchez Muñoz, ha adoptado una metodología ágil, utilizando herramientas como Trello para la organización del trabajo y sprints semanales para asegurar un avance estructurado y eficiente.

A lo largo del proyecto, se ha prestado especial atención a la optimización del código y la gestión eficiente de recursos, mitigando riesgos asociados como la incapacidad del chatbot para encontrar respuestas adecuadas y el alto costo computacional y económico de implementación. Esta estrategia ha sido crucial para garantizar que el proyecto se mantenga en línea con los objetivos de Sabentis y que la solución desarrollada sea robusta, escalable y alineada con las mejores prácticas de la industria.

Material

Documentación: Seis manuales de Seguridad y Salud en el Trabajo (SST) y un documento de Preguntas Frecuentes (FAQ) proporcionados por Sabentis, que constituyen la base de datos principal para el entrenamiento del chatbot.

Arquitectura

Rewrite.

Sabentis ya consta con una arquitectura propia donde proporciona a sus usuarios una plataforma web, donde aloja ya sus documentos PDFs, etc.

El futuro de este proyecto es implementar el chatbot en su plataforma, pero es una decisión final de Sabentis.

Por el momento nuestro proyecto necesitaría acceder al directorio ya sea en un disco duro físico, o en un servicio en la nube pero “montado” para poder acceder a los PDFs.

Acceso de escritura a un disco o a otro servicio online para alojar ahí los textos procesados y una nueva base de datos basada en estos datos procesados y sus diferentes embeddings.

Herramientas de Inteligencia Artificial y NLP:

NO son de IA o NLP.

Python: Lenguaje de programación utilizado para el desarrollo del chatbot.

PyMuPDF: Biblioteca para la extracción y procesamiento de texto de documentos PDF.

NLTK (Natural Language Toolkit): Biblioteca para el procesamiento de lenguaje natural.

SpaCy: Biblioteca avanzada para el procesamiento de lenguaje natural.

Hugging Face Transformers: Plataforma que proporciona modelos de lenguaje preentrenados como BERT.

Trello: Herramienta de gestión de proyectos utilizada para la organización del trabajo y la planificación de sprints.

Docker: Herramienta para la creación de contenedores que facilita el despliegue y la gestión del entorno de desarrollo.

Modelos de Lenguaje: Modelos de lenguaje basados en algoritmos avanzados como Word2vec, TF-IDF, BERT y OpenAI, que se utilizan para procesar y generar respuestas precisas y contextuales.

Metodología

NO son language models. Text encoding / Text representation / Text embedding models.

Investigación Inicial: Revisión de la literatura y documentación técnica para identificar las mejores prácticas en el desarrollo de chatbots. Se exploraron estudios de caso exitosos y se analizaron las características y limitaciones de diferentes enfoques tecnológicos.

Selección de Soluciones Técnicas: Evaluación de diversas tecnologías de IA, incluyendo ChatGPT y Llama2, y elección del modelo óptimo basado en algoritmos avanzados. La selección se basó en criterios de precisión, eficiencia, capacidad de integración y escalabilidad.

Desarrollo del Chatbot: Implementación de un pipeline de Retrieve and Generate (RAG) para procesar y generar respuestas precisas. El desarrollo incluyó la creación de embeddings de texto, la configuración de sistemas de recuperación de información y la integración de modelos generativos.

Pruebas y Validación: Configuración de pruebas utilizando métricas de similitud como Similitud de Coseno y Distancia Euclídea para comparar la eficacia de los diferentes modelos. Se realizaron pruebas de usabilidad y rendimiento para asegurar que el chatbot cumpliera con los estándares de calidad y eficacia.

Es metodología ??? Rewrite,

→ ???

Contribución del proyecto al Ámbito Profesional, Científico y Social

El desarrollo de este chatbot aporta una solución innovadora que mejora significativamente la gestión de la seguridad y salud en el trabajo (SST) en Sabentis. Al automatizar respuestas y gestionar múltiples interacciones simultáneamente, el chatbot libera al personal para que se enfoque en tareas más complejas y estratégicas, incrementando así la productividad y reduciendo los costos operativos.

La disponibilidad continua del chatbot garantiza que los usuarios reciban soporte en tiempo real, lo cual aumenta la satisfacción y la lealtad del cliente, fortaleciendo la posición de Sabentis en el mercado. Además, la adopción de esta tecnología optimiza la gestión de consultas relacionadas con SST, permitiendo una mayor eficiencia operativa y una mejor asignación de recursos humanos.

Este proyecto representa un avance importante en el campo de la inteligencia artificial aplicada a la gestión de SST. La implementación y evaluación de modelos avanzados como RAG en un entorno práctico proporcionan insights valiosos sobre la eficacia de estas tecnologías en aplicaciones reales. Además, el uso de técnicas de procesamiento de lenguaje natural para mejorar la precisión y personalización de las respuestas ofrece un caso de estudio relevante para futuras investigaciones en el área de chatbots y automatización. Los resultados y metodologías desarrolladas en este proyecto pueden servir como referencia para la integración de tecnologías similares en otros sectores, ampliando su impacto y relevancia.

La implementación del chatbot tiene un impacto significativo en el negocio de Sabentis. Estudios recientes indican que en 2024, los chatbots podrían generar ahorros anuales de hasta 11 mil millones de dólares en sectores clave como el retail, la banca y la salud, gracias a la automatización de tareas repetitivas y la mejora en la eficiencia del servicio al cliente. ([Tidio](#)) ([Business Wire](#)).

La capacidad del chatbot para ofrecer soporte continuo 24/7 es esencial para manejar emergencias y consultas urgentes de SST, mejorando la respuesta y mitigación de riesgos. En consecuencia, se espera que Sabentis obtenga beneficios económicos significativos, optimizando sus operaciones y fortaleciendo su competitividad en el mercado. ([Ubique Digital Solutions](#)).

Como hemos mencionado anteriormente la automatización de las respuestas a consultas frecuentes mediante el chatbot tiene un impacto positivo en los empleados de Sabentis que previamente realizaban estas tareas manualmente. Al liberar a estos empleados de tareas repetitivas y rutinarias, se les permite concentrarse en actividades más complejas y enriquecedoras, lo que puede aumentar su satisfacción laboral y su desarrollo profesional. Esta transición promueve un ambiente de trabajo más motivador y menos estresante, contribuyendo a la retención de talento y mejorando el bienestar general de los empleados.

La integración de chatbots también tiene un impacto positivo en el medio ambiente. Al reducir la necesidad de infraestructura física y recursos asociados, como electricidad y espacio de oficina, se contribuye a una operación más sostenible y verde. Este enfoque no sólo es económicamente beneficioso, sino que también refuerza el compromiso de Sabentis con la responsabilidad ambiental y la optimización de procesos. Al reducir la huella de carbono y promover prácticas sostenibles, Sabentis se posiciona como una empresa responsable y consciente del medio ambiente, mejorando su reputación y atrayendo a clientes y socios comerciales comprometidos con la sostenibilidad.

3. Descripción de Sabentis



Sabentis es una empresa líder en la transformación digital de la gestión de la seguridad y salud en el trabajo (SST). Su software optimiza la prevención de riesgos laborales mediante la automatización y centralización de información clave, utilizando inteligencia artificial generativa y Power BI. Esto permite a las organizaciones predecir riesgos, mejorar protocolos de seguridad y tomar decisiones informadas.

Sabentis ofrece un conjunto integral de 43 módulos diseñados para la gestión de la seguridad y salud en el trabajo, agrupados en cuatro categorías principales:

1. Módulos transversales
2. Gestión organizativa y planificación
3. Seguridad, salud y bienestar
4. Comunicación, capacitación y cumplimiento

De los cuales Sabentis dispone de más de 100 manuales relacionados con la Seguridad y la Salud en el trabajo. Para mejorar aún más la gestión y facilitar la interacción de los usuarios con el sistema, surge la necesidad de integrar un chatbot en su software.

Este chatbot permitirá a los usuarios acceder rápidamente a la información, recibir asistencia en tiempo real y gestionar eficientemente las tareas relacionadas con la seguridad y salud en el trabajo.

4. Planificación del Reto

El proyecto se centra en el desarrollo de un chatbot personalizado, diseñado específicamente para integrarse con la plataforma de Sabentis, proporcionando un servicio de atención y asistencia directa, optimizado para las necesidades específicas de la empresa y sus usuarios. Este sistema no está concebido como un chatbot de propósito general, enfocado en mantener conversaciones genéricas, sino como una herramienta especializada, capaz de ofrecer respuestas y soluciones pertinentes y de alto valor añadido a las consultas específicas de los usuarios.

Aún siendo un chatbot personalizado, este chatbot utiliza Retrieval Augmented Generation (RAG), con lo cual no es un chatbot entrenado exclusivamente con los PDFs otorgados, sino que es un chatbot escalable. Sabentis le puede introducir más PDFs y seguirá funcionando eficientemente, y a su vez, cualquier otra empresa que utilice una documentación en PDF similar a Sabentis puede utilizar el código fuente, solo basta con ajustar ciertos parámetros dependiendo de cómo la información esté clasificada y sería capaz de responder a las preguntas de otra empresa.

4.1. Cronograma del Proyecto

El cronograma del proyecto se desarrolló meticulosamente para asegurar un avance organizado y eficiente en cada fase del desarrollo del chatbot para Sabentis. A continuación, se detalla de forma extensa y clara cómo transcurrieron los trabajos a lo largo de varios meses, describiendo cada hito importante y las actividades realizadas por el equipo.

Fase de Planificación y Configuración Inicial

14/02/24:

- **Definición de roles y configuración inicial del entorno de trabajo:**

- Asignación de roles específicos a cada miembro del equipo.
- Creación de un repositorio en GitHub para la gestión del código fuente.
- Configuración de Trello para la gestión de tareas y planificación de sprints.
- Uso de Google Drive para el intercambio y almacenamiento de documentos.

Fase de Análisis y Comprensión de Necesidades

19/02/24:

- **Kickoff meeting con Sabentis:**

- Discusión de las necesidades específicas del cliente.
- Establecimiento de objetivos y expectativas del proyecto.
- Revisión de la documentación proporcionada por Sabentis.

