Genís Martínez Garrido

Creació, Avaluació i Millora d'un Sistema RAG per a la Gestió d'Aigües a AquaCIS CF

TREBALL DE FI DE GRAU

dirigit per Marc Sánchez

Grau d'Enginyeria Informàtica

UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2024

Agraïments

A la meva família pel suport mostrat en tot moment, sense el qual aquest projecte no hauria estat possible. Al Marc Sánchez per haver-me fet de guía en aquest treball. A totes les persones que han fet possible aquest treball dins de Synectic: Daniel Pastrana, Jose Luis Guasch, David Porta, Segismundo Pareja i altres membres de Synectic. Finalment, vull expressar la meva gratitud a tots aquells que d'una manera o altra han contribuït al meu creixement acadèmic i personal, brindant-me les eines necessàries per aconseguir aquest objectiu.

Resum

Aquest projecte es centra en la creació i avaluació d'un sistema de Generació Augmentada per Recuperació (RAG) aplicat al software AquaCIS CF, utilitzat en la gestió d'aigües dins de l'organització VEOLIA. El RAG ha estat dissenyat per millorar l'eficiència en la recuperació d'informació sobre impagaments i fraus, àrees crítiques en aquest context. A través de la integració de tècniques avançades d'intel·ligència artificial, com els models de llenguatge i la IA generativa, s'ha desenvolupat una eina capaç de generar respostes precises i rellevants. L'avaluació del sistema s'ha dut a terme mitjançant mètriques com la distància cosinus, la distància euclidiana, i les puntuacions ROUGE, garantint la qualitat i precisió de les respostes generades pel sistema en comparació amb respostes humanes revisades.

Resumen

Este proyecto se centra en la creación y evaluación de un sistema de Generación Aumentada por Recuperación (RAG) aplicado al software AquaCIS CF, utilizado en la gestión de aguas dentro de la organización VEOLIA. El RAG ha sido diseñado para mejorar la eficiencia en la recuperación de información sobre impagos y fraudes, áreas críticas en este contexto. A través de la integración de técnicas avanzadas de inteligencia artificial, como los modelos de lenguaje y la IA generativa, se ha desarrollado una herramienta capaz de generar respuestas precisas y relevantes. La evaluación del sistema se ha llevado a cabo mediante métricas como la distancia coseno, la distancia euclidiana y las puntuaciones ROUGE, garantizando la calidad y precisión de las respuestas generadas por el sistema en comparación con respuestas humanas revisadas.

Abstract

This project focuses on the creation and evaluation of a Retrieval-Augmented Generation (RAG) system applied to the AquaCIS CF software, used in water management within the VEOLIA organization. The RAG has been designed to improve efficiency in retrieving information on unpaid invoices and fraud, which are critical areas in this context. Through the integration of advanced artificial intelligence techniques, such as language models and generative AI, a tool capable of generating precise and relevant responses has been developed. The system's evaluation has been carried out using metrics such as cosine distance, Euclidean distance, and ROUGE scores, ensuring the quality and accuracy of the responses generated by the system compared to reviewed human responses.

Índex

Índex de figures

1. Introducció

1.1. Context

En l'era de la informació, l'accés eficient i precís al coneixement és un requisit fonamental per a qualsevol organització. Els sistemes de Recuperació d'Informació (RAG, per les seves sigles en anglès) han emergit com a eines essencials per processar grans volums de dades i oferir respostes precises a preguntes específiques, utilitzant fonts de coneixement predefinides. Aquest treball se centra en la creació, l'avaluació i la millora d'un sistema RAG, amb l'objectiu de garantir una major precisió i rellevància en les respostes proporcionades per a un àmbit específic de la nostra empresa.

Aquest projecte es desenvolupa en el context de l'organització Veolia, on he creat i analitzat la capacitat del sistema RAG per respondre preguntes sobre temes tan crítics com els impagaments i els fraus, utilitzant com a base de coneixement la documentació corporativa disponible. He implementat un enfocament metodològic que combina l'avaluació "human-in-the-loop" amb mètriques automàtiques de similitud i qualitat com la distància cosinus, la distància euclidiana, i les mètriques ROUGE. Aquest enfocament m'ha permès identificar les limitacions del sistema i aplicar millores mitjançant la implementació de prompts específics per millorar la qualitat de les respostes.

Els resultats obtinguts mostren una millora significativa en la precisió de les respostes després de la implementació dels prompts, validada tant per mètriques automàtiques com per l'avaluació humana. Aquesta millora no només augmenta la fiabilitat del sistema, sinó que també destaca la importància de l'ajust fi (finetuning) dels models de llenguatge per a l'optimització de sistemes de recuperació d'informació.

En aquest treball, descriuré detalladament el procés de creació, avaluació i millora del sistema RAG, presentant els resultats obtinguts i discutint les implicacions d'aquests en l'ús pràctic del sistema dins de l'organització.

1.2. Objectius del treball

Aquest treball té com a objectiu general crear, avaluar i optimitzar la funcionalitat d'un sistema de Recuperació d'Informació (RAG) en el context de l'organització Veolia, centrant-nos en la millora de la precisió, rellevància i completitud de les respostes generades. En primer lloc, es busca crear un sistema de Recuperació d'Informació (RAG) capaç de generar respostes eficients a partir de la documentació interna de l'organització. Aquest sistema ha d'estar adequadament integrat amb les fonts de dades rellevants, especialment amb la documentació sobre impagaments i fraus, per tal de garantir que les respostes generades siguin informades i precises.

En segon lloc, es pretén avaluar la precisió del sistema RAG existent abans d'implementar qualsevol millora. Aquesta avaluació inclou la mesura de la qualitat de les respostes generades mitjançant l'ús de mètriques automàtiques, com la distància cosinus

i la distància euclidiana, així com mètodes d'avaluació humana ("human-in-the-loop"). L'objectiu és identificar àrees de millora en les respostes proporcionades pel sistema, considerant aspectes com la correcció factual, la rellevància, la completitud i la claredat.

El tercer objectiu és implementar millores en el sistema RAG mitjançant l'ajust dels prompts. Això implica el desenvolupament i la integració de prompts específics que permetin millorar la qualitat de les respostes generades, assegurant que aquestes siguin més precises i alineades amb les necessitats de l'organització. Posteriorment, s'avaluarà l'impacte d'aquestes millores utilitzant les mateixes mètriques automàtiques i mètodes d'avaluació humana emprats en l'avaluació inicial.

El quart objectiu consisteix a aplicar mètriques de qualitat textual avançades, com les mètriques ROUGE-N i ROUGE-L, per avaluar la qualitat de les respostes en termes de similitud textual respecte a les respostes de referència proporcionades per experts. Els resultats obtinguts abans i després de la implementació de les millores es compararan per quantificar l'efectivitat de les intervencions realitzades.

A continuació, s'ha de documentar el procés i els resultats obtinguts. Això inclou la presentació clara i estructurada de les passes seguides durant l'avaluació i optimització del sistema RAG, detallant la metodologia emprada, els resultats obtinguts i les conclusions extretes. A més, es proporcionaran recomanacions per a futures millores basades en els resultats i les observacions realitzades durant el procés.

1.3. Importància del projecte

Aquest projecte és de vital importància per a l'organització, ja que té la capacitat de transformar i optimitzar significativament la manera en què es gestiona la recuperació d'informació crítica. En primer lloc, la millora de l'eficiència operativa és un dels beneficis més destacats. Implementar un sistema de Recuperació d'Informació (RAG) optimitzat permetrà reduir de manera dràstica el temps necessari per accedir a informació rellevant i precisa. Aquesta rapidesa és crucial en entorns on les decisions s'han de prendre de forma àgil, com en la gestió de fraus i impagaments. Gràcies a aquest sistema, l'organització podrà prendre decisions més informades en un temps més curt, augmentant així la seva eficàcia global.

En segon lloc, la precisió i fiabilitat de les decisions també es veuran millorades. Un sistema RAG que generi respostes fiables i correctes reduirà la possibilitat d'errors, augmentant la confiança en les dades utilitzades per prendre decisions. Aquest aspecte és especialment rellevant en àrees sensibles com la gestió financera, on una decisió equivocada pot tenir conseqüències significatives. A més, la reducció de costos i l'estalvi de recursos són altres avantatges clau d'aquest projecte. Un sistema automatitzat i eficient minimitza la necessitat d'intervenció humana, cosa que es tradueix en una reducció dels

costos operatius i una menor possibilitat d'errors. Això implica un estalvi directe en costos i un augment de la productivitat dels empleats.

La capacitat d'adaptació i resiliència en un entorn canviant és un altre aspecte que subratlla la importància d'aquest projecte. L'entorn empresarial és dinàmic, i un sistema RAG flexible i adaptable permetrà a l'organització ajustar-se ràpidament a les noves necessitats d'informació, mantenint-se competitiva i preparada per als reptes futurs. També cal destacar que aquest projecte pot ser una font d'innovació i avantatge competitiu. Tenir un sistema de recuperació d'informació avançat i optimitzat pot situar l'organització en una posició preferent davant dels competidors, aportant un valor afegit que altres potser no poden oferir.

Finalment, aquest projecte contribuirà a millorar la satisfacció tant dels empleats com dels clients. Per als empleats, un accés més àgil a la informació necessària per al seu treball diari redueix l'estrès i millora la productivitat. Per als clients, la capacitat de l'organització per proporcionar respostes ràpides i precises augmentarà la seva confiança i satisfacció amb els serveis prestats.

2. Descripció de l'empresa i departament

2.1. Descripció de Synectic

Synectic és la companyía tecnològica d'AGBAR. Amb un enfocament centrat en la innovació i l'excel·lència, Synectic es dedica a proporcionar serveis i eines que impulsen l'eficiència operativa, la presa de decisions basada en dades, i la millora continua dels processos dins del grup AGBAR.

Fundada amb la missió de ser un partner estratègic per a les empreses que busquen liderar en el seu sector mitjançant l'adopció de noves tecnologies, Synectic ha desenvolupat una àmplia gamma de serveis que inclouen la consultoria tecnològica, el desenvolupament de solucions d'intel·ligència artificial, la gestió de dades, i la optimització de processos.

L'empresa compta amb un equip multidisciplinari d'experts en tecnologia, que treballen estretament amb els clients per entendre les seves necessitats específiques i desenvolupar solucions personalitzades que aportin valor real i mesurable. A més, Synectic es caracteritza per la seva capacitat d'adaptar-se a les tendències canviants del mercat i de la tecnologia, oferint sempre solucions a l'avantguarda de la innovació.

Amb una sòlida presència en el mercat i un compromís ferm amb la qualitat i la satisfacció dels clients, Synectic continua creixent i ampliant la seva oferta de serveis, posicionant-se com un líder en la indústria de la tecnologia de la informació i la transformació digital.

2.2. Descripció dels departaments dins de Synectic

2.2.1. Dirección Tecnologías de Información

- **Responsable**: Pastrana Perez, Daniel
- **Funcions**: Aquest departament és el nucli central de les operacions tecnològiques de Synectic. Supervisa totes les funcions tecnològiques i s'assegura que les estratègies tecnològiques alineïn amb els objectius generals de l'empresa. Coordina i integra les activitats dels diferents subdepartaments, proporcionant una visió unificada i estratègica.

2.2.2. Arquitectura IT

- **Responsable**: Clavera Gispert, David
- **Funcions**: El departament d'Arquitectura IT s'encarrega de la planificació, disseny i implementació de l'arquitectura tecnològica de l'empresa. Això inclou l'estructura dels sistemes, xarxes i aplicacions. El departament treballa per assegurar que tots els sistemes siguin interoperables, escalables i alineats amb les necessitats empresarials a llarg termini.

2.2.3. Dirección Data, IA y RPA

- **Responsable**: Porta Alonso, David
- **Funcions**: Aquest departament es centra en la gestió i anàlisi de dades, la implementació d'intel·ligència artificial i la robòtica de processos automatitzats. Les seves funcions inclouen la recollida, neteja i anàlisi de grans volums de dades per extreure informació valuosa per a la presa de decisions. També implementa solucions d'IA per automatitzar processos empresarials i millorar l'eficiència operativa.

2.2.3.1. Data Scientists

- **Responsable**: Ibarra Gomez, David
- **Funcions**: Els científics de dades s'encarreguen de desenvolupar models analítics i algoritmes per analitzar dades complexes i proporcionar informació accionable. Utilitzen tècniques d'aprenentatge automàtic i estadístiques avançades per identificar patrons i tendències.

2.2.3.2. Gobierno y servicio Datahub y BI

- **Responsable**: Rodriguez Aranega, Oscar
- **Funcions**: Aquest subdepartament es dedica a la gestió i governança de les dades, així com als serveis de Business Intelligence (BI). S'assegura que les dades es gestionin de manera segura, complint amb les normatives i estàndards pertinents. També desenvolupa i manté solucions de BI per permetre als usuaris empresarials prendre decisions informades basades en dades.

2.2.3.3. Datahub y BI

- **Responsable**: Galindo Lozano, Montserrat
- **Funcions**: Similar al subdepartament anterior, aquest equip se centra específicament en la infraestructura i les aplicacions del Datahub i les solucions de BI. Implementen i mantenen plataformes que permeten la integració i anàlisi de dades de diverses fonts.

2.2.4. Gestión de la Demanda

- **Responsable**: Millet Gallego, Javier
- **Funcions**: Aquest departament s'encarrega de gestionar la demanda de serveis i recursos tecnològics dins de l'empresa. Això inclou la priorització de projectes, la planificació de recursos i la coordinació amb altres departaments per assegurar que les necessitats tecnològiques es compleixin de manera eficient.

2.2.5. Dirección Planificación y Control

- **Responsable**: Goma Clotet, Marta
- **Funcions**: El departament de Planificació i Control s'encarrega de la planificació estratègica i el control de les operacions tecnològiques. Desenvolupa plans a llarg termini per al desenvolupament i implementació de noves tecnologies i controla l'execució de projectes per assegurar que es compleixin els objectius i es mantinguin dins del pressupost.

2.2.6. Dirección Estrategia IT

- **Responsable**: Cejudo Anton, Ainhoa

- **Funcions**: Aquest departament desenvolupa i implementa l'estratègia IT de l'empresa. Això inclou la definició d'objectius tecnològics, la identificació de noves oportunitats tecnològiques i la coordinació amb altres departaments per assegurar que l'estratègia IT s'alineï amb els objectius generals de l'empresa.

2.2.7. Dirección Aplicaciones de Negocio

- **Responsable**: Ejarque Monserrate, Pascual
- **Funcions**: El departament d'Aplicacions de Negocio gestiona el desenvolupament, implementació i manteniment d'aplicacions empresarials. Això inclou ERP, CRM i altres aplicacions que suporten les operacions empresarials diàries. Treballa per assegurar que les aplicacions empresarials siguin eficients, fiables i alineades amb les necessitats de l'empresa.

2.2.8. Dirección IS&T

- **Responsable**: Marti Climent, Miquel
- **Funcions**: Aquest departament s'encarrega de la infraestructura de sistemes i tecnologies de la informació. Això inclou la gestió de xarxes, servidors, emmagatzematge i seguretat de la informació. Assegura que la infraestructura IT sigui robusta, segura i capaç de suportar les operacions empresarials.

3. Marc teòric

3.1. Intel·ligència artificial generativa

3.1.1. Definició i concepte

La intel·ligència artificial generativa és una branca de la intel·ligència artificial que es centra en la creació de contingut nou a partir de patrons i dades preexistents. Aquesta capacitat per generar contingut de manera autònoma diferencia la IA generativa d'altres tècniques de IA que principalment analitzen, classifiquen o fan prediccions basades en dades existents. La IA generativa, per contra, té el potencial d'introduir una dimensió creativa en la manera com interactuem amb la tecnologia, oferint solucions innovadores i personalitzades a problemes complexos.

3.1.1.1. Principis bàsics de la IA generativa

Els principis bàsics de la IA generativa se sustenten en una sèrie de conceptes i tecnologies avançades que permeten a aquests sistemes no només analitzar dades, sinó també crear contingut nou i original. La IA generativa es basa en algoritmes molt sofisticats que tenen la capacitat d'aprendre patrons i estructures complexes a partir de grans volums de dades. Aquests algoritmes poden identificar relacions subtils entre les dades i utilitzar aquest coneixement per generar contingut que és coherent i rellevant respecte a les dades originals. Això implica que la IA generativa no només imita el que ha après, sinó que també és capaç de crear nous continguts que s'ajusten a aquests patrons.

Un dels aspectes més distintius de la IA generativa en comparació amb la IA tradicional és la seva capacitat creativa. Mentre que la IA tradicional sovint es limita a analitzar i processar informació per a la presa de decisions, la IA generativa està dissenyada per produir nous continguts. Això pot incloure la generació de text, imatges, sons, vídeos i altres formes de contingut digital, ampliant significativament les aplicacions potencials d'aquesta tecnologia en àmbits com la creació de contingut multimèdia, el disseny, la publicitat i més.

La qualitat del contingut generat per la IA depèn en gran mesura de la quantitat i la qualitat de les dades utilitzades per entrenar els models. Els models d'IA generativa es beneficien d'entrenaments amb grans volums de dades que cobreixen una àmplia varietat de casos d'ús i situacions. Aquest entrenament extens permet que l'algoritme entengui profundament els matisos, les variacions i les estructures presents en les dades, millorant així la seva capacitat per generar contingut que sigui realista, divers i creatiu.

Un altre pilar fonamental de la IA generativa és l'ús de tècniques de deep learning. Aquests models utilitzen múltiples capes de xarxes neuronals artificials per processar les dades d'entrada, on cada capa aprèn a reconèixer característiques més abstractes i complexes de les dades. Aquest procés de deep learning permet als models d'IA generativa captar les relacions intrínseques i les característiques essencials de les dades, permetent-los generar contingut nou que reflecteix fidelment les complexitats del món real.

3.1.1.2. Exemples de IA generativa

En l'actualitat, la IA generativa ha trobat aplicació en diversos camps, mostrant una versatilitat increïble a l'hora de crear contingut digital. Un dels exemples més coneguts d'IA generativa és el de l'àmbit de text, com pot ser GPT-4 (Generative Pre-trained Transformer 4). Aquest model de llenguatge és capaç de generar text coherent i contextuat a partir d'una entrada específica, com pot ser una frase o una pregunta. Les aplicacions d'aquest tipus d'IA inclouen la redacció automàtica d'articles, la creació de resums, la resposta a preguntes en assistents virtuals, i fins i tot la generació de contingut

per a xarxes socials i blogs. El que fa que aquests models siguin tan potents és la seva capacitat per mantenir la coherència temàtica i estilística, fent que el text generat sembli escrit per un humà.

