Disseny i desenvolupament d'un xatbot de suport a l'usuari basat en models generatius

Pol Fombona Delgado

Resum— Els xatbots són programes d'ordinador basats comunment en models de processament de llenguatge natural (NLP) que simulen conversacions humanes i, en entorns de suport a l'usuari, es poden usar com eines per a proporcionar ajuda i resoldre dubtes senzills de forma automatitzada. Els avenços en els models generatius en els últims anys han permès desenvolupar models més eficients i complexes que no estan condicionats per les limitacions dels models tradicionals com poden ser les respostes genèriques o la dificultat per entendre el context d'una conversació. Aquest projecte té com a objectiu avaluar la viabilitat d'utilitzar un model generatiu per a ajudar a un xatbot basat en regles en aquelles preguntes en les quals es confon o no genera una resposta adient.

Paraules clau— xatbot, processament de llenguatge natural, models generatius, suport a l'usuari, GPT, openaAI, experiència d'usuari, intel·ligència artificial, IA conversacional, satisfacció d'usuari, seguretat, Vicuna, LLaMA, Facebook, codi obert, Koala.

Abstract—Chatbots are computer programs commonly based on natural language processing (NLP) models that simulate human conversations and, in user support environments, can be used as tools to provide help and resolve simple queries in an automated way. Advances in generative models in recent years have allowed the development of more efficient and complex models that are not conditioned by the limitations of traditional models such as generic responses or difficulty in understanding the context of a conversation. This project aims to evaluate the feasibility of using a generative model to assist a rule-based chatbot in those qüestions where it gets confused or cannot generate an adequate response.

Index Terms— chatbot, natural language processing, generative models, user support, GPT, OpenAI, user experience, artificial intelligence, conversational AI, user satsifcation, security, Vicuna, LLaMA, Facebook, Open Source, Koala.

1 Introducció - Context del treball

A Mesura que les empreses busquen proporcionar atenció als usuaris i clients més eficient i personalitzada, la implementació i ús de xatbots s'ha estès de manera general a tots els àmbits, des d'empreses de comerç en línia fins a companyies telefòniques. Els xatbots tradicionals, com poden ser IBM Watson [1] o Noa de CaixaBank [2], tenen limitacions a causa de la seva natura, com pot ser la generació de respostes genèriques o la inhabilitat de comprendre el context d'una conversa. Com a conseqüència, l'interès en models alternatius, amb la capacitat d'interactuar de manera més humana, ha augmentat.

L'objectiu principal d'aquest projecte consisteix a proporcionar una visió sobre la viabilitat de complementar un xatbot basat en regles amb un model generatiu per a millorar la precisió de les respostes en aquelles preguntes en què el model de regles no les entén o que la resposta generada no és adient. Específicament, s'avaluarà l'efectivitat dels models generatius per a entendre la intenció de la conversa de l'usuari basant-se en el context d'aquesta, la generació de respostes adaptades a les necessitats d'acord amb les polítiques de l'empresa i, mitjançant mètriques, l'eficiència de la implementació. A més a més, també s'estudiarà les possibles implicacions de seguretat de fer servir aquests models, com pot ser extreure informació confidencial manipulant el xatbot.

El resultat d'aquest projecte proporcionarà dades útils a les empreses per a prendre decisions informades sobre l'ús de models generatius per a complementar els xatbots tradicionals i idees per al desenvolupament de sistemes de suport al client més efectius i eficients.

2 Objectius

Per a dur a terme aquest projecte, s'ha definit un objectiu global: proporcionar una visió sobre la viabilitat de complementar un xatbot basat en regles amb un model generatiu per a millorar la precisió de les respostes. També s'han definit un seguit de subobjectius relacionats amb el global:

1. Comparar l'efectivitat dels xatbots tradicionals amb els basats en models generatius.
2. Avaluar la capacitat de comprensió de la intenció de l'usuari i generar respostes adequades.
3. Identificar possibles riscos de seguretat.