21/02/24:

- Recepción de los manuales de seguridad y salud en el trabajo en formato PDF y el documento de preguntas frecuentes (FAQ).

Fase de Lectura y Análisis de la Documentación

Semana del 03/03/24:

- **Comienzo de la fase de lectura y análisis de los manuales recibidos:**
 - Asignación de secciones específicas a cada miembro del equipo para resumir:
 - **Victor Aranda:** Sección de Estructura Organizativa.
 - **Brandon Maldonado Alonso:** Secciones de Identificación y Evaluación de Riesgos, y Planes de Emergencia.
 - **Verónica Sánchez Muñoz:** Secciones de Auditorías y Ausentismo.

Fase de Evaluación Inicial y Metodología de Limpieza de Datos

02/03/24:

- **Reunión con los tutores:**
 - Discusión de los avances realizados.
 - Definición de los pasos a seguir basados en los hallazgos iniciales.

Semana del 04/03/24:

- **Evaluación de ChatGPT mediante la API:**
 - Introducción de los manuales en PDF para determinar la adecuación de sus respuestas a los objetivos del proyecto.
 - Identificación de limitaciones y exploración de alternativas tecnológicas.

Fase de Limpieza de Datos y Desarrollo Inicial

Semana del 11/03/24:

- **Reunión de equipo para discutir la metodología óptima de limpieza de documentos:**
 - Conversión de PDFs a texto (txt) para facilitar su manipulación.
 - Limpieza de datos irrelevantes.
- **Tareas asignadas:**
 - **Verónica:** Creación del documento de proyecto, integrando inputs y feedback recibido durante la semana de Victor y Brandon.
 - **Brandon:** Desarrollo de un script en Python para la limpieza masiva de los textos, preparándose para análisis posteriores.
 - **Victor:** Aplicación de los conocimientos de Word2vec adquiridos en clase para transformar los textos en vectores, permitiendo al sistema ofrecer respuestas relevantes a las consultas de los usuarios.

Fase de Mejora de Extracción de Texto y Pruebas de Modelos

Semana del 18/03 - 24/03:

- **Distribución de tareas para mejorar la extracción de texto de los documentos PDF y desarrollo de métodos para crear textos unitarios a partir del texto extraído:**
 - **Brandon:** Iniciar la mejora en la extracción de texto de los documentos PDF para asegurar una mayor precisión y calidad en el texto extraído. Iniciar pruebas con Word2vec preentrenado para comparar su rendimiento frente a modelos no preentrenados.
 - **Victor:** Crear clases en Python para cada modelo de prueba (Word2vec, TF-IDF, combinación de ambos) y preparar el entorno de prueba. Comenzar las pruebas con el modelo Word2vec para evaluar su efectividad en el manejo de un rango amplio de preguntas.
 - **Verónica:** Crear clases en Python para el modelo basados en Bert preentrenado con otros embeddings.

Fase de Pruebas Comparativas y Ajustes Finales

Semana del 25/03 - 01/04:

- **Continuación de las pruebas y comparaciones entre los resultados de Word2vec, TF-IDF, sus combinaciones y Bert para identificar los modelos más óptimos:**
 - Integración con ChatGPT para experimentar con la transmisión de los outputs de estos modelos y generar respuestas más conversacionales y ajustadas a las necesidades del proyecto.

02/04/24:

- **Reunión con los tutores:**
 - Presentación de los avances.
 - Discusión de los resultados de las pruebas de modelos y recepción de feedback.
 - Los tutores transmiten los puntos claves que debe tener la presentación.
 - Planificación de los siguientes pasos basados en el feedback recibido, enfocándose en la actualización automática de contenidos y el desarrollo de una interfaz web.

Fase de Revisión y Análisis de Modelos

11/04/24:

- **Reunión con los tutores para mostrar la presentación:**
 - Validación de la estructura y contenido de la presentación.
 - Recepción de feedback para ajustes finales.

Del 14/04 al 07/05/24:

- **Análisis de los modelos para determinar el más óptimo:**
 - **Brandon:** Análisis de Word2vec.
 - **Victor:** Análisis de TF-IDF y OpenAI.
 - **Verónica:** Análisis de Bert.
 - Revisión de documentación y papers para sustentar los análisis.

08/05/24:

- **Reunión con los tutores para compartir avances:**
 - Presentación de los hallazgos y avances en el análisis de los modelos.
 - Discusión sobre la integración y viabilidad de los modelos analizados.

22/05/24:

- **Reunión con los tutores para muestra de presentación con nuevos apartados:**
 - Inclusión de las últimas mejoras y actualizaciones.
 - Ajustes finales antes de la presentación definitiva.

5. Estudio de los Resultados Conocidos

En esta sección, se realiza un análisis exhaustivo de los resultados obtenidos en estudios previos y la literatura existente relacionada con el uso de chatbots y modelos de procesamiento de lenguaje natural (NLP) en el ámbito de la seguridad y salud en el trabajo (SST). Este estudio proporciona un contexto teórico y práctico que fundamenta las decisiones metodológicas y técnicas adoptadas en el desarrollo del chatbot para Sabentis.

5.1. Revisión de la Literatura

Se llevó a cabo una revisión sistemática de la literatura para identificar investigaciones y estudios relevantes que hayan abordado la implementación de chatbots en diversos sectores, con un enfoque particular en la SST. Las principales fuentes consultadas incluyeron artículos académicos, conferencias, tesis de maestría y estudios de caso publicados en bases de datos reconocidas como IEEE Xplore, Springer, Elsevier y Google Scholar.

Principales hallazgos:

- **Eficiencia Operativa:** Diversos estudios han demostrado que los chatbots pueden mejorar significativamente la eficiencia operativa al automatizar respuestas a preguntas frecuentes y gestionar múltiples interacciones simultáneamente. Por ejemplo, el trabajo de Jain et al. (2018) destaca cómo los chatbots pueden reducir la carga de trabajo administrativo en el sector salud, permitiendo a los profesionales de la salud centrarse en tareas más complejas y críticas. Similarmente, un estudio de Adamopoulou y Moussiades (2020) muestra cómo la implementación de chatbots en empresas puede liberar recursos humanos para actividades estratégicas, mejorando así la eficiencia general de la organización.
- **Reducción de Costos:** La implementación de chatbots ha sido asociada con una reducción notable en los costos operativos, particularmente en áreas de atención al cliente y soporte técnico. Un estudio realizado por Deloitte (2019) en el sector financiero muestra que los chatbots pueden gestionar hasta el 80% de las consultas de clientes sin intervención humana, resultando en ahorros significativos en costos operativos. Otro estudio de Gartner (2020) proyecta que para 2024, los chatbots podrían generar ahorros anuales de hasta 11 mil millones de dólares en sectores clave como el retail, la banca y la salud.
- **Mejora en la Satisfacción del Usuario:** Los chatbots que utilizan modelos avanzados de NLP han mostrado mejorar la satisfacción del usuario al proporcionar respuestas rápidas y precisas, ajustadas a las consultas específicas de los usuarios. Un estudio de McKinsey (2021) en el sector de servicios muestra que los clientes valoran la rapidez y precisión de las respuestas proporcionadas por chatbots, lo que resulta en una mejora significativa en la satisfacción del cliente y en su lealtad.

It should have a brief overview of what was the technical approach to implement chatbot let's say 5 years ago and clearly write that currently there are plenty research about it.

5.2. Estudios de Caso Relevantes

Se analizaron varios estudios de caso donde se implementaron chatbots en diferentes industrias, destacando aquellos que utilizaron tecnologías similares a las consideradas en este proyecto.

Estudio de Caso 1: Tesla



En el sector manufacturero, Tesla implementó un chatbot para asistir en la gestión de consultas sobre seguridad y salud en el trabajo. El chatbot proporcionaba información sobre normativas de seguridad, procedimientos de emergencia y prevención de riesgos.

La implementación del chatbot en Tesla resultó en varios beneficios notables:

- **Reducción del Tiempo de Respuesta:** El tiempo dedicado a responder consultas repetitivas disminuyó significativamente, permitiendo a los empleados enfocarse en tareas más complejas.
- **Precisión de la Información:** Se mejoró la precisión de la información proporcionada, lo que llevó a una mayor conformidad con las normativas de seguridad y una reducción en incidentes relacionados con la SST.
- **Cultura de Seguridad Proactiva:** El chatbot facilitó una cultura de seguridad proactiva al permitir a los empleados acceder a información crítica en tiempo real.
- **Mejora en los Índices de Seguridad:** Tesla reportó una mejora en sus índices de seguridad, con una reducción del 50% en la tasa de lesiones por vehículo producido en comparación con el año anterior. Además, la tasa de lesiones totales registrables (TRIR) en la fábrica de Fremont mejoró y se mantuvo un 5% por debajo del promedio de la industria para grandes fabricantes según la Oficina de Estadísticas Laborales (BLS) ([Tesla](#)).

Estudio de Caso 2: Alcoa



En la industria del aluminio, Alcoa implementó un chatbot para mejorar la gestión de seguridad laboral, proporcionando información sobre procedimientos de seguridad y salud, y normativas internas.

Alcoa reportó varios beneficios significativos tras la implementación del chatbot:

- **Reducción de Incidentes de Seguridad:** Se observó una significativa reducción en los incidentes de seguridad laboral. La empresa logró alcanzar su objetivo de cero accidentes en varias de sus plantas gracias a la mejora en la gestión de la seguridad y la rapidez en la difusión de información crítica.
- **Eficiencia Operativa Mejorada:** La utilización del chatbot permitió una rápida difusión de la información y consultas en tiempo real, lo que resultó en una mayor conformidad y cumplimiento de las normativas de seguridad.
- **Cultura de Seguridad:** Alcoa ha cultivado una cultura de seguridad robusta, apoyada por su programa "Stop for Safety", que permite a cualquier empleado detener el trabajo si se identifica un riesgo de seguridad, reforzando su compromiso con la salud y seguridad de su personal ([Alcoa](#)) ([Alcoa](#)).

Estudio de Caso 3: DuPont



DuPont, una empresa del sector químico, introdujo un chatbot para gestionar la seguridad y salud ocupacional, facilitando información sobre manejo de materiales peligrosos, primeros auxilios y procedimientos de emergencia.