Les xarxes generatives adversàries (GANs) són un altre exemple prominent d'IA generativa. Aquestes xarxes poden crear imatges realistes a partir de descripcions textuals o altres imatges. Un dels usos més destacats és la generació de cares humanes que, tot i no existir realment, semblen fotografies autèntiques. Les GANs també s'utilitzen en l'art digital, la moda, i el disseny de productes, on poden generar conceptes visuals únics a partir de poques indicacions. Aquesta capacitat de crear imatges detallades i realistes té un potencial enorme en sectors com la publicitat, els mitjans de comunicació i l'entreteniment.

Per altre banda, els algoritmes d'IA generativa també s'han aplicat amb èxit en la creació de sons i música. Aquests sistemes poden analitzar patrons musicals existents per generar noves peces musicals o efectes sonors. Això permet composar música personalitzada per a esdeveniments, pel·lícules, jocs, o simplement per a l'escolta personal. La capacitat de l'IA per entendre les estructures musicals i crear-ne de noves que siguin agradables a l'oïda humana obre la porta a una nova era de creativitat musical assistida per màquina.

Finalment, la IA generativa també està fent avenços en la creació de vídeos i animacions. Algoritmes especialitzats poden generar seqüències de vídeo realistes o animacions a partir de dades existents. Això inclou la capacitat de crear vídeos que semblen autèntics, però que són completament generats per un algoritme, sense la necessitat d'una gravació prèvia. Aquesta tecnologia s'està utilitzant en la producció de pel·lícules, sèries, publicitat i continguts digitals, permetent la creació de materials audiovisuals que abans haurien requerit molts recursos humans i tècnics.

3.1.1.3. Aplicacions pràctiques de la IA generativa

Una de les aplicacions més rellevants de la IA generativa és en el camp del màrqueting digital i la publicitat. Les empreses poden utilitzar IA generativa per crear contingut personalitzat per a les seves campanyes, adaptant els anuncis, descripcions de productes i publicacions en xarxes socials a les preferències específiques dels seus usuaris. Això no només millora la rellevància dels missatges publicitaris, sinó que també augmenta la probabilitat d'interacció i conversió, ja que el contingut creat ressona millor amb les necessitats i interessos de l'audiència objectiu.

En l'àmbit del desenvolupament web, la IA generativa s'està convertint en una eina poderosa per a la generació automàtica de codi i dissenys de pàgines web. Els desenvolupadors poden utilitzar aquesta tecnologia per crear plantilles de llocs web, components d'interfície d'usuari, i fins i tot llocs web complets de manera ràpida i eficient. Això no només redueix el temps i els costos associats al desenvolupament web, sinó que també permet als desenvolupadors centrar-se en tasques més complexes i creatives.

La IA generativa també està transformant la indústria dels jocs i l'entreteniment. Aquesta tecnologia permet la creació de mons virtuals, personatges i narratives que s'adapten en temps real a les accions i decisions dels jugadors. Això genera una experiència de joc molt més immersiva i dinàmica, on cada partida pot ser única. A més, la IA pot generar entorns de joc detallats i històries interactives sense la necessitat d'una programació manual exhaustiva, oferint als desenvolupadors la possibilitat d'explorar nous territoris creatius.

En el camp de l'educació, la IA generativa està canviant la manera en què es crea i es distribueix el contingut educatiu. Aquesta tecnologia pot ser utilitzada per generar materials d'estudi personalitzats, exercicis pràctics, i resums de lectures adaptats a les necessitats individuals dels estudiants. Això permet una experiència d'aprenentatge molt més efectiva i personalitzada, on cada alumne pot avançar al seu propi ritme i segons les seves pròpies necessitats, millorant així els resultats educatius.

3.1.1.4. Futur de la IA generativa

El futur de la IA generativa es presenta com un horitzó ple de possibilitats i reptes. Encara que aquesta tecnologia es troba en una fase de desenvolupament inicial, les seves aplicacions ja estan començant a impactar significativament en una àmplia gamma d'indústries. En els propers anys, s'espera que la IA generativa esdevingui una eina cada vegada més potent i accessible, capaç de transformar la manera com creem i interaccionem amb el contingut digital. Una de les àrees amb més potencial és l'automatització de processos creatius. A mesura que els algoritmes d'IA generativa es perfeccionin, es preveu que puguin assumir rols més complexos en la creació de continguts, no només generant text, imatges o sons, sinó també contribuint a la innovació en àmbits com la ciència, la medicina i l'enginyeria. Això podria incloure des de la generació de nous dissenys de productes fins a la creació d'informes científics basats en dades en temps real. Tanmateix, el futur de la IA generativa no està exempt de reptes. Un dels principals és la qüestió de l'autenticitat i la seguretat del contingut generat. A mesura que aquesta tecnologia es faci més sofisticada, es fa necessari establir mecanismes per assegurar que el contingut generat per IA sigui identificable i no es pugui utilitzar de manera malintencionada, com ara en la creació de notícies falses o contingut enganyós. A més, les implicacions ètiques del desenvolupament i ús de la IA generativa són un altre aspecte crític a considerar. La protecció de la privacitat, la gestió de dades sensibles i l'impacte de la IA en el mercat laboral són qüestions que requeriran una atenció especial per part de legisladors, desenvolupadors i la societat en general.

3.2. Algoritmes i tècniques principals

Els algoritmes i tècniques utilitzats en la intel·ligència artificial generativa són essencials per entendre com aquestes tecnologies són capaces de crear contingut nou i original. Aquestes tècniques, que van des de models de llenguatge fins a xarxes generatives adversàries, permeten que els sistemes d'IA no només analitzin dades, sinó que també les generin de manera autònoma. A continuació, es detallen alguns dels algoritmes i tècniques més importants en aquest camp.

3.2.1. Models de llenguatge (Language Models)

Els models de llenguatge són un component fonamental en la intel·ligència artificial generativa, ja que estan dissenyats per processar i generar text de manera coherent a partir d'unes dades d'entrada específiques. Aquests models han revolucionat la manera en què interactuem amb la tecnologia, permetent que les màquines siguin capaces de produir text que no només és gramaticalment correcte, sinó que també té sentit i rellevància en contextos diversos. A través de l'aprenentatge profund (deep learning) sobre el qual ja hem parlat anteriorment, aquests models poden analitzar i reproduir patrons del llenguatge natural, la qual cosa els permet entendre no només el significat de les paraules, sinó també les relacions complexes entre elles.

Un dels exemples més avançats d'aquests models és el GPT-4 (Generative Pre-trained Transformer 4), desenvolupat per OpenAI. Aquest model és capaç de generar text amb un nivell de sofisticació que sovint és indistingible del text escrit per un ésser humà. GPT-4 no només és capaç de generar respostes a preguntes simples, sinó que també pot redactar articles complets, crear diàlegs realistes, i fins i tot mantenir una conversa fluida sobre temes complexos. El seu funcionament es basa en un entrenament previ amb una enorme quantitat de text disponible públicament, el que li permet tenir un vast coneixement de molts temes diferents. A més, aquest model es pot afinar per a tasques específiques, el que li dóna una gran versatilitat en diferents aplicacions industrials i comercials.

La seva capacitat per comprendre i generar text en funció del context el fa especialment útil en aplicacions que requereixen una alta coherència i cohesió, com ara la creació de continguts personalitzats, la traducció automàtica o el desenvolupament d'assistents virtuals capaços de mantenir converses naturals amb els usuaris. La seva versatilitat i eficàcia han portat a una adopció àmplia en diverses indústries, des del màrqueting i la publicitat fins a l'educació i l'entreteniment. Amb tot, els models de llenguatge com GPT-4 estan transformant la manera en què les empreses i els individus interactuen amb la informació i la tecnologia, obrin noves possibilitats per a la creativitat, la productivitat i la comunicació.

Característiques clau:

- **Pre-entrenament i ajust fi:** Els models de llenguatge com GPT-4 es pre-entrenen amb grans quantitats de text disponible públicament, com ara llibres, articles i llocs web. Després, poden ser ajustats finament amb dades específiques per a tasques concretes.
- **Comprensió del context:** Aquests models poden mantenir la coherència del text generat gràcies a la seva capacitat per comprendre el context de les frases i les paraules.
- **Versatilitat:** Els models de llenguatge poden ser utilitzats per a una àmplia gamma d'aplicacions, com ara la generació automàtica de respostes, la redacció d'articles, la traducció automàtica, i molt més.

Aplicacions:

- **Assistents virtuals:** Utilitzats en chatbots i assistents de veu per generar respostes naturals a les preguntes dels usuaris.
- **Redacció automàtica:** Eines que poden escriure articles, resums, o contingut per a xarxes socials basant-se en inputs específics.
- **Generació de diàlegs:** Creació de diàlegs realistes per a personatges de jocs de vídeo o simulacions de conversa per a la formació.

3.2.2. Xarxes generatives adversàries (GANs)

Les xarxes generatives adversàries (GANs) representen una de les innovacions més significatives en el camp de la intel·ligència artificial generativa. Introduïdes per Ian Goodfellow i els seus col·laboradors el 2014, les GANs han revolucionat la manera en què es poden generar dades sintètiques amb un alt grau de realisme. Aquestes xarxes es basen en un enfocament competitiu, on dos models, anomenats generador i discriminador, treballen junts però amb objectius oposats per tal de millorar constantment les seves capacitats.

El generador té la tasca de crear noves mostres de dades que imitin les dades reals de la manera més fidel possible. Aquest model, entrenat per crear imatges, sons, textos o altres formes de dades, busca enganyar el discriminador fent que les seves creacions siguin indistingibles de les mostres reals. D'altra banda, el discriminador està entrenat per identificar si una mostra prové del conjunt de dades real o si ha estat creada pel generador. Aquesta dinàmica de competició contínua entre el generador i el discriminador permet que ambdós models millorin progressivament les seves capacitats: el generador crea mostres cada vegada més realistes, mentre que el discriminador es fa més precís en la seva tasca de diferenciació.

Aquest procés d'entrenament, on ambdós models s'ajusten de manera iterativa, continua fins que el generador arriba a un punt en què és capaç de crear dades tan realistes que el discriminador no pot distingir-les de les mostres reals. Aquesta capacitat de generar dades gairebé indistingibles de les reals ha obert un ampli ventall d'aplicacions en diverses indústries. Per exemple, les GANs s'utilitzen àmpliament en la creació d'imatges realistes de cares humanes, objectes o paisatges que no existeixen en la realitat. També són utilitzades en la generació de vídeos sintètics, que poden semblar autèntics i són molt útils en àmbits com l'animació i els efectes visuals en pel·lícules i videojocs.

A més de la generació de contingut, les GANs també tenen aplicacions en la transferència de l'estil, on poden aplicar l'estil visual d'una imatge a una altra, transformant, per exemple, una fotografia en una pintura amb l'estil d'un artista famós. Aquesta capacitat de transformar i crear noves dades visuals amb alta qualitat i fidelitat ha fet que les GANs siguin una eina poderosa i versàtil en el camp de la creativitat digital, el disseny i l'entreteniment.

Característiques clau:

- **Generador:** Aquest model crea noves mostres de dades intentant que siguin el més similars possibles a les dades reals.
- **Discriminador:** Aquest model intenta distingir entre les dades reals i les generades pel generador. El seu objectiu és identificar les mostres falses creades pel generador.
- **Entrenament:** Durant l'entrenament, el generador i el discriminador competeixen entre si. El generador intenta millorar les seves creacions per enganyar el discriminador, mentre que el discriminador millora les seves habilitats per detectar falsificacions. Aquest procés continua fins que el generador crea mostres tan realistes que el discriminador no pot distingir-les de les dades reals.

Aplicacions:

- **Generació d'imatges:** Creació d'imatges realistes de cares humanes, objectes, paisatges, etc., que no existeixen en la realitat.
- **Vídeos sintètics:** Generació de seqüències de vídeo que semblen autèntiques, utilitzades en animació i efectes visuals.
- **Transferència de l'estil:** Aplicació de l'estil visual d'una imatge a una altra, per exemple, transformant una fotografia en una pintura d'estil impressionista.

3.2.3. Autoencoders variacionals (VAEs)

Els autoencoders variacionals (VAEs) són una tècnica sofisticada dins del camp de la intel·ligència artificial generativa que combina conceptes de compressió de dades i generació probabilística per crear noves mostres de dades. A diferència dels autoencoders convencionals, que simplement comprimeixen i reconstrueixen les dades, els VAEs introdueixen un enfocament probabilístic que els permet generar variacions noves i realistes de les dades d'entrada. Els VAEs funcionen a través d'un procés de codificació i descodificació que implica la creació d'una representació latent comprimida de les dades d'entrada. Aquesta representació latent no és fixa, sinó que es modela com una distribució probabilística, cosa que permet que el model generi noves mostres extraient valors d'aquesta distribució. Aquesta característica és especialment valuosa en aplicacions creatives, ja que permet la generació de variacions noves que mantenen una semblança amb les dades originals, però amb la flexibilitat d'explorar nous espais de possibilitats.

Per exemple, en la generació d'imatges, un VAE podria entrenar-se amb un conjunt de dades d'imatges de cares humanes. Durant el procés de codificació, el model comprimiria les imatges en un espai latent, on cada punt de l'espai representa una variació potencial d'una cara humana. Després, durant el procés de descodificació, el model podria generar noves imatges a partir de qualsevol punt en aquest espai latent, produint cares que no existeixen en la realitat, però que semblen autèntiques. Aquesta capacitat de generar noves mostres a partir de la distribució latent fa que els VAEs siguin extremadament útils en la síntesi d'imatges, sons, i altres formes de dades. A més de la generació de noves mostres, els VAEs són útils per a la interpolació, una tècnica que permet crear transicions suaus entre diferents mostres dins de l'espai latent. Per exemple, si es tenen dues imatges diferents, un VAE pot interpolar entre les dues per crear una seqüència contínua de transició que mostri com una imatge es transforma gradualment en l'altra. Aquesta característica és especialment valuosa en aplicacions com l'animació, on es poden crear transicions fluides entre diferents fotogrames o estats.

Característiques clau:

- **Codificador (encoder):** Comprimeix les dades d'entrada en una representació latent més petita.
- **Descodificador (decoder):** Reconstrueix les dades d'entrada a partir de la representació latent.
- **Espai latent:** En comptes de comprimir les dades en una única representació específica, els VAEs generen una distribució probabilística en l'espai latent. Això permet la generació de noves mostres a partir d'aquesta distribució.

Aplicacions:

- **Generació de noves mostres:** Creació de noves dades similars a les originals, útil en la síntesi d'imatges, sons, i altres formes de dades.

- **Interpolació:** Creació de transicions suaus entre diferents mostres dins de l'espai latent, per exemple, interpolar entre dues imatges diferents per crear una seqüència de transició.

3.2.4. Xarxes neuronals recurrents (RNN) i LSTM

Les xarxes neuronals recurrents (RNN) són un tipus d'arquitectura de xarxa neuronal especialment dissenyada per treballar amb dades seqüencials, on l'ordre de les dades és important. A diferència de les xarxes neuronals tradicionals, que tracten les dades com si fossin independents, les RNN tenen la capacitat de mantenir una "memòria" del que han processat anteriorment, fet que les fa idònies per a tasques on el context és crucial, com ara el processament de text, la parla i altres formes de seqüències temporals. Tot i aquestes ventatges, les RNN simples presenten algunes limitacions, especialment en la seva capacitat per recordar informació a llarg termini. Aquest problema es coneix com a "problema de la desaparició del gradient", que fa que les RNN tinguin dificultats per aprendre dependències a llarg termini. Per superar aquestes limitacions, es van desenvolupar les unitats de memòria a llarg termini (LSTM), una variant de les RNN que incorpora mecanismes especials, com ara "portes" d'entrada, sortida i oblit, per controlar el flux de la informació. Aquestes portes permeten a la xarxa decidir quina informació mantenir, quina actualitzar i quina oblidar, millorant així la seva capacitat per recordar informació rellevant durant períodes de temps més llargs.

Les LSTM han revolucionat el camp de les RNN en permetre que aquestes xarxes siguin molt més eficients en la gestió de seqüències llargues, fent-les especialment adequades per a una àmplia gamma d'aplicacions pràctiques. Per exemple, en la generació de text, les LSTM poden crear paràgrafs sencers, poemes i altres formes de text continu, mantenint la coherència i fluïdesa al llarg de llargues seqüències. Això les fa molt útils per a la redacció automàtica i la generació de contingut creatiu. A més, les RNN amb LSTM són també excel·lents per a la predicció de seqüències, com ara la predicció de les següents notes en una peça musical o les següents paraules en una frase. Aquesta capacitat és essencial per a aplicacions en el camp de la música generativa o la traducció automàtica, on el context a llarg termini és fonamental per a la qualitat de la sortida generada.

Un altre àmbit on les RNN i les LSTM són àmpliament utilitzades és en l'anàlisi de sèries temporals. Per exemple, en aplicacions financeres, aquestes xarxes poden predir moviments futurs del mercat basant-se en dades històriques, proporcionant una eina poderosa per als analistes financers. De la mateixa manera, en el camp de la meteorologia, les RNN i les LSTM poden ser utilitzades per predir canvis climàtics, analitzant patrons temporals en les dades de temperatura, humitat, pressió, entre altres.

Característiques clau:

- **Memòria a llarg termini:** Capacitat per recordar informació durant períodes de temps llargs, cosa que és crucial per a la generació de text coherent en llargues seqüències.
- **Control de flux de dades:** Utilitzen portes especials per controlar el flux de dades entrant i sortint de les cel·les de memòria, permetent una gestió més eficient de la informació.

Aplicacions:

- **Generació de text:** Creació de paràgrafs, poemes, i altres formes de text continu.
- **Predicció de seqüències:** Predicció de les següents notes en una peça musical o les següents paraules en una frase.
- **Anàlisi de sèries temporals:** Ús en aplicacions financeres per predir moviments de mercat o en meteorologia per predir canvis climàtics.

3.3. Ús d'Embeddings en la Intel·ligència Artificial

3.3.1. Què són els embeddings?

Els embeddings són representacions numèriques de text utilitzades en el processament del llenguatge natural (NLP). Es tracta de vectors en un espai multidimensional que capturen el significat semàntic del text. Paraules o frases similars es troben més a prop en aquest espai vectorial, mentre que les dissimilars es troben més lluny. En el context dels NLP, els embeddings transformen paraules, frases o documents en vectors de nombres reals en un espai multidimensional. Aquesta transformació és crucial perquè les màquines puguin comprendre i processar el text de manera efectiva. En lloc de treballar amb paraules individuals que poden tenir múltiples significats depenent del context, els embeddings permeten capturar relacions semàntiques complexes. Per exemple, considerem les frases "Quin és el camí cap al supermercat?" i "Podria obtenir indicacions per arribar a la botiga?". Tot i que utilitzen paraules diferents, ambdues frases tenen una intenció semàntica similar: demanar direccions per arribar a un lloc. Els embeddings mapegen aquestes frases a vectors en l'espai vectorial que estan molt a prop l'un de l'altre, reflectint així la seva similitud semàntica.

