

Alexia Vidal Pons

INTERACCIÓ INTEL·LIGENT AMB DOCUMENTS

TREBALL DE FI DE GRAU

dirigit per el Sr. Esteban Herreros

Grau d'Enginyeria Informàtica



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2024

Resum

Les empreses sovint enfronten dificultats en la gestió d'un volum gran de documents emmagatzemats tant localment com en plataformes de núvol. Aquests documents, que poden ser en formats diversos com PDF, Excel, Word o PowerPoint, poden arribar a contenir molta informació de diferents àmbits, cosa que fa complicada la seva organització, accés o localització.

Aquest projecte aplica tecnologies d'intel·ligència artificial (IA) per simplificar aquesta gestió de documents. Utilitzant bases de dades vectorials (DeepLake), models de llenguatge de gran escala (GPT d'OpenAI), i *frameworks* com Streamlit i LangChain s'ha desenvolupat una interfície de xat que permet als usuaris interactuar amb els documents carregats al núvol, principalment en Google Drive.

La solució innovadora que es proposa facilita la recerca d'informació dins dels fitxers, permetent als usuaris estalviar temps i millorar l'eficiència en les seves tasques. La interfície de xat intuïtiva contribueix a un aprenentatge ràpid i eficaç de la tecnologia per part dels usuaris.

Aquest projecte demostra l'èxit de l'aplicació de la IA en la millora de la gestió de documents en l'àmbit d'empreses. Aquesta aproximació innovadora transforma la manera en què les organitzacions accedeixen i fan servir la informació, obrint camí per a futures aplicacions de tecnologies similars en altres àrees de negoci.

Resumen

Las empresas a menudo enfrentan dificultades en la gestión de un gran volumen de documentos almacenados tanto localmente como en plataformas en la nube. Estos documentos, que pueden estar en varios formatos como PDF, Excel, Word o PowerPoint, pueden llegar a contener información de diferentes ámbitos, lo que complica su organización, acceso y localización.

Este proyecto aplica tecnologías de inteligencia artificial (IA) para simplificar esta gestión documental. Utilizando bases de datos vectoriales (DeepLake), modelos de lenguaje a gran escala (GPT de OpenAI) y *frameworks* como Streamlit y LangChain, se ha desarrollado una interfaz de chat que permite a los usuarios interactuar con los documentos cargados en la nube, principalmente en Google Drive.

La solución innovadora propuesta facilita la búsqueda de información dentro de los documentos, permitiendo a los usuarios ahorrar tiempo y mejorar la eficiencia en sus tareas. La interfaz de chat intuitiva contribuye a un aprendizaje rápido y efectivo de la tecnología por parte de los usuarios.

Este proyecto demuestra el éxito de la aplicación de la inteligencia artificial en la mejora de la gestión de documentos en el ámbito empresarial. Este enfoque innovador transforma la forma en que las organizaciones acceden y utilizan la información, abriendo camino para futuras aplicaciones de tecnologías similares en otras áreas de negocio.

Abstract

Nowadays, companies oftentimes face challenges in managing a large volume of documents stored both locally and on cloud solutions. These documents, which can be in various formats like PDF, Excel, Word, or PowerPoint, contain information from different fields, making their organization and access complex.

This project applies artificial intelligence (AI) technologies to simplify this document management. Using vector databases (DeepLake), large-scale language models (GPT from OpenAI), and frameworks like Streamlit and LangChain, a chat interface has been developed that allows users to interact with documents loaded in the cloud, mainly on Google Drive.

The innovative solution proposed here facilitates the search for information within documents, allowing users to save time and improve efficiency in their tasks. The intuitive chat interface contributes to a rapid and effective learning of the technology by users.

This project demonstrates the success of applying artificial intelligence in improving document management in the business sector. This innovative approach transforms how organizations access and use information, paving the way for future applications of similar technologies in other business areas.

ÍNDEX

RESUM.....	I
RESUMEN	II
ABSTRACT	III
1. INTRODUCCIÓ.....	5
1.1. VISIÓ GENERAL	5
1.2. OBJECTIUS DEL PROJECTE	5
2. DESCRIPCIÓ GENERAL DEL PROJECTE.....	6
2.1. INTRODUCCIÓ A L'ENTORN.....	6
2.1.1. <i>MODEL DE LLENGUATGE A GRAN ESCALA</i>	7
2.1.2. <i>PROMPT ENGINEERING</i>	11
2.1.3. <i>BASES DE DADES VECTORIALS I EMBEDDINGS</i>	12
2.2. ENTORN DE DESENVOLUPAMENT	13
2.3. PREVISIONS D'ÚS.....	16
3. REQUISITS	16
3.1. REQUISITS FUNCIONALS	16
3.2. REQUISITS NO FUNCIONALS.....	16
4. DISSENY	17
4.1. ARQUITECTURA DE L'APLICACIÓ.....	17
4.2. DISSENY INTERFÍCIE GRÀFICA	18
4.2.1. <i>FRAMEWORK STREAMLIT</i>	18
4.2.2. <i>APARTATS DE LA INTERFÍCIE</i>	18
5. IMPLEMENTACIÓ.....	21
5.1. DETALLS DE LA TECNOLOGIA	21
5.2. ALGORITMES ESPECÍFICS.....	28
6. AVALUACIÓ.....	28
7. CONCLUSIÓ.....	32
8. BIBLIOGRAFIA	33
9. ANNEX.....	37

ÍNDIX DE FIGURES

FIGURA 1. DADES D'ENTRENAMENT I APLICACIONS DELS LLM. IMATGE EXTRETA DE [42].	7
FIGURA 2. ARQUITECTURA BASADA EN TRANSFORMADORS QUE SEGUEIX EL GPT-2. IMATGE EXTRETA DE [43].	8
FIGURA 3. FUNCIONAMENT D'UN PROMPT AMB UN LLM.	11
FIGURA 4. CONVERSIÓ DELS TEXT A EMBEDDINGS I EMMAGATZEMATGE A LES BASES DE DADES VECTORIALS.	12
FIGURA 5. DIAGRAMA DELS ELEMENTS NECESSARIS PARA EL PROJECTE I COM SE RELACIONARIEN.	13
FIGURA 6. DIAGRAMA DE L'ARQUITECTURA DE L'APLICACIÓ.....	17
FIGURA 7. DIFERENTS ESTATS EN ELS QUE ES POT TROBAR EL SIDEBAR DE L'APLICACIÓ. ...	19
FIGURA 9. VISUALITZACIÓ DE L'ESPAI DE XAT MENTRE ES TRACTEN ELS DOCUMENTS.....	20
FIGURA 8. ESTAT INICIAL DEL XAT DE L'APLICACIÓ.	20
FIGURA 10. AVATARS QUE ACOMPANYEN ELS MISSATGES.	21
FIGURA 11. VISUALITZACIÓ UN COP ES PERMET FER PREGUNTES I COM ES VEU MENTRE CONTESTEN.....	21
FIGURA 12. GRÀFIC COMPARATIU DE NOTES I PREUS FET PERSONALMENT PER AVALUAR EL MODEL GPT-4 I EL GPT-3.5-TURBO-16K.	23
FIGURA 13. EL FUNCIONAMENT DE LES VARIABLES <code>CHUNK_SIZE</code> I <code>CHUNK_OVERLAP</code> EN LA FRAGMENTACIÓ DE DOCUMENTS.	26
FIGURA 14. MISSATGE DEL SISTEMA AL XAT QUAN ES CARREGUEN NOUS DOCUMENTS	31
FIGURA 15. MISSATGE D'ERROR EN CAS D'INTRODUIR L'OPENAI KEY INCORRECTA.	32

ÍNDIX DE CODIS

CODI 1. CODI PER AUTENTIFICAR-SE I CREAR UNA CONNEXIÓ AMB L'API DE GOOGLE DRIVE	24
CODI 2. FRAGMENT DE CODI LA FUNCIO DOWNLOAD_FILES QUE S'ENCARREGA DE LA DESCARREGA DE DOCUMENTS	25
CODI 3. DEFINICIÓ DE L'ESTRUCTURA DEL XAT I LA CRIDA PER ACONSEGUIR UNA RESPOSTA A "QUERY".....	27
CODI 4. LÍNIES AFEGIDES A LA FUNCIO LOAD_DOC PER SABER SI UN DOCUMENT HA ESTAT CARREGAT.	28

ÍNDEX DE TAULES

TAULA 1. TAULA COMPARATIVA DELS PREUS DELS TOKENS DELS MODELS D'OPENAI I EL MÀXIM QUE ACCEPTEN.	22
TAULA 2. TAULA AMB POSSIBLES PREGUNTES, LES SEVES CORRESPONENTS RESPOSTES I EL CONTINGUT QUE HAURIA DE TENIR LA RESPOSTA PER CONSIDERAR-SE CORRECTA.	28

1. Introducció

1.1. Visió general

La intel·ligència artificial (IA) es troba cada cop més en la nostra vida diària, oferint una àmplia varietat d'utilitats per facilitar tasques quotidianes. Amb l'objectiu d'aprofitar la gran quantitat de capacitats de la IA i afrontar el problema que suposa la gran quantitat de documents digitals que solen tenir les empreses, s'ha proposat una solució per gestionar tot aquest enrenou de documents, moltes vegades desordenats, amb noms no adequats i difícils de trobar. Així com per facilitar la tasca de buscar el document, també es vol facilitar la tasca d'entendre, ja que dins d'una mateixa empresa no tots els membres tenen els mateixos coneixements; un desenvolupador no té els mateixos que un de vendes, i es busca que sigui més fàcil entendre allò que s'està llegint, ja sigui una factura, un pressupost, un catàleg, un prototip, etc. Amb aquesta solució s'aconseguiria estalviar temps i treballar de manera més ràpida i eficient.

Per aconseguir-ho, volem oferir una interfície que utilitzi tecnologies emergents, com és el cas de la IA generativa, per poder interactuar directament amb els documents i preguntar-los directament allò que es busca o es vol saber, millorant la manera com els usuaris es relacionen amb la seva informació escrita. L'aplicació que hem volgut desenvolupar està orientada a les empreses, i en la gran majoria d'entorns professionals on es treballa amb documents digitals, aquests solen estar compartits i gestionats en alguna plataforma al núvol, com ara pot ser el cas de Google Drive o Dropbox. Tenint en compte aquesta característica, hem desenvolupat la solució per accedir a documents carregats en una carpeta al núvol.

Aquest projecte busca presentar-se com una solució innovadora per proporcionar una interfície de conversa amb documents, permetent als usuaris interactuar amb les seves dades de forma més intuïtiva i pràctica.

1.2. Objectius del projecte

Els objectius a l'hora de realitzar aquest projecte eren ampliar el coneixement sobre noves eines innovadores i en apogeu, com és la IA generativa, millorant la productivitat en la manipulació de documents a través d'aquesta interfície conversacional i intuïtiva. En aquesta interfície, els usuaris podran dur a terme consultes, extraccions i altres tasques relacionades amb els documents de manera més natural i sense la necessitat de conèixer detalladament on trobar-ho. A més de la IA, que ja de per si és una tecnologia molt potent, s'ha buscat complementar-ho amb noves eines com el processament de llenguatge natural utilitzant LLM (*Large Language Models*) per poder oferir respostes contextuais i precises. També s'ha incorporat l'ús de tecnologies de vectorització i emmagatzematge eficients per gestionar grans volums de dades, ajudant a indexar els continguts i facilitar la recuperació d'informació específica. Així mateix, es fa ús de llibreries LangChain, que ofereixen un marc per al desenvolupament amb cadenes, agents i estratègies de recuperació que conformen l'arquitectura cognitiva d'una aplicació.

2. Descripció general del projecte

2.1. Introducció a l'entorn

Per desenvolupar el projecte, s'han utilitzat diverses eines. En primer lloc, per saber quines escollir d'entre la gran varietat existent, es va fer una investigació prèvia sobre la IA, més concretament la IA generativa. La IA correspon a un camp de la informàtica que es dedica a la creació de sistemes i programes encarregats d'imitar la intel·ligència humana. Aquests sistemes poden executar tasques complexes com el raonament, la planificació, la percepció, l'aprenentatge, la presa de decisions, la interacció lingüística o resolució de problemes de manera autònoma [1]. La IA usa algorismes i models matemàtics per processar les dades i generar resultats intel·ligents. Els sistemes de IA actualment es poden trobar en diversos camps com en economia, medicina, enginyeria, o inclús, en la milícia [2].

L'arquitectura i els processos pels quals les IA aprenen, es milloren i implementen en alguna àrea d'interès segons la utilitat que se li vulgui donar. Entre els mètodes d'aprenentatge existeixen els següents tipus:

- **Aprenentatge Supervisat:** Aquest mètode fa servir dades d'entrada i sortida etiquetades per entrenar els seus algorismes. L'objectiu és aprendre la relació que existeix entre les entrades i les sortides per fer prediccions sobre noves dades [3].
- **Aprenentatge No Supervisat:** En aquest cas, la IA treballa amb dades no etiquetades i busca trobar patrons o estructures en aquestes dades per si mateix. No requereix una supervisió externa per aprendre [3].
- **Aprenentatge Semisupervisat:** Aquesta és una combinació de l'aprenentatge supervisat i no supervisat. Fa ús de dades etiquetades i no etiquetades per entrenar els seus algorismes i millorar el seu rendiment [3].
- **Aprenentatge per Reforç:** Aquest mètode implica que la IA aprèn a través d'observar les interaccions en un entorn. La IA pren accions i rep una resposta de tornada o penalitzacions basades en el seu rendiment. Amb l'objectiu de maximitzar les recompenses obtingudes, va ajustant el seu comportament [3].