La implementación del chatbot en DuPont resultó en:

- **Reducción de Incidentes Relacionados con la Seguridad:** Hubo una disminución del 25% en los incidentes relacionados con la seguridad debido a la disponibilidad inmediata de información precisa.
- **Mejora en la Satisfacción y Confianza de los Empleados:** Los empleados mostraron una mayor satisfacción y confianza en las prácticas de seguridad de la empresa, lo que también mejoró la moral y la retención del personal.
- **Eficiencia en la Comunicación de Seguridad:** El chatbot permitió una gestión más eficiente de la información de seguridad, contribuyendo a una respuesta rápida y efectiva en situaciones de emergencia.

5.3. Evaluación de Modelos de Lenguaje

Se revisaron estudios específicos sobre los modelos de lenguaje utilizados en el desarrollo de chatbots, como Word2vec, TF-IDF, BERT y modelos de OpenAI, para evaluar su desempeño en tareas de comprensión y generación de texto.

Word2vec:

- **Ventajas:** La investigación de Mikolov et al. (2013) destaca la eficiencia de Word2vec en la generación de vectores de palabras, lo que permite una rápida similitud semántica.
- **Limitaciones:** Estudios posteriores, como el de Levy y Goldberg (2014), señalan que Word2vec tiene una menor capacidad para capturar relaciones contextuales complejas en comparación con modelos más avanzados como BERT.

TF-IDF:

- **Ventajas:** Salton y McGill (1986) demostraron la simplicidad y efectividad de TF-IDF en la recuperación de información basada en la relevancia de términos.
- **Limitaciones:** Sin embargo, TF-IDF no puede captar relaciones semánticas entre términos y contextos, lo que limita su aplicabilidad en tareas más complejas de NLP.

BERT:

- **Ventajas:** Devlin et al. (2018) mostraron que BERT tiene una capacidad avanzada para comprender el contexto de palabras en una oración, lo que mejora significativamente la precisión de las respuestas generadas.
- **Limitaciones:** BERT requiere mayor capacidad computacional y tiempo de procesamiento, lo que puede ser una barrera para su implementación en entornos con recursos limitados.

Modelos de OpenAI:

- **Ventajas:** Radford et al. (2019) demostraron que los modelos de OpenAI, como GPT-2, tienen una alta capacidad para generar texto coherente y relevante, incluso en contextos complejos.
- **Limitaciones:** Estos modelos requieren recursos computacionales significativos y pueden presentar problemas de "alucinación", donde el modelo genera respuestas inexactas o irrelevantes.

5.4. Aplicación de Resultados al Proyecto

Los resultados de los estudios analizados proporcionan una base sólida para las decisiones técnicas y metodológicas adoptadas en este proyecto. Al comprender las fortalezas y limitaciones de cada modelo de lenguaje y las experiencias previas de implementación de chatbots en diferentes sectores, se ha diseñado un sistema que maximiza la eficiencia operativa, mejora la satisfacción del usuario y se adapta a las necesidades específicas de Sabentis.

Maximización de la Eficiencia Operativa

Los casos de Tesla, Alcoa y DuPont demuestran cómo la implementación de chatbots puede reducir significativamente el tiempo de respuesta a consultas repetitivas y mejorar la precisión de la información proporcionada. En el contexto de Sabentis, estas mejoras se traducirán en una gestión más ágil y precisa de las consultas sobre seguridad y salud en el trabajo (SST). La automatización de respuestas a preguntas frecuentes permitirá al personal enfocarse en tareas más complejas y estratégicas, mejorando la productividad general de la empresa ([Tesla](#)) ([Alcoa](#)) ([Alcoa](#)).

Mejora de la Satisfacción del Usuario

Los chatbots, al proporcionar respuestas rápidas y precisas, han demostrado ser efectivos en mejorar la satisfacción del usuario. Los estudios de McKinsey (2021) y otros indican que los usuarios valoran la rapidez y exactitud en las respuestas, lo cual es crucial para mantener una alta satisfacción y lealtad del cliente. En el caso de Sabentis, esto se traducirá en una mejor experiencia para los usuarios, quienes podrán acceder a información crítica de SST en tiempo real y recibir soporte continuo, especialmente en situaciones de emergencia .

Adaptabilidad a las Necesidades Específicas de Sabentis

Los estudios revisados destacan la importancia de personalizar los chatbots para adaptarse a las necesidades específicas de cada organización. En Sabentis, el chatbot será diseñado para manejar consultas relacionadas con más de 100 manuales explicativos de SST y 43 módulos especializados. Esta personalización asegurará que el chatbot pueda proporcionar información precisa y relevante, mejorando la efectividad de las respuestas y facilitando una cultura de seguridad proactiva dentro de la empresa.

Fortalezas y Limitaciones de los Modelos de Lenguaje

La revisión de modelos como Word2vec, TF-IDF, BERT y los modelos de OpenAI ha sido fundamental para seleccionar la tecnología más adecuada para Sabentis. Cada modelo tiene sus propias fortalezas y limitaciones:

- **Word2vec:** Eficiente para generar vectores de palabras, pero con limitaciones en capturar relaciones contextuales complejas .
- **TF-IDF:** Simple y efectivo para recuperar información basada en la relevancia de términos, pero limitado en su capacidad para captar relaciones semánticas .
- **BERT:** Avanzado en la comprensión del contexto de palabras, pero con alta demanda computacional .
- **Modelos de OpenAI:** Capaces de generar texto coherente y relevante en contextos complejos, aunque requieren significativos recursos computacionales y pueden generar respuestas inexactas o irrelevantes .

La implementación del chatbot en Sabentis se beneficia de los aprendizajes y resultados observados en otros sectores. Al integrar las mejores prácticas y tecnologías disponibles, el proyecto no solo maximiza la eficiencia operativa y mejora la satisfacción del usuario, sino que también se adapta perfectamente a las necesidades específicas de Sabentis, asegurando una gestión efectiva y proactiva de la seguridad y salud en el trabajo. Este enfoque integral posiciona a Sabentis como líder en la transformación digital del sector SST, reforzando su compromiso con la innovación y la excelencia operativa.

6. Análisis de Datos

6.1. Visión General de los Datos

Sabentis proporcionó seis manuales en PDF y un documento de preguntas frecuentes (FAQ) que sirven como base para el desarrollo del chatbot. Estos documentos incluyen los siguientes temas:

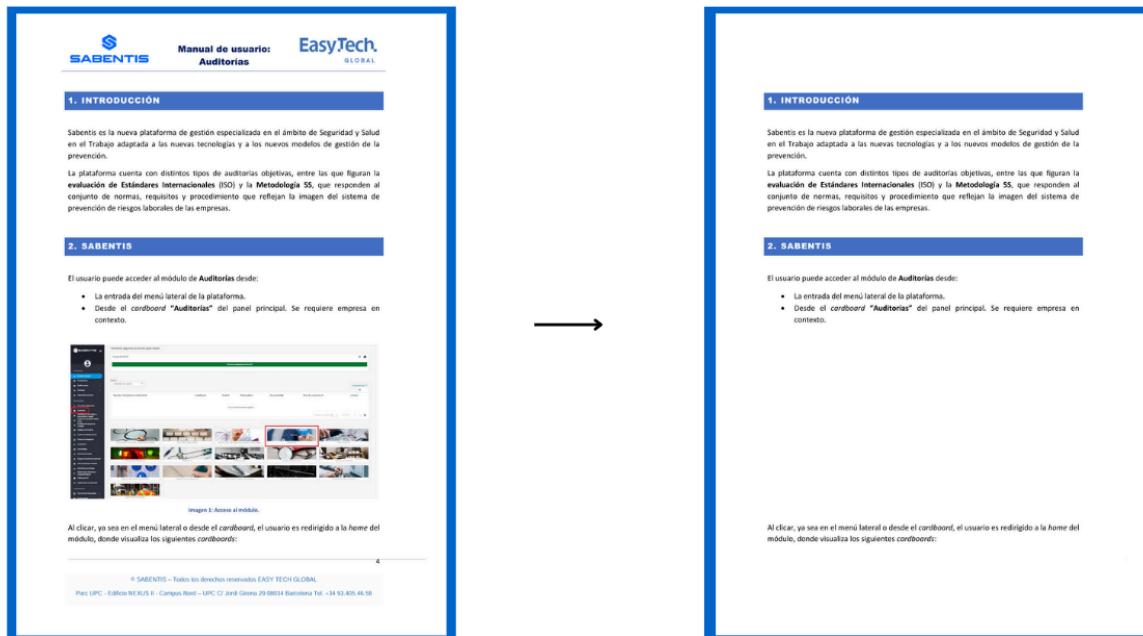
1. Auditorías
2. Ausentismo
3. Estructura Organizativa
4. Identificación y Evaluación de Riesgos
5. Información Documentada
6. Planes de Emergencia

Estos documentos constituyen el corpus principal sobre el cual se entrenará y evaluará el chatbot para garantizar que pueda proporcionar respuestas precisas y útiles a las consultas relacionadas con la seguridad y salud en el trabajo.

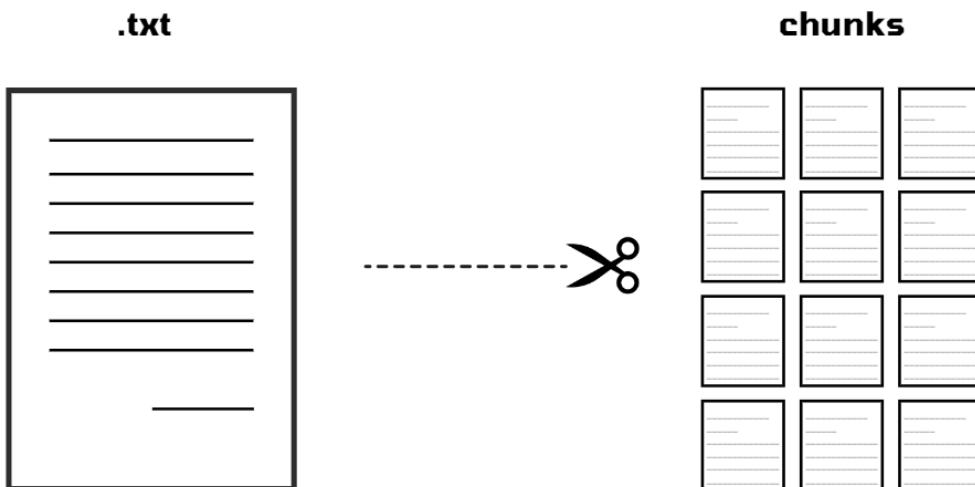
6.2. Preprocesamiento de Datos Específicos

El preprocesamiento de los datos se llevó a cabo utilizando Python y la biblioteca PyMuPDF. Los principales pasos del preprocesamiento incluyen:

1. **Extracción de Texto:** Se extrajo el texto de los documentos PDF, eliminando elementos no textuales como encabezados, pies de página e imágenes que no aportan valor a la información requerida.