Per assolir aquests objectius, es proposa adaptar un model de xatbot GPT-3.5 d'OpenAI [3] conjuntament amb un script de Python per a gestionar els processos, com es mostra en la figura 1. Aquest xatbot es basa en models de llenguatge gran (LLM, per les seves sigles en anglés), els quals consisteixen en xarxes neuronals amb milers de

- *E-mail de contacte: pol.fombona@autonoma.cat*
- *Menció realitzada: Enginyeria de Dades*
- *Treball tutoritzat per: Antonio Espinosa Morales (Departament d'Arquitectura de Computadors i Sistemes Operatius)*
- *Curs 2022/23*

milions de pesos i paràmetres que utilitzen generalment l'arquitectura de transformadors, el que permet que s'adaptin a una àmplia gamma de tasques en comptes de ser entrenats per a una tasca específica.

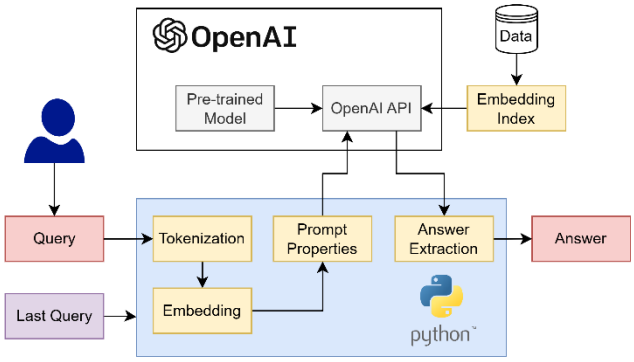


Fig. 1: Proposta d'arquitectura del projecte

L'element principal del disseny és el model de xatbot que s'utilitzarà per a processar les preguntes, buscar informació relacionada i generar una resposta coherent. Hi ha diferents models disponibles que es poden fer servir, però per a fer les proves es treballarà amb el model "Curie", ja que és un model balancejat respecte al cost i poder computacional. Aquest model, per defecte, no té accés a la informació necessària per a poder respondre les preguntes perquè només pot accedir a la informació amb la qual es va entrenar i, per la qual cosa, per afegir noves dades, se seguirà un procés anomenat *embedding*, que consisteix a generar una representació vectorial del text de manera que es pot obtenir una mesura de la relació entre dos texts mitjançant el càlcul de la "distància" entre vectors. Per cada registre del conjunt de les dades d'entrenament, es generarà la seva representació vectorial utilitzant l'API d'OpenAI [4] i es desarà en un dataset (*embedding index*). Aquest índex posteriorment permetrà comparar els *embeddings* de les preguntes amb les dades d'entrenament i trobar les dades més similars per a generar un context. La creació d'aquest índex permetrà que el model "assimili" noves dades sense haver de modificar l'estructura o els pesos del model original, no obstant això, l'ús d'*embeddings* presenta limitacions com la pèrdua d'informació numèrica, ja que estan dissenyats principalment per identificar relacions entre variables textuals, o una possible disminució de l'eficàcia quan els vectors generats presenten una dimensionalitat molt elevada.

Una alternativa utilitzada quan es vol entrenar a un model amb informació nova és la metodologia de *fine-tuning* que implica l'adaptació dels pesos de la xarxa, de manera total o parcial, a partir de noves dades per a obtenir millors resultats per a una tasca especifica. S'ha decidit no fer servir aquesta metodologia en el projecte, degut al fet que no només volem augmentar el coneixement del model sobre un tòpic en concret, sinó que també volem limitar el coneixement al qual té accés únicament a aquest tòpic perquè només respongui amb les dades proveïdes durant l'entrenament, i això el *finetuning* no ho permet.