Entre alguna de les IA més conegudes i usades en l'actualitat trobem assistents virtuals com Alexa o Siri, traductors automàtics com el de Google i DeepL, sistemes de recomanació com el de YouTube o xatbots com ChatGPT [2]. El desenvolupament d'aquestes ha comportat una gran inversió, tant monetària com de temps, ja que són processos molt costosos si es vol que donin els millors resultats.

Per aquest projecte el tipus de IA que més ens interessa és la IA generativa, aquesta fa servir algorismes d'aprenentatge no supervisat o semisupervisat per crear contingut nou i original [4]. És utilitzada principalment utilitzada per crear o dur a terme tasques que tradicionalment només els humans eren capaços de fer, el seu ús ha estalviat temps i diners a les empreses amb la seva aplicació. Entre les tasques capaces de realitzar la IA generativa trobem la redacció i generació de text, la simplificació de continguts, la creació d'imatges i vídeos, la composició musical, entre altres.

2.1.1. Model de llenguatge a gran escala

Un model de llenguatge de gran escala o *Large Language Model* (LLM, per les seves sigles en anglès) es refereix a una classe de models d'intel·ligència artificial generativa que s'entrenen amb grans quantitats de dades textuais, com articles, llibres, llocs web, i moltes altres formes de contingut escrit, per entendre i generar llenguatge semblant a l'humà. Aquests models utilitzen algorismes d'aprenentatge profund o *Deep Learning* per processar i entendre patrons de llenguatge natural, caracteritzant-se per la seva capacitat de probabilitat i aleatorietat. Analitzen relacions estadístiques entre paraules, expressions i frases, creant així xarxes neuronals que els ajuden a predir la següent paraula d'una frase. La principal diferència entre els models de llenguatge simples i els LLM és la quantitat de dades amb les quals són entrenats, arribant a crear xarxes neuronals de dimensions molt majors, fins i tot de trilions de nodes [5]. La seva capacitat d'entendre i generar text de manera similar a com ho faria una persona els converteix en eines valuoses per a diverses aplicacions com generació de text, traduccions, assistents virtuals, generació de resums, sistemes de preguntes i respostes, entre d'altres [6].

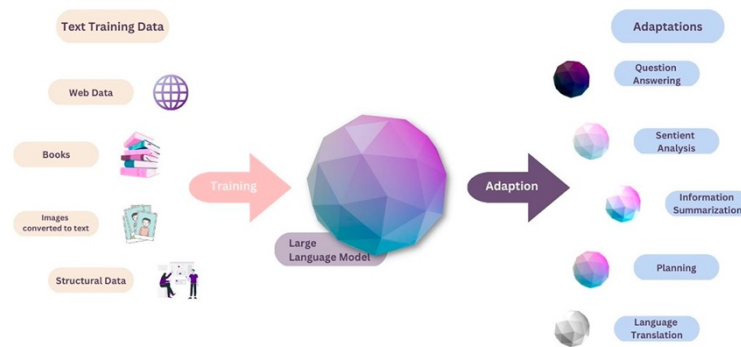


Figura 1. Dades d'entrenament i aplicacions dels LLM. Imatge extreta de [42].

Per escollir quin LLM és l'indicat per al nostre projecte, s'han contemplat característiques com l'arquitectura dels models, les dades d'entrenament utilitzades, el seu funcionament, la precisió en les respostes, la capacitat d'afinació i personalització, així com consideracions ètiques. Cada model pot tenir diferents arquitectures, cadascuna amb els seus punts forts i febles:

- Model basat en auto-codificadors: Aquests models codifiquen el text d'entrada en una representació de baixa dimensió, generant text nou basant-se en aquesta representació. Són especialment eficaços en tasques com resumir o generar contingut [7].
- Model seqüència-a-seqüència: Són models que prenen una seqüència d'entrada, com una frase, i generen una seqüència de sortida, com ara una traducció a una altra llengua. S'utilitzen sovint en traducció automàtica i resum de text [7].
- Model basat en transformadors: Actualment, són els models més populars, tenen una arquitectura de xarxa neural excel·lent per entendre dependències a llarg termini entre dades textuais. Són útils per una gran varietat de tasques de llenguatge, com ara generar text, fer traduccions i respondre preguntes [7].

- Models basats en xarxes neuronals recursives: Dissenyats per tractar dades estructurades, com ara arbres de descomposició, són útils per analitzar sentiments i la inferència del llenguatge natural [7].
- Models jeràrquics: Dissenyats per gestionar text a diferents nivells, com frases, paràgrafs i documents, aquests models es fan servir en tasques de classificació de documents [7].

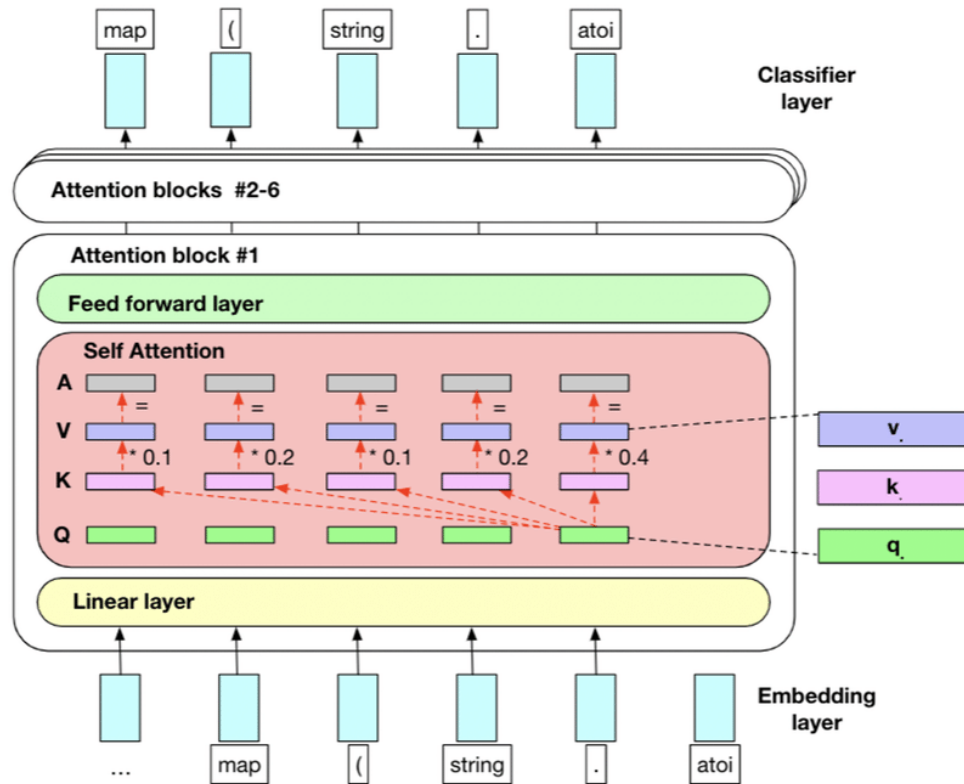


Figura 2. Arquitectura basada en transformadors que segueix el GPT-2. Imatge extreta de [43].

L'arquitectura més utilitzada en els LLM més coneguts és la del model basat en transformadors. Aquest model té quatre passos principals (Figura 2), els quals se segueixen cada cop que s'introdueixen dades d'entrada:

1. El model realitza un embeddings les paraules. Converteix les paraules en representacions vectorials d'alta dimensió on paraules similars són agrupades de manera més propera. Aquest ajuda que el model entengui el significat de les paraules i pugui fer prediccions basades en aquest coneixement.
2. Es fa una codificació posicional (Linear Layer). Aquest procés ajuda el model a entendre on es troben les paraules en una seqüència. No es tracta del significat de la paraula, sinó que se centra a fer un seguiment de l'ordre de les paraules. Poder entendre l'ordre de les paraules és molt important per a tasques com la traducció, resums o respondre preguntes.
3. Després, les dades passen a través d'unes quantes capes de transformadors. Cadascuna d'aquestes capes serveixen per millorar l'habilitat del LLM per modelar un ampli rang de dependències en el llenguatge natural. Aquesta treballa

processant la seqüència d'entrada en paral·lel, en lloc de fer-ho de forma seqüencial. Aquesta està conformada per dos components essencials: el mecanisme d'auto-atenció (*self-attention*) i la xarxa neuronal de *feedforward*. La primera assigna un pes a la paraula segons que valuosa és per la futura predicció. La segona, rep la sortida del primer component, i processa independentment cada posició passant la seva representació vectorial per diverses capes connectades generant noves representacions. Tot aquest procés serveix per poder aprendre patrons i relacions complexes entre paraules.

4. Finalment, un cop el LLM ha estat entrenat i refinat és capaç de generar respostes de text sofisticades a preguntes o prompts. El model utilitza els patrons apresos per generar respostes coherents i contextualitzades a partir de l'entrada donada.

És important també entendre alguns conceptes de com els models tracten o interpreten les dades d'entrada, un dels més importants, és el de token. Els LLM processen el text en fragments que anomenem tokens. Aquests poden ser des d'un sol caràcter fins a una paraula sencera en anglès, i poden ser encara més curts o més llargs en altres idiomes. Per exemple, la frase "ChatGPT is great!" es codifica en sis tokens com ["Chat", "G", "PT", " is", " great", "!"] [8]. En alguns dels models investigats aquest concepte és important, ja que la quantitat d'aquests pot arribar a afectar el funcionament d'aquests.

Empreses de renom com Meta, Google o OpenAI han desenvolupat els seus propis LLM. Examinem a continuació algunes d'aquestes contribucions:

- BERT (*Bidirectional Encoder Representations from Transformers*): Desenvolupat per Google AI l'any 2018, es troba entre els LLM més influents que ha aportat avenços significatius en la comprensió del llenguatge natural. BERT utilitza una arquitectura bidireccional de transformador, introduint el concepte d'entrenament bidireccional, permetent que el model consideri el context dret i esquerre d'una paraula, donant lloc a una millor comprensió del context i la relació entre paraules. Va ser entrenat amb 3,3 bilions de paraules, extretes de Wikipedia i BookCorpus de Google per ser capaç de dur dues tasques simultàniament: modelatge del llenguatge i predicció de la pròxima frase. En l'actualitat, BERT ajuda a millorar els resultats de cerques de Google des de novembre de 2020 [9].
- RoBERTa (*Robustly Optimized BERT Approach*): Desenvolupat per Meta AI l'any 2019, consisteix en una versió de BERT millorada després d'observar l'èxit d'aquest. Les optimitzacions busquen millorar el procés de pre-entrenament eliminant la tasca de predicció de la pròxima frase i usant un conjunt de dades major amb més iteracions. L'arquitectura i les fonts de les dades s'han mantingut. S'ha convertit en una eina per la classificació i generació de text, es tracta d'un sistema capaç d'analitzar grans quantitats de text no estructurat i extraure automàticament informació valuosa [10].
- T5 (*Text-to-text Transfer Transformer*): Creat per Google AI l'any 2019, és un LLM versàtil que és capaç de dur a terme tasques de llenguatge natural com un

problema de text a text. Pot dur a terme tasques tractant-les de manera uniforme com a tasques de generació de text, donant lloc a resultats consistents i impressionants en diversos àmbits. T5 es basa en una arquitectura de transformador i va ser entrenat en l'objectiu d'omplir el buit, és a dir, per predir les paraules faltant en un fragment danyat de text. Té un ampli ventall d'aplicacions un sistema que pot traduir no sols entre idiomes, sinó també entre diferents formats de text, com resums, preguntes i respostes, o inclús codi de programació [5].