2. **Limpieza de Datos:** Se eliminaron datos redundantes y se corrigieron errores en el texto extraído para asegurar la coherencia y precisión de la información.
 3. **Chunking:** Los documentos fueron divididos en fragmentos (chunks) manejables. Este proceso permite una mejor gestión y análisis de la información contenida en los documentos.



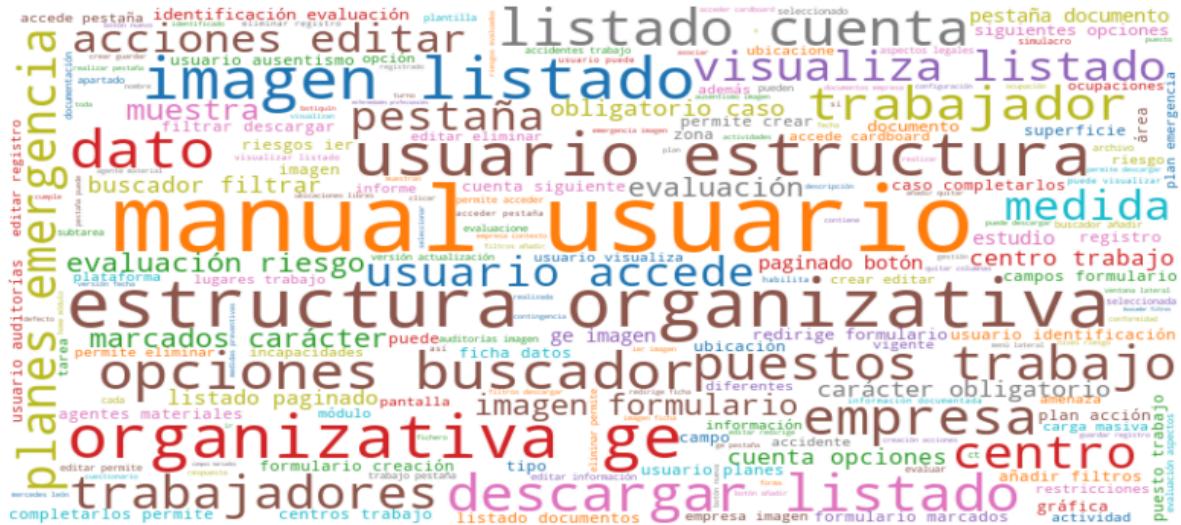
Para dividir los documentos en fragmentos, se utilizó la librería `RecursiveCharacterTextSplitter` de Python, con un tamaño de fragmento (chunk size) de 500 caracteres y un solapamiento de 60 caracteres entre fragmentos para mantener el contexto.

6.3. Análisis Exploratorio de Datos

El análisis exploratorio de los datos incluyó varias técnicas para entender mejor la estructura y contenido de los documentos. Algunas de las técnicas utilizadas fueron:

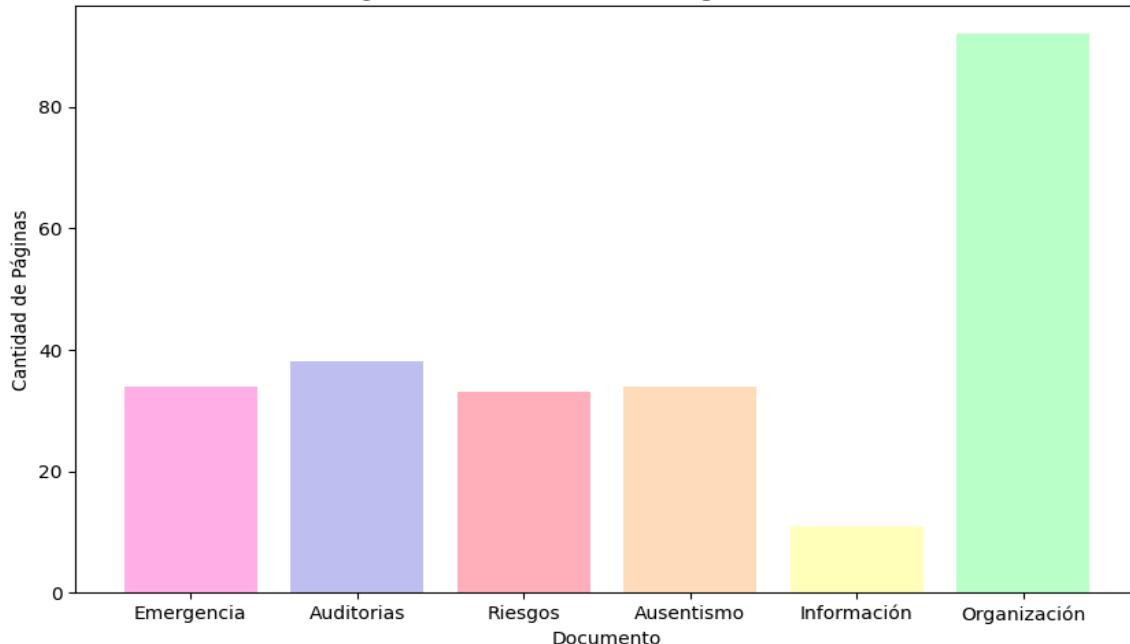
1. **Wordclouds:** Se generan nubes de palabras (wordclouds) para identificar términos relevantes y frecuentes en los documentos. Aunque inicialmente no proporcionaron información relevante, ayudaron a entender la frecuencia de ciertos términos clave.

¿Qué palabras predominan en los documentos?



2. **Análisis de la longitud de los documentos:** Se analizaron las longitudes de los documentos para detectar si todos tenían las mismas características.

¿Los manuales tienen una longitud similar?



3. **Análisis de Patrones:** Se exploraron los documentos visualmente para detectar patrones, información innecesaria y secciones clave que podrían influir en el rendimiento del chatbot.

Durante el análisis, se observó que los documentos tenían una cantidad de páginas similar, con la excepción de dos documentos que tenían una cantidad de páginas significativamente diferente. Esta observación ayudó a determinar si esos documentos podrían considerarse outliers en el conjunto de datos.

Analizamos cada manual por separado y en conjunto, que palabras son mas y menos habituales, que unión de 2 y 3 palabras son mas y menos habituales, por tal de entender mejor el contenido de los documentos y sacar insights a la hora de poder mejorar modelos como TF-IDF. Se muestran palabras de la unión de todos los documentos.

Detectamos palabras residuales que aparecen o en un solo documento o muy pocas veces, y palabras muy generales que aparecen en todos los documentos.

	Palabra	Frecuencia
0	listado	288
1	trabajo	196
2	usuario	179
3	pestaña	143
4	permitir	141
5	editar	138
6	evaluación	127
7	acción	120
8	formulario	116
9	visualizar	110
10	acceder	105
11	trabajador	96
12	centro	89
13	descargar	88
14	empresa	86
15	eliminar	78
16	puesto	73
17	opción	73
18	añadir	70
19	medida	65

Palabras más frecuentes

	Palabra	Frecuencia
0	nombre	6
1	cardboard	6
2	generado	6
3	archivo	7
4	documentos	8
5	información	8
6	módulo	8
7	plataforma	8
8	informe	9
9	medidas	10
10	nivel	10
11	evaluar	10
12	plantilla	11
13	riesgos	11
14	columna	12
15	quitar	12
16	filtro	12
17	iso	13
18	cerrar	14
19	legal	14

Palabras menos frecuentes

	Bigram	Frecuencia
0	(visualizar, listado)	71
1	(opción, buscador)	70
2	(descargar, listado)	67
3	(puesto, trabajo)	61
4	(acción, editar)	56
5	(centro, trabajo)	52
6	(usuario, acceder)	51
7	(siguiente, opción)	44
8	(buscador, filtrar)	44
9	(obligatorio, caso)	43
10	(listado, paginado)	42
11	(redirigir, formulario)	39
12	(listado, opción)	36
13	(formulario, creación)	34
14	(carácter, obligatorio)	32
15	(filtrar, descargar)	32
16	(marcado, carácter)	31
17	(permitir, crear)	29
18	(caso, completarlo)	28
19	(completarlo, permitir)	27

	Trigram	Frecuencia
0	(marcado, carácter, obligatorio)	46
1	(siguiente, opción, buscador)	44
2	(opción, buscador, filtrar)	44
3	(carácter, obligatorio, caso)	43
4	(descargar, listado, paginado)	42
5	(obligatorio, caso, completarlo)	40
6	(caso, completarlo, permitir)	38
7	(formulario, marcado, carácter)	36
8	(campo, formulario, marcado)	35
9	(redirigir, formulario, creación)	33
10	(buscador, filtrar, descargar)	32
11	(filtrar, descargar, listado)	32
12	(completarlo, permitir, crear)	32
13	(listado, siguiente, opción)	26
14	(listado, opción, buscador)	26
15	(pestaña, visualizar, listado)	19
16	(permitir, crear, editar)	19
17	(crear, editar, registro)	19
18	(acción, editar, eliminar)	19
19	(usuario, acceder, pestaña)	19

Unión de 2 palabras más frecuentes

Unión de 3 palabras más frecuentes

Al tratarse de manuales de uso de la plataforma hay muchas referencias a acciones, elementos de la propia UI de la web de Sabentis, ya que muchos de los documentos aportados no solo muestran información de reglas y normas generales en el entorno de la seguridad y salud, si no ayuda al usuario para poder realizar trámites sobre la seguridad y salud en su centro de trabajo mediante el entorno de Sabentis.