Respecte a les preguntes dels usuaris, es realitzarà un procés similar a l'explicat per a l'entrenament, el text d'entrada es dividirà en tokens i, posteriorment, es farà un *embedding*, que s'utilitzarà per generar el context i la resposta.

Com que el model no pot processar tot el text directament, ja que té una mida màxima d'entrada determinada per tokens, que són la unitat bàsica de mesura de quantitat de text, prèviament als processos dels diferents mòduls s'ha de dividir el text en fragments de mida inferior a aquest límit.

Addicionalment, també es realitzaran proves amb dos models generatius de codi obert, els xatbots "Koala 13B" [5] i "Vicuna 13B" [6], els quals es basen en el model de llenguatge "LLaMA" [7]. El primer està entrenat mitjançant *fine-tuning* sobre el model LLaMA amb dades conversacionals. Aquest dataset d'entrenament, a diferència dels utilitzats per altres xatbots com ChatGPT o Bard, està format per dades conversacionals de llicència oberta (*Open Source Data*) i conversacions entre usuaris i LLM. El xatbot Vicuna, també està desenvolupat a partir d'aplicar *fine-tuning* al model LLaMA, però a diferència del model Koala, només ha estat entrenat amb dades generades per conversacions entre usuaris i ChatGPT. Fer servir aquests dos models permetrà fer comparacions en termes de precisió, eficiència i costos, amb l'objectiu de garantir que els resultats obtinguts i la implementació del sistema no estiguin limitats a un únic model propietari, sinó que siguin adaptables i aplicables a altres models.

3 Metodologia i Planificació

La metodologia que se seguirà en aquest projecte és el Producte Mínim Variable (*MVP*). Aquesta metodologia permet enfocar els esforços a desenvolupar un producte amb el conjunt mínim de funcions necessàries per a satisfer l'objectiu principal, això permet validar l'estat del producte i afegir nous canvis basats en els comentaris de l'empresa col·laboradora. Per a garantir el correcte funcionament de les funcions bàsiques, el projecte es dividirà en cicles, els quals inicialment estan centrats a desenvolupar les funcions bàsiques i primordials i posteriorment les funcionalitats addicionals, com poden ser noves directives o mesures de seguretat.

Durant les primeres setmanes del projecte, el desenvolupament estarà dividit en intervals d'una o dues setmanes segons la complexitat de les tasques a realitzar com es pot veure en l'annex A1 (Taula 1). Posteriorment, entrarà en una etapa d'iteracions a on es durà a terme un

seguit de tasques per a desenvolupar un prototip del xatbot i aquest s'avaluarà, identificant les possibles millores i problemes i aplicant aquest coneixement en la següent iteració.

Per a desenvolupar el codi del projecte s'utilitzarà l'eina Google Colab per a tenir un entorn de desenvolupament amb els requisits de computació necessaris i es farà servir Microsoft Teams per a comunicar-se amb el tutor i l'empresa col·laboradora durant la realització del projecte.

4 Estat de l'art

Actualment, és freqüent trobar implementacions de xatbots en diversos sectors, com ara el comerç en línia o educació. La implementació d'aquesta tecnologia ajuda a les empreses a agilitzar l'atenció als usuaris, reduint costos i eliminant tasques repetitives dels treballadors d'atenció al client.