- GPT-3 (*Generative Pre-Trained Transformer-3*): Desenvolupat per OpenAI, una de les empreses capdavanteres en l'àmbit de la intel·ligència artificial, l'any 2020. És un dels LLM més grans que existeix amb 175 bilions de paràmetres. Tot i tenir una estructura com la dels seus predecessors (GPT-2), una arquitectura basada en transformadors, millora la seva precisió s'atribueix a la seva capacitat i dades d'entrenament format per uns 500 bilions de paraules extretes de Common Crawl¹, WebText2, Books1, Books2 i Wikipedia fins al 2019. Aquest model és capaç de generar text coherent i rellevant en una gran varietat de contexts, ja siguin científics, literaris, etc. La seva capacitat per entendre i produir llenguatge natural l'ha portat a diverses aplicacions, des d'assistents virtuals que no sols responen a preguntes, sinó que també et fa sentir com si estiguessis parlant en un ésser humà, fet que el converteix en una fita important en la IA conversacional. La subclasse de GPT-3.5 creada l'any 2022, són una millora més capaç que les versions prèvies i entrenades amb dades de fins al juny de 2021, és aquest mateix que s'encarrega de donar respostes en el famós ChatGPT. Tot i ser de la mateixa sèrie, l'empresa no inclou GPT-3.5 dintre del GPT-3, de fet el GPT-3 a la pàgina de l'empresa apareix com a obsolet [11].
- LLaMA (*Large Language Model Meta AI*): creat per Meta AI a principis de febrer de 2023. Segueix una arquitectura basada en transformadors. Existeixen dos models, el LLaMA 1 va ser entrenat amb 1,4 trilions de paraules trobades en fonts de domini públic com són pàgines web, repositoris de GitHub, Wikipedia, llibres, articles científics, etc. En canvi, LLaMA 2 es va entrenar amb 2 trilions amb dades que exclouen pàgines web que contenen dades personals i va augmentar les fonts que considerava fiables [12].
- GPT-4 (*Generative Pre-Trained Transformer-4*): també desenvolupat per OpenAI i disponible a partir del març de 2023, és la següent generació de GPT. Aquest model sobrepasa els seus predecessors tenint fins a 100 trilions de paràmetres superant els 175 bilions del GPT-3, seguint també una arquitectura basada en transformadors. El principal avantatge, similar al GPT-3, es troba en el pre-entrenament compost per un enorme conjunt de dades textuales, permetent-li aprendre una gran varietat de característiques i relacions lingüístiques. Com a resultat, el GPT-4 pot ser utilitzat per a tasques específiques del processament del llenguatge natural convertint-lo en una excepcionalment eficient i versàtil. Per

¹ Common Crawl: organització sense ànim de lucre que rastreja la web i proporciona lliurement els seus arxius i conjunts de dades al públic. El seu arxiu web consta de pentabytes de dades recollides des de 2011.

poder fer una comparativa de les capacitats s'ha de considerar el fet que el GPT-4 és 500 cops més potent que el GPT-3 [8], [13]. Una de les innovacions més cridaneres és la capacitat de processar imatges igual que text, generant respostes basades en una combinació d'informació visual i textual. Entre les aplicacions trobem que es pot usar en la plataforma ChatGPT, però pagant una subscripció mensual. Les seves capacitats més avançades permeten que generi certs continguts que els predecessors erraven o no eren gaire precisos com esborranys de demandes, aprovar exàmens estandarditzats, construir pàgines web a partir d'un esbós i més. Certs detalls tècnics com les dades d'entrenament, els costos d'energia o especificacions de maquinari no s'han publicat completament per l'empresa per por als competidors en el mercat de la IA [14].

Cada model té les seves particularitats i aplicacions, des de la millora dels resultats de cerca fins a l'assistència virtual avançada, demostrant l'abast i la versatilitat de la IA generativa en el camp del processament del llenguatge natural.

Les recerques actuals en els models de llenguatge de gran escala (LLM) suggereixen que aquests continuaran millorant en la seva capacitat per entendre i respondre als humans. S'espera que aviat siguin tan eficients que podran ser utilitzats en gairebé qualsevol dispositiu, inclosos els telèfons mòbils. A més, es preveu que aquests models es convertiran en experts en àrees específiques com la medicina o les lleis, i no només treballaran amb text, sinó que també podran processar imatges i sons.

2.1.2. *Prompt Engineering*

D'altra banda, en l'àmbit de la IA generativa, trobem el concepte emergent de *Prompt Engineering*. Però primer saber que un prompt és una instrucció o text inicial que es proporciona a una eina de IA generativa, per tal de guiar la seva generació de respostes o resultats. Aquest pot ser una frase, una pregunta o una petició escrita per aconseguir una contestació específica [15].

La disciplina de *Prompt Engineering*, relativament nova, se centra a refinar i dissenyar prompts per fer més eficients els models de llenguatge, obtenint així els resultats desitjats. Consisteix a crear instruccions o consultes específiques que guien el comportament del LLM, dirigint les seves respostes sense necessitat d'actualitzar els pesos del model [16]. L'habilitat de maximitzar l'eficiència del model mitjançant prompts ben estructurats i clars pot guiar al LLM per produir respostes precises i rellevants [17].



Figura 3. Funcionament d'un prompt amb un LLM.

El *Prompt Engineering* pot millorar el rendiment en tasques específiques, proporcionant respostes més matisades i conscients del context, com en els casos de traducció d'idiomes, anàlisi de sentiments o generació de text. A més, és una eina valuosa per comprendre les limitacions del model. Mitjançant un refinament iteratiu dels prompts i l'anàlisi de les respostes, podem identificar les fortaleses i debilitats dels models [18]. Aquesta tècnica també augmenta la seguretat dels LLM, una preocupació important en aplicacions públiques. Un prompt mal dissenyat pot conduir a la generació de respostes amb contingut inadequat o nociu, mentre que un prompt ben creat pot prevenir aquests problemes, fent que la IA sigui més segura.

2.1.3. Bases de Dades Vectorials i Embeddings

Durant la revolució de la IA, han sorgit nous reptes, sobretot en el processament i emmagatzematge de dades en aplicacions que utilitzen LLM, IA generativa o cerques semàntiques. Un dels desenvolupaments clau ha estat l'ús dels embeddings, representacions numèriques d'una paraula, frase o altres fragments de dades [19]. En l'àmbit de la IA i el processament del llenguatge natural, aquests embeddings faciliten l'anàlisi i manipulació de dades textuals, gràcies a la generació de vectors contextualitzats per frases completes per part dels mateixos LLM [20].

Els textos convertits en embeddings vectorials, que contenen informació semàntica, permeten a la IA executar tasques com fer preguntes, classificar, aplicar algoritmes d'aprenentatge automàtic o mantenir memòria a llarg termini. Aquests vectors tenen una gran quantitat d'atributs i dimensions, essencials per entendre els patrons, les relacions i estructures subjacents [21].

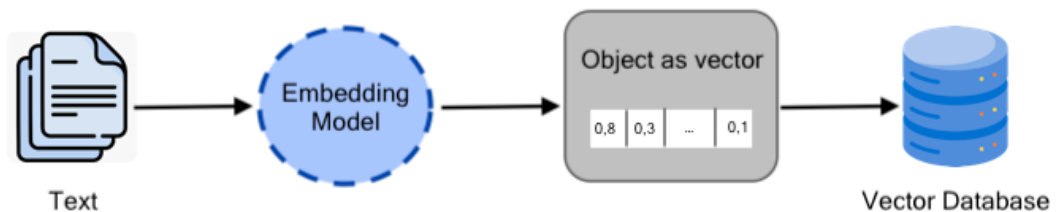


Figura 4. Conversió dels text a embeddings i emmagatzematge a les bases de dades vectorials.

La complexitat en la gestió d'aquests embeddings requereix l'ús de bases de dades especialitzades, com les bases de dades vectorials o índexs vectorials. Les primeres combinen les capacitats d'una base de dades tradicional amb l'especialització en vectors, oferint funcionalitats com inserció, eliminació i actualització d'embeddings [19]. A més, suporten l'emmagatzematge de metadades, la filtració, l'escalabilitat, actualitzacions en temps real, còpies de seguretat, integració amb altres eines d'IA i la seguretat de les dades.

Les bases de dades vectorials operen diferentment de les bases de dades tradicionals, que busquen coincidències exactes. Aquestes utilitzen mètriques de semblança per

identificar els vectors més semblants a la consulta, utilitzant algoritmes com PQ², LSH³ o HNSW⁴, essencials en els processos d'indexació, consulta i post-processament [19]. Les mesures de similitud entre vectors es fan servir per determinar com són de semblants dos vectors en l'espai, com per exemple a través de la distància cosinus, la distància euclidiana i el producte escalar. La selecció d'aquestes mètriques influeix directament en els resultats de les consultes.

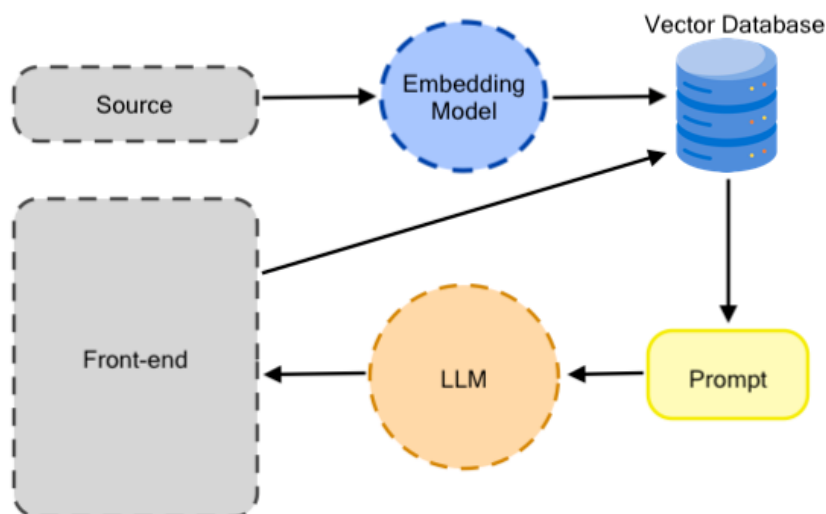


Figura 5. Diagrama dels elements necessaris per al projecte i com se relacionarien.

2.2. Entorn de desenvolupament

LLM

En la fase de preparació del nostre projecte, hem identificat els elements clau per al desenvolupament del prototip de l'aplicació que busquem crear. Per aquest propòsit, principalment hem optat per les eines desenvolupades per l'empresa OpenAI, que han demostrat ser les més adequades per a les nostres necessitats. També tenir en compte que és l'única empresa que ofereix el conjunt complet d'eines necessàries, ja que si no s'haurien d'utilitzar diferents proveïdors. A més, usar tots els productes que té OpenAI ens ajuda a evitar problemes de compatibilitat. OpenAI, va ser fundada l'any 2015 per Elon Musk, Sam Altman, Greg Brockman i Ilya Sutskever, és una empresa de recerca i desenvolupament d'IA amb la missió de beneficiar a tota la humanitat amb la IA general. El seu treball s'ha centrat en el desenvolupament de sistemes que pensen i raonen com humans, resultant en productes com la sèrie LLM de GPT, on s'inclouen GPT-3 i GPT-4. Aquests models han demostrat capacitats impressionants en el processament del llenguatge natural i en la generació de text humà, com es veu en la seva aplicació més coneguda, ChatGPT [11].

² PQ (Product Quantization): parteix el vector inicial en fragments, entrena un llibre de codis per a cada fragment usant un cluster k-mean, codificant cada fragment amb un codi concret del llibre, fent el mateix amb el vector de la consulta i usant el codis per trobar els vectors més semblants.

³ LSH (Locality-sensitive hashing): vectors similars són mapejats en "buckets" utilitzant funcions de hash, permetent que les cerques per trobar el més semblant sigui més ràpida i comparant el vector de la consulta amb els de la taula.

⁴ HNSW (Hierarchical Navigable Small World): construeix una estructura d'arbre on els nodes representen conjunts de vectors i on les rames denoten la similitud.

En el nostre projecte, després d'una anàlisi detallada, hem decidit treballar amb el model gpt-3.5-turbo-16k. En l'apartat 5.1 explicarem la comparativa realitzada i com s'ha arribat a aquesta decisió.

Embeddings

Pel que fa a l'aspecte pràctic, utilitzarem el mòdul OpenAIEmbeddings per crear representacions numèriques en vectors que capturin significats semàntics i de context de paraules, frases o documents [22]. Aquest mòdul genera els embeddings vectorials fent ús de models de llenguatge.

Per accedir als productes d'OpenAI, s'ha hagut de realitzar una inversió monetària, ja que l'ús dels LLM i els embeddings té cert cost associat. El factor cost ha estat una de les consideracions més important a l'hora de decidir quins productes utilitzar, perquè de cara al client a menor preu major interès.

Tractament de dades

Finalment, la llibreria LangChain, també desenvolupada per OpenAI, serà un recurs fonamental en el nostre marc de treball. LangChain està dissenyada per simplificar el desenvolupament, producció i desplegament d'aplicacions basades en models de llenguatge [23]. Ofereix un conjunt de llibreries, plantilles i eines per a la construcció d'aplicacions conscients del context i que poden raonar basant-se en aquest [24]. Aquesta llibreria té una àmplia gamma d'aplicacions, com xatbots, assistents virtuals, sistemes de resposta a preguntes, generació de contingut, traducció de llenguatge, entre altres [25]. Proporciona la infraestructura necessària per aprofitar al màxim les capacitats dels LLM, simplificant també així el procés de treball i permetent als desenvolupadors centrar-se en la construcció d'aplicacions amb context sense haver de gestionar les complexitats de la integració i desplegament del model [26].

Prompt

Per gestionar i ajustar les respostes i el comportament del projecte, s'ha creat un prompt. Aquest prompt es pot canviar i perfeccionar en qualsevol moment per garantir respostes més específiques o en el format que millor s'adapti a les nostres necessitats. Aquesta funcionalitat està destinada exclusivament als desenvolupadors, permetent afinar la capacitat de resposta i adaptant-lo als requisits particulars. No necessita cap intervenció de l'usuari.

Base de dades vectorial

En quant al tractament i emmagatzematge de dades ens hem decidit per una base de dades vectorial, ja que amb aquestes és més fàcil integrar a l'entorn amb el qual treballem amb eines d'IA com és la llibreria LangChain [27]. Durant la fase de selecció d'eines per al nostre projecte, vam analitzar diverses bases de dades vectorials que ens van cridar l'atenció, inclosos Pinecone i DeepLake. Després d'una comparativa exhaustiva, vam decidir utilitzar DeepLake, ja que s'ajustava millor a les nostres necessitats i proporcionava els resultats desitjats de manera més eficaç. El factor decisiu va ser la facilitat de connexió amb la base de dades i el tractament de les dades, així com el fet

que DeepLake és gratuït, a diferència de Pinecone que tenia un cost associat a l'augmentar les dimensions dels documents [28].