Entendemos pues que la complejidad no está en la seguridad y salud, si no en darle al usuario la capacidad de aprender a usar la plataforma de una forma mas fácil gracias a contar con una inteligencia artificial que desde cualquier lugar de la página web pueda responder a cualquier pregunta.

7. Selección de Soluciones Técnicas

Investigación Inicial

Para determinar el proceso más adecuado para el desarrollo del chatbot, comenzamos investigando diversos artículos académicos y documentación técnica disponible en internet.

Buscamos identificar las mejores prácticas y las metodologías más eficaces para implementar un chatbot capaz de manejar información específica contenida en los manuales de Sabentis.

~~Primeras Opciones Evaluadas~~

ChatGPT LLM sin Información Directa:

Nuestra primera opción fue implementar directamente ChatGPT LLM. Sin embargo, descartamos esta solución de inmediato ya que, sin la información directa de los manuales, el sistema tomaba datos de internet, lo cual resultaba en respuestas inexactas o inventadas ("alucinaciones").



Realizamos la prueba preguntando a ChatGPT sin contexto. Como era previsible, ChatGPT alucinaba en muchas respuestas, aunque algunas respuestas oficiales como normativas ISO eran correctas. No podíamos confiar en esta aproximación para proporcionar información correcta y exclusiva de Sabentis.

Add example of some answers.

LLM con Información de los Manuales:

La siguiente opción fue utilizar ChatGPT LLM añadiendo los manuales de Sabentis. Aunque esta solución mejoró ligeramente la precisión, seguía generando respuestas alucinadas, lo que la hizo inadecuada para nuestras necesidades.



Cargamos un entorno OpenWebUI usando Docker, donde se le pueden añadir diferentes LLM locales y cargar documentos.

+ **Llm finetuned :**



Here write that the model itself can be retrained with specific documents.

We did not try this method.

1) Because the model should be retrained once new document come

2) the results can be unstable
and it has a lot of risk

Subimos distintos PDFs a Ollama, y a Mixtral, y le hicimos preguntas sobre ellos. Aunque la diferencia con ChatGPT sin contexto fue significativa, Ambos modelos aunque Mixtral era más eficiente que Ollama al estar entrenado en español aún se inventaban respuestas.

Retrieval-Augmented Generation (RAG)

Finalmente, después de un análisis exhaustivo y de la revisión de varios artículos clave, decidimos utilizar la metodología de Retrieval-Augmented Generation (RAG). Los siguientes artículos fueron particularmente influyentes en nuestra decisión:

Una encuesta sobre RAG encuentra modelos de lenguaje grandes: hacia modelos de lenguaje grandes aumentados por recuperación por Yujuan Ding y otros. Este documento ofrece una visión completa de cómo las técnicas aumentadas por recuperación se integran con modelos de lenguaje grandes, destacando su impacto y aplicación en diversas tareas de IA.

Generación aumentada por recuperación (RAG): de la teoría a LangChain. Este artículo proporciona información sobre las aplicaciones prácticas de RAG, explorando su integración con cadenas de lenguaje para mejorar las tareas de PNL.

Utilizar RAG para desarrollar un chatbot que busque información en un conjunto de documentos tiene varias ventajas importantes:

- **Acceso a Información actualizada y relevante:** RAG permite al chatbot recuperar información de una base de datos o un conjunto de documentos en tiempo real, asegurando que las respuestas sean actuales y relevantes para las consultas del usuario. Esto es especialmente útil en dominios donde la información cambia rápidamente.
- **Personalización de respuestas:** Gracias al componente de recuperación, RAG puede ajustar sus respuestas basándose en los datos más pertinentes extraídos de documentos específicos, ofreciendo una personalización que los modelos de generación pura no pueden alcanzar.
- **Eficiencia en el procesamiento:** En vez de generar respuestas desde cero, el modelo RAG busca primero los fragmentos más relevantes de texto, que pueden servir como base para generar la respuesta final. Esto es más eficiente desde el punto de vista computacional.
- **Mejora continua del sistema:** Incorporando un mecanismo de retroalimentación, los sistemas basados en RAG pueden mejorar con el tiempo, aprendiendo qué documentos son más útiles para diferentes tipos de preguntas y afinando sus métodos de recuperación y generación.
- **Reducción de sesgos y errores:** Al basar las respuestas en información recuperada de fuentes confiables y verificadas, se reduce el riesgo de generar respuestas incorrectas o sesgadas, comunes en modelos entrenados exclusivamente en datos de internet.

Para aplicar la metodología de RAG en nuestro proyecto, seguimos estos pasos y consideraciones:

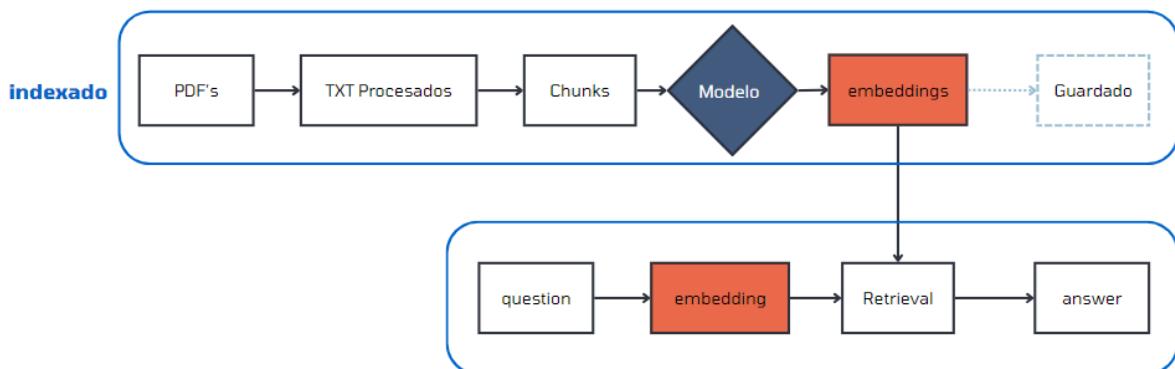
1. Indexación:

- Recogemos y organizamos los datos en un formato que sea fácilmente accesible. Utilizamos herramientas como vectorizadores de documentos para transformar el texto en representaciones numéricas que pueden ser indexadas y recuperadas rápidamente.

2. Recuperación y Generación:

- Durante la ejecución, el sistema recupera la información relevante basada en la entrada del usuario y luego utiliza un modelo de lenguaje para generar una respuesta basada en esta información.
- Observamos que las respuestas iniciales del sistema eran poco "humanas". Para mejorar esto, añadimos un prompt y realizamos llamadas a la API de ChatGPT.

RAG



El uso de RAG ofrece un enfoque robusto y adaptable para la creación de chatbots que necesitan buscar y utilizar información de un conjunto específico de documentos. Este método maximiza la relevancia, precisión y utilidad de las respuestas generadas, mejorando la interacción con los usuarios y la eficiencia operativa del sistema de gestión de seguridad y salud en el trabajo de Sabentis.

Finally, add one more diagram with
ChatGPT generating answer and
explain why it is necessary -

This is final technical selection. Add diagram
of the complete solution (with streamlit)
and explain with example

8. Configuración de la Comparación

Para comparar la eficacia de los diferentes modelos de incrustación y algoritmos de recuperación de información, se configuraron pruebas utilizando dos métricas de similitud principales: la Similitud de Coseno y la Distancia Euclídea. Estas pruebas son esenciales para evaluar cómo los diferentes modelos manejan las consultas y recuperan la información pertinente de los manuales.

Hacemos unos ejercicios sencillos para comprender el funcionamiento de las 2 métricas que queremos probar, que explicamos en un poco mas de detalle en el siguiente punto.

Para hacer este ejercicio elegimos 3 frases sencillas, 2 de ellas son muy parecidas gramaticalmente, y 1 de ellas completamente diferente, así podemos ver los resultados y como una métrica interpreta los diferentes valores. Ponemos aquí los resultados obtenidos para TF-IDF y OpenAI embeddings..

```
import openai
import numpy as np
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity,
euclidean_distances

# Frases de ejemplo
sentences = [
    "La casa es de color rojo",
    "La casa es de color azul",
    "Fui de vacaciones a Tokyo"
]

# Convertir frases a embeddings usando TF-IDF
def tfidf_embeddings(sentences):
    vectorizer = TfidfVectorizer()
    tfidf_matrix = vectorizer.fit_transform(sentences)
    return tfidf_matrix.toarray()

# Convertir frases a embeddings usando OpenAI
def openai_embeddings(sentences):
    openai.api_key = 'editada por seguridad'
    embeddings = []
    model = "text-embedding-ada-002"
    for sentence in sentences:
        response = openai.Embedding.create(input=sentence,
model=model)
        embeddings.append(response['data'][0]['embedding'])
    return np.array(embeddings)

# Calcular y mostrar las similitudes
def calculate_similarities(embeddings, method):
    cos_sim = cosine_similarity(embeddings)
    euc_dist = euclidean_distances(embeddings)

    print(f"Similitud coseno ({method}):")
```

```

        print(f"Entre 'La casa es de color rojo' y 'La casa es de color azul': {cos_sim[0, 1]}")
        print(f"Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo': {cos_sim[0, 2]}")
        print()

        print(f"Distancia euclídea ({method}):")
        print(f"Entre 'La casa es de color rojo' y 'La casa es de color azul': {euc_dist[0, 1]}")
        print(f"Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo': {euc_dist[0, 2]}")
        print()

# Embeddings con TF-IDF
tfidf_embeds = tfidf_embeddings(sentences)
print("Embeddings TF-IDF:")
print(tfidf_embeds)
print()
calculate_similarities(tfidf_embeds, "TF-IDF")

# Embeddings con OpenAI
openai_embeds = openai_embeddings(sentences)
print("Embeddings OpenAI:")
print(openai_embeds)
print()
calculate_similarities(openai_embeds, "OpenAI")

```

Embeddings TF-IDF:

```

[[0.          0.39740155 0.39740155 0.30861775 0.39740155 0.
  0.39740155 0.52253528 0.          0.          ],
 [0.52253528 0.39740155 0.39740155 0.30861775 0.39740155 0.
  0.39740155 0.          0.          0.          ],
 [0.          0.          0.          0.32274454 0.          0.54645401
  0.          0.          0.54645401 0.54645401]]