Els xatbots més comuns per a suport a l'usuari són els basats en regles, ja que són models relativament simples de desenvolupar i implementar. Aquests són molt efectius per negocis que tenen un gran volum de peticions per a dur a terme tasques simples com podria ser el restabliment de contrasenyes, seguiment d'ordres o dubtes sobre polítiques de l'empresa, entre d'altres. Un exemple d'aplicació d'un xatbot es pot trobar al banc "Bank of America" que utilitza Erica [8] com a suport a l'usuari. Erica és un xatbot dissenyat amb l'objectiu de proporcionar ajuda personalitzada als clients amb tasques com poden ser comprovar la quantitat de fons, pagar factures o transferir diners a altres comptes i, fins i tot, es pot demanar assessorament financer, com es pot veure en l'annex A2. Altres exemples d'implementacions de xatbots basats en regles es poden trobar en el sector de transport de mercaderies, com l'empresa UPS amb el xatbot UPS Bot [9]. No obstant això, aquests models presenten algunes limitacions, com no entendre missatges molt complexos o llargs el que causa que no es pugui resoldre el problema de l'usuari. Una altra limitació és la dificultat per a tractar amb dades molt específiques, ja que aquests models no tenen capacitat per a raonar i, per tant, quan l'usuari té alguna consulta especifica, com per exemple, un dubte sobre el cost d'un paquet amb unes mesures de pes i volum exactes, el xatbot no entén la pregunta i genera una resposta inadequada.

Per a solucionar aquestes limitacions s'han desenvolupat xatbots generatius basats en LLM. Tot i que l'ús d'aquests models en el camp de suport a l'usuari és poc comú actualment, es pot fer una aproximació del seu funcionament mitjançant el xatbot conversacional "ChatGPT" d'OpenAI [10]. Aquest xatbot s'ha desenvolupat a partir de l'arquitectura de transformadors generatius preentrenats (amb acrònim en anglès GPT) de la classe de LLM, específicament, s'ha desenvolupat a partir del model GPT-3.5 d'OpenAI mitjançant *fine-tuning* i una combinació d'aprenentatge supervisat i per reforç utilitzant dades conversacionals.

Respecte al processament de la informació, l'ús de dades en format de text no es eficient per al processament en grans volums utilitzant models de llenguatge natural. Per aquest motiu, s'opta per tècniques d'*embedding*, que proporcionen una representació més efectiva i eficient del text. Com s'ha mencionat en l'apartat 2, els *embeddings* són representacions numèriques de paraules en un espai vectorial de dimensió finita que contenen informació semàntica i estructural. Els mètodes més populars per a generar aquests vectors són Word2Vec [11] per les paraules i Doc2Vec [12] per els documents. Aquests consisteixen a entrenar una xarxa neuronal amb una gran quantitat de text per a aprendre les relacions entre les paraules i posteriorment assignar un vector a cada paraula en un espai vectorial definit. Aquests vectors estan organitzats de manera que paraules similars o amb context compartit es troben properes mentre que paraules no relacionades es troben a distàncies més grans.

5 Dades

Les dades d'entrenament i validació són un element clau en els models de xatbot, tant generatius com basats en regles, ja que aquestes permeten al model aprendre i comunicar-se millor. La qualitat del conjunt de dades té un impacte directe en la precisió de les respostes i, per tant, influència directa en la satisfacció dels usuaris.

L'empresa col·laboradora ha proporcionat dos datasets en castellà per a dur a terme el projecte. El primer consisteix en un registre que conté informació rellevant sobre les preguntes més recurrents (conegudes com a FAQ, acrònims de *Frequently Asked Questions* en anglès) i el segon consisteix en una selecció de converses telefòniques entre usuaris i el xatbot de l'empresa. Cal destacar que aquesta selecció no constitueix una mostra representativa de la distribució real de les preguntes, sinó que s'ha focalitzat en incloure una àmplia gamma dels diferents temes de preguntes que solen ser formulades al xatbot.

A partir del primer dataset s'obtindrà un índex d'embeddings (*embedding index*) que servirà com a base de coneixement pel xatbot, mentre que el segon dataset s'utilitzarà per a simular les preguntes (*query*) dels usuaris durant la validació. Aquestes dades corresponen a l'entrada del mòdul *Embedding Module* de l'arquitectura mostrada en la figura 2.