Els embeddings es creen mitjançant l'entrenament de models de llenguatge que analitzen grans quantitats de text per identificar patrons i relacions entre les paraules. Un dels models més utilitzats per generar embeddings és el model Word2Vec, desenvolupat per Google. Aquest model entrena una xarxa neuronal per predir una paraula basant-se en el seu context (mètode de skip-gram) o per predir el context d'una paraula (mètode CBOW, Continuous Bag of Words). Un altre model popular és el GloVe (Global Vectors

for Word Representation), desenvolupat per Stanford. GloVe utilitza informació global de coocurrència de paraules en un corpus de text per aprendre representacions vectorials.

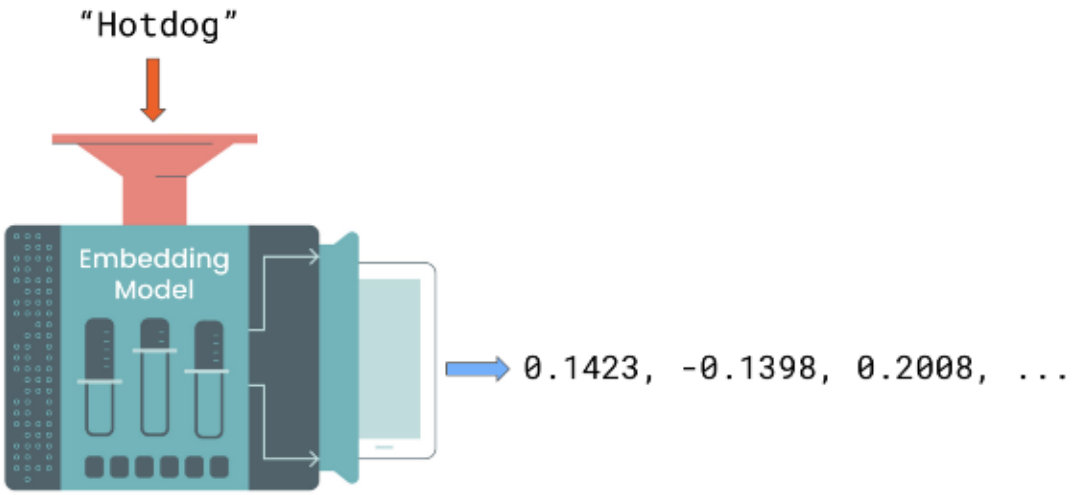

Ilustración 1. Paraules transformades a embeddings

Propietats dels embeddings:

- **Dimensionalitat**: Els vectors d'embeddings poden tenir diferents dimensions, que es determinen durant el procés d'entrenament del model. Per exemple, un model pot generar vectors de 50, 100 o fins a 300 dimensions. Els vectors amb més dimensions poden capturar més informació, però també poden ser més complexos i costosos de calcular. Hi aprofundirem més endavant.
- **Similitud**: La similitud entre dos embeddings es mesura generalment utilitzant la distància coseno o la similitud coseno. Aquesta mesura indica l'angle entre dos vectors i proporciona una manera de quantificar la seva similitud semàntica. Vectors que representen paraules o frases similars tindran una distància coseno petita (proper a zero).
- **Arrelament semàntic**: Els embeddings permeten que les màquines capturin relacions semàntiques complexes, com ara sinonímia (paraules amb significats similars) i antonímia (paraules amb significats oposats), així com relacions de tipus part-whole i altres associacions contexturals.

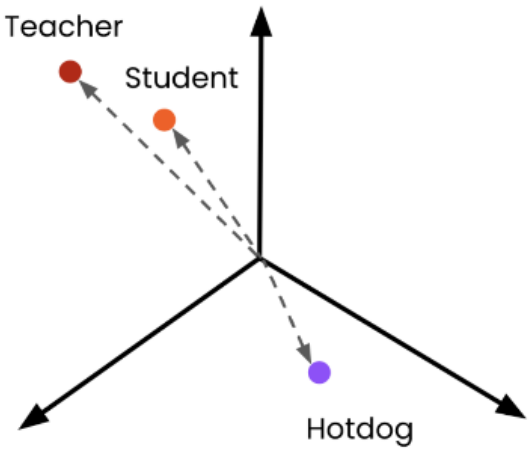

Ilustración 2. Representació vectorial

Els embeddings es poden aplicar a frases senceres, com ara titulars de notícies (headlines). En aquest context, un model de llenguatge pot transformar un titular complet en un vector d'embeddings que captura la seva essència semàntica. Això és especialment útil per a aplicacions com la classificació de notícies, la recomanació de continguts i la cerca semàntica. Per exemple, un titular com "Apple llança un nou producte innovador" podria estar representat per un vector que destaca termes com "Apple", "producte" i "innovador", reflectint la seva importància relativa i la connexió entre ells.

La llargada del vector d'embeddings, també coneguda com la dimensionalitat, és un factor crucial en la seva eficàcia. Models com Word2Vec i GloVe permeten definir la llargada del vector durant l'entrenament. Per exemple, Word2Vec sovint utilitza vectors de 100 a 300 dimensions, mentre que els models més recents, com els de la sèrie OpenAI's GPT, poden utilitzar vectors de fins a 1536 dimensions, com en el cas del model "text-embedding-ada-002". Una major dimensionalitat permet capturar més matisos i relacions complexes, però també augmenta la complexitat computacional.

En algunes aplicacions, és necessari generar embeddings per múltiples entrades alhora. Això es pot fer combinant embeddings de diferents paraules, frases o documents en un sol vector que representa la informació combinada. Un exemple d'això és l'ús de xarxes neuronals recurrents (RNN) o transformadors (Transformers), que poden processar seqüències de text i generar un embedding per tota la seqüència. Això és útil per a tasques com la traducció automàtica, on es necessita comprendre i representar la relació entre múltiples paraules en una frase. Un altre enfocament és l'ús d'agregació d'embeddings, on es generen embeddings individuals per cada entrada i després es combinen mitjançant operacions com la mitjana, la suma o una xarxa neuronal addicional per obtenir un embedding representatiu de les múltiples entrades. Aquest mètode s'utilitza sovint en aplicacions com la cerca de documents, on un conjunt de paraules clau o una frase de consulta es converteix en un sol vector d'embeddings per trobar documents rellevants.

3.3.2. Aplicacions dels embeddings:

Els embeddings són una eina poderosa que permet la representació de textos com a vectors en un espai multidimensional, capturant el significat semàntic del text. Això fa que siguin especialment útils en diverses aplicacions del processament del llenguatge natural (NLP), com ara la cerca semàntica, els sistemes de recomanació i la classificació de textos.

3.3.2.1. Cerca semàntica

Els motors de cerca tradicionals es basen en la coincidència de paraules clau per trobar informació. Aquesta metodologia pot perdre la intenció real d'una consulta i no capturar les variacions de paraules. En canvi, els motors de cerca semàntica utilitzen embeddings per comprendre el context i la intenció darrere del text, permetent obtenir resultats més precisos i rellevants. Per exemple, considerem les següents consultes: "Quin és el camí cap al supermercat?" i "Podria obtenir indicacions per arribar a la botiga?". Encara que utilitzen paraules diferents, ambdues frases tenen una intenció semàntica similar: demanar direccions. Els embeddings mapegen aquestes frases a vectors en l'espai vectorial que estan molt a prop l'un de l'altre, reflectint la seva similitud semàntica.

Un exemple real d'aquesta aplicació és Google Search. Google utilitza tècniques avançades d'embeddings per millorar la precisió dels resultats de cerca. En comptes de simplement buscar coincidències de paraules clau, Google comprèn la intenció de les consultes dels usuaris, la qual cosa permet obtenir resultats més rellevants i útils. Aquesta tecnologia permet que els usuaris trobin la informació que necessiten de manera més ràpida i eficient, millorant significativament l'experiència de cerca.

Exemple de codi per implementar una cerca semàntica amb embeddings:

```
from openai import OpenAI
from scipy.spatial import distance
import os

client = OpenAI(api_key=os.environ["OPENAI"])


def create_embeddings(texts, model="text-embedding-ada-002"):
    response = client.embeddings.create(
        model=model,
        input=texts
    )
    response_dict = response.model_dump()
    return [data['embedding'] for data in response_dict['data']]


def semantic_search(query, documents):
```

```
    embeddings = create_embeddings([query] + documents)
    query_embedding = embeddings[0]
    doc_embeddings = embeddings[1:]
    similarities = [1 - distance.cosine(query_embedding,
doc_embedding) for doc_embedding in doc_embeddings]
    sorted_docs = sorted(zip(documents, similarities), key=lambda x:
x[1], reverse=True)
    return sorted_docs


documents = ["El supermercat és a la cantonada del carrer principal.",
             "Podeu trobar la botiga girant a la dreta al segon
semàfor.",
             "La biblioteca està situada al costat de l'ajuntament."]

query = "Com arribo al supermercat?"

results = semantic_search(query, documents)
for doc, similarity in results:
    print(f"Document: {doc} - Similitud: {similarity}")
```

En aquest codi, primer es configuren les biblioteques necessàries i l'API d'OpenAI amb la clau d'API. La funció create_embeddings pren una llista de textos i un model (per defecte, "text-embedding-ada-002") com a entrada, fa una crida a l'API d'OpenAI per crear embeddings per als textos proporcionats i retorna una llista d'embeddings. La funció semantic_search pren una consulta (query) i una llista de documents, genera embeddings per a la consulta i els documents, calcula la similitud coseno entre l'embedding de la consulta i els embeddings dels documents, i retorna els documents ordenats per similitud. Finalment, es defineixen alguns documents d'exemple i una consulta, s'executa la cerca semàntica i es mostren els documents ordenats per similitud amb la consulta.

3.3.2.2. Sistemes de recomanació

En sistemes de recomanació, els embeddings es poden utilitzar per suggerir elements similars basats en la similitud dels vectors. Per exemple, en recomanacions de llocs de treball, es poden recomanar posicions basades en les descripcions prèviament visualitzades, mitigant les variacions en els títols dels llocs de treball. Un exemple real d'aquesta aplicació és Netflix. Netflix utilitza embeddings per recomanar pel·lícules i sèries als seus usuaris. Aquests embeddings es generen a partir de les preferències de visualització dels usuaris i les característiques dels continguts, permetent a Netflix suggerir títols que probablement agradaran a cada usuari. Aquesta tecnologia millora l'experiència de l'usuari en oferir recomanacions personalitzades, augmentant així la satisfacció i el temps de visualització a la plataforma.

Exemple de codi per a la recomanació de llocs de treball:

```
from scipy.spatial import distance


def recommend_jobs(job_descriptions, new_job_description):
    embeddings = create_embeddings(job_descriptions +
[new_job_description])
    new_job_embedding = embeddings[-1]
    similarities = [1 - distance.cosine(new_job_embedding, embedding)
for embedding in embeddings[:-1]]
    similar_jobs = sorted(zip(job_descriptions, similarities),
key=lambda x: x[1], reverse=True)
    return similar_jobs[:5]


job_descriptions = ["Enginyer de programari amb experiència en Python
i Machine Learning.",
                    "Analista de dades amb coneixements avançats en
SQL i estadística.",
                    "Desenvolupador web especialitzat en JavaScript i
React."]

new_job_description = "Enginyer de dades amb habilitats en Python i
anàlisi de dades."

recommended_jobs = recommend_jobs(job_descriptions,
new_job_description)
for job, similarity in recommended_jobs:
    print(f"Lloc de treball: {job} - Similitud: {similarity}")
```

En aquest exemple de codi, primer es carrega la biblioteca necessària per calcular la distància coseno. La funció recommend_jobs pren una llista de descripcions de llocs de treball i una nova descripció de lloc de treball, genera embeddings per a totes les descripcions, calcula la similitud coseno entre la nova descripció i les altres, i retorna les descripcions més similars. Es defineixen algunes descripcions de llocs de treball d'exemple i una nova descripció, s'executa la recomanació i es mostren les descripcions de llocs de treball més similars a la nova descripció.

3.3.2.3. Classificació de textos

Els embeddings també es poden utilitzar per classificar textos en categories basades en la similitud amb descripcions de categories predefinides. Per exemple, es poden classificar titulars de notícies en diferents categories temàtiques com ara "Negocis", "Ciència", "Tecnologia" i "Esports". Un exemple real d'aquesta aplicació és el sistema de detecció de correu brossa de Gmail. Gmail utilitza embeddings per ajudar a classificar els correus electrònics com a spam o no spam. Aquests embeddings capten les característiques semàntiques dels correus i permeten a Gmail identificar patrons i contextos que són típics dels correus de spam. Aquesta tecnologia ajuda a mantenir les bústies dels usuaris lliures de correu brossa, millorant així la seguretat i la satisfacció dels usuaris.

Un exemple de codi per a la classificació de textos podria ser el que he creat a continuació per exemplificar-ho:

```
def classify_headlines(headlines, category_descriptions):
    category_embeddings = create_embeddings(category_descriptions)
    headline_embeddings = create_embeddings(headlines)
    classifications = []
    for headline_embedding in headline_embeddings:
        similarities = [1 - distance.cosine(headline_embedding,
category_embedding) for category_embedding in
                        category_embeddings]
        category_index = np.argmax(similarities)
        classifications.append(category_descriptions[category_index])
    return classifications


categories = ["Negocis", "Ciència", "Tecnologia", "Esports"]

headlines = ["Apple llança un nou producte innovador", "Descobriment
científic revoluciona la medicina"]
classifications = classify_headlines(headlines, categories)
for headline, category in zip(headlines, classifications):
    print(f"Títol: {headline} - Categoria: {category}")
```

En aquest exemple de codi, la funció classify_headlines pren una llista de titulars i una llista de descripcions de categories, genera embeddings per a les categories i els titulars, calcula la similitud coseno entre cada titular i les categories, i classifica els titulars en la categoria més similar. Es defineixen algunes categories temàtiques d'exemple i alguns titulars de notícies d'exemple, s'executa la classificació i es mostren els titulars de notícies amb les seves categories corresponents. En el següent apartat podrem observar aquest estudi de forma gràfica d'una manera clara.

3.3.3. Visualització d'embeddings

Visualitzar embeddings és una tècnica útil per entendre com es distribueixen els vectors en l'espai multidimensional i per obtenir informació sobre les relacions semàntiques entre paraules, frases o documents. Aquesta tècnica permet veure com els embeddings capten les similituds i diferències semàntiques, i com les dades es clusteritzen de manera natural. Com que els embeddings solen tenir una alta dimensionalitat (per exemple, 300 o 1536 dimensions), no es poden visualitzar directament. Per aquest motiu, s'utilitzen tècniques de reducció de dimensionalitat per projectar els vectors en un espai de dues o tres dimensions, que es poden visualitzar fàcilment. Algunes de les tècniques més utilitzades són:

- **PCA (Principal Component Analysis):** Aquesta tècnica transforma les dades a un nou sistema de coordenades, on les noves variables (components principals)

són combinacions lineals de les variables originals. PCA redueix la dimensionalitat mantenint la màxima variància possible en les dades.

- **t-SNE (t-Distributed Stochastic Neighbor Embedding):** És una tècnica de reducció de dimensionalitat que es centra en mantenir la relació de veïnatge entre els punts en l'espai original. És especialment útil per visualitzar dades d'alta dimensionalitat en dues o tres dimensions, ja que preserva les estructures locals de les dades.

3.3.3.1. Implementació de t-SNE

A continuació he creat un exemple de codi per implementar t-SNE en Python utilitzant la biblioteca sklearn per reduir la dimensionalitat dels embeddings de text. Aquest codi redueix la dimensionalitat dels embeddings a dues dimensions per a una visualització més fàcil.

```
from sklearn.manifold import TSNE
import numpy as np

embeddings = [article['embedding'] for article in articles]

tsne = TSNE(n_components=2, perplexity=5)
embeddings_2d = tsne.fit_transform(np.array(embeddings))
```

Figura 6. Exemple de codi per a reduïr la dimensionalitat dels embeddings

Per implementar t-SNE, primer importo les biblioteques necessàries. sklearn.manifold conté la classe TSNE que utilitzo per aplicar l'algoritme t-SNE, mentre que numpy em permet manipular arrays numèrics. A continuació, preparo els embeddings a partir de la meva llista d'articles. Cada article és un diccionari que conté una clau 'embedding' amb el vector d'embeddings corresponent. Creo una llista amb aquests vectors d'embeddings. Després, creo una instància de la classe TSNE especificant dos paràmetres: n_components i perplexity. n_components defineix el nombre de dimensions resultants després de la reducció de dimensionalitat. En aquest cas, redueixo la dimensionalitat dels embeddings a dues dimensions per facilitar la visualització. perplexity és un paràmetre utilitzat per l'algoritme que afecta l'equilibri entre les estructures locals i globals en les dades. Aquest valor ha de ser menor que el nombre de punts de dades. Un cop creada la instància de t-SNE, aplico l'algoritme als embeddings generats transformant-los a un espai de dues dimensions. Aquest procés redueix la dimensionalitat dels meus embeddings originals, permetent-me visualitzar-los en un pla bidimensional. Ara puc veure com els diferents articles es distribueixen en aquest espai, i com els embeddings capten les relacions semàntiques entre ells.

És important tenir en compte alguns aspectes a l'hora d'implementar t-SNE. Primer, la variable n_components defineix el nombre de dimensions resultants després de la reducció de dimensionalitat. En aquest exemple, l'he establert a 2 per facilitar la visualització en un pla bidimensional. Escollir un valor més alt podria ser útil per a altres tipus d'anàlisis, però podria ser més difícil de visualitzar. Segon, la variable perplexity és crucial per a l'algoritme t-SNE. La perplexitat es refereix al nombre d'efectius veïns que cada punt considera. Ha de ser inferior al nombre de punts de dades. Un valor típic sol estar entre 5 i 50, i pot requerir algun ajustament per obtenir els millors resultats segons el conjunt de dades. Finalment, reduir la dimensionalitat dels embeddings de 300 o 1536 dimensions a només dues dimensions inevitablement resulta en una pèrdua d'informació. t-SNE intenta minimitzar aquesta pèrdua mantenint les relacions de veïnatge tant com sigui possible, però algunes estructures globals poden no ser perfectament preservades.