Localització de dades

Les dades a processar, convertides en embeddings, provenen de diversos formats com Excel, Word, PDF, PowerPoint, text, etc., i es troben en una carpeta anomenada “test” a Google Drive. El Google Drive es tracta d'un servei d'emmagatzematge i sincronització al núvol desenvolupat per Google, el qual facilita el treball en equip i el compartiment de documents en una varietat de formats de forma ràpida.

Accés de dades

Durant el desenvolupament, vam trobar dificultats en l'accés directe a la carpeta de Google Drive. La solució va ser l'ús de Google Cloud, un servei de *cloud computing* ofert per Google. Aquest inclou una àmplia varietat de serveis, com màquines virtuals, computació sense servidors, emmagatzematge escalable, bases de dades gestionades, serveis de BigQuery, i eines per al desenvolupament d'aprenentatge automàtic, així com serveis d'identitat i control d'accés, i la gestió de dispositius IoT. A més, proporciona un ampli ventall d'API i SDK per integrar serveis i plataformes [29].

Per accedir a aquests serveis, ens vam registrar a la plataforma, la qual requereix una targeta de crèdit, però l'ús que li vam donar va ser completament gratuït [30]. Entre les possibles solucions a implementar, vam trobar l'opció d'utilitzar els serveis d'API de Google, que permeten crear una pantalla de consentiment OAuth perquè els usuaris puguin accedir amb els seus comptes de Google [31]. Una altra opció va ser fer ús dels serveis d'identitat i control d'accés que s'ofereixen, creant un compte de servei especial vinculat a l'aplicació per a l'autenticació segura [32], [33].

Desenvolupament de front-end

Respecte a la interfície d'usuari, vam optar per Streamlit [34], un *framework* ideal per treballar amb eines d'IA generativa i apropiat per als nostres coneixements bàsics de desenvolupament *front-end*. Streamlit permet crear interfícies d'usuari simples, interactives i fàcils d'usar [35].

Entorn de treball i control de versions

Per al desenvolupament del codi, vam utilitzar l'IDE Visual Studio Code, juntament amb el llenguatge de programació Python 3.11 per les seves característiques. Les més importants la gran versatilitat i portabilitat del llenguatge. A més, de la gran quantitat de biblioteques i *frameworks* que poden ajudar al tractament intern de dades en el desenvolupament d'aplicacions.

A més, vam adoptar Git com a sistema de control de versions, integrat dins de Visual Studio Code, per a un seguiment eficient del codi i l'accés a versions anteriors del projecte. El repositori del projecte es va crear a GitHub, donant-nos la facilitat de connectar-lo amb l'eina de Git de Visual Studio Code, permetent-nos així observar les diferents etapes que ha tingut el projecte.

2.3. Previsions d'ús

El prototip que hem desenvolupat està destinat a empreses, especialment aquelles de gran envergadura com les multinacionals, que gestionen una gran quantitat de documents en diversos àmbits. Aquestes empreses, amb múltiples departaments, podrien beneficiar-se significativament de la nostra solució.

Imaginem, per exemple, un escenari on el personal de vendes d'una empresa identifica un possible nou client. Aquesta persona de vendes necessita trobar ràpidament informació rellevant sobre els productes que podrien interessar al client, així com detalls específics sobre com aquests productes poden ser d'utilitat al client potencial i informació rellevant sobre l'àmbit de negoci del client. Normalment, aquesta informació estaria dispersa en diversos documents, alguns dels quals podrien contenir termes tècnics o informació fora de la seva àrea d'expertesa.

El nostre prototip facilita aquesta tasca. Permetria al personal de vendes trobar les respostes de manera ràpida i eficient, sense haver de dedicar temps excessiu a cercar a través de muntanyes de documents. Això no només estalviaria temps valuós, sinó que també milloraria la precisió i la qualitat de la informació proporcionada al client.

En resum, aquest prototip està dissenyat per optimitzar el procés de recerca i anàlisi de documents en empreses amb una gran quantitat de dades i informació. Facilitarà als empleats l'accés a la informació necessària, millorant així la seva eficiència i capacitat per respondre de manera informada a les necessitats dels clients.

3. Requisites

3.1. Requisites funcionals

1. Que els usuaris puguin introduir la seva pròpia clau OpenAI.
2. Que els usuaris puguin introduir el seu usuari de DeepLake.
3. Que els usuaris puguin introduir la seva clau de DeepLake.
4. Que els usuaris puguin carregar múltiples tipus de documents (Excel, Word, PowerPoint, PDF, TXT, CSV).
5. Que els usuaris puguin afegir nous documents a la carpeta.
6. Que els usuaris puguin consultar la última data de consulta a la carpeta.
7. Que l'usuari pugui fer preguntes sobre la informació dels documents carregats.
8. Que l'usuari pugui visualitzar la resposta donada per l'assistent.
9. Que el sistema notifiqui al usuari si s'ha afegit informació sobre nous documents.
10. Que l'usuari pugui visualitzar l'historial de preguntes fetes en tot moment
11. Que l'usuari pugui fer les preguntes en diferents idiomes.

3.2. Requisites no funcionals

1. Garantir temps de resposta ràpida i eficient per agilitzar la conversació amb l'usuari.
2. Assegurar disponibilitat del sistema amb temps d'inactivitat mínims, consistent per als usuaris.
3. Adaptabilitat a diferents tipus de documents sense problemes mantenint coherència amb els resultats.

4. Emmagatzematge dels documents ja carregats per evitar duplicació d'informació.
5. Fer el codi adaptable estructurant-lo de forma clara per facilitar futurs canvis.
6. Interfície d'usuari intuïtiva i fàcil d'usar.
7. Garantir la seguretat de les dades introduïdes per l'usuari.
8. El sistema ha de ser escalable per gestionar majors càrregues de treball sense perdre rendiment o eficàcia en donar respostes.

4. Disseny

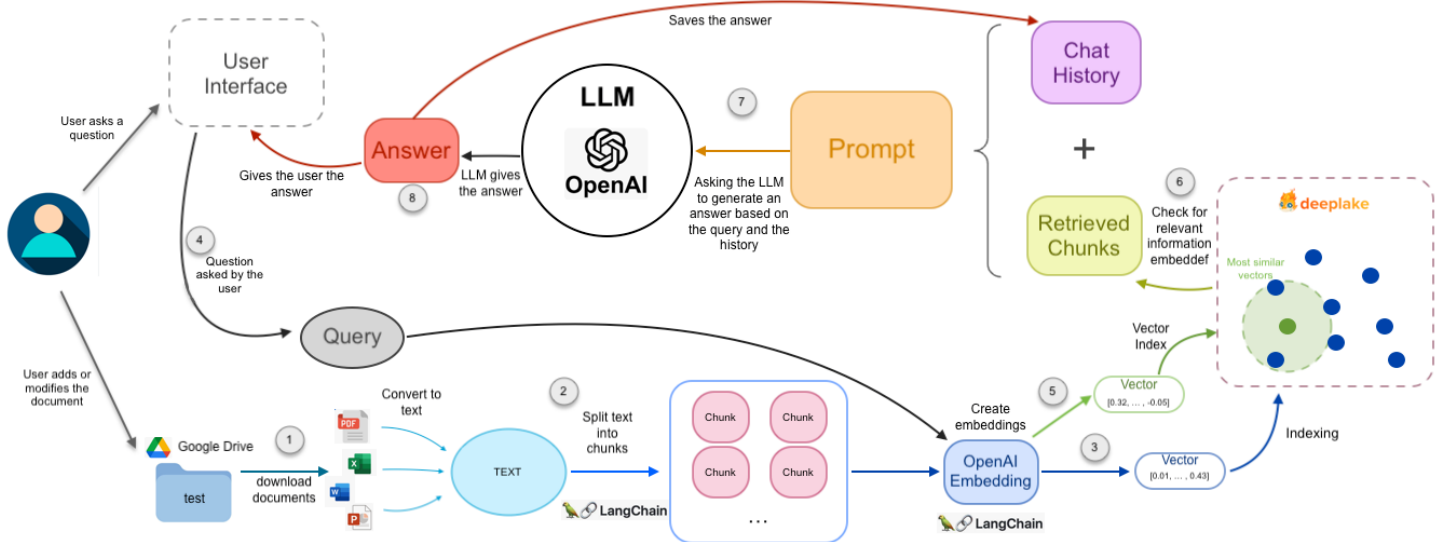


Figura 6. Diagrama de l'arquitectura de l'aplicació.

4.1. Arquitectura de l'aplicació

La nostra aplicació segueix l'arquitectura representada en la **Figura 6**. Aquesta descriu com es gestionen els documents dins de l'aplicació:

1. Càrrega de Documents: Els documents es carreguen des d'una carpeta anomenada "test" a Google Drive. Des d'aquesta carpeta es descarreguen aquestes dades per al seu posterior processament.
2. Processament de Documents: Els documents baixats són convertits a text per estandarditzar la informació i fragmentats en peces més petites per facilitar la seva anàlisi.
3. Creació d'Embeddings: Per a cada fragment, es crea el seu corresponent embedding que conté la informació pertinent. Aquests embeddings són emmagatzemats en una base de dades vectorial. Aquest procés es realitza una sola vegada a l'inici de l'execució de l'aplicació, i es repeteix només per a nous documents que s'afegeixin a la carpeta.
4. Interfície d'Usuari: L'usuari fa consultes a través d'una interfície d'usuari, on introdueix les seves preguntes.
5. Generació de Vectors de Consulta: A partir de la pregunta de l'usuari, es crea un vector que es fa servir per a cercar similituds dins la base de dades vectorial.

6. Selecció de Vectors Similars: Es trien els vectors de la base de dades que presenten una major similitud amb el vector de la consulta.
7. Interacció amb el LLM: Els fragments seleccionats, juntament amb l'historial de la conversa (si n'hi ha), es passen al LLM amb les especificacions del prompt, indicant com ha de respondre.
8. Resposta de la IA: La IA generativa processa aquesta informació i proporciona una resposta, la qual és després mostrada a l'usuari i emmagatzemada en l'historial de la conversa.

Aquest procés es repeteix cada vegada que l'usuari realitza una nova consulta, proporcionant un cicle de feedback continu per a la millora de les respostes i l'experiència d'usuari.

4.2. Disseny interfície gràfica

Encara que no és el punt més important del projecte, s'ha buscat fer una interfície agradable visualment, que sigui intuïtiva i fàcil d'aprendre a usar per a qualsevol usuari.

4.2.1. *Framework Streamlit*

Per desenvolupar el *front-end* de l'aplicació, s'ha fet servir el *framework* proporcionat per Streamlit. Aquesta eina és un codi obert de Python que va ser creada l'any 2019 i està dissenyada per permetre a equips que treballen en aprenentatge automàtic i ciències de dades crear i compartir aplicacions de dades de manera ràpida i senzilla [34]. L'hem escollit per les següents característiques que ofereix són del nostre interès:

- La seva API simple permet als desenvolupadors crear la seva aplicació amb només un parell de línies de codi de Python, cosa que la fa accessible per persones experimentades o novells.
- Està completament creat en Python.
- Permet als desenvolupadors fer ús de les tecnologies més innovadores, com és el cas de la IA en el nostre projecte.
- Ofereix àmplia varietat de components per la interfície que són molt intuïtius fàcilment personalitzables.
- Compta amb una plataforma de comunitat al núvol que facilita l'execució el control i la compartició fàcil de les aplicacions creades. Aquesta comunitat és molt activa i està plena de professionals.

Streamlit està evolucionant constantment i s'afegeixen contínuament noves característiques i funcionalitats.

4.2.2. *Apartats de la Interfície*

Sidebar:

Un *sidebar* o una barra lateral és una secció vertical que trobem en un lloc web o aplicació que molts cops conté enllaços, eines i informació addicional. En el nostre cas, el *sidebar* es troba al costat esquerre de la pantalla i conté tres camps per introduir text, on l'usuari podrà introduir les seves credencials, com ara l'OpenAI

key, Activeloop username i Activeloop token (**Figura 7a**). Un cop introduïdes les dades aquelles que són personals es mostren encriptades (**Figura 7b**). Quan els documents han estat carregats i indexats, apareix sota els camps de text la data i l'hora de l'última consulta a la carpeta del Drive, un botó per si es vol tornar a consultar i un enllaç al repositori GitHub amb el codi del projecte per si es vol consultar (**Figura 7c**). Finalment, si es consulta la carpeta i es troben nous documents, apareix un missatge informant que els nous documents s'estan carregant a la base de dades (**Figura 7d**).