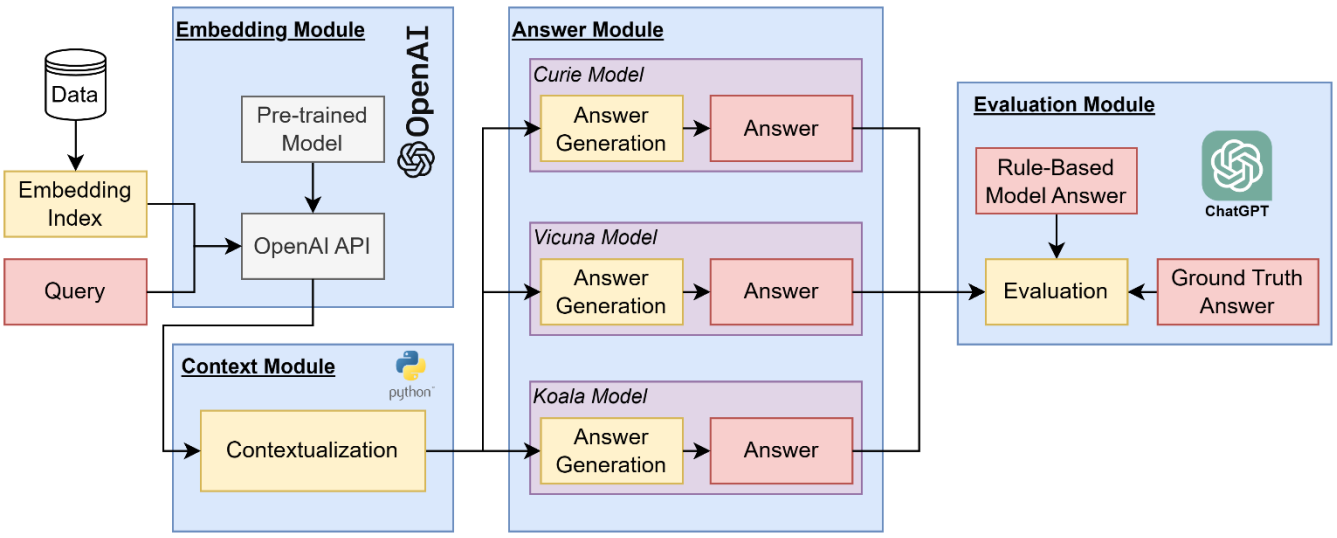


Fig. 2: Diagrama modular de l'arquitectura del projecte

6 Arquitectura

Per a dur a terme la implementació del projecte, s'ha proposat l'ús del diagrama representat a la figura 2, que està compost per 4 mòduls: *embedding*, contextualització, respostes i avaluació, cadascun amb una tasca específica.

El mòdul d'*embedding* (*Embedding Module*) té com a finalitat generar una representació vectorial de les dades d'entrada, utilitzant l'API d'OpenAI [4] i el model d'*embedding* "text-embedding-ada-002". Aquesta tasca es pot dividir en dues etapes: en primer lloc, es realitza el procés de vectorització (*embedding*) del dataset de preguntes i respostes, generant un dataset amb la representació vectorial que es desa localment, així, en afegir o eliminar dades al dataset només és necessari modificar l'arxiu local sense haver de tornar a generar-lo sencer. En la segona etapa, es genera l'*embedding* específic per a cada pregunta independentment.

El segon mòdul (*Context Module*) es dedica a la generació del context de les preguntes, és a dir, a l'obtenció de les parts del text d'entrenament que estan relacionades amb la pregunta. Aquest procés es duu a terme mitjançant el càlcul de "distàncies" entre les representacions vectorials. Posteriorment, es genera un objecte que conté la pregunta i el context que serveix com a entrada per al següent mòdul. Aquestes dades poden ser compartides entre els tres models generatius, de manera que només és necessari dur a terme el procés una sola vegada, el que suposa una optimització del temps i dels recursos, i afavoreix la consistència dels resultats assolits.