3.3.3.2. Visualització

Per poder tenir una representació visual dels embeddings, he fet servir la biblioteca matplotlib. A continuació, mostro un exemple de codi per crear un gràfic de dispersió (scatter plot) dels embeddings reduïts a dues dimensions. Aquest codi també afegeix etiquetes a cada punt del gràfic per indicar la categoria temàtica corresponent a cada article. He creat aquest petit exemple per poder mostrar de forma senzilla com podriem visualitzar els embeddings.

```
import matplotlib.pyplot as plt

plt.scatter(embeddings_2d[:, 0], embeddings_2d[:, 1])

topics = [article['topic'] for article in articles]
for i, topic in enumerate(topics):
    plt.annotate(topic, (embeddings_2d[i, 0], embeddings_2d[i, 1]))

plt.show()
```

Exemple de les dades:

```
def create_article_text(article):
return f"""Headline: {article['headline']}
Topic: {article['topic']}
Keywords: {' , ' .join(article['keywords'])}"""


article_texts = [create_article_text(article) for article in articles]
current_article_text = create article text(current_article)
print(current_article_text)
```

```
articles = [
    {"headline": "Economic Growth Continues Amid Global Uncertainty",
     "topic": "Business",
     "keywords": ["economy", "business", "finance"]},
    ...
    {"headline": "1.5 Billion Tune-in to the World Cup Final",
     "topic": "Sport",
     "keywords": ["soccer", "world cup", "tv"]}
]

current_article = {"headline": "How NVIDIA GPUs Could Decide Who Wins the AI Race",
                   "topic": "Tech",
                   "keywords": ["ai", "business", "computers"]}
```

Per visualitzar els embeddings, faig servir la biblioteca matplotlib. Primer, importo la biblioteca matplotlib, que em permet crear gràfics en Python. Utilitzo el mòdul pyplot de matplotlib per generar els gràfics. Creo un gràfic de dispersió utilitzant els embeddings reduïts a dues dimensions. La funció plt.scatter pren les coordenades x i y dels punts (les dues dimensions resultants de t-SNE) i crea el gràfic de dispersió. Després, preparo les etiquetes que s'afegiran a cada punt del gràfic. Creo una llista de les categories temàtiques (topics) corresponents a cada article. Utilitzo un bucle for per afegir etiquetes a cada punt del gràfic. La funció plt.annotate afegeix una etiqueta a cada punt, indicant la categoria temàtica de l'article. En aquest bucle, la funció enumerate em permet accedir a l'índex i al valor de cada element en la llista de temàtiques, de manera que puc posicionar correctament cada etiqueta en el gràfic. Finalment, utilitzo la funció plt.show per mostrar el gràfic. Això obre una finestra amb el gràfic de dispersió, permetent-nos veure com es distribueixen els embeddings en l'espai bidimensional. El resultat és el següent:

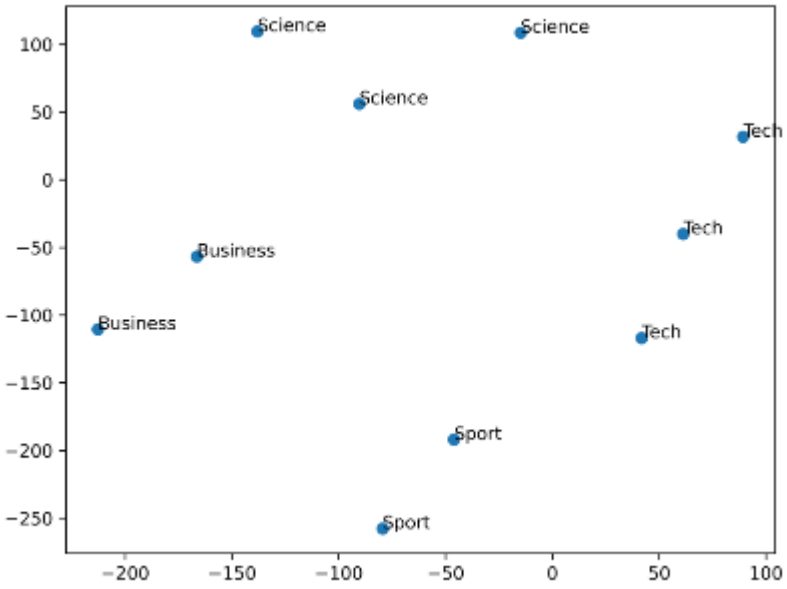

Ilustración 3. Distribució dels embeddings

Com podem observar en la gràfica de dispersió resultant, els embeddings de diversos articles es distribueixen en un espai bidimensional després d'aplicar la tècnica t-SNE. Cada punt en el gràfic representa un article, i les etiquetes corresponents indiquen la categoria temàtica de cada article: Business, Science, Tech, i Sport.

Els resultats mostren que els articles de temàtiques similars tendeixen a clusteritzar-se junts. Per exemple, els articles etiquetats com "Business" es troben propers entre ells, indicant que els embeddings han capturat bé les similituds semàntiques entre aquests articles. De manera similar, els articles de la categoria "Science" es concentren en una àrea específica del gràfic, mostrant una agrupació clara.

Els articles etiquetats com "Tech" també es troben agrupats, tot i que hi ha una certa dispersió entre ells, la qual cosa podria suggerir una variabilitat més gran en els continguts tecnològics representats. Finalment, els articles de la categoria "Sport" es troben junts en la part inferior del gràfic, indicant una bona captació de la seva similitud semàntica.

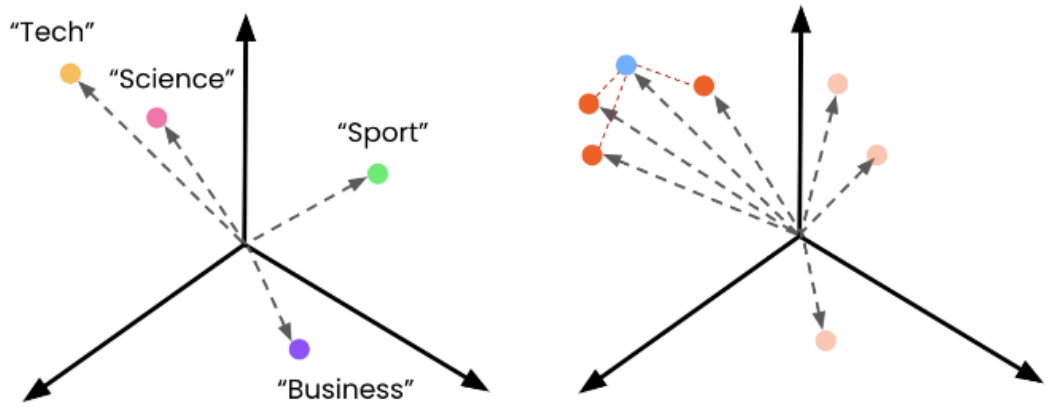

Ilustración 4. Representació vectorial

3.4. Bases de Dades Vectorials

3.4.1. Introducció a les Bases de Dades Vectorials

Les bases de dades vectorials són sistemes de gestió de dades específicament dissenyats per emmagatzemar i gestionar vectors d'embeddings de manera eficient. Aquests vectors són essencials per a moltes aplicacions de processament del llenguatge natural (NLP), com ara la cerca semàntica, els sistemes de recomanació, i la classificació de textos com bé hem explicat en l'apartat anterior. Ara ens centrarem en com s'emmagatzemen aquests vectors.

Diferències entre Bases de Dades NoSQL i SQL:

Bases de Dades NoSQL:

- **Estructura Flexible:** Les bases de dades NoSQL ofereixen una estructura de dades més flexible en comparació amb les bases de dades SQL tradicionals. Això permet emmagatzemar dades no estructurades o semi-estructurades, com ara documents JSON, de manera més eficient.
- **Escalabilitat Horitzontal:** Les bases de dades NoSQL estan dissenyades per escalar horitzontalment, el que significa que es poden afegir més nodes al sistema per gestionar un volum de dades creixent i demandes de trànsit més altes.
- **Velocitat de Consulta:** Amb una estructura de dades flexible, les bases de dades NoSQL poden proporcionar temps de resposta més ràpids per a certs tipus de consultes, especialment aquelles que no requereixen la integritat i consistència estrictes que ofereixen les bases de dades SQL.

Bases de Dades SQL:

- **Estructura Estandarditzada:** Les bases de dades SQL utilitzen una estructura de dades rígida basada en taules, files i columnes. Aquesta estructura és ideal per a dades altament estructurades que requereixen una integritat i consistència estrictes.
- **Integritat de Dades:** Les bases de dades SQL ofereixen funcionalitats avançades d'integritat de dades, com ara restriccions de claus primàries i estrangeres, i transaccions ACID (Atomicitat, Consistència, Aïllament, Durabilitat), que garanteixen que les operacions de base de dades es completin correctament.
- **Consulta Estandarditzada:** Utilitzant el llenguatge SQL (Structured Query Language), les bases de dades SQL proporcionen un mitjà poderós i estandarditzat per consultar i manipular dades.

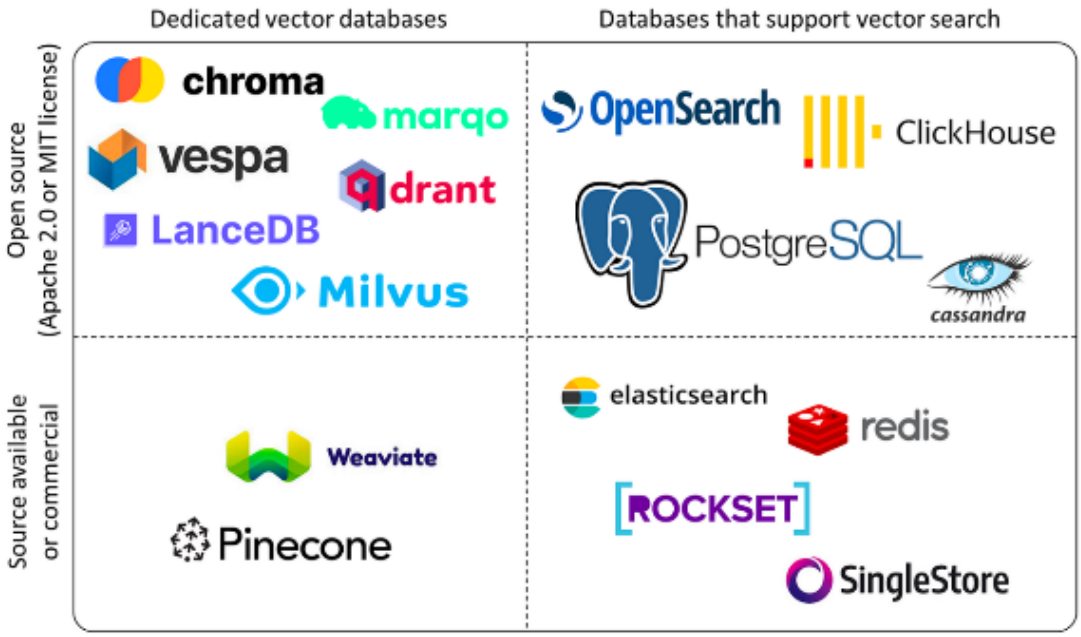

Ilustración 5. Esquema de la classificació BBDD

Components Necessaris per Emmagatzemar Embeddings en una base de dades vectorial:

Per gestionar embeddings de manera eficient en una base de dades vectorial, es necessiten diversos components clau:

1. **Vectors d'Embeddings:** Aquests són els vectors numèrics generats per models de llenguatge que representen el significat semàntic de textos. Cada vector d'embeddings pot tenir centenars o milers de dimensions.
2. **Textos Font:** Els textos originals dels quals es generen els embeddings. Aquests poden ser frases, paràgrafs, documents sencers, etc.

3. **Metadades:** Informació addicional associada amb cada embedding, com ara la data de creació, la font del text, l'autor, i altres dades rellevants que poden ser útils per a les consultes i el filtratge.
4. **IDs i Referències:** Identificadors únics per als documents i embeddings que permeten referenciar-los fàcilment en les operacions de consulta i actualització.

Per exemple, una base de dades vectorial podria contenir embeddings generats a partir de descripcions de productes, on cada embedding està associat amb el text original de la descripció del producte, metadades com la categoria del producte i l'ID únic del producte. Aquesta configuració permetria realitzar cerques semàntiques eficients per trobar productes similars basats en les seves descripcions.

3.4.2. Cost, Eficiència i Consultes Avançades d'Embeddings

Generar embeddings utilitzant models avançats com "text-embedding-ada-002" té un cost associat que depèn del nombre de tokens processats. Cada token és una unitat de text, com una paraula o un símbol, i el cost es calcula en funció de la quantitat de tokens.

Per estimar el cost, primer hem de comptar el nombre total de tokens en els documents que volem processar. Això es pot fer utilitzant biblioteques com tiktoken, que permeten codificar textos en tokens. A continuació, es multiplica el nombre de tokens pel cost per 1.000 tokens establert pel model.

Per exemple, si el model "text-embedding-ada-002" té un cost de $0.0001 per 1.000 tokens, i tenim 444.463 tokens en total, el càlcul del cost es faria de la següent manera:

```
import tiktoken

enc = tiktoken.encoding_for_model("text-embedding-ada-002")

total_tokens = sum(len(enc.encode(text)) for text in documents)

cost_per_1k_tokens = 0.0001

cost = cost_per_1k_tokens * total_tokens / 1000
print('Total tokens:', total_tokens)
print('Cost:', cost)
```

Aquest càlcul ens ajuda a planificar i gestionar el pressupost necessari per generar embeddings, especialment quan treballem amb grans volums de dades.

Optimització de la Memòria i Rendiment:

La gestió de memòria i el rendiment són aspectes crítics quan treballem amb embeddings, especialment en aplicacions que requereixen temps de resposta ràpids i manipulació de grans volums de dades. Algunes tècniques per optimitzar l'ús de memòria i el rendiment inclouen:

- **Emmagatzematge Eficient:** Utilitzar bases de dades vectorials que permeten l'emmagatzematge eficient de grans quantitats d'embeddings, reduint la necessitat de carregar totes les dades en memòria simultàniament.
- **Indexació:** Crear índexs sobre els embeddings per accelerar les consultes. Això inclou l'ús d'algoritmes de cerca eficients com HNSW (Hierarchical Navigable Small World) per trobar vectors similars ràpidament.
- **Batch Processing:** Processar embeddings en lots (batches) per aprofitar millor la memòria disponible i reduir els temps de càlcul.
- **Parallelisme:** Utilitzar tècniques de paral·lelisme per distribuir la càrrega de treball entre múltiples processadors o màquines, millorant així la velocitat de processament.

Consultes Avançades i Filtratge:

Les consultes avançades i el filtratge permeten extreure el màxim valor dels embeddings emmagatzemats. Aquestes tècniques són essencials per optimitzar la recuperació de dades i garantir que els resultats obtinguts siguin els més rellevants i específics possibles. A continuació, es descriuen algunes tècniques i exemples per a la recuperació i filtratge de dades.

Consultes Utilitzant Múltiples Textos:

En algunes aplicacions, és útil realitzar consultes basades en múltiples textos per obtenir recomanacions més precises. Per exemple, si volem recomanar pel·lícules basant-nos en diverses preferències de l'usuari, podem utilitzar embeddings de diversos textos de consulta per recuperar els ítems més rellevants. Aquest enfocament permet combinar les preferències de múltiples fonts per generar recomanacions més acurades.

Per tal d'implementar aquesta tècnica, he creat aquest exemple, on primer obtinc els textos de referència dels identificadors corresponents i després, faig una consulta, utilitzant aquests textos de referència per obtenir els resultats més similars basats en els embeddings generats.

```
reference_ids = ['s8170', 's8103']
reference_texts = collection.get(ids=reference_ids)["documents"]

result = collection.query(
    query_texts=reference_texts,
    n_results=3
```

```
)
print(result)
```

En aquest codi, reference_ids conté els identificadors dels documents que volem utilitzar com a referència. A continuació, es recuperen els textos corresponents amb collection.get(), i es realitza una consulta amb collection.query(), especificant els textos de consulta i el nombre de resultats desitjats. Finalment, es mostren els resultats obtinguts.

Afegir i Utilitzar Metadades:

Afegir metadades als embeddings permet refinar les consultes i obtenir resultats més específics. Les metadades poden incloure informació addicional com el tipus de document, l'any de publicació, l'autor, etc. Aquesta informació addicional es pot utilitzar per aplicar filtres a les consultes, millorant així la precisió dels resultats.

Per exemple, si volem filtrar els resultats per tipus de document i any de publicació, primer actualitzem les metadades dels embeddings amb la informació rellevant. Després, fem una consulta utilitzant aquestes metadades com a criteris de filtratge. Tornem al mateix exemple d'abans, realitzant unes petites modificacions.

```
collection.update(ids=ids, metadatas=metadatas)

result = collection.query(
    query_texts=reference_texts,
    n_results=3,
    where={
        "type": "Movie"
    }
)
```

En aquest increment del codi anterior, utilitzo collection.update() per afegir o actualitzar les metadades dels documents amb els identificadors especificats. Després, collection.query() realitza una consulta utilitzant els textos de referència i aplico un filtre per tipus de document (type), en aquest cas, per recuperar només els elements que són pel·lícules.

Operadors de Filtratge:

Els operadors de filtratge permeten refinar les consultes de manera flexible, aplicant condicions específiques sobre les metadades dels embeddings. Alguns dels operadors més comuns inclouen:

- **$eq:** Igual a
- **$ne:** Diferent de
- **$gt:** Major que
- **$gte:** Major o igual que
- **$lt:** Menor que
- **$lte:** Menor o igual que

Aquests operadors poden ser utilitzats per crear condicions complexes que s'apliquen a les metadades dels documents. Per exemple, podem filtrar els resultats per tipus de document i any de publicació utilitzant múltiples operadors. Un exemple d'ús seria:

```
where={
  "type": "Movie",
  "release_year": {"$gt": 2020}
}
```

Aquest codi aplica dos filtres: el tipus de document ha de ser "Movie" i l'any de publicació ha de ser posterior a 2020. Aquesta combinació de condicions ens permet obtenir resultats que compleixen criteris molt específics.

4. Disseny del RAG

4.1. Definició, Objectiu i Beneficis del RAG

La Generació Augmentada per Recuperació (RAG) és un enfocament innovador que combina el poder dels sistemes de recuperació d'informació amb els models de llenguatge gran (LLM). Aquest mètode integra un component "retriever", que recupera fragments de documents rellevants d'un corpus gran, amb un component generatiu que utilitza aquesta informació recuperada per produir respostes informades. En essència, el RAG permet als models consultar fonts de dades externes per millorar la qualitat i la precisió de les respostes generades.

L'objectiu principal del RAG és millorar la precisió i rellevància de les respostes proporcionades pels models de llenguatge gran. En comptes de confiar exclusivament en el coneixement preentrenat del model, el RAG permet la consulta de bases de dades externes en temps real. Aquest enfocament és especialment útil per a aplicacions que

requereixen informació actualitzada o molt específica, com ara sistemes d'assistència al client, recomanacions personalitzades, i consultes sobre bases de coneixement dinàmiques.