```

Similitud coseno (TF-IDF) :

Entre 'La casa es de color rojo' y 'La casa es de color azul':

0.7269568815850795

Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo':

0.09960469563447014

Distancia euclídea (TF-IDF) :

Entre 'La casa es de color rojo' y 'La casa es de color azul':

0.7389764792128647

Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo':

1.341935396630948

Embeddings OpenAI:

```
[ [-0.00956719  0.00406699 -0.00672557 ... -0.02097089 -0.01195589
-0.01218545]
[-0.00796125  0.01537644 -0.00401786 ... -0.00297073 -0.01737451
-0.01797021]
[-0.01781411 -0.00605167 -0.0057139 ... -0.0027037   0.01396106
-0.01084609]]
```

Similitud coseno (OpenAI) :

Entre 'La casa es de color rojo' y 'La casa es de color azul':

0.9427289746594278

Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo':

0.7868335420496669

Distancia euclídea (OpenAI) :

Entre 'La casa es de color rojo' y 'La casa es de color azul':

0.33844062500745853

Entre 'La casa es de color rojo' y 'Fui de vacaciones a Tokyo':

0.652941752495682

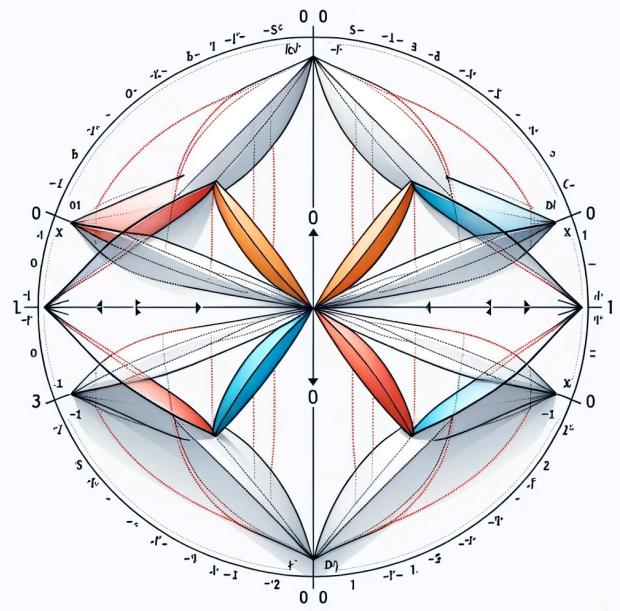
9. Métricas de Comparación

Similitud de Coseno

La Similitud de Coseno mide el coseno del ángulo entre dos vectores proyectados en un espacio vectorial. El valor resultante varía entre -1 y 1:

- **1** indica que los vectores están en la misma dirección (máxima similitud).
- **0** indica que los vectores son ortogonales (sin similitud).
- **-1** indica que los vectores están en direcciones opuestas (máxima disimilitud).

En este caso, al medir el ángulo entre dos vectores y determinar si están en la misma dirección o en otra, un valor de similitud mayor representa que el vector de la pregunta se parece más al vector de la posible respuesta (chunk de texto del manual).



Esta imagen ilustra la similitud de coseno entre dos vectores en un espacio vectorial. Fuente: Creada con IA a través de Chat GPT.

La fórmula de la similitud de coseno entre dos vectores **A** y **B** en un espacio vectorial es:

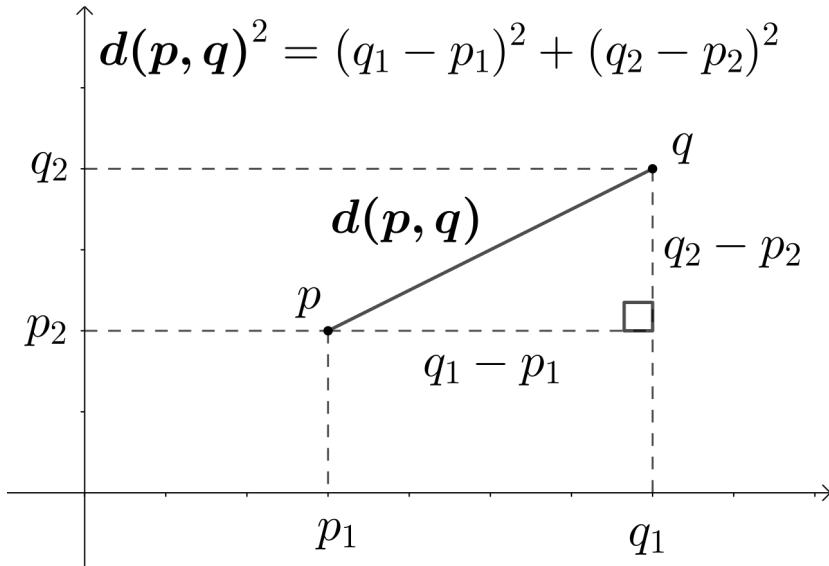
$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Donde:

- $\mathbf{A} \cdot \mathbf{B}$ es el producto punto (o producto escalar) de los vectores **A** y **B**.
- $\|\mathbf{A}\|$ es la magnitud (o norma) del vector **A**.
- $\|\mathbf{B}\|$ es la magnitud (o norma) del vector **B**.

Distancia Euclídea

La Distancia Euclídea es una medida de la longitud del segmento de línea recta que conecta dos puntos en un espacio euclidiano. En términos matemáticos, es la distancia "ordinaria" entre dos puntos, calculada usando el teorema de Pitágoras.



*En la imagen se usa el teorema de Pitágoras para calcular la distancia euclídea bidimensional. Fuente Wikipedia.

En este contexto, cómo mide la distancia entre dos puntos, un valor menor de distancia indica que los vectores están más cerca, resultando en que los vectores de posibles respuestas están más cerca del vector de la pregunta correspondiente. Así, un valor menor indica mayor similaridad entre los vectores.

La fórmula de la distancia euclídea entre dos vectores \mathbf{A} y \mathbf{B} en un espacio vectorial es:

$$d(\mathbf{A}, \mathbf{B}) = \sqrt{\sum_{i=1}^n (A_i - B_i)^2}$$

Para dos vectores en un espacio n -dimensional, $\mathbf{A}=(A_1, A_2, \dots, A_n)$ y $\mathbf{B}=(B_1, B_2, \dots, B_n)$, la fórmula se puede expresar como:

$$d(\mathbf{A}, \mathbf{B}) = \sqrt{(A_1 - B_1)^2 + (A_2 - B_2)^2 + \dots + (A_n - B_n)^2}$$

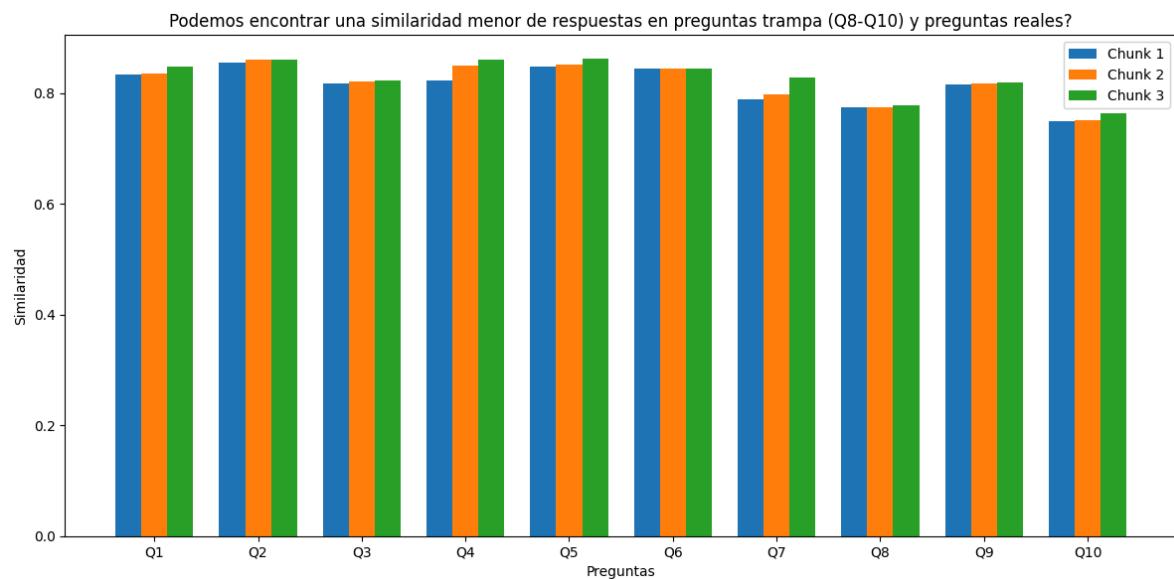
Métrica Elegida

No se observó una diferencia significativa entre los resultados obtenidos con la Similitud de Coseno y la Distancia Euclídea. Debido a este motivo, junto con la facilidad de interpretación y la amplia utilización de esta métrica en el procesamiento del lenguaje natural (NLP), optamos por utilizar la **Similitud de Coseno** para este proyecto.

Cálculo del Threshold Óptimo

El proceso de Retrieval-Augmented Generation siempre devuelve información para cualquier pregunta, incluso si esta no está relacionada con los documentos. Por lo tanto, es crucial establecer un mecanismo de validación para determinar la relevancia de los chunks de texto.

Para ello, analizamos cómo el modelo OpenAI Embeddings (ADA) maneja los resultados positivos y falsos positivos. Realizamos preguntas al modelo, algunas extraídas del documento de Preguntas Frecuentes y otras inventadas, que no tenían relación con el ecosistema de Sabentis ni con la seguridad y salud en el entorno laboral.



Usamos la métrica de Similitud de Coseno para comparar las respuestas y observamos una ligera diferencia entre las respuestas a las preguntas con contexto real y las preguntas fuera de contexto. Aunque la diferencia no es notable, pudimos observar una tendencia a la baja en las respuestas a las preguntas fuera de contexto.