Respecte al tercer mòdul (*Answer Module*), aquest és l'encarregat de generar les respostes a partir de la pregunta i el context obtingut en el mòdul anterior. En aquest mòdul és on s'utilitzen els diferents models (Curie, Koala i Vicuna), és a dir, és l'única part de l'arquitectura que és diferent per a cada model. La resposta generada és la que veuria l'usuari en el cas de la implementació real, i aquesta s'envia al mòdul d'avaluació per a poder avaluar el rendiment i eficàcia de cadascun dels models.

Finalment, el mòdul d'avaluació (*Evaluation Module*) té com a finalitat proporcionar mètriques, manuals i automàtiques, del funcionament i rendiment dels diferents models i processos per a poder obtenir una visió sobre la viabilitat d'utilitzar models generatius.

7 Configuració Experimental

El projecte en conjunt pot ser desglossat en les següents parts: obtenció de dades i preprocessament, transformació del dataset d'entrenament en *embeddings*, contextualització, implementació de models generatius, generació de la resposta i avaluació.

7.1 Obtenció de dades i preprocessament

Com s'ha esmentat en l'apartat 5, per a dur a terme el projecte s'han utilitzat dos datasets proporcionats per l'empresa col·laboradora. El primer conté informació sobre les preguntes més freqüents i es farà servir per a fer l'entrenament amb *embeddings*. Aquest dataset conté informació addicional no rellevant com poden ser les traduccions o la categoria del producte i, per tant, s'ha simplificat de manera que només s'ha mantingut la pregunta i la resposta. Respecte a la resposta, s'ha desat de dues maneres diferents, la primera només conté la primera interacció amb l'usuari sense cap ramificació, per exemple: "Todas las tarjetas tienen un límite establecido [...]. Si quieres más información de la tarjeta Y di 1, si quieres más información sobre la tarjeta Z di 2.", d'ara endavant anomenada resposta simplificada. La segona conté una resposta per a cada opció, és a dir, es desa cada ramificació individualment, d'ara endavant resposta desenvolu-

pada. Això permet fer un estudi amb més detall sobre el rendiment del model generatiu segons el tipus de pregunta i resposta a generar. Posteriorment, s'han eliminat els caràcters especials, com els salts de línia, i s'ha convertit tot el text a minúscules per a facilitar el procés d'anonimització.

Considerant la sensibilitat de la informació amb què es treballa en aquest projecte i l'ús d'eines en línia que poden retenir la informació compartida, s'ha dut a terme un procés d'anonimització de les dades. Aquest procés ha implicat la substitució de tota informació personal que pugui identificar a l'empresa per dades fictícies. D'aquesta manera, les dades resultants són de naturalesa genèrica i no permeten cap procés d'identificació directa.

En relació amb el segon dataset, aquest consisteix en una mostra de 200 registres de converses telefòniques entre clients i el xatbot i serà utilitzat per a realitzar el testeig i avaluació del xatbot desenvolupat. Com amb el dataset anterior, s'ha dut a terme un preprocessament de les dades mantenint només les dades essencials, és a dir, la pregunta, la resposta generada i la validesa d'aquesta i s'han anonimitzat. El conjunt final de dades consta de 74 preguntes amb respostes generades correctes i 126 amb respostes incorrectes o inadequades, fent un total de 200, tanmateix, no hi ha cap registre sense resposta generada.

7.2 Transformació del dataset d'entrenament en embeddings

Per a obtenir la representació vectorial de les dades d'entrenament o *embedding*, s'ha optat per utilitzar l'API d'OpenAI [4], més concretament la funció "Embedding.create", que rep com a entrada el text i el model a utilitzar. En aquest cas, es farà servir el model "text-embedding-ada-002", ja que és un model econòmic, potent i fàcil de fer servir. Respecte al text d'entrada, com que la mida màxima és de 8191 tokens, abans de fer la crida a la funció s'ha de realitzar un procés per a determinar la mida en tokens de les dades, mitjançant la llibreria de Python "tiktoken" [13] i, si alguna entrada supera la mida permesa, s'ha de dividir en parts més petites mantenint l'estructura de pregunta-resposta. Finalment, es crida a la funció i aquesta retorna la representació vectorial del text.