Els RAG ofereixen diversos beneficis importants. Primer, permeten l'actualització en temps real de les respostes del model, assegurant que la informació utilitzada estigui sempre actualitzada i rellevant. Això és especialment valuós en entorns on les dades canvien freqüentment. Segon, ajuden a reduir les al·lucinacions, un problema comú en els models de llenguatge gran, ja que les respostes es basen en informació recuperada de fonts verificades. Tercer, són altament eficients pel que fa a les dades, ja que poden funcionar de manera efectiva amb conjunts de dades limitats. A més, els RAG són versàtils i es poden adaptar fàcilment a una àmplia gamma d'aplicacions i casos d'ús, oferint una solució robusta i flexible. Finalment, els RAG poden proporcionar una major transparència, permetent als usuaris veure quines fonts d'informació s'han utilitzat per generar una resposta, augmentant així la confiança i la interpretabilitat.

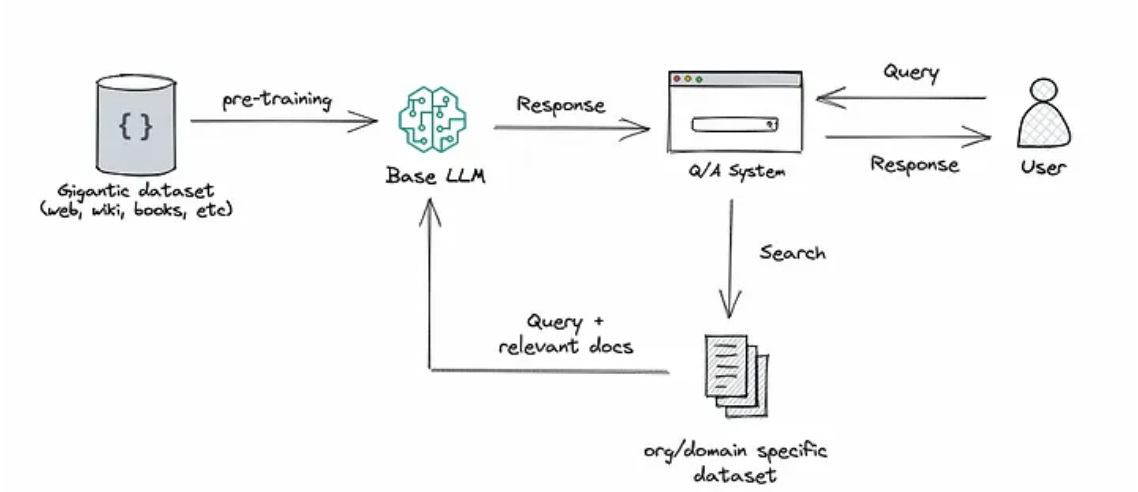

Ilustración 6. Esquema de funcionament RAG

Per altra banda, el Finetuning, o ajust fi, és una tècnica que permet personalitzar un model de llenguatge gran (LLM) per a tasques específiques ajustant els seus pesos amb un conjunt de dades més petit i específic. Aquesta metodologia té avantatges i desavantatges en comparació amb la Generació Augmentada per Recuperació (RAG).

El finetuning permet adaptar el model als matisos, tons i terminologies específiques del domini o tasca. Això resulta en un rendiment superior per a tasques concretes, ja que el model es pot ajustar per satisfer exactament les necessitats de l'aplicació. Aquesta capacitat de personalització és especialment valuosa en entorns on és important que el model reflecteixi el llenguatge i els coneixements específics del domini. Amb el finetuning, els desenvolupadors tenen un major control sobre com respon el model a diferents entrades. Això permet ajustar la manera en què el model processa la informació i genera respostes, fent possible una alineació més precisa amb les

expectatives de l'usuari final. També pot millorar significativament el rendiment del model en tasques específiques per a les quals s'ha entrenat. Això es deu a que el model aprèn a reconèixer patrons i contextos particulars del conjunt de dades específic utilitzat per al seu entrenament.

No obstant això, el finetuning té desavantatges significatius. Requereix grans quantitats de dades etiquetades per a l'entrenament, així com recursos computacionals considerables. Aquest requisit pot ser una barrera significativa, especialment en àmbits on les dades etiquetades són escasses o difícils d'obtenir. El procés de finetuning requereix recursos computacionals considerables. Entrenar un model gran amb dades específiques pot ser costós en termes de temps i recursos de computació, fent que aquesta opció no sigui viable per a totes les organitzacions. En entorns on les dades subjacents canvien freqüentment, el model ha de ser reentrenat regularment per mantenir la seva precisió i rellevància. Això implica un cost continu i pot ser poc pràctic per a aplicacions on la informació es actualitza constantment. Tot i que el finetuning pot reduir la probabilitat d'al·lucinacions (respostes inventades o inexactes) en comparació amb un model no ajustat, no elimina completament aquest problema. El model pot seguir produint respostes incorrectes quan es troba amb entrades que no estan ben representades en el conjunt de dades d'entrenament.

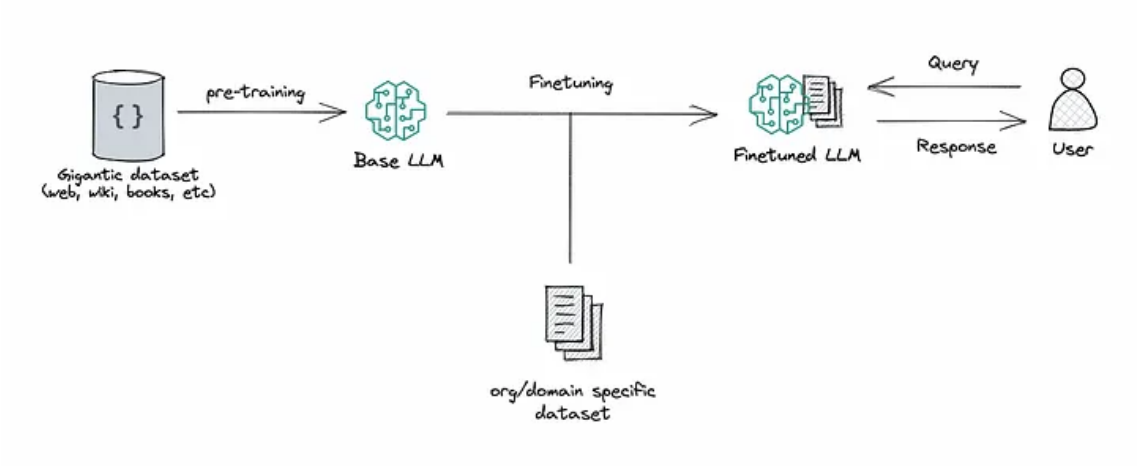

Ilustración 7. Esquema de funcionament del finetuning

La principal diferència entre el RAG i el finetuning és com gestionen la informació i s'adapten a les necessitats de l'aplicació. El RAG és ideal per a aplicacions que requereixen accedir a informació externa o actualitzada constantment, ja que pot recuperar dades en temps real. El finetuning, en canvi, és més adequat per a tasques amb dades relativament estàtiques i ben definides, on no és necessari accedir a noves fonts d'informació contínuament. El RAG pot funcionar amb conjunts de dades etiquetades limitats i requereix menys recursos computacionals perquè se centra en la recuperació d'informació existent. El finetuning, per contra, necessita grans quantitats de dades

específiques i és més costós en termes de computació. El RAG ofereix una major transparència, ja que pot mostrar quins documents o dades externes han influït en una resposta concreta, permetent una traçabilitat que el finetuning no proporciona. Això és important en aplicacions que requereixen alta responsabilitat i verificació. El finetuning proporciona un major control sobre el comportament del model, permetent una personalització detallada de les respostes. El RAG, tot i que és potent en la recuperació d'informació, no ajusta intrínsecament el comportament estilístic o específic del domini del model. Tot i això, com bé comentarem més endavant, també hem aplicat tècniques de finetuning al nostre RAG per tal de millorar el seu rendiment un cop realitzats els primers anàlisis.

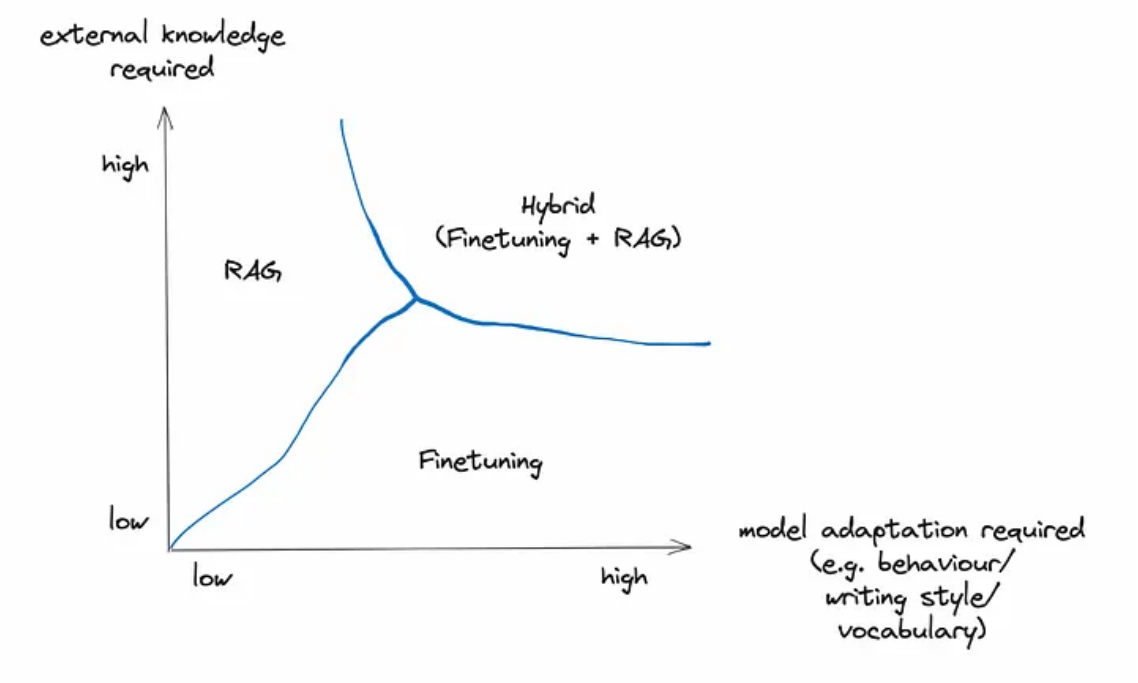

Ilustración 8. RAG VS finetuning

4.2. Descripció d'AquaCIS CF

AquaCIS CF és una solució avançada desenvolupada per AB Software, dedicada a la gestió integral del cicle de l'aigua. Aquesta plataforma està dissenyada per abordar totes les necessitats operatives de les empreses de subministrament d'aigua, des de la gestió de clients i la facturació fins a la comptabilitat i les operacions tècniques. El seu objectiu principal és millorar l'eficiència operativa i la qualitat del servei a través d'una administració integral i eficient dels recursos hídrics.

La implementació d'un sistema de Generació Augmentada per Recuperació (RAG) en AquaCIS CF sorgeix de la necessitat de proporcionar als treballadors que configuren el software una eina que els permeti consultar ràpidament qualsevol pregunta que tinguin sobre el sistema. En particular, aquesta necessitat és crítica en dos àmbits específics: impagaments i fraus.

En el cas dels impagaments, AquaCIS CF ja ofereix eines per a la gestió de cobraments. No obstant això, la diversitat i complexitat dels casos d'impagament sovint requereixen una intervenció manual intensiva, que pot ser costosa i lenta. Els retards en els pagaments poden provocar desequilibris financers importants per a les empreses de subministrament d'aigua. La integració d'un sistema RAG permet automatitzar la recuperació d'informació rellevant sobre polítiques de cobrament i procediments legals. Això permet als treballadors respondre ràpidament i amb precisió a les seves consultes, millorant la gestió dels impagaments i augmentant l'eficiència del personal.

Pel que fa als fraus, la detecció i gestió són essencials per protegir els ingressos i mantenir la integritat del sistema de subministrament d'aigua. Els fraus poden manifestar-se en formes diverses, com ara la manipulació de comptadors o connexions il·legals. Tot i que AquaCIS CF té capacitats per identificar patrons sospitosos i alertar els administradors, gestionar cada cas específic pot ser complex. Amb un sistema RAG, es pot accedir ràpidament a dades històriques, informes d'inspeccions i directrius de seguretat, permetent al personal tècnic respondre de manera més precisa i ràpida. Això ajuda a reduir els casos de frau i a mantenir un sistema de subministrament segur i eficient.

He decidit acotar el projecte als àmbits d'impagaments i fraus degut a la gran quantitat de documentació que disposem sobre aquestes àrees específiques del software AquaCIS CF. Aquesta decisió permet una millor gestió de les dades i una avaluació més precisa del sistema. La riquesa de la documentació disponible facilita la creació de bases de coneixement detallades, que són essencials per al correcte funcionament del sistema de Generació Augmentada per Recuperació (RAG). Focalitzant-nos en aquests dos àmbits, podem assegurar-nos que el sistema proporcioni respostes acurades i eficients als treballadors que configuren el software, millorant així la seva eficiència i precisió en la gestió d'aquestes situacions crítiques.

4.3. Creació del RAG

La creació del sistema de Generació Augmentada per Recuperació (RAG) per al software AquaCIS CF ha requerit diverses etapes clau. Aquest procés ha començat amb la recopilació i organització de la documentació existent sobre els àmbits d'impagaments i fraus, assegurant-nos que tota la informació rellevant estigués disponible i ben estructurada. La gran quantitat de documentació disponible ha estat essencial per alimentar el sistema de RAG, proporcionant una base sòlida de coneixement sobre aquests dos aspectes crítics del software.

El primer pas ha consistit en l'extracció de dades rellevants dels manuals, guies d'usuari, casos d'ús i altres documents tècnics associats a AquaCIS CF. Aquesta informació s'ha processat i s'ha emmagatzemat en dues bases de coneixement (KB) separades, una per a impagaments i l'altra per a fraus. Cada base de coneixement s'ha dissenyat per incloure descripcions detallades, procediments, preguntes freqüents i exemples pràctics, amb l'objectiu de cobrir totes les possibles consultes que els treballadors podrien tenir.

Un cop recopilada, la documentació s'ha organitzat i classificat de manera sistemàtica. S'ha creat una estructura jeràrquica clara per a cada KB, assegurant que la informació fos fàcilment accessible i navegable. Per exemple, la KB d'impagaments s'ha dividit en seccions com ara definició d'impagaments, procediments de seguiment, estratègies de recuperació i preguntes freqüents. De la mateixa manera, la KB de fraus s'ha estructurat en categories com identificació de fraus, mesures preventives, procediments d'investigació i casos d'estudi.

Després de l'organització inicial, la informació s'ha processat per assegurar que fos clara, concisa i pertinent. Això ha implicat la revisió dels documents per eliminar redundàncies, actualitzar informació obsoleta i assegurar la coherència terminològica i estilística. Aquest refinament ha estat crucial per garantir que les respostes proporcionades per les KB fossin d'alta qualitat i directament aplicables a les necessitats dels usuaris.

Un cop la informació ha estat ben organitzada i refinada, s'ha carregat en el sistema de gestió de coneixement de l'empresa. Aquest sistema, ja existent, permet l'accés ràpid i eficient a la informació per part dels models de llenguatge utilitzats en el RAG. La integració amb el sistema de recuperació, ja desenvolupat a l'empresa, ha permès aprofitar la infraestructura existent per oferir respostes generades basades en la informació de les KB.

4.4. Justificació del model escollit per al càlcul dels embeddings

La decisió d'utilitzar el model "text-embedding-3-small" en el nostre projecte es fonamenta en diversos factors que el fan especialment adequat per als nostres objectius, tal com es desprèn de les últimes actualitzacions proporcionades per OpenAI. En primer lloc, el model "text-embedding-3-small" ofereix un rendiment significativament superior en comparació amb el seu predecessor, "text-embedding-ada-002". Els resultats dels benchmarks indiquen una millora notable en l'eficàcia de recuperació en múltiples idiomes, amb un increment en la puntuació mitjana del 31,4% al 44,0% en el benchmark MIRACL. Aquesta millora es reflecteix també en les tasques en anglès, on la puntuació mitjana ha augmentat del 61,0% al 62,3% en el benchmark MTEB. Aquesta millora en el rendiment és crucial per garantir la precisió en la generació d'embeddings en el nostre

sistema de Generació Augmentada per Recuperació (RAG), on es requereix una alta qualitat en la representació semàntica.

Un altre aspecte clau és l'eficiència econòmica del model "text-embedding-3-small". OpenAI ha reduït substancialment el preu d'aquest model en comparació amb el seu predecessor, oferint una reducció de costos de fins a cinc vegades. Aquesta reducció en el cost, juntament amb la millora del rendiment, fa que aquest model sigui una opció òptima per a la implementació a gran escala en un projecte com el nostre, on l'eficiència en costos és tan important com l'eficàcia tècnica. A més, el "text-embedding-3-small" ha estat dissenyat per ser altament eficient i adequat per a aplicacions que requereixen l'ús de recursos limitats, cosa que ens permet integrar-lo fàcilment en el nostre sistema sense la necessitat de desplegar infraestructures costoses o complexes. La seva mida compacta i la seva capacitat per mantenir la qualitat en la generació d'embeddings, fins i tot amb recursos computacionals més baixos, el converteixen en l'elecció ideal per al nostre entorn d'implementació.

Finalment, cal destacar que, tot i que el model "text-embedding-3-large" ofereix una capacitat encara més gran per crear embeddings amb fins a 3072 dimensions, la nostra elecció del model "text-embedding-3-small" respon a la necessitat de trobar un equilibri entre la precisió, la mida dels embeddings, i els costos associats. Això ens permet assegurar que el nostre sistema RAG sigui escalable i sostenible a llarg termini, sense comprometre la qualitat de les respostes generades.

5. Metodologia d'avaluació

En aquest apartat es descriu la metodologia emprada per avaluar l'eficàcia i la precisió del sistema de Generació Augmentada per Recuperació (RAG) implementat per al software AquaCIS CF. Per tal d'assegurar una avaluació exhaustiva, s'ha dissenyat un Workflow específic que permet analitzar el rendiment del RAG des de diverses perspectives, incloent-hi tant mètriques quantitatives com avaluacions qualitatives realitzades per experts humans.

5.1. Workflow d'Avaluació

El primer pas en la metodologia d'avaluació ha implicat la creació d'un workflow que utilitzarem per guiar-nos durant tot el procés de validació del sistema RAG. Per crear aquest workflow m'he inspirat en esquemes de decisió estandarditzats, utilitzats sovint en la indústria per garantir una avaluació exhaustiva i sistemàtica. L'objectiu principal és assegurar que es consideren totes les variables i metodologies pertinents per avaluar el RAG en diferents contextos i escenaris d'ús.