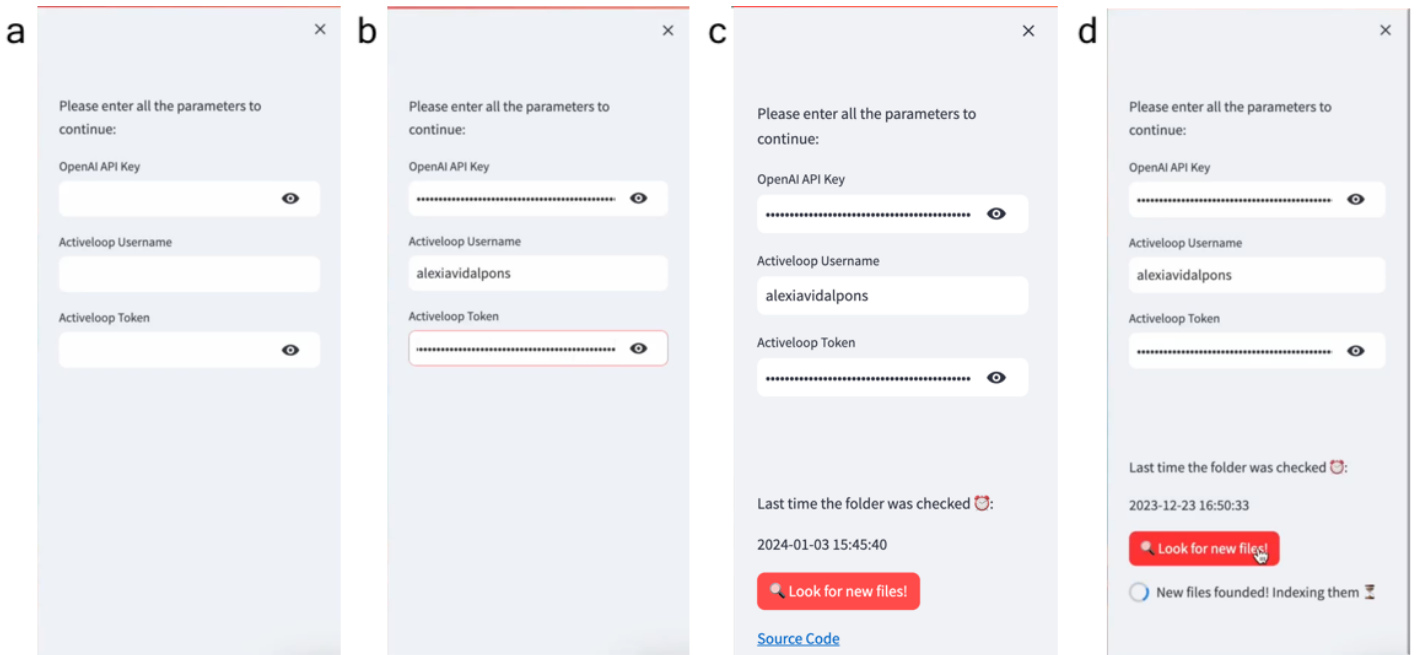


Figura 7. Diferents estats en els que es pot trobar el sidebar de l'aplicació.

Espai de xat:

A l'aplicació, l'espai que no està ocupat pel *sidebar* forma part de l'espai on es pot xatejar amb els documents, i aquesta zona també pot passar per diferents estats. Aquest espai està compost, primer, per un títol i una captura on ens diu que es tracta d'un xat amb els teus documents, seguit de l'àrea on es podrà observar la conversa. El primer estat que podem trobar és el d'inici, on es demana que s'introdueixin les credencials d'usuari per poder continuar (**Figura 9**), ja que són necessàries per dur a terme tot el processament i iniciar l'aplicació.

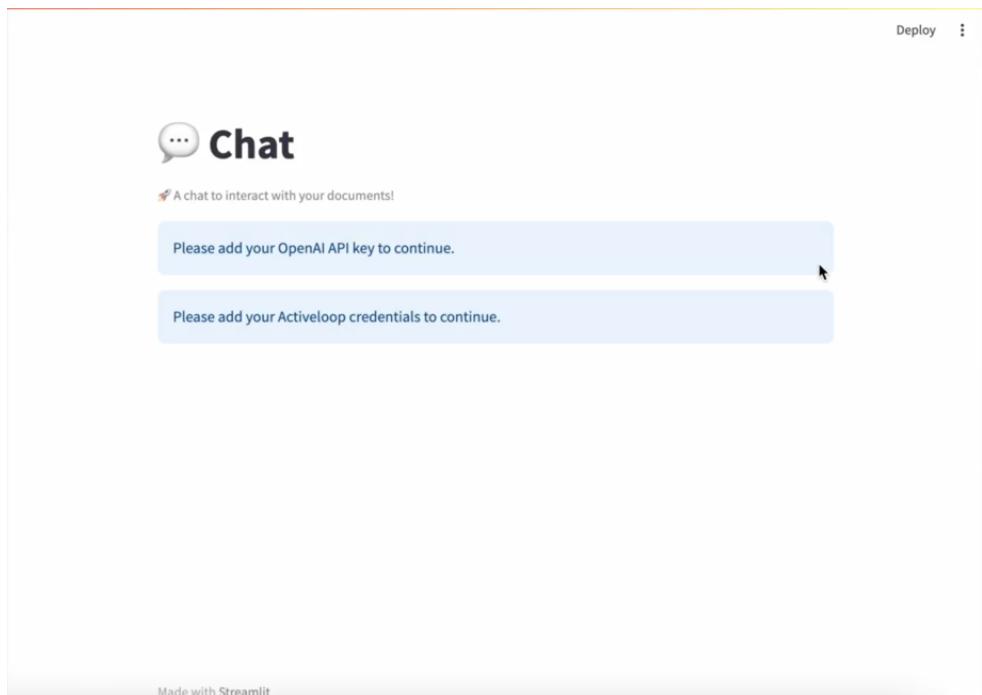


Figura 9. Estat inicial del xat de l'aplicació.

Un cop les credencials són introduïdes, comença el processament dels documents, i així ho indica a l'espai amb el missatge que els documents s'estan indexant i que aquest procés pot trigar una estona (**Figura 8**).

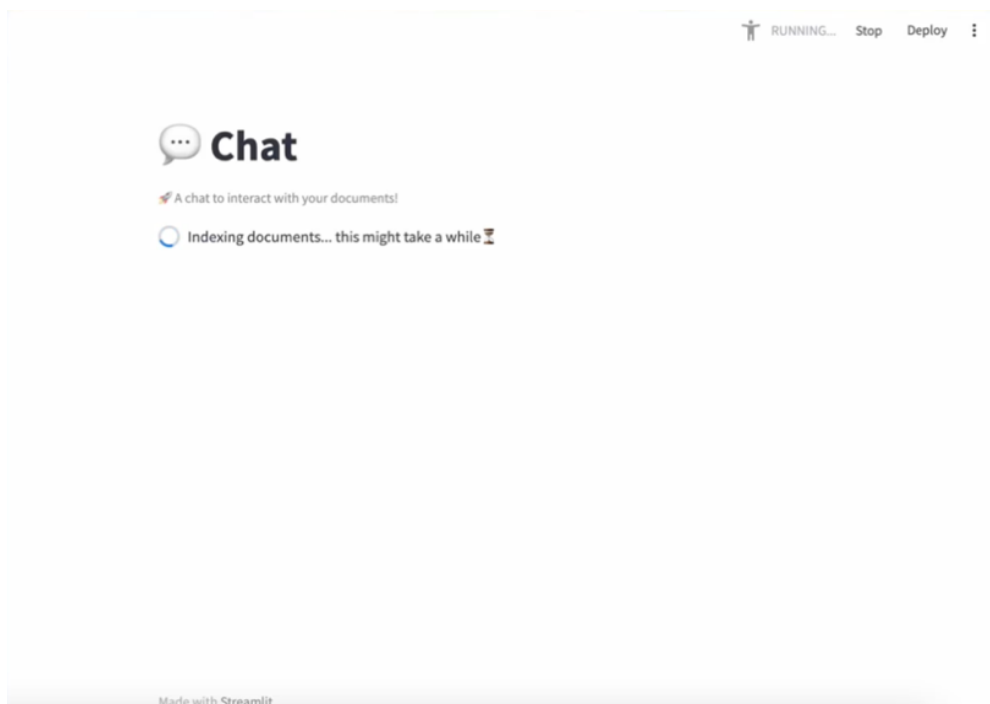


Figura 8. Visualització de l'espai de xat mentre es tracten els documents.

Un cop es finalitza la càrrega de documents, es pot començar el procés de xatejar. Primer, apareix a la part inferior un espai perquè l'usuari pugui escriure les seves preguntes (**Figura 11a**). Després d'enviar la pregunta, es mostra el missatge que la seva pregunta s'està contestant (**Figura 11b**).

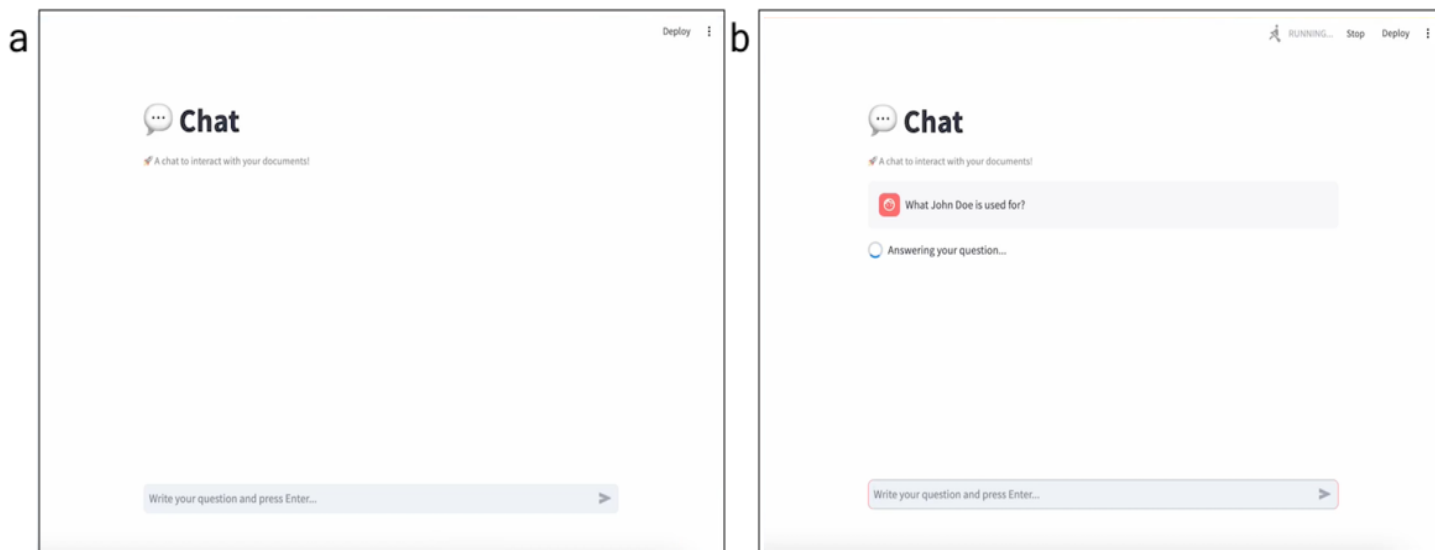


Figura 11. Visualització un cop es permet fer preguntes i com es veu mentre contesten.

Un cop es rep la resposta per part de la IA, es mostra en l'espai sota la pregunta, l'usuari pot introduir una altra pregunta repetint el procés tants cops com es vulgui. Els missatges s'identifiquen segons l'avatar que apareix davant del missatge escrit: el roig identifica l'usuari, el groc a l'assistent i en alguns casos pot aparèixer el sistema identificat en un avatar blanc amb la lletra "S" en negre (**Figura 10**).



Figura 10. Avatars que acompanyen els missatges.

5. Implementació

5.1. Detalls de la tecnologia

Per al desenvolupament d'aquest projecte s'han utilitzat varies tecnologies, de les quals no tenia gaire experiència prèvia, s'han hagut de d'escollir algunes tecnologies o funcions per sobre d'altres, especificar certs paràmetres d'aquestes funcions per un motiu concret o canviar com fer alguna cosa per aconseguir el millor producte possible.

Base de Dades Vectorial

Dintre de l'ampli ventall de possibilitats que existeix de bases de dades vectorials, van ser les tres següents sobre les quals es va investigar amb més profunditat per poder decidir quina podria adaptar-se millor a les nostres necessitats. La comparativa entre les tres bases és la següent:

- Pinecone: és una base de dades vectorial i un sistema d'indexació dissenyat per un emmagatzematge eficient, capaç de recuperar dades i de fer cerques de similitud entre vectors multidimensionals. Va ser creada per Pinecone Systems Inc., empresa centrada a crear una infraestructura de cerca de vectors escalable. Treballa completament al núvol on els usuaris poden operar i treballar en les

seves solucions d'IA sense necessitat de mantenir l'estructura o monitorar el servei [36]. No és de codi obert, per tant, no es poden fer consultes. A més a més, ofereix una API fàcil d'usar i llibreries de client per simplificar la integració en sistemes existents.

- Chroma: és una base de dades de codi obert que treballa amb embeddings vectorials buscant simplificar el procés de creació d'aplicacions amb LLM. Va ser creada per l'equip de Chroma amb funcionalitats d'afegir, recuperar, actualitzar, inserir, eliminar i consultar vectors [36]. Permet fer cerques de similitud, permetent buscar el vector més semblant a una consulta.
- DeepLake: és una base de dades vectorial multimodal creada per Actueloop, aquesta ha treballat ja amb els reptes que suposa ordenar i treballar amb vectors a gran escala en aplicacions d'aprenentatge automàtic [37]. DeepLake no sols emmagatzema embeddings sinó que també guarda les dades originals. Aquest proporciona una arquitectura sense servidors, eliminant la necessitat de serveis addicionals. Permet un control de versions automàtic, assegurant la integritat de les dades i rastrejament d'aquestes. En ser descrita com multimodal indica que no sols és capaç d'emmagatzemar vectors de text sinó d'altres tipus com poden ser àudio, imatges, PDF, etc. [38]. A més a més, d'oferir els serveis com les altres ja descrites de fer consultes i comparar la similitud entre vectors.