7.3 Contextualització

Com que els models utilitzats no poden accedir directament a les dades d'entrenament, s'afegeix un petit text, anomenat context, per ajudar al model a formular una resposta adient, on aquest serà el mateix pels tres models utilitzats. El procediment per a obtenir aquest context es pot dividir en etapes com es pot observar en la figura 3. En primer lloc, és necessari generar l'*embedding* de la pregunta realitzada, de manera similar a com s'ha generat per a les dades d'entrenament. A continuació, es calcula la similitud de cosinus i s'ordenen les entrades del dataset d'entrenament de manera ascendent segons aquesta mètrica i se seleccionen les "n" entrades superiors, on "n" ve determinada per la mida màxima d'entrada de text més petita entre tots els models. D'aquesta manera, aconseguim un petit text de preguntes i respostes que està molt relacionat amb la pregunta feta i que ajudarà als models a generar una resposta adient.

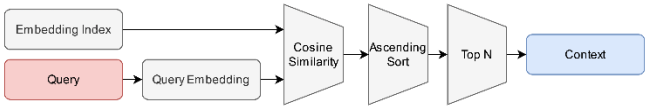


Fig. 3: Procés per a generar el context d'una pregunta

7.4 Implementació models generatius

La implementació dels tres models utilitzats en aquest projecte presenta diferències específiques per a cadascun. Respecte al model "Curie" d'OpenAI, s'ha desenvolupat mitjançant un *script* de Python que realitza crides a l'API d'OpenAI per a gestionar el model. Així, no es requereix una gran capacitat de còmput localment per a la seva execució, però està limitat a les restriccions de l'API, com poden ser el nombre de crides per minut, que s'explica més detalladament en l'apartat 9. *L'script* de Python està estructurat en un objecte classe, com es pot veure en l'annex A3, que conté diverses funcions que serveixen per processar dades, generar *embeddings*, calcular tokens, crear context i generar *prompts* entre altres.

Pel que fa als altres dos models, Koala i Vicuna, no tenen disponible cap API per a executar-los mitjançant crides i requereixen una gran capacitat computacional per a executar-los en l'àmbit local. És per això que s'ha optat per una implementació en línia anomenada "FastChat" [14], que és una plataforma oberta per a l'entrenament i avaluació de diferents models de xatbots entre els quals es troben els dos esmentats anteriorment. Aquesta implementació ha permès obtenir una idea general del funcionament d'aquests dos models abans de fer cap inversió econòmica per a executar-los en l'àmbit local, tanmateix, això comporta uns riscos i limitacions que s'han de tenir en compte.

7.5 Generació de respostes

Un aspecte de gran importància en els models generatius és el que es coneix com a "prompt", que fa referència a l'entrada o instrucció proporcionada al model per a guiar la generació de la resposta. El prompt pot adoptar diverses formes, com ara una pregunta, un conjunt d'instruccions o un text incomplet, i la qualitat i la claredat d'aquest són fonamentals, ja que té un impacte directe en la rellevància i coherència de les respostes generades pel model. En el marc d'aquest projecte, el prompt estarà compost per tres elements: instrucció, context i pregunta. La instrucció fa referència al text que indica al xatbot com ha d'actuar, és a dir, especifica que ha d'actuar com un

xatbot d'una entitat comercial i respondre a la pregunta basant-se en el context proporcionat. El context, tal com s'ha exposat en l'apartat 7.3, consisteix en un petit fragment de text que inclou les preguntes i respostes més similars a la pregunta de l'usuari i té com a finalitat permetre que el model generi una resposta adequada. Finalment, la pregunta és la formulada per l'usuari. Addicionalment, aquest prompt consta d'atributs que es poden modificar per tal d'ajustar la generació de respostes com poden ser la mida i el grau d'aleatorietat.