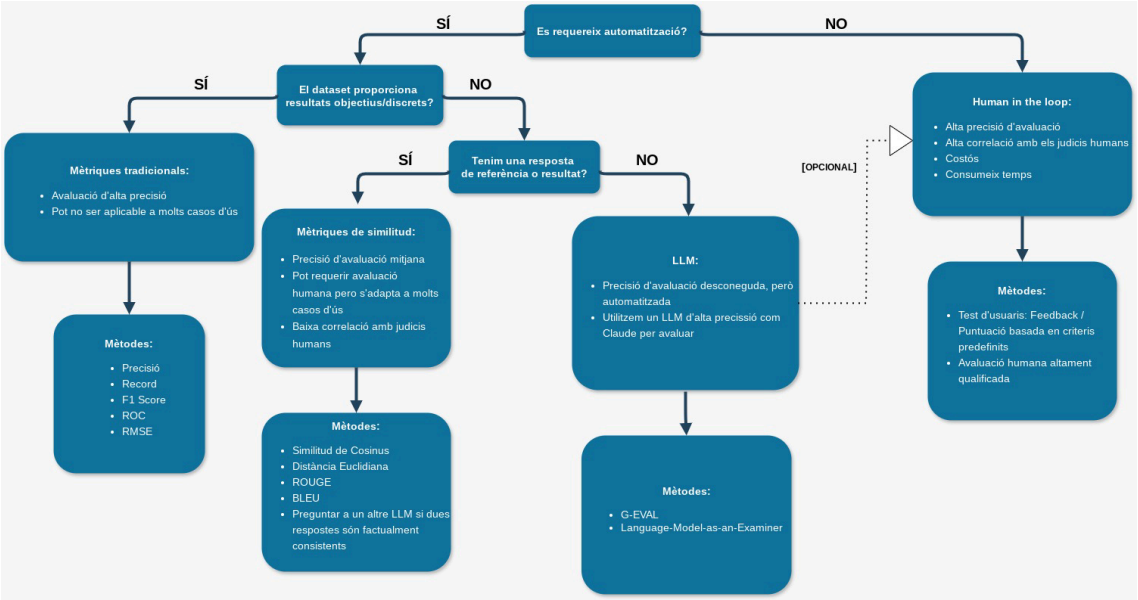

Ilustración 9. Workflow d'avaluació d'un RAG

El workflow comença amb la pregunta inicial: "Es requereix automatització?" Aquesta pregunta és crucial perquè defineix si el procés d'avaluació es pot gestionar de manera eficient mitjançant eines automatitzades, o si caldrà una avaluació manual per garantir la qualitat. Si es decideix que l'automatització és necessària, el següent pas és construir un dataset d'avaluació adequat que permeti la comparació automatitzada entre les respostes del RAG i les respostes de referència.

Un cop definit si s'utilitza o no automatització, el workflow es bifurca depenent de si "El dataset proporciona resultats objectius o discrets?" Aquesta bifurcació és important perquè determina la metodologia d'avaluació que s'ha d'utilitzar. Si el dataset té resultats objectius o discrets, es poden aplicar mètriques tradicionals, com la precisió, el record, el F1 Score, el ROC o l'RMSE. Aquestes mètriques són ideals per a casos d'ús on es pot mesurar directament l'eficàcia del RAG comparant les respostes generades amb les respostes correctes conegudes.

Si el dataset no proporciona resultats discrets, el workflow es mou cap a l'ús de mètriques de similitud, que permeten una avaluació basada en la comparació de les respostes generades pel RAG amb respostes de referència conegudes. Les mètriques de similitud inclouen la similitud de Cosinus, la distància Euclidiana, ROUGE, BLEU, i fins i tot l'ús d'un altre LLM per verificar la consistència factual de les respostes. Aquestes mètriques proporcionen una precisió d'avaluació mitjana i sovint necessiten la intervenció humana per validar els resultats, ja que les correlacions amb els judicis humans poden ser baixes.

Si es disposa d'una resposta de referència o resultat conegut, el workflow permet l'avaluació mitjançant un altre LLM d'alta precisió, com ara Claude, per comparar i verificar l'eficàcia del RAG en un procés més automatitzat. Aquesta part del workflow és

particularment útil quan es requereix una verificació ràpida i precisa, sense la necessitat d'un cost elevat associat a l'avaluació manual.

Si no es té una resposta de referència clara, o si es necessita una avaluació més precisa, s'opta per la metodologia de Human in the loop. Aquesta part del workflow implica l'avaluació directa per part d'experts humans que revisen les respostes generades pel RAG. Aquest enfocament és més costós i consumeix temps, però ofereix una correlació molt alta amb els judicis humans, cosa que és crucial per a casos d'ús d'alt risc o alta sensibilitat. En aquesta fase, els mètodes inclouen tests d'usuaris amb feedback i puntuacions basades en criteris predefinits, així com una avaluació detallada i qualificada per part d'experts en el tema.

5.2. Human in the loop

En l'apartat de Human in the Loop, s'ha implementat una metodologia exhaustiva per validar el funcionament del sistema RAG en comparació amb respostes generades per un model de llenguatge preentrenat. Aquesta metodologia s'ha basat en la intervenció activa d'experts humans per avaluar la qualitat de les respostes proporcionades pel RAG, i s'ha desenvolupat mitjançant una sèrie de passos estructurats que garanteixen una avaluació rigorosa i precisa.

Per dur a terme aquesta validació, s'han creat dos fulls de càlcul separats, un dedicat a l'àmbit dels impagaments i l'altre a l'àmbit dels fraus, que són les dues àrees principals sobre les quals s'ha centrat el nostre RAG. Aquests fulls de càlcul contenen tres columnes inicials essencials: "Preguntes", "Respuesta GPT" i "Respuesta KB".

El procés ha començat amb la generació de preguntes i respostes a partir de la documentació existent. Utilitzant el chat corporatiu de Veolia, se li ha demanat que generi 10 preguntes i les seves respectives respostes per cada document. Aquestes preguntes i respostes inicials s'han registrat en les columnes "Preguntes" i "Respuesta GPT" respectivament.

Posteriorment, s'ha posat a prova el RAG, ja nodrit amb tota la documentació disponible, però sense accedir als documents directament en aquell context específic del chat. S'ha fet la mateixa pregunta al RAG i s'ha registrat la resposta generada en la columna "Respuesta KB". Aquesta part del procés ha permès comparar directament les respostes generades pel RAG amb les respostes generades inicialment pel model GPT.

Per garantir la fiabilitat de l'avaluació, les "Respuestas GPT" han estat revisades i corregides per professionals experts en el software AquaCIS. Aquests experts han aplicat una rúbrica de quatre criteris per puntuar les respostes:

1. Correctitud Factual
2. Rellevància

3. Completitu
4. Claredat

Cada resposta ha estat puntuada segons aquests criteris, i s'ha calculat una mitjana de les puntuacions obtingudes per obtenir una Puntuació Total. Els resultats han sigut els següents:

Taula 1: Resultats de Puntuació per "IMPAGADOS"

Pregunta	Correctitud Factual	Relevancia	Completitud	Claridad	Puntuación Total
Pregunta 1	3	4	4	3	3.5
Pregunta 2	4	4	4	4	4.0
Pregunta 3	3	3	3	3	3.0
Pregunta 4	4	4	3	3	3.5
Pregunta 5	3	3	3	3	3.0
Pregunta 6	4	4	4	4	4.0
Pregunta 7	3	4	3	4	3.5
Pregunta 8	4	3	4	3	3.5
Pregunta 9	3	4	3	4	3.5
Pregunta 10	4	4	4	4	4.0

Els resultats de la taula d'impagats han mostrat un rendiment consistent del sistema RAG en aquest àmbit, amb puntuacions que oscil·len entre 3.0 i 4.0 en els quatre criteris avaluats: Correctitud Factual, Relevància, Completitud i Claritat. La puntuació més baixa l'hem obtingut en les Preguntes 3 i 5, amb una mitjana de 3.0, cosa que suggereix que en aquests casos les respostes han estat correctes però no completament adequades o clares. En canvi, les respostes a les Preguntes 2, 6 i 10 han destacat, amb una puntuació total de 4.0, indicant que el RAG ha proporcionat respostes molt precises, rellevants, completes i clares. Aquests resultats suggereixen que el sistema és efectiu en la majoria de les situacions, tot i que hi ha certs àmbits on podríem millorar la claredat o la completitud de la informació proporcionada.

Taula 2: Resultats de Puntuació per "FRAUDES"

Pregunta	Correctitud Factual	Relevancia	Completitud	Claridad	Puntuación Total
Pregunta 1	4	4	4	4	4.0
Pregunta 2	3	3	3	3	3.0
Pregunta 3	4	4	3	3	3.5
Pregunta 4	3	4	3	4	3.5
Pregunta 5	4	3	4	3	3.5
Pregunta 6	3	4	3	4	3.5

Pregunta 7	4	4	4	4	4.0
Pregunta 8	3	3	3	3	3.0
Pregunta 9	4	4	4	4	4.0
Pregunta 10	3	4	3	4	3.5

En l'àmbit de fraus, les puntuacions obtingudes pel sistema RAG han estat similars a les de l'àmbit d'impagats, amb una mitjana que varia entre 3.0 i 4.0. Les respostes a les Preguntes 1, 7 i 9 han rebut la puntuació màxima de 4.0, indicant una excel·lent precisió factual, rellevància, completitud i claredat. Això reflecteix que el RAG ha estat particularment eficaç en proporcionar respostes d'alta qualitat en aquests casos. No obstant això, hi ha respostes, com les de les Preguntes 2 i 8, que han obtingut una puntuació de 3.0, indicant que el sistema podria haver proporcionat respostes menys completes o clares en aquestes instàncies. En conjunt, aquests resultats suggereixen que el RAG ha funcionat bé en la majoria dels casos relacionats amb fraus, tot i que encara hi ha espai per a la millora, especialment pel que fa a la consistència en la completitud i claredat de les respostes.

5.3. Mètriques de similitut

5.3.1. Distància Euclidiana

La distància euclidiana és una de les mètriques més utilitzades per mesurar la similitud entre dos vectors en un espai vectorial. Aquesta distància es defineix com la longitud del camí més curt entre dos punts en aquest espai, i es calcula mitjançant la fórmula de la distància euclidiana, que és la suma de les diferències al quadrat entre els components corresponents dels dos vectors, seguida de l'arrel quadrada del resultat.

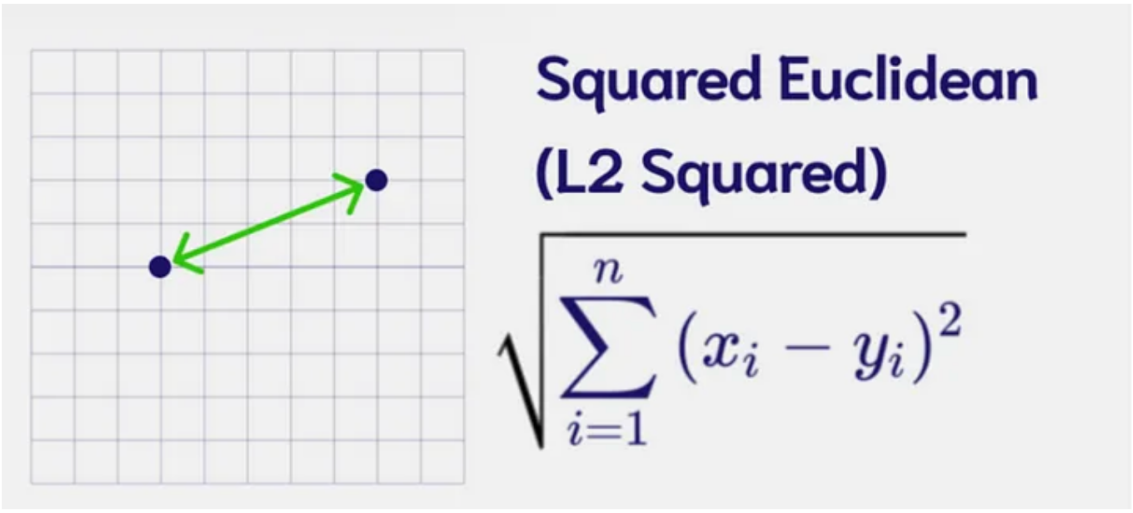

Ilustración 10. Fòrmula distància euclidiana

Aquesta mètrica és especialment útil en aplicacions on la magnitud i les diferències absolutes entre els elements dels vectors són importants. Per exemple, en l'àmbit del processament del llenguatge natural (NLP), la distància euclidiana es pot utilitzar per comparar les representacions vectorials de frases o paraules, conegudes com embeddings, per determinar la similitud entre elles. Un valor més baix de la distància euclidiana indica que els dos vectors són més propers en l'espai vectorial, cosa que significa que les seves representacions són més similars.

Tot i això, la distància euclidiana també té les seves limitacions. Per exemple, és sensible a l'escala dels dades, el que significa que si les característiques tenen diferents escales, la distància euclidiana pot estar dominada per la característica amb l'escala més gran. A més, en espais d'alta dimensionalitat, la distància euclidiana pot perdre significat a causa del fenomen conegut com la "maledicció de la dimensionalitat", on totes les distàncies tendeixen a convergir.

En el nostre cas, hem utilitzat la distància euclidiana per comparar les respostes generades per un sistema GPT i el sistema RAG abans i després d'introduir un prompt específic. Els resultats obtinguts ens han permès avaluar la millora en la precisió i la similitud de les respostes després de l'aplicació del prompt, mostrant com es redueix la distància euclidiana entre les respostes generades pel GPT i les respostes generades pel RAG. Això ens indica que les respostes del RAG són més coherents i properes a les generades pel GPT després de l'ajustament, millorant així la qualitat global del sistema. Els resultats han sigut el següents:

Pregunta	Distancia Euclidiana (PRE)	Distancia Euclidiana (POST)
Pregunta 1	0,863805001	0,386657342
Pregunta 2	0,965892431	0,442054191
Pregunta 3	0,703044293	0,449060479
Pregunta 4	0,514160324	0,708408986
Pregunta 5	0,273059067	0,537777485
Pregunta 6	0,82299609	0,611512204
Pregunta 7	0,190119312	0,391719534
Pregunta 8	0,46257653	0,37226844
Pregunta 9	0,674766809	0,514605582
Pregunta 10	0,778575016	0,546282724

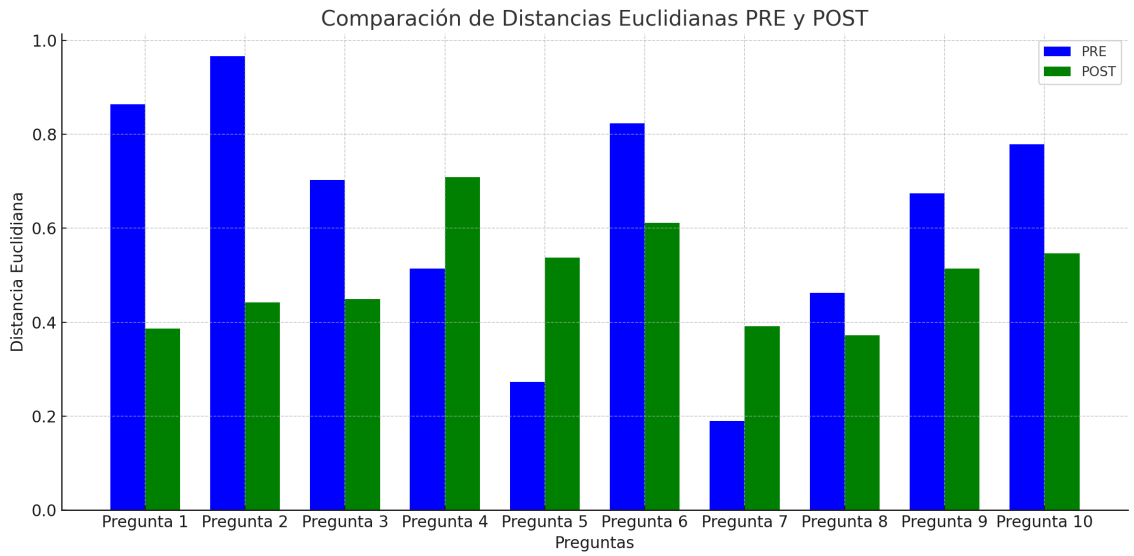

Ilustración 11. Comparació de distàncies euclidianes PRE i POST

Els resultats indiquen que, en la majoria de les preguntes, les distàncies euclidianes han disminuït significativament després d'introduir el prompt. Això suggereix que les respostes generades pel RAG després de l'ús del prompt són més properes a les respostes esperades, reflectint una millora en la precisió i la coherència de les respostes del sistema.

En concret, hem observat una reducció notable de la distància en preguntes com la 1, 2, 3 i 6, la qual cosa demostra que el prompt ha ajudat a afinar la resposta, augmentant-ne la qualitat. En canvi, en altres preguntes, com la 5, la diferència entre les distàncies PRE i POST és menys accentuada, inclus perdent un mica de rendiment, indicant que l'impacte del prompt en aquests casos ha empitjorat la resposta.

5.3.2. Distància Cosinus

A diferència de la distància euclidiana, que mesura la distància rectilínia entre dos punts, la distància cosinus es centra en la similitud de la direcció dels vectors, independentment de la seva magnitud. Això fa que sigui una mètrica molt robusta en espais d'alta dimensionalitat, on la distància euclidiana pot perdre significació. En altres paraules, la distància cosinus és especialment valuosa quan es vol mesurar com de similars són dos vectors en termes de la seva orientació en l'espai, més que no pas per la distància absoluta entre ells. A més, la distància cosinus és senzilla d'entendre i calcular, proporcionant una mesura directa de la similitud en termes de direcció, cosa que la fa molt aplicable en motors de cerca i sistemes de recomanació.

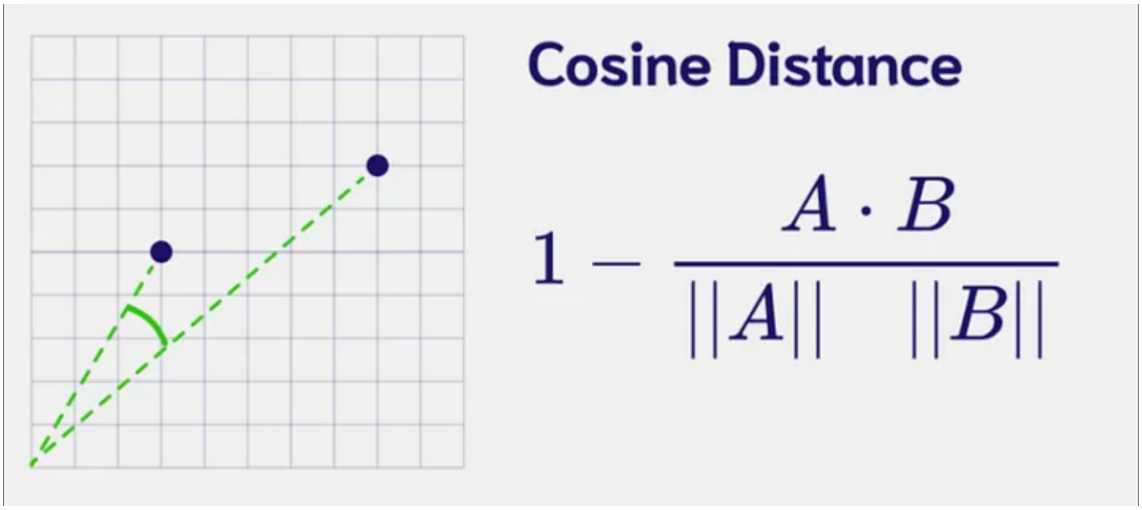

Ilustración 12. Fòrmula distància cosinus

No obstant això, la distància cosinus també té desavantatges. Un dels més importants és que ignora la magnitud dels vectors, la qual cosa pot ser un inconvenient en situacions on aquesta magnitud és rellevant. A més, tot i ser eficient en espais d'alta dimensionalitat, pot ser sensible a vectors molt esparsos, on petites variacions poden afectar desproporcionadament la mesura. També és important destacar que la distància cosinus no capta informació sobre la magnitud, la qual cosa pot limitar la seva utilitat en contextos on aquesta informació és crucial.