Finalment, ens vam decidir per la base de dades de DeepLake, perquè les característiques que proporciona són les que més s'ajusten als nostres interessos per al projecte. El fet que és gratuït és un factor que pot ajudar a vendre el nostre producte al client.

LLM

Un cop es va decidir utilitzar algun dels models de llenguatge que ofereix OpenAI es va haver d'escollir quin. Com ja s'ha vist els tokens és com interpreten els models de paraules i la seva quantitat en una crida és rellevant. En el cas dels models d'OpenAI el nombre de tokens influeix en el cost de la crida de l'API perquè es paga per token. També en la durada de la crida perquè com més tokens més temps. Finalment, determina si la crida funciona correctament, perquè certs models tenen un nombre limitat de tokens (gpt-3.5-turbo té 4.097 tokens). Aquests factors es veuen tan afectat pels tokens d'entrada com els de sortida, és a dir, si el missatge d'entrada té 10 tokens i el de sortida 20 tokens es facturen 30 tokens. S'ha de consultar per cada model el preu per token, ja que pot variar entre els d'entrada i sortida. En el nostre cas estem interessats en els següents models que tenen aquests preus [39]:

Taula 1. Taula comparativa dels preus dels tokens dels models d'OpenAI i el màxim que accepten.

Model	Entrada	Sortida	Max tokens
gpt-4	\$0.03/1000 tokens	\$0.06/1000 tokens	8.192 tokens
gpt-3.5-turbo	\$0.003/1000 tokens	\$0.006/1000 tokens	4.097 tokens
gpt-3.5-turbo-16k	\$0.003/1000 tokens	\$0.006/1000 tokens	16.385 tokens

Un cop contemplat el factor monetari la decisió estava entre gpt-4 i gpt-3.5-turbo-16, ja que ens interessa que es puguin fer preguntes llargues igual que les respostes que es generin.

El següent que ens vam contemplar, va ser si la diferència de preu significava un canvi en la qualitat de les respostes que ens donava. Vam observar que el gpt-4 té un millor rendiment en les avaluacions externes a les quals l'ha sotmès l'empresa, demostrant que és més capaç de seguir amb cura instruccions complexes. Es va trobar una avaluació que es va fer amb tots dos models resolent exàmens d'admissió del Brasil, on van resoldre exàmens del 2009-2017 el gpt-3.5-turbo-16k va treure una mitjana de 81,32 sobre 100 i el gpt-4 un 94,56 [40]. Els resultats que es mostraven no tenien un empitjorament significatiu, però per sortir de dubtes vam fer la nostra pròpia prova.

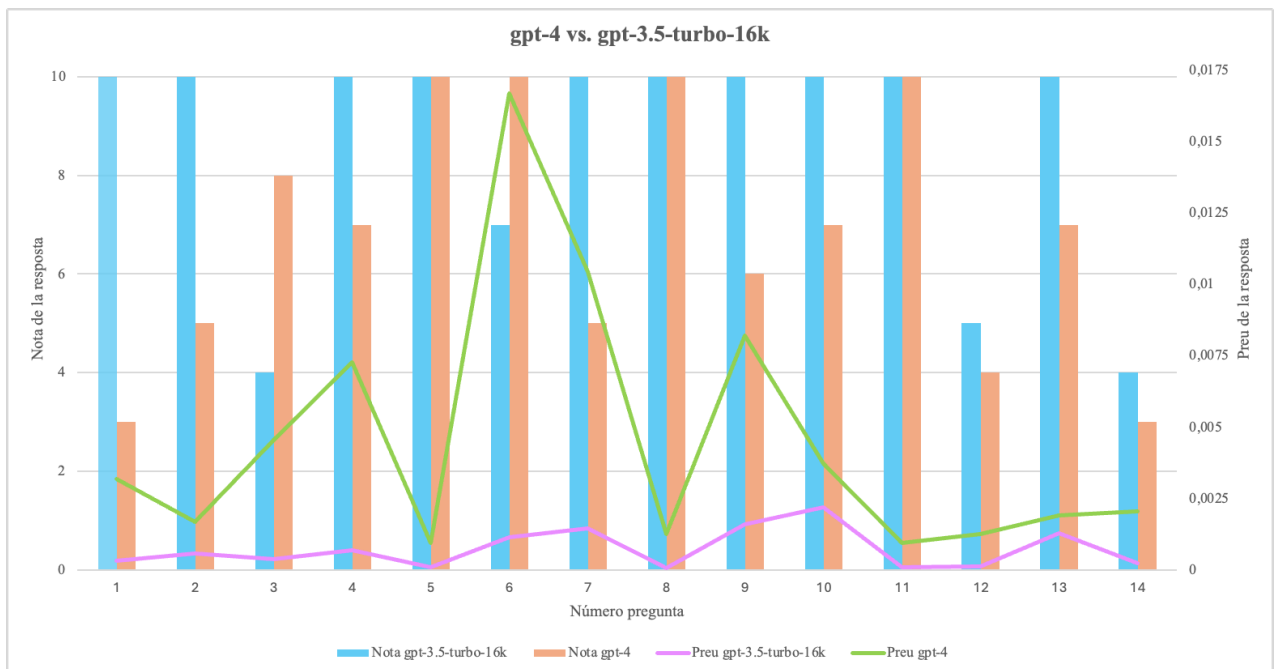


Figura 12. Gràfic comparatiu de notes i preus fet personalment per avaluar el model gpt-4 i el gpt-3.5-turbo-16k.

En aquesta vam avaluar tots dos models contestant un total de 14 preguntes simples, de les quals sabíem la resposta correcta pel context carregat, i vam poder posar nota a la qualitat resposta. Per mesurar la qualitat es va mirar el contingut, el format expressat i la longitud d'aquesta. Com es pot veure en el gràfic resultant de l'avaluació (**Figura 12**), tots dos models van donar respostes correctes, però la qualitat de les donades pel model gpt-3.5-turbo-16k tenien major qualitat. Un cop finalitzat vam poder analitzar el preu que va tenir avaluar, obtenint que el gpt-3.5-turbo 16k va costar \$0.04 i el gpt-4 va tenir un cost de \$0.32, en aquest preu s'inclou el preu d'entrada i sortida. Tal com ens permet observar el gràfic la relació qualitat-preu el del model gpt-3.5-turbo-16k va ser millor en el nostre context. A més a més, al preu del model s'ha de sumar el preu dels embeddings que és de \$0.0001/1000 tokens, tot i ser molt poc si tenim molts documents pot arribar a pujar bastant si hi ha molts documents.

A més a més, si mirem el temps de resposta que tenen tots dos models observem que el gpt-3.5-turbo-16k té menys temps latència, ja que el gpt-4 en tenir tantes capacitats computacionals tot i no utilitzar-les augmenta el temps[41].

Finalment, contemplant tots els paràmetres s'ha conclòs que tot i el gpt-4 ser superior en les avaluacions vistes, es farà servir el gpt-3.5-turbo-16k que ha donat millors resultats en la nostra avaluació. Si a més, es contempla el factor preu l'augment significatiu que suposa el canvi de model, no suposa cap millora significativa per al nostre projecte.

Accés i descarregues a la carpeta Google Drive

Tal com s'ha comentat prèviament, per accedir a la carpeta del Google Drive on es troben els documents, es proposen dues possibles solucions. La primera sent utilitzar OAuth i la segona és creant un compte de servei. Tenint en compte les necessitats del nostre projecte, s'ha decidit que la millor és usar un compte de servei per diverses raons. L'OAuth implica sovint una fase d'autorització que requereix interacció amb l'usuari, fet que el compte evita i facilita fer operacions automatitzades i accedir a recursos sense la necessitat d'interacció amb un usuari. Com que els documents en una carpeta que pot estar compartida entre diversos usuaris, es permet donar permisos adequats al compte perquè pugui accedir i descarregar els documents, simplificant la gestió d'autoritzacions i el manteniment del sistema.

Per poder fer ús d'un compte de servei, és necessari registrar-se al Google Cloud i crear un projecte. Després, a la secció IAM i administració, cal accedir a l'apartat de comptes de servei. Un cop dins, és tan fàcil com crear un compte de servei al qual li podem donar el nom relacionat amb la tasca que realitzarà. Arribats aquest punt, és necessari obtenir la clau JSON associada al compte de servei. També serà necessari activar l'API Google Drive per al projecte. Finalment, serà necessari que la carpeta on es troben els documents es comparteixi amb el correu del compte de servei com es compartiria amb qualsevol altre usuari.

```
scope = ['https://www.googleapis.com/auth/drive']
service_account_json_key = 'credential-key.json'
credentials = service_account.Credentials.from_service_account_file(
    filename=service_account_json_key,
    scopes=scope)
service = build('drive', 'v3', credentials=credentials)
```

Codi 1. Codi per autenticar-se i crear una connexió amb l'API de Google Drive

El **Codi 1** mostra com s'autentifica i com es crea una connexió amb l'API de Google Drive utilitzant un compte de servei. Primerament, es defineix l'abast d'autorització del compte de servei i es defineix el nom que té el fitxer JSON que conté la clau del compte de servei. A continuació, es creen les credencials usant les claus i l'abast, aquestes credencials s'utilitzen posteriorment per crear l'objecte servei, que ofereix una interfície per fer les crides a l'API del Google Drive utilitzant les credencials proporcionades per compte de servei. També especifiquem la versió de l'API amb la qual volem interactuar.

Un cop creat el servei, es pot demanar un llistat dels documents de la carpeta. Aquest llistat ens permet obtenir l'identificador, nom i tipus de fitxer que es tracta. Aquesta informació la guardarem per posteriors comparacions, i a aquesta s'afegeix una columna per indicar si el document ha estat carregat a la base de dades, a l'inici aquest valor sempre és fals. A continuació, es descarrega cadascun dels fitxers, i es guarden a la carpeta “docs”, que es troba en el directori en local de l'aplicació, tal com es pot veure en el **Codi 2**.

```
try:
    request_file = service.files().get_media(fileId=row["id"])
    file = io.BytesIO()
    downloader = MediaIoBaseDownload(file, request_file)
    done = False
    while done is False:
        done = downloader.next_chunk()
except HttpError as error:
    print(F'An error occurred: {error}')

file_retrieved: str = file.getvalue()
with open(f"docs/{row['name']}", 'wb') as f:
    f.write(file_retrieved)
```

Codi 2. Fragment de codi la funció `download_files` que s'encarrega de la descarrega de documents

Tractament de documents

Quant al tractament de documents, s'ha creat una funció especial anomenada “`load_doc`” que consulta la carpeta passada per paràmetre, en el nostre cas, “docs”. S'obté un llistat dels documents i s'observa l'extensió del document. Depenent d'aquesta extensió, s'utilitza un loader, encarregat d'estandarditzar la informació, específic per cada tipus de fitxer. S'utilitzen els loaders proporcionats per LangChain per poder transformar les dades en un format amb el qual puguem treballar correctament:

- PDF: S'usa “PyMuPDFLoader” de la llibreria PyMuPDF, que proporciona diverses funcionalitats per treballar amb PDFs. En el nostre cas, li passem el paràmetre `extract_images` com a vertader perquè faci l'extracció d'imatges.
- TXT: S'utilitza “TextLoader” un mòdul de LangChain que permet carregar les dades dels fitxers de tipus text. Suposem que els fitxers que carreguem estaran codificats en el format utf-8.
- CSV: Es fa servir “CSVLoader”, que extreu les dades i les representa un document de LangChain. S'ha definit en l'argument `csv_arg` que el delimitador és ‘,’ i el caràcter per citar són els ‘“’.
- XLS i XLSX: Utilitzen “UnstructuredExcelLoader”, un loader creat específicament per a Microsoft Excel usant la llibreria Unstructured. Permet

extraure les dades i representar-les com un document. La variable `elements` tracta cada fulla de l'Excel com un document separat.

- DOCX: S'utilitza el “Docx2txtLoader” que carrega Microsoft Word fent servir la llibreria `docx2txt` extraient el contingut del document.
- PPTX: S'utilitza el “UnstructuredPowerPointLoader”, que al igual que amb l'Excel, utilitza la llibreria `Unstructured` per extreure les dades del fitxer. La variable `elements` permet que cada element del PowerPoint sigui representat i tractat com si fos un independent.

Un cop carregats els documents en un format estàndard, és necessari trencar els documents en fragments més petits. Per fer-ho, s'utilitza la funció “`split_documents`” de la classe “`CharacterTextSplitter`”, la qual cridem amb els paràmetres `chunk_size = 2048` i `chunk_overlap = 24`. La variable `chunk_size` indica la mida dels fragments que es crearan del document estandarditzat, i `chunk_overlap` indica el solapament entre cadascun dels fragments (**Figura 13**).

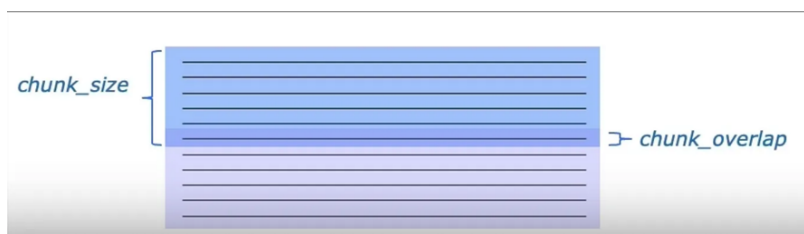


Figura 13. El funcionament de les variables `chunk_size` i `chunk_overlap` en la fragmentació de documents.