Trazamos un threshold de **0.80** de Similitud de Coseno. Este umbral nos permite, sin perder mucha información relevante, establecer una línea de corte:

- Si el modelo devuelve respuestas con una similitud inferior a 0.80, estas no serán presentadas al usuario.
- Si las respuestas superan el threshold de 0.80, continuarán con los siguientes pasos en el procesamiento y generación de respuestas.

Este threshold es crucial para asegurar que las respuestas ofrecidas por el modelo sean precisas y relevantes, mejorando así la experiencia del usuario y la eficacia del chatbot.

10. Comparación de Modelos

Bag of words → Words Embedding's → RNN based models → LSTM based models → Attention based → Transformers

Cronología de los modelos de procesamiento del lenguaje natural. Fuente: Wikipedia

10.1. BERT

BERT (Bidirectional Encoder Representations from Transformers) es una técnica avanzada de modelado de lenguaje basada en la arquitectura de transformers, que se distingue por utilizar mecanismos de atención para captar el contexto de una palabra en todas las posiciones de un texto ingresado.

A diferencia de modelos anteriores, BERT se entrena simultáneamente en dos tareas: predicción de la siguiente oración y enmascaramiento de palabras (MLM). Esta capacidad para comprender el contexto bidireccional de las palabras es fundamental para generar respuestas coherentes y contextualmente adecuadas en aplicaciones como los chatbots.

Para la evaluación de modelos en este trabajo, se utilizó un corpus proporcionado por los manuales de SST de Sabentis, abarcando temas diversos como auditorías, ausentismo y planes de emergencia. Mediante herramientas de Python como PyMuPDF, se extrajo y limpió el texto de estos manuales, eliminando elementos no textuales que podrían interferir con el procesamiento del lenguaje natural.

Selección del Modelo "dccuchile/bert-base-spanish-wwm-uncased"

Para la implementación específica, se seleccionó el modelo "dccuchile/bert-base-spanish-wwm-uncased", una adaptación de BERT preentrenada específicamente para el idioma español. Este modelo, conocido comúnmente como BETO, se ha entrenado utilizando la técnica de Enmascaramiento de Palabra Completa (Whole Word Masking), lo cual mejora la comprensión del modelo sobre las palabras completas en lugar de fragmentos de palabras. Este enfoque es particularmente beneficioso para el español debido a su rica morfología y variaciones conjugadas ([GitHub](#)) ([Spark NLP](#)).

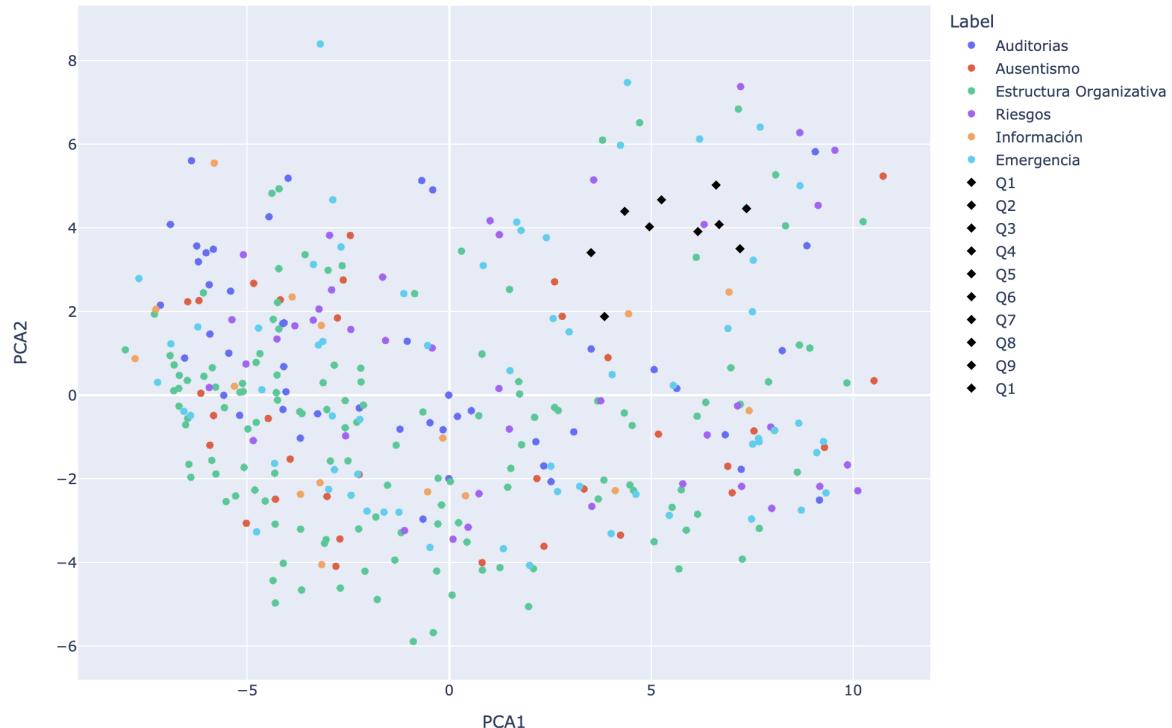
BETO ha demostrado superar a otros modelos multilingües en tareas como el reconocimiento de entidades nombradas (NER) y el análisis de sentimientos, especialmente en contextos en español ([GitHub](#)). Esto se debe a su entrenamiento en un extenso corpus en español, permitiéndole captar una amplia gama de usos lingüísticos y contextos que son esenciales para entender y procesar el lenguaje técnico y específico encontrado en los manuales de SST.

La integración de este modelo se facilita mediante la biblioteca de Transformers de Hugging Face, utilizando BertTokenizer para la tokenización y BertModel para la generación de embeddings. La elección de un modelo preentrenado se justifica por su probada capacidad para procesar y comprender el español a un nivel que es crucial para el análisis efectivo de textos técnicos y específicos.

Visualización de Embeddings

Se generaron embeddings para segmentos de texto extraídos, los cuales luego se visualizaron utilizando técnicas de reducción de dimensionalidad como PCA y t-SNE, implementadas a través de sklearn. Estas técnicas permiten una exploración visual de la distribución de los embeddings y ayudan a verificar la capacidad del modelo para agrupar información relacionada de manera significativa.

Reducción de Dimensionalidad con PCA (2D)



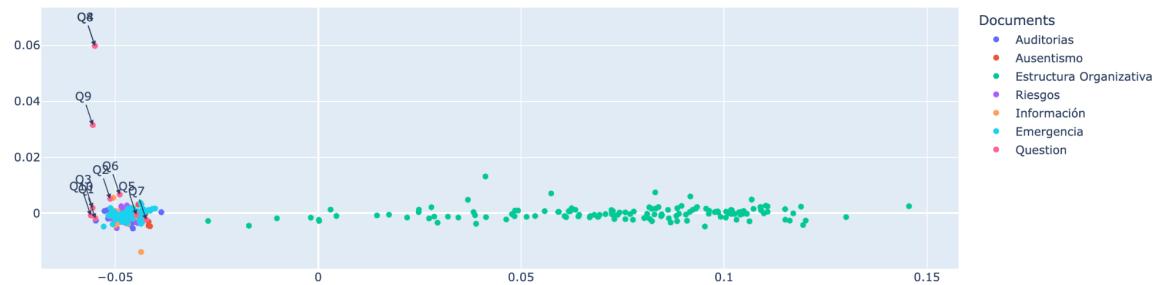
[AQUÍ VA LAS DOS GRÁFICAS] Falta la gráfica TSNE y arreglar la gráfica de PCA que aparecen dos Q1.

10.2. Word2vec

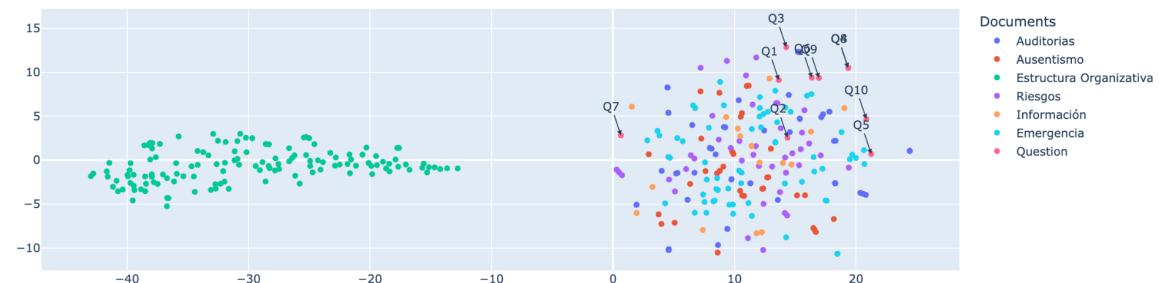
Word2vec es una técnica de procesamiento de lenguaje natural (NLP) que se utiliza para convertir palabras en vectores de números (vectores de palabras) que capturan el significado semántico de las palabras. Las palabras que tienen significados similares tienen representaciones vectoriales similares.

El modelo Word2vec fue aplicado a los textos extraídos de los manuales de Sabentis para transformar las palabras en vectores numéricos. Este enfoque permitió al sistema ofrecer respuestas relevantes a las consultas de los usuarios basándose en la similitud semántica entre las palabras de la consulta y los fragmentos de texto de los manuales.