Per a l'anàlisi de la distància cosinus entre les respostes generades pel model de GPT i les respostes obtingudes mitjançant el sistema RAG en els àmbits d'impagats i fraus, hem utilitzat una comparació entre les distribucions PRE i POST implementació del prompt. Aquestes distàncies cosinus ens permeten avaluar la similitud direccional entre els vectors d'embeddings corresponents a les respostes, on una menor distància resultarà una major similitud.

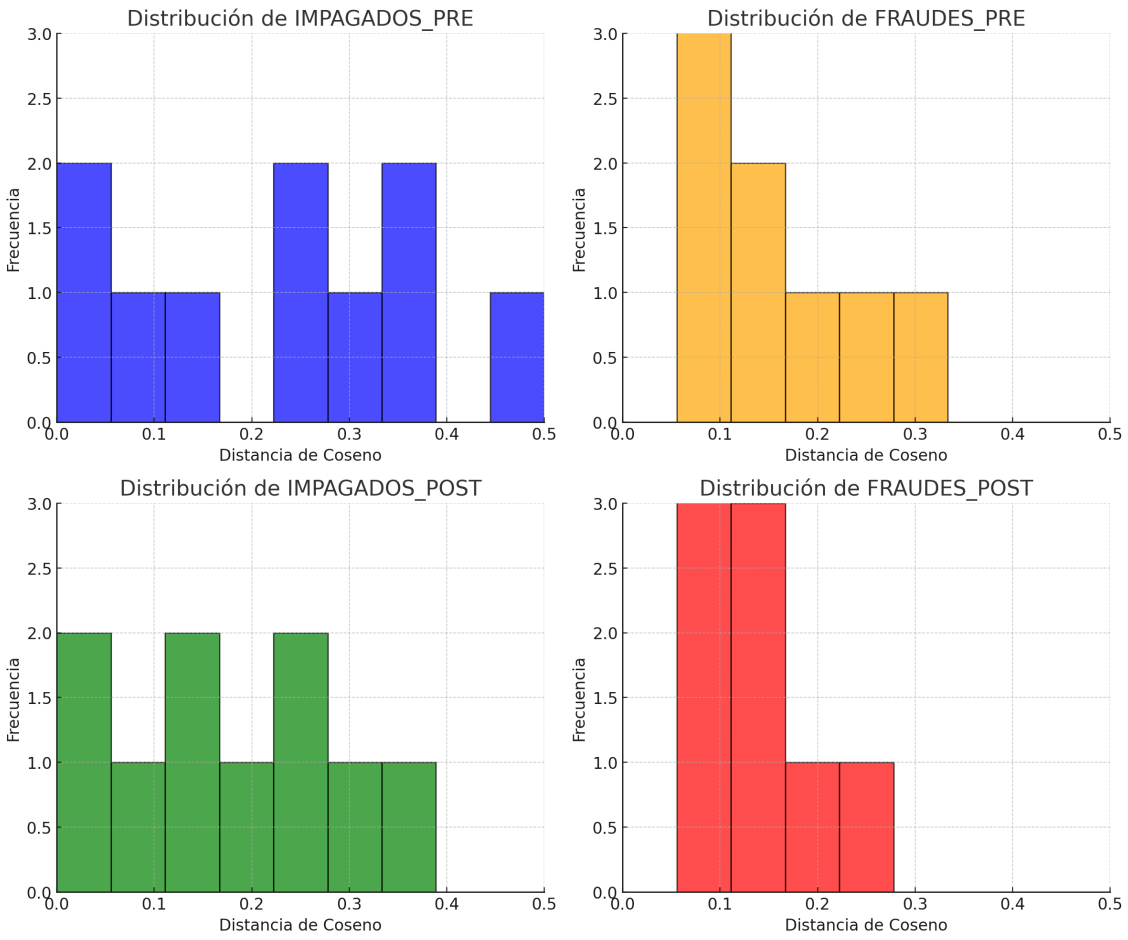

Ilustración 13. Distribucions de resultats distància cosinus PRE i POST

En primer lloc, els gràfics que mostren les distàncies cosinus abans de la implementació del prompt (IMPAGADOS_PRE i FRAUDES_PRE) ens mostren una distribució més variada en el cas d'impagats, amb valors que s'estenen des de 0,0 fins a 0,5. Aquesta variabilitat indica que, inicialment, les respostes de GPT i les obtingudes mitjançant el sistema RAG eren més heterogènies en termes de similitud, especialment per a impagats. En canvi, per a fraus, les distàncies cosinus estaven més concentrades en valors baixos, entre 0,0 i 0,33, suggerint una major uniformitat en les respostes des de l'inici.

Després de la implementació del prompt al sistema RAG (IMPAGADOS_POST i FRAUDES_POST), s'observa una tendència cap a una major concentració de les distàncies cosinus en intervals més baixos per ambdós casos, amb valors predominantment situats entre 0,1 i 0,3. Aquesta concentració de valors indica que les respostes generades s'han tornat més similars entre si després de la implementació del sistema RAG, la qual cosa suggereix una millora en la precisió i consistència de les respostes.

Els resultats obtinguts mostren un percentatge de millora clar després de la implementació del prompt. En el cas dels impagats, el percentatge de millora és particularment notable, amb una reducció aproximada del 20% en les distàncies cosinus més elevades. Això indica una major concentració de les distàncies cap als valors més baixos en el gràfic POST, la qual cosa suggereix que les respostes generades pel sistema són ara més similars a les esperades i, per tant, de major qualitat.

Pel que fa als fraus, també s'ha observat un percentatge de millora d'entre un 10% i un 15%. Encara que les distàncies cosinus ja eren relativament baixes en la fase PRE, la implementació del prompt ha aconseguit reduir-les encara més en el gràfic POST. Aquest resultat reforça la idea que el sistema ha estat efectiu, oferint respostes més coherents i ajustades a les necessitats tant en l'àmbit dels impagats com en el dels fraus.

5.3.3. ROUGE N i L

El ROUGE (Recall-Oriented Understudy for Gisting Evaluation) és una família de mètodes utilitzats per avaluar la qualitat de textos generats automàticament en comparació amb textos de referència. Dins d'aquesta família, les variants més comunes són ROUGE-N i ROUGE-L. EL ROUGE-N es basa en la coincidència de n-grams, que són seqüències de n paraules consecutives. Així, ROUGE-1 mesura la coincidència de paraules individuals (unigrams), mentre que ROUGE-2 avalua la coincidència de parelles de paraules (bigrams) entre el text generat i el text de referència. Per altre banda, el ROUGE-L es centra en la longitud de la subseqüència comuna més llarga (LCS) entre el text generat i el de referència, considerant la subseqüència més llarga que respecta l'ordre d'aparició. Aquesta mètrica és útil per captar la similitud en l'estructura general de les frases, més enllà de la coincidència de paraules individuals o bigrames.

Un exemple del càlcul del ROUGE-1 seria:

Referència:

The weather is cold outside.

Output by LLM:

The weather is cold.

Fòrmules per al càlcul del Rouge:

$$\text{Rouge} - 1\ (\text{Recall}) = \frac{unigram\ matches}{unigram\ in\ reference}$$

$$\text{Rouge} - 1\ (\text{Precision}) = \frac{unigram\ matches}{unigram\ in\ output}$$

Rouge-1 (Recall) = 4/5 => 0.8

Rouge-1 (Precision) = 4/4 => 1

Un exemple del càlcul del ROUGE-2 seria:

Referència:

The weather is cold outside.

Bigram:

(the, weather), (weather, is), (is, cold), (cold, outside)

Output by LLM:

The weather is cold.

Bigram:

(the, weather), (weather, is), (is, cold).

Formules per al càlcul del Rouge:

$$\text{Rouge} - 1\ (\text{Recall}) = \frac{unigram\ matches}{unigram\ in\ reference}$$

$$\text{Rouge} - 1\ (\text{Precision}) = \frac{unigram\ matches}{unigram\ in\ output}$$

Rouge-2 (Recall) = 3/4 => 0.75

Rouge-2 (Precision) = 3/3 => 1

He creat un codi en Python per calcular les mètriques ROUGE-N (ROUGE-1 i ROUGE-2) i ROUGE-L. El que he fet ha estat desenvolupar un script que llegeix les respostes generades pel sistema RAG i les respostes de referència generades pel model GPT des d'un fitxer Excel. A partir d'aquí, el codi compara aquestes respostes utilitzant les mètriques ROUGE per quantificar la similitud entre els textos. He utilitzat la llibreria rouge_score, que em permet calcular diverses variants de ROUGE. El codi recorre cada parell de respostes (RAG i GPT) i calcula les f-measures per ROUGE-1, ROUGE-2 i ROUGE-L, que són indicadors de la superposició de n-grams entre les respostes, i per tant, de la seva similitud en termes de contingut i estructura. Després de calcular aquestes mètriques, he afegit les puntuacions al mateix fitxer Excel per a una posterior anàlisi. A més, he generat dos gràfics que mostren la distribució de les puntuacions ROUGE per a cada pregunta, facilitant així la visualització de la qualitat i similitud de les respostes generades pel sistema RAG en comparació amb les respostes del model GPT. Aquests han estat els resultats:

ROUGE-1 i ROUGE-2 (PRE):

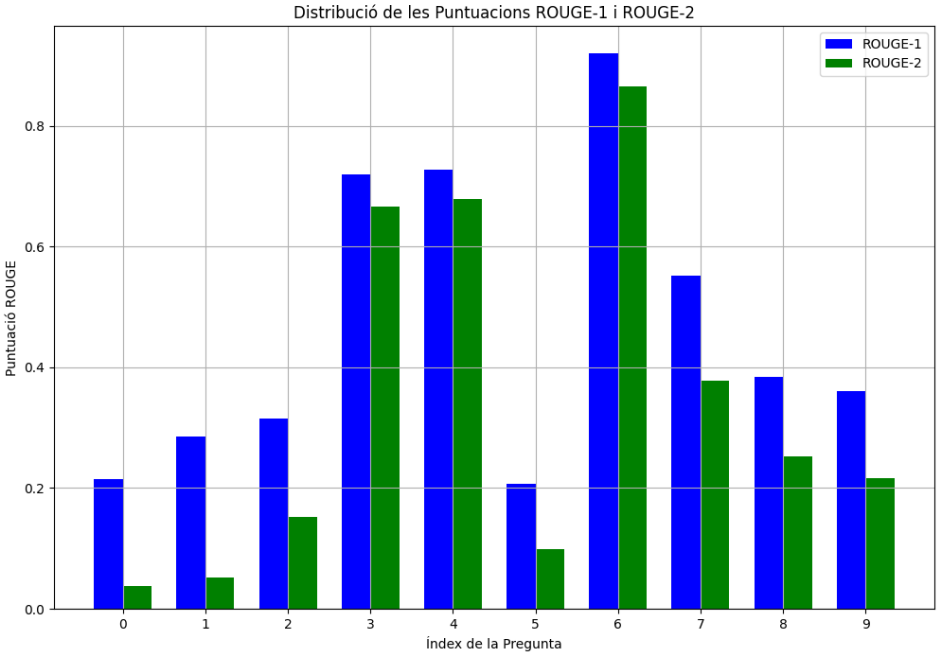

Ilustración 14. Distribució de les puntuacion ROUGE-1 i ROUGE-2 (PRE)

ROUGE-1 i ROUGE-2 (POST):

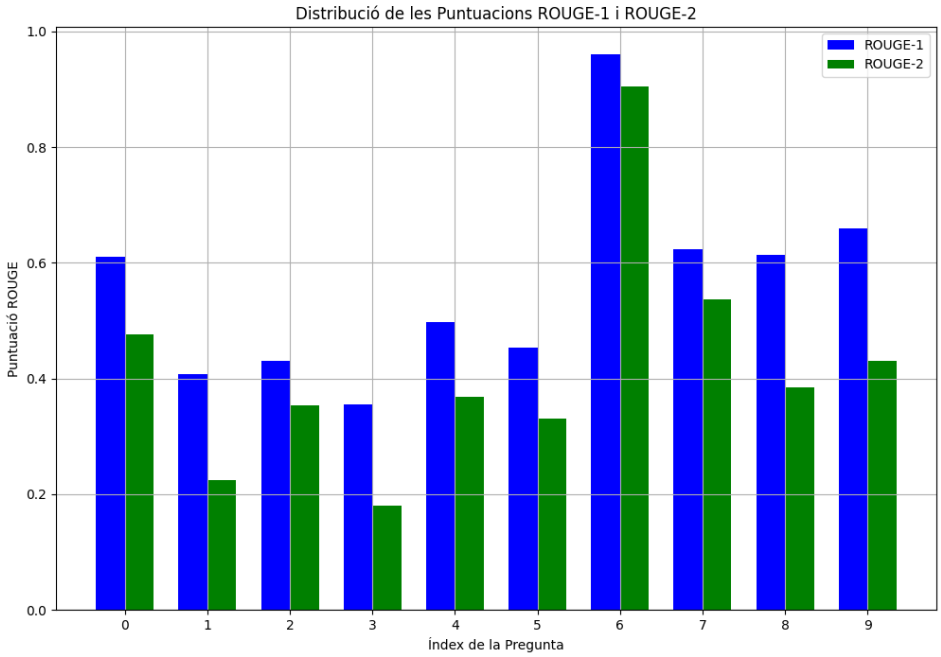

Ilustración 15. Distribució de les puntuacions ROUGE-1 i ROUGE-2 (POST)

En els resultats PRE, observem que les puntuacions de ROUGE-1 i ROUGE-2 mostren variabilitat significativa entre les diferents preguntes. Les puntuacions més altes es concentren en les preguntes 3, 4, i 6, indicant una major semblança entre les respostes generades pel model de llenguatge i les respostes esperades en aquests casos específics. Tanmateix, hi ha altres preguntes, com les preguntes 1 i 5, on les puntuacions són més baixes, la qual cosa suggereix que les respostes no són tan alineades amb les respostes de referència.

Després d'implementar el prompt, les puntuacions de ROUGE-1 i ROUGE-2 (POST) mostren una millora general en la qualitat de les respostes. Les puntuacions per a la majoria de les preguntes han augmentat, amb un increment notable en la consistència de les puntuacions entre ROUGE-1 i ROUGE-2. Aquest augment en les puntuacions reflecteix una millor alineació entre les respostes generades pel model i les respostes de referència, especialment després d'haver ajustat el model amb el prompt específic.

En general, els resultats POST indiquen que la implementació del prompt ha estat efectiva per millorar la qualitat de les respostes generades pel model, tal com es demostra amb les puntuacions més altes i més consistents de ROUGE-1 i ROUGE-2 en comparació amb els resultats PRE

ROUGE-L (PRE):

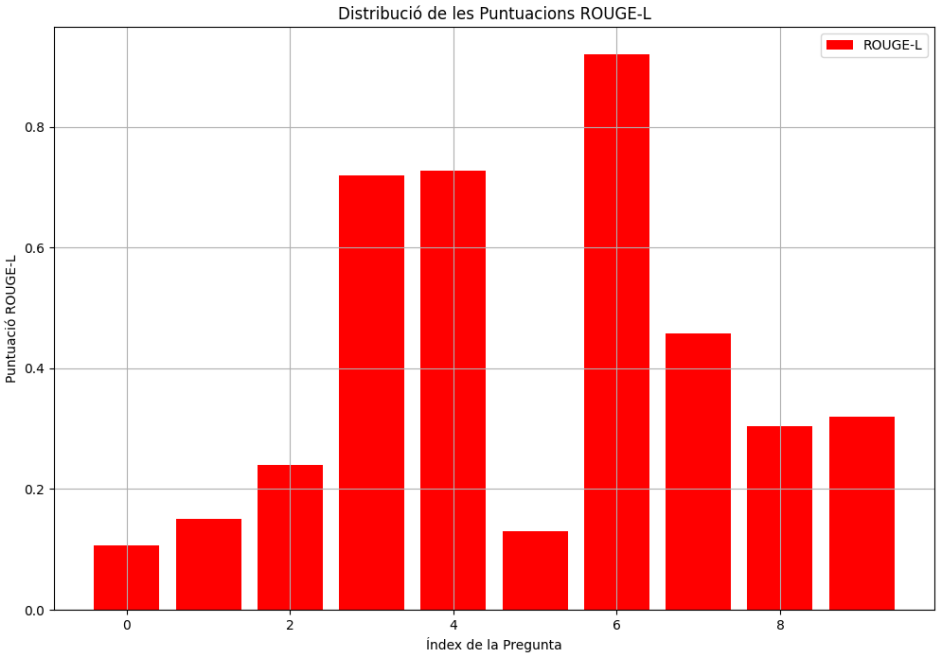

Ilustración 16. Distribució de les puntuacions ROUGE-L (PRE)

ROUGE-L (POST):

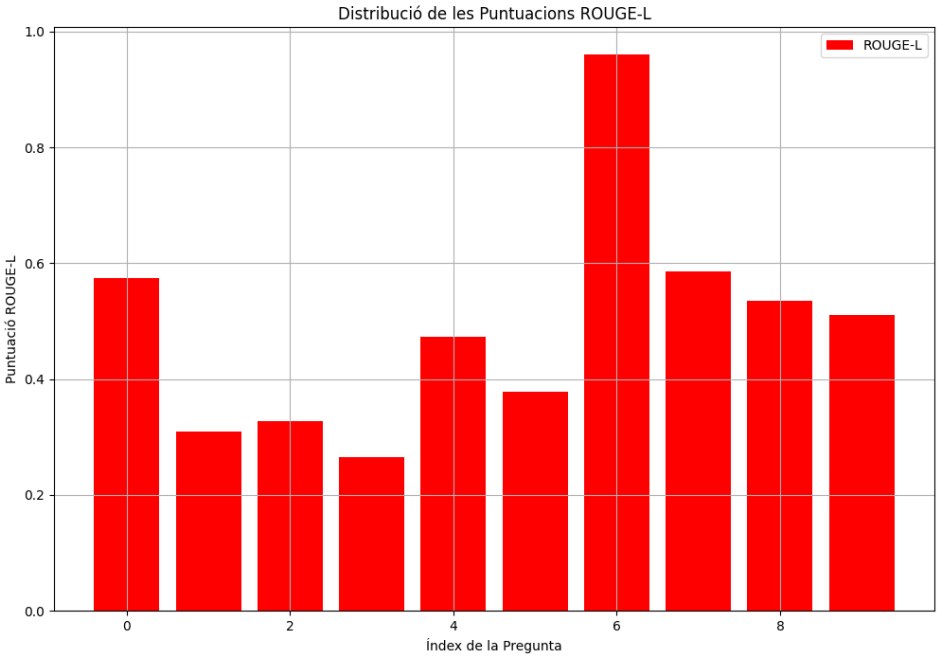

Ilustración 17. Distribució de les puntuacions ROUGE-L (POST)

En el gràfic PRE, podem observar que les puntuacions ROUGE-L varien significativament entre les diferents preguntes. Tot i així, hi ha una tendència general a tenir valors moderats, amb puntuacions que oscil·len entre 0,2 i 0,8. Això indica que, abans de la implementació del prompt, la coincidència entre les respostes generades pel GPT i les respostes correctes del KB era acceptable, però no excel·lent, amb algunes preguntes obtenint puntuacions més baixes.