Seguit de la fragmentació, hi segueix el procés de guardar a la base de dades vectorial DeepLake. Per fer-ho, es fa la crida a la funció de la mateixa base passant per paràmetre la ruta definida per al conjunt de dades, els embeddings i el token d'ActiveLoop. En el nostre cas, s'ha fet una crida a “OpenAIEmbeddings” fora la funció “`load_doc`” i s'ha passat com a paràmetre el resultat per no fer múltiples crides, ja que es necessiten per altres funcions. Aquesta variable embeddings serà l'encarregada de transformar els fragments de documents en els vectors que es guardaran en la base de dades.

Construcció del Xat

Un cop finalitzada la construcció de la base de dades amb el context de treball que necessitem, és moment de construir l'estructura del xat encarregat de donar-nos les respostes a les preguntes de l'usuari. LangChain ofereix diverses opcions per crear el xat; una de les primeres que trobem és la classe “`RetrievalQA`” o “`ConversationalRetrievalChain`”, classe de cadenes senzilles que responen preguntes en funció del context proporcionat.

Per inicialitzar les cadenes, es necessiten dos elements. En primer lloc, el context, el qual és la base de dades vectorial creada prèviament. Però, per poder-la passar, es transforma a una interfície genèrica amb la funció de la base de dades “`as_retriever`”, que s'encarregarà de retornar els documents més semblants a la consulta. A la funció també se li passa com paràmetre el prompt creat per nosaltres, on s'especifica com ha d'actuar per contestar correctament.

El següent element és el LLM. En el nostre cas, es crea fent una crida a la funció “OpenAIChat” amb els paràmetres `model = gpt-3.5-turbo-16k`, `temperature = 0.2` i la clau d’OpenAI. El paràmetre de `temperature` és l’encarregat de controlar com acurada és una resposta. Els valors que pot prendre van del 0 a l’1. Com major sigui el valor, més creativa i aleatòria és la resposta. En canvi, com més baix sigui el valor, la resposta és més centrada i determinista, donant respostes més conservadores i predictibles. L’únic inconvenient de posar un valor molt baix és que augmenta el temps de resposta. En el nostre cas com ens interessa que doni respostes raonables, però no augmentar el temps d’espera s’ha assignat a un valor de 0,2.

Vistos els dos elements que es necessiten, veiem les diferències entre totes dues components de LangChain. “RetrievalQA” es centra en la resposta a preguntes basades en la recuperació d’un índex. Està dissenyat per extreure respostes a preguntes concretes basades en els vectors recuperats en la cerca de vectors similars a la consulta feta en la base de dades vectorial.

Per l’altra banda, el “ConvesationalRetrievalChain” amplia les capacitats de “RetrievalQA” per donar suport a interaccions conversacionals. Permet mantenir un historial de xat i generar respostes de manera conversacional. També permet interactuar amb les preguntes i respostes del xat, on el mateix historial és l’encarregat de proporcionar context per generar una resposta més adequada.

```
memory = ConversationBufferMemory (memory_key="chat_history",
return_messages=True)

qa = ConversationalRetrievalChain.from_llm ( ChatOpenAI(model="gpt-3.5-
turbo-16k", temperature=0.2, streaming=True,
openai_api_key=openai.api_key),
retriever=db.as_retriever(qa_template=qa_template), memory=memory )

...

response = qa({"question": query, "chat_history": history})

...
```

Codi 3. Definició de l’estructura del xat i la crida per aconseguir una resposta a “query”.

Mirant en els nostres interessos, ens convé més utilitzar el “ConvesationalRetrievalChain”, ja que volem que el xat sigui el més humà possible i, per tant, ens interessa conservar l’historial del xat. Ajuntant totes les parts, el xat es construiria i respondria a les preguntes tal com es mostra en el **Codi 3**.

Altres canvis

Un dels canvis fets entre l’entrega i la primera fase és la càrrega automàtica de variables d’entorn o que el mateix usuari introdueixi les seves variables. Inicialment, les variables de la clau d’OpenAI, l’usuari i el token d’Actueloop es trobaven en un fitxer `.env` i es carregaven automàticament. No obstant això, per fer-ho més intuïtiu per l’usuari l’acció d’introduir les seves pròpies credencials, s’ha decidit que aquest les introdueixi manualment de la interfície d’usuari.

5.2. Algoritmes específics

Botó de consulta per nous documents

El botó afegit en les últimes fases de desenvolupament s'ha definit un comportament concret en cas que l'usuari li faci clic. Aquest procediment va fer que es creés una nova funció anomenada "check_new_files", que torna a consultar la carpeta del Google Drive aconseguint el llistat de documents que hi ha. Es verifica si no es troba a la llista prèviament creada i, en cas de no trobar-se, es descarrega aquell document. S'indica que no ha estat carregat a la base i s'actualitza l'antiga llista amb les noves incorporacions. Un cop es surt de la nova funció, es passa a la funció "load_doc", la qual es va modificar per identificar si un fitxer ja havia estat carregat o no a la base de dades, tal com es mostra a **Codi 4**. Tot aquest control es pot fer adequadament a què el llistat de documents es guarda en format de Dataframe i no una simple llista.

```
result = files.query(f'name == "{file}" and loaded == True')

if not result.empty:
    continue
```

Codi 4. Línies afegides a la funció load_doc per saber si un document ha estat carregat.

6. Avaluació

De tots els elements que componen l'aplicació, la que major importància té i que ens interessa que funcioni correctament és la interacció conversacional amb l'usuari. Per poder veure millor un exemple d'ús per aquesta aplicació, s'han aconseguit documents reals de Clarivate, una empresa global especialitzada a proporcionar perspectives i analítiques per accelerar el ritme de la innovació. Ofereix eines per ajudar organitzacions i individus a prendre decisions informades i impulsar la innovació en diverses indústries. Les principals àrees en les quals treballa són la recerca científica i acadèmica en l'àmbit de ciències de la vida i farmacèutica, propietat intel·lectual i analítiques empresarials.

En el nostre cas, suposarem que formem part de l'equip de vendes i tenim un nou client, el cap d'I+D d'un equip que vol desenvolupar un nou producte per al tractament l'alopecía als Estats Units. Dintre dels documents, tenim gran varietat d'informació, especialment amb tecnicismes científics que no estan dintre dels coneixements del personal de vendes, i farem preguntes per informar-nos sobre els productes i preparar la millor experiència de venda per al client. Algunes de les preguntes que es poden plantejar són les següents:

Taula 2. Taula amb possibles preguntes, les seves corresponents respostes, el contingut que hauria de tenir la resposta per considerar-se correcta i la nota que li donem a la qualitat de la resposta.

	Preguntes	Resultat
1	Pregunta: <i>Which is the therapy area of the alopecia?</i> Resposta: <i>The therapy area of alopecia is Dermatology.</i>	10/10 Resposta correcta

	Contingut que hauria de tenir la resposta correcta: Dermatology	
2	<p>Pregunta: <i>When was the last published article about alopecia in the US?</i></p> <p>Resposta: <i>The most recent article published about alopecia in the US according to "Alopecia Executive Insights US" was in 2023.</i></p> <p>Contingut que hauria de tenir la resposta correcta: 2023</p>	<p>10/10</p> <p>Resposta correcta, inclou justificació</p>
3	<p>Pregunta: <i>What Clarivate key solution or product can I use to find out the competitive landscape of alopecia?</i></p> <p>Resposta: <i>You can use Clarivate's Cortellis Competitive Intelligence solution to find out the competitive landscape of alopecia. It provides comprehensive information on assets and competitors, including early stage assets and orphan drugs. It also offers extended global coverage and access to drug programs and companies from over 250 countries. Additionally, it provides exclusive content such as broker research and SWOTs and offers pipeline forecasting with machine-learning enabled Drug Timeline & Success Rates for higher accuracy.</i></p> <p>Contingut que hauria de tenir la resposta correcta: hauria de nombrar algun dels següents productes (DTS consulting, CortellisDrug Discovery Intelligence, CortellisCompetitive Intelligence, OFF-X, BioWorldScience o Drug discovery intelligence APIs)</p>	<p>10/10</p> <p>Resposta correcta, proposa solució i l'explica resumidament</p>
4	<p>Pregunta: <i>Any other possible key solution?</i></p> <p>Resposta: <i>Yes, there are other key solutions available to find out the competitive landscape of alopecia in the US. Some of these solutions include:</i></p> <ol style="list-style-type: none"> 1. <i>Cortellis Competitive Intelligence: This solution provides comprehensive information on competitors in the market, including their pipeline products, partnerships, and market strategies.</i> 2. <i>BioWorld: BioWorld provides news and analysis on the biopharmaceutical industry, including updates on competitors in the alopecia market.</i> 3. <i>Drug discovery intelligence APIs: These application programming interfaces provide access to real-time competitive intelligence data, allowing users to gather information on competitors in the alopecia market.</i> 	<p>6/10</p> <p>Repeteix resposta, Cortellis Competitive Intelligence ja l'havia proposat.</p>

It's important to note that the specific availability and features of these solutions may vary, so it's recommended to reach out to the respective providers for more information.

Contingut que hauria de tenir la resposta correcta: hauria de nombrar algun dels següents productes (DTS consulting, CortellisDrug Discovery Intelligence, CortellisCompetitive Intelligence, OFF-X, BioWorldScience o Drug discovery intelligence APIs)

Pregunta: *For what can be used BioWorld?*

Resposta: *BioWorld can be used for the following purposes:*

- 1. Staying up to date on industry news: BioWorld provides daily, weekly, and breaking news updates on the biopharmaceutical, medical technology, and early science sectors. It covers the complete lifecycle of therapies in development, companies behind those candidates, business development transactions, and regulatory hurdles.*
- 2. Analyzing recent news: BioWorld offers deep analysis and trends in the biopharmaceutical and medtech industries. Decision makers can use this analysis to develop dynamic strategies and make informed business decisions.*
- 3. Understanding the life sciences landscape: BioWorld provides comprehensive coverage of global biopharma, medtech, early science, and Asia-focused news. It helps users gain insights into the current state of the industry, including important trends and developments.*
- 4. Monitoring competitors: BioWorld allows users to track and monitor what their competition is doing. It provides information on the activities, advancements, and strategies of other companies in the biopharma and medtech sectors.*
- 5. Generating leads: Any news reported in BioWorld can be a prospective call with a talking point. Users can use the information and insights from BioWorld to generate their own leads and identify potential business opportunities.*

Overall, BioWorld serves as a valuable source of news, analysis, and data for professionals in the biopharmaceutical, medical technology, and early science industries.

5

10/10
Resposta correcta, nombra els seus usos i ho explica

	<p>Contingut que hauria de tenir la resposta correcta: s'hauria de nombrar algun dels exemples d'ús (stay up to date on industry, analyze recent news, understand the life science landscape and monitor competitors).</p>	
6	<p>Pregunta: <i>For which industry can be useful?</i></p> <p>Resposta: <i>BioWorld can be useful in the biopharmaceutical and medical technology industry.</i></p> <p>Contingut que hauria de tenir la resposta correcta: Biopharma & medtech</p>	<p>10/10</p> <p>Resposta correcta, entén context</p>
7	<p>Pregunta: <i>With which frequency is updated the data?</i></p> <p>Resposta: <i>The data in BioWorld is updated on a daily, breaking, and weekly basis.</i></p> <p>Contingut que hauria de tenir la resposta correcta: Frequency of updates: Daily, Breaking and Weekly</p>	<p>10/10</p> <p>Resposta correcta, escriu el mateix que s'especifica en els documents</p>
8	<p>Pregunta: <i>who are the competitors of Competitive Cortellis Intelligence?</i></p> <p>Resposta: <i>The competitors of Competitive Cortellis Intelligence include Citeline (Datamonitor Healthcare), GlobalData, and Evaluate.</i></p> <p>Contingut que hauria de tenir la resposta correcta: Datamonitor, GlobalData, Evaluate</p>	<p>7/10</p> <p>Resposta parcialment correcta, inclou informació no completament acurada.</p>

Com es pot veure, les respostes obtingudes per part de l'assistent virtual són bastant precises, realistes i proporcionen les respostes corresponents a la informació que s'està buscant dins dels documents. A més, s'ha pogut comprovar el bon funcionament al fer preguntes relacionades entre elles on ha trobat correctament el context.

Altres proves

A part del bon funcionament del xat, es comprova que les funcionalitats proposades, com la consulta de si s'han afegit nous documents, funcionen de manera adequada. Es descarreguen els nous fitxers, s'afegeixen a la base de dades vectorial, es notifica a l'usuari que s'ha afegit nova informació (**Figura 14**) i es modifica l'última hora de consulta a la carpeta.



Figura 14. Missatge del sistema al xat quan es carreguen nous documents

Entre altres controls que es produeixen és si l'usuari introdueix incorrectament les credencials d'OpenAI; per tant, no es podria accedir a l'API i no es podrien contestar les preguntes de l'usuari. En cas de donar-se aquest error, es notifica amb un missatge d'error corresponent (**Figura 15**) i el sistema indica que la clau és incorrecta. Un altre cas d'error es podria donar si hi ha un error en la descàrrega d'algun document, que també mostraria el seu propi missatge d'error.