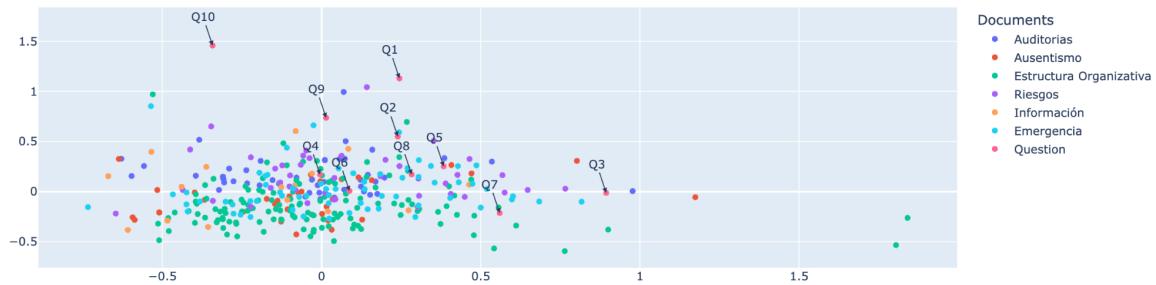
Embeddings Visualization with 2D PCA



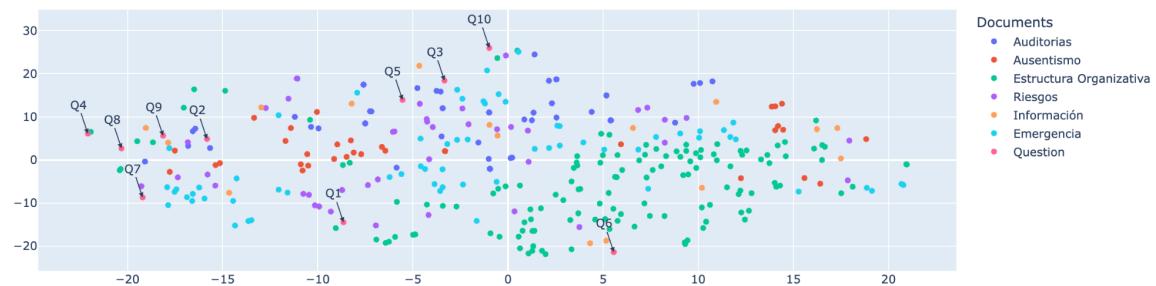
Embeddings Visualization with 2D t-SNE



Embeddings Visualization with 2D PCA



Embeddings Visualization with 2D t-SNE



EXPLICACION MAS DETALLADA Y GRAFICAS DE WORD2VEC AQUI

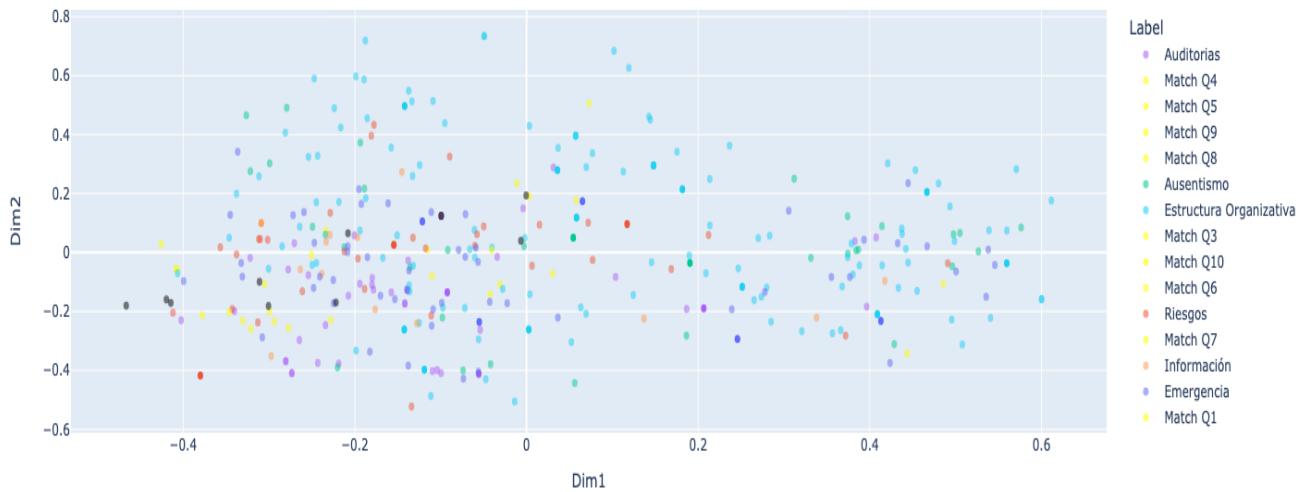
10.3. TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) es un modelo focalizado en la recuperación de información que evalúa la importancia de una palabra en un conjunto de documentos. Combina dos métricas: la frecuencia de términos (TF) y la frecuencia inversa de documentos (IDF).

- **Frecuencia de términos (TF):** Cuántas veces aparece una palabra en el documento, dividido por la cantidad de palabras en el documento.
- **Frecuencia inversa de documentos (IDF):** El número total de documentos dividido entre los documentos que contienen el término buscado en formato logarítmico.

Para mejorar la eficiencia del modelo, se aplicaron técnicas adicionales como lemmatization (reducción de las palabras a su raíz) y la eliminación de stopwords (palabras comunes y frecuentes que no aportan valor a la frase). Este modelo, aunque sencillo de implementar, tiene limitaciones en cuanto a la semántica y el orden de las palabras.

Cómo se relacionan las preguntas con los manuales? (PCA 2D)



En las pruebas realizadas con TF-IDF vemos como el manual estructura organizativa abarca gran parte del espacio, incluso mezclados en el espacio junto con chunks de otros manuales, aun así se detectan diferentes áreas localizadas de manuales, vemos como las preguntas se van repartiendo en el espacio y las respuestas correspondientes a esas preguntas están en muchos casos aproximadas a esas preguntas. Aun así TF-IDF da relevancia a cuántas veces aparece una palabra en un chunk de texto con lo que interpreta que los chunks en los que aparece repetida múltiples veces una palabra contenida en la pregunta tiene más relevancia que aquel chunk que solo la tiene una vez, la exploración de estos chunks cuando pasamos el cursor por encima de ello nos lo confirma.

10.4. OpenAI

OpenAI Embeddings (ADA) es un modelo avanzado entrenado para entender decenas de miles de palabras, ofreciendo una ventaja significativa frente a Word2vec y TF-IDF. Es capaz de entender la semántica, los diferentes significados de las palabras y sus sinónimos, generando una amplia versatilidad.

Como se distribuye cada manual y sus respuestas con las preguntas? (PCA)



En las pruebas realizadas, OpenAI Embeddings mostró ser rápido y eficiente en la generación de embeddings de cada documento, proporcionando las respuestas más idóneas gracias a su capacidad para entender el contexto y los sinónimos.

11. Despliegue

Sabentis ya cuenta con una arquitectura propia donde proporciona a sus usuarios una plataforma web que aloja sus documentos en PDF. El futuro de este proyecto es implementar el chatbot en su plataforma, permitiendo un acceso rápido y eficiente a la información. Se desarrollará una función en Python para detectar cambios en los documentos fuente, generando textos actualizados para el modelo.

Para el despliegue, el proyecto necesitará acceder al directorio de PDFs y contar con un servicio de almacenamiento en la nube o físico para alojar los textos procesados y la nueva base de datos basada en estos datos procesados y sus diferentes embeddings.

12. Resumen y Trabajo Futuro

El proyecto concluye con una serie de recomendaciones para la implementación y despliegue del chatbot, así como propuestas para futuras mejoras y optimizaciones. Estas incluyen:

- **Actualización Automática de Contenidos:** Desarrollar una función en Python que automáticamente detecte cambios o adiciones en los documentos fuente, generando textos actualizados para el modelo.
- **Desarrollo de una Interfaz Web:** Planificación y diseño de una interfaz web que integre el chatbot, facilitando la interacción de los usuarios con el sistema.
- **Evaluación de Tecnologías Avanzadas:** Basándose en el progreso y resultados hasta la fecha, considerar la evaluación de tecnologías avanzadas como Transformers para mejorar aún más el proyecto.

13. Referencias bibliográficas

Natural Language Processing with Transformers - Lewis Tunstall, Leandro von Werra, Thomas Wolf
Generative AI with Langchain - Ben Auffarth
Designing Large Language Model Applications - Suhas Pai
https://huggingface.co/docs/transformers/model_doc/rag
Is Cosine-Similarity of Embeddings Really About Similarity? - Netflix Research
Grafica distancia euclidiana https://en.wikipedia.org/wiki/Euclidean_distance

Referencias relacionadas con la parte de investigación.

1. Jain, R., et al. (2018). "Chatbots in Healthcare: A Study on the Benefits of Implementing AI in Medical Practices." *Journal of Healthcare Informatics*.
2. Adamopoulou, E., & Moussiades, L. (2020). "An Overview of Chatbot Technology." *Springer*.
3. Deloitte. (2019). "The Rise of Chatbots in Banking: A Deloitte Report." *Deloitte Insights*.

4. Gartner. (2020). "Chatbots Will Save \$11 Billion Annually by 2024." *Gartner Research*.
5. McKinsey & Company. (2021). "Enhancing Customer Experience with AI: The Role of Chatbots." *McKinsey Insights*.
6. Mikolov, T., et al. (2013). "Efficient Estimation of Word Representations in Vector Space." *arXiv preprint arXiv:1301.3781*.
7. Levy, O., & Goldberg, Y. (2014). "Neural Word Embedding as Implicit Matrix Factorization." *Advances in Neural Information Processing Systems*.
8. Salton, G., & McGill, M. (1986). "Introduction to Modern Information Retrieval." *McGraw-Hill*.
9. Fit For Work. (2023). "Case Studies in Workplace Safety: Real-Life Examples of Successful Safety Initiatives." *Fit For Work*. Retrieved from [Fit For Work](#).
10. Devlin, J., et al. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." *arXiv preprint arXiv:1810.04805*.
11. Radford, A., et al. (2019). "Language Models are Unsupervised Multitask Learners." *OpenAI*.
12. Tesla, Inc. (2023). "Annual Safety Report." *Tesla*. Retrieved from Tesla Annual Report.
13. Office of Statistics Labor (2023). "Injury Rates for Manufacturing Sector." *BLS*. Retrieved from BLS Statistics.
14. Alcoa, Inc. (2023). "Sustainability and Safety Report." *Alcoa*. Retrieved from [Alcoa Safety Report](#).

Referencias relacionadas con el modelo "dccuchile/bert-base-spanish-wwm-uncased"

BERT en español:

1. Hugging Face Model Repository, BETO: Spanish BERT. Disponible en: [Hugging Face](#).
2. GitHub Repository for BETO, dccuchile. Disponible en: [GitHub](#).
3. Información sobre BETO en Spark NLP. Disponible en: [Spark NLP](#).
4. Paper sobre BETO, publicado en ICLR 2020. Disponible en: [arXiv](#).

5. Descripción técnica de BETO en la plataforma de modelos de Hugging Face.

Disponible en: [Hugging Face](#).