Per altre banda, el en gràfic POST, podem observar que les puntuacions ROUGE-L varien significativament entre les diferents preguntes. Tot i així, hi ha una tendència general a tenir valors moderats, amb puntuacions que oscil·len entre 0,2 i 0,8. Això indica que, abans de la implementació del prompt, la coincidència entre les respostes generades pel GPT i les respostes correctes del KB era acceptable, però no excel·lent, amb algunes preguntes obtenint puntuacions més baixes.

5.4. Avaluació mitjançant Models de Llenguatge Gran (LLM)

En aquest apartat es detalla la metodologia d'avaluació del sistema RAG utilitzant Models de Llenguatge Gran (LLM) per automatitzar el procés de valoració de les respostes generades. Aquesta estratègia complementa les metodologies anteriors, aportant una solució eficient i escalable per avaluar la qualitat de les respostes en funció de diversos criteris clau.

5.4.1. Objectiu de l'avaluació amb LLM

L'objectiu principal d'utilitzar un LLM en el procés d'avaluació és obtenir una valoració objectiva i consistent de les respostes generades pel sistema RAG sense la necessitat d'intervenció humana directa en cada cas. Aquesta metodologia permet:

- Escalabilitat: Avaluar grans volums de dades de manera ràpida i eficient.
- Consistència: Mantenir criteris d'avaluació uniformes en totes les respostes.
- Objectivitat: Reduir el biaix subjectiu que pot sorgir en l'avaluació humana.
- Rapidesa: Accelerar el procés d'avaluació, facilitant iteracions més ràpides en el desenvolupament del sistema.

5.4.2. Metodologia d'avaluació

Per dur a terme l'avaluació amb LLM, s'ha seguit un procés estructurat que involucra els següents passos:

1. Selecció del Model de Llenguatge Gran:

S'ha utilitzat un model GPT-4 degut a la seva capacitat avançada de comprensió i generació de llenguatge natural, així com la seva competència per avaluar textos en funció de múltiples criteris de qualitat.

2. Definició dels Criteris d'Avaluació:

S'han establert quatre criteris fonamentals per avaluar cada resposta:

- Correctitud Factual: Mesura si la resposta és precisa i es basa en informació correcta.
- Rellevància: Avalua si la resposta és pertinent i està directament relacionada amb la pregunta formulada.
- Completitud: Determina si la resposta cobreix tots els aspectes importants de la pregunta.
- Claredat: Valora si la resposta és clara, ben estructurada i fàcil d'entendre.

Cada criteri es puntua en una escala de 0 a 5, on 0 representa una manca total de qualitat en el criteri, i 5 indica una excel·lència completa.

3. Preparació del Dataset per a l'Avaluació:

S'han recopilat parelles de preguntes i respostes generades pel sistema RAG en els àmbits d'impagaments i fraus. Aquestes dades han estat organitzades en un format estructurat per facilitar el procés d'avaluació automatitzada.

4. Disseny del Prompt per al LLM:

S'ha dissenyat un prompt específic i detallat per instruir el LLM sobre com avaluar cada resposta segons els criteris definits. El prompt inclou:

- Instruccions clares: Especificant la tasca d'avaluació i els estàndards esperats per a cada criteri.
- Exemples il·lustratius: Proporcionant mostres de respostes amb puntuacions associades per guiar el model en l'aplicació correcta dels criteris.
- Format de sortida estructurat: Indicant al model com presentar les puntuacions per a cada criteri i el càlcul de la puntuació total.

Exemple de Prompt Utilitzat:

Imagina que ets un professional d'AquaCIS CF. A continuació trobaràs una pregunta i la seva resposta generada pel sistema RAG. Avalua la resposta segons els següents criteris, assignant una puntuació de 0 a 5 per a cadascun:

1. Correctitud Factual

2. Rellevància

3. Completitud

4. Claredat

Després, calcula la puntuació total com la mitjana de les quatre puntuacions. Presenta els resultats en el següent format:

Pregunta: [Aquí va la pregunta]

Resposta: [Aquí va la resposta]

Avaluació:

- Correctitud Factual: X

- Rellevància: Y

- Completitud: Z

- Claredat: W

- Puntuació Total: V

5. Procés d'Avaluació Automatitzada:

S'ha implementat un script automatitzat que envia cada parella de pregunta i resposta al LLM utilitzant el prompt definit. El model retorna les puntuacions per a cada criteri i la puntuació total corresponent. Aquest procés es repeteix per a tot el conjunt de dades, recopilant les puntuacions de manera sistemàtica.

6. Recopilació i Anàlisi de Resultats:

Les puntuacions generades pel LLM s'han recopilat en fulls de càlcul per facilitar l'anàlisi estadística i la visualització dels resultats. S'han elaborat taules i gràfics per identificar tendències, fortaleses i àrees de millora en les respostes del sistema RAG.

5.4.3. Resultats de l'avaluació

Taula 3: Resultats de Puntuació per "IMPAGADOS" mitjançant LLM

Pregunta	Correctitud Factual	Rellevància	Completitud	Claredat	Puntuació Total
Pregunta 1	4	5	4	5	4.5
Pregunta 2	5	5	5	5	5.0
Pregunta 3	3	4	3	4	3.5
Pregunta 4	4	4	4	4	4.0
Pregunta 5	3	3	3	3	3.0
Pregunta 6	5	5	5	5	5.0
Pregunta 7	4	4	4	4	4.0
Pregunta 8	3	4	3	4	3.5
Pregunta 9	4	5	4	5	4.5
Pregunta 10	5	5	5	5	5.0

Les puntuacions obtingudes demostren una excel·lent qualitat en la majoria de les respostes generades pel sistema RAG en l'àmbit d'impagats. Especialment destacables són les puntuacions de les Preguntes 2, 6 i 10, que han obtingut la màxima puntuació de 5.0, indicant una resposta impecable en tots els criteris.

Algunes preguntes com la 3, 5 i 8 han obtingut puntuacions inferiors, suggerint àrees de millora especialment en completitud i correctitud factual. Aquests resultats proporcionen una guia clara per enfocar esforços en l'optimització del sistema.

Taula 4: Resultats de Puntuació per "FRAUDES" mitjançant LLM

Pregunta	Correctitud Factual	Rellevància	Completitud	Claredat	Puntuació Total
Pregunta 1	5	5	5	5	5.0
Pregunta 2	4	4	4	4	4.0
Pregunta 3	3	3	3	3	3.0
Pregunta 4	4	5	4	5	4.5
Pregunta 5	3	4	3	4	3.5
Pregunta 6	5	5	5	5	5.0
Pregunta 7	4	4	4	4	4.0
Pregunta 8	3	3	3	3	3.0
Pregunta 9	5	5	5	5	5.0
Pregunta 10	4	5	4	5	4.5

En l'àmbit de fraus, les respostes del sistema RAG també han demostrat un alt nivell de qualitat, amb puntuacions màximes en les Preguntes 1, 6 i 9. Això indica que el sistema és capaç de proporcionar respostes altament precises i rellevants en temes crítics.

Les puntuacions més baixes en les Preguntes 3, 5 i 8 apunten a necessitats de millora similars a les identificades en l'àmbit d'impagats, especialment en termes de completitud i correctitud factual. Aquestes observacions seran essencials per orientar futures millores del sistema.

5.4.4. Comparació amb l'avaluació humana

L'objectiu d'aquesta comparació és identificar les discrepàncies i coincidències entre les puntuacions atorgades pel LLM i les valoracions humanes, i així avaluar si el LLM pot servir com una alternativa viable o complementària a l'avaluació tradicional. En aquest context, es va demanar al LLM que puntués una sèrie de respostes generades pel RAG segons quatre criteris: Correctitud Factual, Rellevància, Completitud i Claredat, amb una escala de puntuació de 0 a 5. Aquestes mateixes respostes van ser avaluades de manera independent per un grup d'experts humans seguint els mateixos criteris.

Els resultats obtinguts mostren una correlació significativa entre les puntuacions atorgades pel LLM i les assignades pels humans. En la majoria dels casos, el LLM va ser capaç d'identificar i puntuar amb precisió les respostes que els experts humans també van considerar de qualitat alta, particularment en els criteris de Rellevància i Claredat, on les puntuacions tendeixen a ser molt properes. Això demostra que el LLM pot captar amb eficàcia els aspectes més evidents de la qualitat d'una resposta, com ara si la resposta és pertinent i ben estructurada.

No obstant això, també es van detectar algunes discrepàncies, especialment en els criteris de Correctitud Factual i Completitud. En aquests casos, el LLM va mostrar algunes limitacions per reconèixer matisos més subtils o errors menors que els experts humans van considerar importants. Això és especialment evident en respostes que requerien una comprensió més profunda del context o que incloïen detalls factuales que el LLM no va identificar amb la mateixa precisió que els humans. Aquestes diferències subratllen la necessitat de mantenir una supervisió humana en l'avaluació de respostes en àmbits crítics, on la precisió factual i la completitud són crucials. Tot i que el LLM pot oferir una solució ràpida i consistent per a l'avaluació de grans volums de dades, la seva capacitat per substituir completament l'avaluació humana encara és limitada en certs aspectes.

En conclusió, la comparació amb l'avaluació humana demostra que els LLMs, com el GPT-4, són eines poderoses per automatitzar i agilitzar el procés d'avaluació, especialment en criteris més superficials com la rellevància i la claredat. Tanmateix, per

garantir la màxima qualitat en la validació de respostes, especialment en contextos on la precisió factual és crítica, és recomanable combinar l'ús de LLMs amb una revisió humana per obtenir els millors resultats possibles. Aquesta col·laboració entre l'avaluació automàtica i humana ofereix un enfocament equilibrat que maximitza tant l'eficiència com la fiabilitat.

6. Conclusions i futures línies de treball

Aquest projecte ha explorat amb profunditat la creació, implementació i avaluació d'un sistema de Generació Augmentada per Recuperació (RAG) aplicat al software AquaCIS CF, utilitzat per VEOLIA en la gestió d'aigües. Les conclusions que he pogut extreure d'aquest treball abasten diversos aspectes tècnics, metodològics i pràctics, així com les implicacions que aquests tenen tant a nivell d'organització com en el camp de la intel·ligència artificial.

La creació del RAG ha suposat un repte tecnològic significatiu, que ha requerit la integració de diverses tècniques d'intel·ligència artificial, com ara els models de llenguatge i la IA generativa. Aquests models han estat essencials per permetre que el sistema pugui comprendre i processar les consultes relacionades amb impagaments i fraus, generant respostes que siguin no només rellevants, sinó també factualment correctes i coherents. En el desenvolupament del sistema, ha estat fonamental la selecció acurada dels algoritmes i models utilitzats. L'ús de models de llenguatge avançats com GPT-4 ha proporcionat una base sòlida per a la generació de respostes contextualitzades, mentre que les tècniques de recuperació de la informació han permès que el sistema accedeixi i recuperi dades específiques del domini d'impagaments i fraus de manera eficient.

La validació del RAG s'ha dut a terme mitjançant una avaluació exhaustiva basada en diverses mètriques establertes en la literatura, com la distància cosinus, la distància euclidiana, i les puntuacions ROUGE. Tot seguint un workflow creat específicament per avaluar el rendiment d'aquest RAG en específic. Aquestes mètriques han estat seleccionades per la seva capacitat de mesurar la similitud i la qualitat de les respostes generades pel RAG en comparació amb les respostes humanes revisades. Els resultats de les avaluacions han mostrat una millora significativa en la qualitat de les respostes després d'aplicar un ajustament fi dels models, que va implicar l'ús d'un prompt específic per a les consultes. Això ha permès que el sistema RAG esdevingui una eina poderosa, capaç de generar respostes que són gairebé indistinguibles de les respostes humanes en termes de precisió, rellevància i coherència.

El desplegament del sistema RAG en l'entorn de VEOLIA representa un avenç significatiu en la gestió de la informació relacionada amb impagaments i fraus. Aquestes àrees són crítiques per a l'operativa de VEOLIA, i l'automatització de la recuperació de la informació permetrà als empleats accedir de manera ràpida i eficient a dades essencials,

millorant així la presa de decisions i la gestió dels recursos. A més, el RAG proporciona un avantatge competitiu a VEOLIA en el sector de la gestió d'aigües, ja que permet a l'organització mantenir un nivell alt de control sobre les seves operacions financeres i de seguretat. La implementació d'aquest sistema també obre la porta a futures integracions amb altres àrees del software AquaCIS CF, ampliant així les capacitats de l'eina per abastar més àrees crítiques de la gestió empresarial.

Durant el desenvolupament del projecte, s'han trobat diversos reptes tècnics i metodològics. Un dels reptes principals ha estat la gestió de la gran quantitat de dades disponibles i la seva adequació per a ser utilitzada en l'entrenament dels models. Aquesta fase ha requerit una acurada normalització de les dades per assegurar-se que només la informació més rellevant i precisa fos utilitzada, cosa que ha tingut un impacte directe en la qualitat de les respostes generades. Un altre repte ha estat l'avaluació de les respostes generades pel RAG, especialment en la comparació amb les respostes humanes. La interpretació de les mètriques de similitud, com la distància cosinus i les puntuacions ROUGE, ha requerit un anàlisi detallat per garantir que els resultats fossin fiables i representatius de la qualitat real del sistema. A nivell metodològic, la necessitat de mantenir un equilibri entre la complexitat del model i la seva usabilitat en un entorn empresarial real ha estat un altre desafiament. És essencial que el RAG sigui fàcil d'utilitzar pels empleats de VEOLIA, cosa que ha condicionat la selecció dels models i l'arquitectura del sistema.

El projecte ha demostrat que la integració de RAG en sistemes empresarials com AquaCIS CF té un gran potencial per transformar la manera com es gestiona i es recupera la informació. Això obre la porta a futures investigacions i desenvolupaments, no només en l'àmbit de la gestió d'aigües, sinó també en altres sectors que requereixen la gestió eficient de grans volums d'informació. En el futur, seria interessant explorar la possibilitat de combinar el RAG amb altres tecnologies emergents, com ara la intel·ligència artificial explicativa (XAI) per millorar la transparència del sistema, o la integració de tècniques d'aprenentatge automàtic per a la millora contínua del model a mesura que es recopilen noves dades. També es podrien investigar noves mètriques d'avaluació que siguin més sensibles a les subtileses del llenguatge, així com la creació de bases de coneixement encara més especialitzades que permetin al RAG oferir respostes encara més precises i detallades.

En resum, aquest projecte ha aconseguit no només desenvolupar i avaluar un sistema RAG efectiu per a VEOLIA, sinó que també ha demostrat la viabilitat i els avantatges de la implementació de sistemes de generació augmentada per recuperació en entorns empresarials. Els resultats obtinguts mostren un camí clar cap a la millora contínua de les eines de suport a la decisió, establint una base sòlida per a futurs desenvolupaments i aplicacions. Amb la seva implementació, VEOLIA està millor equipada per afrontar els desafiaments de la gestió d'informació crítica en el seu sector, consolidant la seva posició com a líder en la gestió de recursos a nivell global.

7. Bibliografia

1. Amanat, M. U. (2023). LLM evaluation with ROUGE. Medium. https://medium.com/@MUmarAmanat/llm-evaluation-with-rouge-0ebf6cf2aed4

2. Bhagyashree. (2023). 2: Deep dive on evaluation of large language models (LLMs). Medium. https://medium.com/@bhagyashree01041994/2-deep-dive-on-evaluation-of-large-language-models-llms-a3bf2929c400

3. CUEMATH. (s.f.). Euclidean distance formula. CUEMATH. https://www.cuemath.com/euclidean-distance-formula/

4. Milana Shxanukova, M. (2023). Cosine distance and cosine similarity. Medium. https://medium.com/@milana.shxanukova15/cosine-distance-and-cosine-similarity-a5da0e4d9ded

5. Ollama. (2023). Embedding models. Ollama. https://ollama.com/blog/embedding-models

6. OpenAI. (2023). Embeddings. OpenAI. https://platform.openai.com/docs/guides/embeddings

7. Planet, N. (2023). Two minutes NLP: Learn the ROUGE metric by examples. Medium. https://medium.com/nlplanet/two-minutes-nlp-learn-the-rouge-metric-by-examples-f179cc285499

8. Torres, J. (2023). Conceptos de inteligencia artificial: qué son las GANs (Redes Generativas Antagonistas). Xataka. https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-gans-redes-generativas-antagonicas

9. Wikipedia. (2023). ROUGE (métrica). Wikipedia. https://es.wikipedia.org/wiki/ROUGE_(métrica)

10. Wikipedia. (2023). Variational autoencoder. Wikipedia. https://en.wikipedia.org/wiki/Variational_autoencoder

11. AWS. (2023). Vector databases. Amazon Web Services. https://aws.amazon.com/es/what-is/vector-databases/

12. AB Software. (s.f.). Portfolio. AB Software. https://www.absoftware.es/portfolio/

13. Data Science at Microsoft. (2023). Evaluating LLM systems: Metrics, challenges, and best practices. Medium. https://medium.com/data-science-at-microsoft/evaluating-llm-systems-metrics-challenges-and-best-practices-664ac25be7e5

14. Towards AI. (2023). Why RAG applications fail in production: A technical deep dive. Medium. https://medium.com/towards-artificial-intelligence/why-rag-applications-fail-in-production-a-technical-deep-dive-15cc976af52c