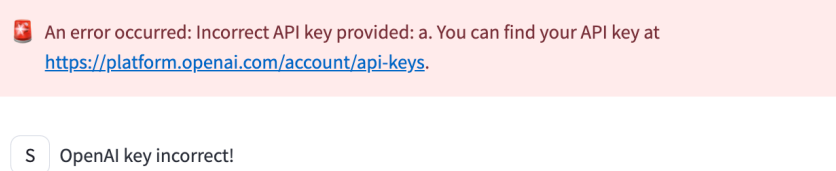


Figura 15. Missatge d'error en cas d'introduir l'OpenAI key incorrecta.

7. Conclusió

En aquest projecte, s'ha abordat el repte de simplificar la gestió de documents en empreses mitjançant la intel·ligència artificial, realitzant una investigació exhaustiva sobre les eines més adients. La selecció i prova d'aquestes eines han estat un procés meticulós, el qual s'ha intentat que quedi reflectit acuradament en aquesta memòria. En un camp tan dinàmic i en evolució com és la IA, on l'accés a informació actualitzada pot ser complicat, s'ha profunditzat en conceptes inicialment desconeguts, com els models de llenguatge a gran escala (LLM), permetent entendre eines actuals com és el ChatGPT. Aquest aprenentatge ha proporcionat una visió més àmplia sobre les capacitats i el potencial futur de les IA.

El projecte també ha permès explorar en detall altres mètodes de tractament de dades, com les bases de dades vectorials. Aquesta investigació ha permès ampliar la comprensió sobre l'organització, els algorismes i el processament de les dades d'aquest format. L'aplicació d'aquestes noves metodologies ha resultat en un enriquiment professional significatiu.

L'experiència de desenvolupar una interfície d'usuari interactiva amb Streamlit ha estat particularment enriquidora, marcant un pas important en el desenvolupament de capacitats en el *front-end*. Aquesta experiència ha demostrat que sempre hi ha espai per a la innovació, especialment en la integració d'eines avançades com la IA generativa amb serveis quotidians com és el Google Drive. Combinar aquests dos elements ha demostrat com tecnologies actuals com la IA pot ser aplicada de maneres creatives i eficients.

Els resultats obtinguts en aquest projecte han complert amb la majoria dels requisits funcionals i no funcionals establerts, considerant-se satisfactoris per a un treball de final de grau. Encara que l'aplicació no estigui preparada per a un desplegament comercial a gran escala, ha demostrat èxit com a projecte acadèmic i ofereix una sòlida base per a futures investigacions i desenvolupaments.

Així doncs, es pot concloure que el projecte ha estat extremadament productiu, tant per l'ampli ventall de coneixements adquirits com per la possibilitat de continuar desenvolupant aquest treball fora de l'entorn acadèmic com a projecte personal. Aquest projecte ha demostrat que és essencial continuar aprenent en l'àmbit de la tecnologia i que introduir noves eines pot portar a innovacions significatives en diversos camps de treball.

8. Bibliografía

- [1] Oracle, “¿Qué es la IA? Conoce la inteligencia artificial,” Oracle Cloud Infrastructure. Accessed: Jan. 13, 2024. [Online]. Available: <https://www.oracle.com/es/artificial-intelligence/what-is-ai/>
- [2] Wikipedia, “Inteligencia artificial,” Viquipèdia. Accessed: Jan. 13, 2024. [Online]. Available: https://ca.wikipedia.org/wiki/Intel%C2%B7lig%C3%A8ncia_artificial
- [3] Tableau, “What are The Top Machine Learning (ML) Methods?,” Salesforce. Accessed: Jan. 13, 2024. [Online]. Available: <https://www.tableau.com/es-es/learn/articles/top-machine-learning-methods>
- [4] Wikipedia, “Generative artificial intelligence,” Wikipedia. Accessed: Jan. 13, 2024. [Online]. Available: https://en.wikipedia.org/wiki/Generative_artificial_intelligence
- [5] B. Banjara, “LLMs in Conversational AI: Building Smarter Chatbots & Assistants,” Analytics Vidhya. Accessed: Jan. 08, 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2023/07/llms-in-conversational-ai/?utm_source=related_WP&utm_medium=https://www.analyticsvidhya.com/blog/2023/03/an-introduction-to-large-language-models-llms/
- [6] codebasics, *LLM Explained | What is LLM*, (Aug. 22, 2023). Accessed: Jan. 04, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=67_aMPDk2zw&ab_channel=codebasics
- [7] M. Ibrahim, “An Overview of Large Language Models (LLMs),” Weights & Biases. Accessed: Jan. 08, 2024. [Online]. Available: <https://wandb.ai/mostafaibrahim17/ml-articles/reports/An-Overview-of-Large-Language-Models-LLMs---VmldzozODA3MzQz>
- [8] OpenAI, “Models.” Accessed: Jan. 04, 2024. [Online]. Available: <https://platform.openai.com/docs/models>
- [9] B. Muller, “BERT 101 🤖 State Of The Art NLP Model Explained,” HuggingFace. Accessed: Jan. 08, 2024. [Online]. Available: <https://huggingface.co/blog/bert-101#3-bert-model-size--architecture>
- [10] Road to Data, “Los mejores modelos de lenguaje de código abierto,” LinkedIn. Accessed: Jan. 08, 2024. [Online]. Available: <https://www.linkedin.com/pulse/los-mejores-modelos-de-lenguaje-c%C3%B3digo-abierto-road-to-data-ncohf/?originalSubdomain=es>
- [11] OpenAI, “New models and developer products announced at DevDay.” Accessed: Jan. 09, 2024. [Online]. Available: <https://openai.com/blog/new-models-and-developer-products-announced-at-devday>

- [12] Wikipedia, “LLaMA,” Wikipedia. Accessed: Jan. 08, 2024. [Online]. Available: <https://en.wikipedia.org/wiki/LLaMA>
- [13] A. Di Nicola, “In-Depth Comparison: GPT-4 vs GPT-3.5,” Kanaries. Accessed: Jan. 10, 2024. [Online]. Available: <https://docs.kanaries.net/articles/compare-gpt-4-gpt-3>
- [14] Wikipedia, “GPT-4,” Wikipedia. Accessed: Jan. 08, 2024. [Online]. Available: <https://en.wikipedia.org/wiki/GPT-4>
- [15] IEBS, “¿Qué es un prompt en IA y cómo crear uno?,” IEBS Blog. Accessed: Jan. 13, 2024. [Online]. Available: <https://www.iebschool.com/blog/que-es-prompt-en-ia-tecnologia/>
- [16] Learn Prompting, “Prompt Engineering Guide.” Accessed: Jan. 08, 2024. [Online]. Available: <https://learnprompting.org/docs/category/-basics>
- [17] OpenAI, “Prompt Engineering,” Documentation OpenAI. Accessed: Jan. 08, 2024. [Online]. Available: <https://platform.openai.com/docs/guides/prompt-engineering>
- [18] A. Takyar, “BRIDGING THE AI-HUMAN COMMUNICATION GAP: A COMPREHENSIVE GUIDE TO PROMPT ENGINEERING,” LeewayHertz. Accessed: Jan. 08, 2024. [Online]. Available: <https://www.leewayhertz.com/prompt-engineering/>
- [19] Roie Schwaber-Cohen, “What is a Vector Database & How Does it Work? Use Cases + Examples.” Accessed: Jan. 08, 2024. [Online]. Available: <https://www.pinecone.io/learn/vector-database/>
- [20] A. Arora, “Demystifying Document Question-Answering Chatbot - A Comprehensive Step-by-Step Tutorial with LangChain,” Aman Arora’s Blog. Accessed: Jan. 10, 2024. [Online]. Available: https://amaarora.github.io/posts/2023-07-27_Document_Question_Answering_with_LangChain.html
- [21] codebasics, *Vector Database Explained | What is Vector Database?*, (Sep. 09, 2023). Accessed: Jan. 08, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=72XgD322wZ8&ab_channel=codebasics
- [22] LangChain, “OpenAI.” Accessed: Jan. 04, 2024. [Online]. Available: <https://python.langchain.com/docs/integrations/platforms/openai>
- [23] G. Neshet, “Introduction to LangChain,” sempahore. Accessed: Jan. 10, 2024. [Online]. Available: <https://semaphoreci.com/blog/langchain>
- [24] codebasics, *LangChain Crash Course For Beginners | LangChain Tutorial*, (Jun. 30, 2023). Accessed: Jan. 08, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=nAmC7SoVLd8&ab_channel=codebasics
- [25] O. Mishra, “Using langchain for Question Answering on Own Data,” Medium. Accessed: Jan. 09, 2024. [Online]. Available:

- <https://medium.com/@onkarmishra/using-langchain-for-question-answering-on-own-data-3af0a82789ed>
- [26] LangChain Inc., “LangChain.” Accessed: Jan. 11, 2024. [Online]. Available: https://python.langchain.com/docs/get_started/introduction
- [27] freeCodeCamp, *Vector Embeddings Tutorial – Code Your Own AI Assistant with GPT-4 API + LangChain + NLP*, (Sep. 03, 2023). Accessed: Jan. 08, 2024. [Online Video]. Available: https://www.youtube.com/watch?v=yfHHvmaMkcA&ab_channel=freeCodeCamp.org
- [28] Pinecone Inc., “Pinecone - Start free, scale effortlessly.” Accessed: Jan. 09, 2024. [Online]. Available: <https://www.pinecone.io/pricing/>
- [29] Google, “Google Cloud.” Accessed: Jan. 09, 2024. [Online]. Available: <https://console.cloud.google.com/welcome/new?project=test-408312>
- [30] M. Daly, “Create a Google Cloud Platform service account,” Medium. Accessed: Jan. 09, 2024. [Online]. Available: <https://medium.com/@matheodaly.md/create-a-google-cloud-platform-service-account-in-3-steps-7e92d8298800>
- [31] GeeksforGeeks, “Get list of files and folders in Google Drive storage using Python.” Accessed: Jan. 09, 2024. [Online]. Available: <https://www.geeksforgeeks.org/get-list-of-files-and-folders-in-google-drive-storage-using-python/>
- [32] M. Daly, “Using Google Drive API with Python and a service account,” Medium. Accessed: Jan. 08, 2024. [Online]. Available: <https://medium.com/@matheodaly.md/using-google-drive-api-with-python-and-a-service-account-d6ae1f6456c2>
- [33] C. Dhongade, “Python and Google Drive: How to List and Create Files and Folders : 2023,” DEV. Accessed: Jan. 03, 2024. [Online]. Available: <https://dev.to/binaryibex/python-and-google-drive-how-to-list-and-create-files-and-folders-2023-2nmm>
- [34] Snowflake Inc, “Streamlit · A faster way to build and share data apps.” Accessed: Jan. 10, 2024. [Online]. Available: <https://streamlit.io/>
- [35] A. Treuille and A. Kelly, “Generative AI and Streamlit: A perfect match.” Accessed: Jan. 09, 2024. [Online]. Available: <https://blog.streamlit.io/generative-ai-and-streamlit-a-perfect-match/>
- [36] lakeFS, “Best 15 Vector Databases for 2024 [Top Picks].” Accessed: Jan. 09, 2024. [Online]. Available: <https://lakefs.io/blog/12-vector-databases-2023/>

- [37] Actueloop, “Deep Lake White Paper.” Accessed: Jan. 09, 2024. [Online]. Available: <https://files.elfsightcdn.com/eafe4a4d-3436-495d-b748-5bdce62d911d/b7733ecd-3d64-44ea-8bc0-e5373bc2402d/Deep-Lake-White-Paper-by-Actueloop.pdf>
- [38] Actueloop, “Deep Lake Docs - Deep Lake.” Accessed: Jan. 09, 2024. [Online]. Available: <https://docs.actueloop.ai/>
- [39] OpenAI, “Pricing,” OpenAI. Accessed: Jan. 10, 2024. [Online]. Available: <https://openai.com/pricing>
- [40] D. Nunes, R. Primi, R. Pires, R. Lotufo, and R. Nogueira, “Evaluating GPT-3.5 and GPT-4 Models on Brazilian University Admission Exams,” Mar. 2023.
- [41] OpenAI, “Overview - OpenAI API.” Accessed: Jan. 11, 2024. [Online]. Available: <https://platform.openai.com/docs/overview>
- [42] F. Alonso, “¿Qué son y como funcionan los Large Language Models?,” Future Space S.A. Accessed: Jan. 13, 2024. [Online]. Available: <https://www.futurespace.es/large-language-models/>
- [43] G. A. Aye, S. Kim, and H. Li, “Learning Autocompletion from Real-World Datasets,” Nov. 2020.

9. Annex

9.1. Instal·lació

Per poder fer ús de l'aplicació creada és necessari descarregar el codi, això es pot fer clonant el repositori de GitHub en l'ordinador personal. Un cop clonat és important instal·lar totes les dependències que té el codi, aquestes estan especificades en el fitxer "requeriments.txt".

9.2. Requisits

Entre els requisits necessaris per fer ús del codi s'ha de tenir Python instal·lat. A més, s'ha de registrar i obtenir:

- Una clau d'OpenAI.
- Credencials Actueloop: un usuari i la clau.

També és important estar registrat a Google Cloud i crear un compte de servei. A aquest compte se li ha de donar permisos d'accés a la carpeta on estan pujats els documents amb els quals volem interactuar.

Nota: per poder tenir una millor experiència és interessant fer una inversió en l'API d'OpenAI.

9.3. Execució

Per executar i accedir al projecte s'ha d'executar el script principal, per fer-ho, en una terminal s'introdueix:

```
streamlit run chat.py
```

Un cop apareix la interfície en una finestra de navegador introdueix les credencials a la barra lateral, i ja està a punt per fer ús.