7.6 Avaluació

Com que l'objectiu d'aquest projecte és fer un estudi sobre la viabilitat d'utilitzar models generatius com a complementaris dels models tradicionals de xatbot, és ideal determinar si el model generatiu és capaç de detectar les respostes incorrectes generades pel xatbot tradicional, per a així poder intervenir i generar una resposta adequada o alertar sobre aquest error. Per tal d'avaluar la competència del model d'OpenAI en aquesta tasca de classificació, es farà servir una matriu de confusió.

Respecte a l'avaluació de la generació del context, es farà servir la mètrica "Mean Reciprocal Rank (MRR)". Aquesta avalua la primera posició del resultat rellevant en un llistat de possibles respostes i, en el cas del projecte, s'avalua a quina posició apareix la resposta més adient a la pregunta en el context generat. El rang d'aquest valor oscil·la entre 0 i 1, sent més proper a 1 quan millor és el context generat.

Avaluar i comparar respostes de diferents models de xatbots és difícil per diversos motius, ja que aquestes depenen de molts factors, com per exemple l'estil de les dades d'entrenament del model, els diferents paràmetres d'entrada o el context de la pregunta. Com s'ha comentat anteriorment, per a facilitar aquesta comparació, s'ha utilitzat el mateix context pels tres models i els mateixos o similars paràmetres d'entrada. Tanmateix, no existeix una mesura objectiva per a avaluar la qualitat d'una resposta, per tant, s'utilitzarà una combinació de tres mètodes d'avaluació.

El primer mètode és la similitud de cosinus, feta servir anteriorment per a generar el context, que permet obtenir una mesura quantitativa de la similitud semàntica entre les respostes generades i l'esperada. Malgrat que aquesta mètrica sembla aplicable de forma independent, això no és del tot cert, a causa del fet que una diferència en un nombre o valor no fa variar molt aquesta similitud, però pot canviar molt el sentit de la resposta. Per aquest motiu, es fa ús del xatbot "ChatGPT" com a segona eina d'avaluació per a les respostes generades. El xatbot rep com a entrada la resposta esperada, així com les respostes generades pels diversos models, i retorna una avaluació quantitativa, en una escala del 1 al 10, comparant-les amb la resposta esperada, oferint una mesura independent i "imparcial" (no se li indica el nom del model que ha generat la resposta per a evitar possibles biaixos). Finalment, el tercer mètode d'avaluació correspon a la validació manual, la qual és indispensable per a validar la fiabilitat i consistència dels resultats obtinguts per les altres mètriques.

La combinació de les tres formes d'avaluació proporciona una anàlisi exhaustiu i precís de la capacitat d'aquests models en comparació amb els tradicionals i de la seva viabilitat per a utilitzar-los en entorns de producció.

8 Resultats

Com s'ha comentat en l'apartat anterior, s'han avaluat els tres aspectes claus del xatbot: classificació de respostes, generació de context i generació de respostes. Mitjançant aquesta aproximació, és possible identificar el punt del procés que és més crític o que té un major potencial d'optimització. Addicionalment, l'avaluació de la generació de respostes s'ha fet per als dos possibles tipus de resposta: la simplificada o la desenvolupada.

8.1 Avaluació classificació de respostes

El conjunt de dades utilitzat per a l'avaluació conté les preguntes formulades pels usuaris, les respostes generades pel xatbot actual de l'empresa, així com la seva veracitat, és a dir, si la resposta és adequada per a la pregunta plantejada. Amb l'objectiu d'avaluar la capacitat del model generatiu per a classificar respostes s'ha generat un prompt per a cada registre, el qual consta de la pregunta, la resposta, un context generat a partir de les dades d'entrenament i la instrucció que indica al model que determini si la resposta és adient o no a la pregunta.

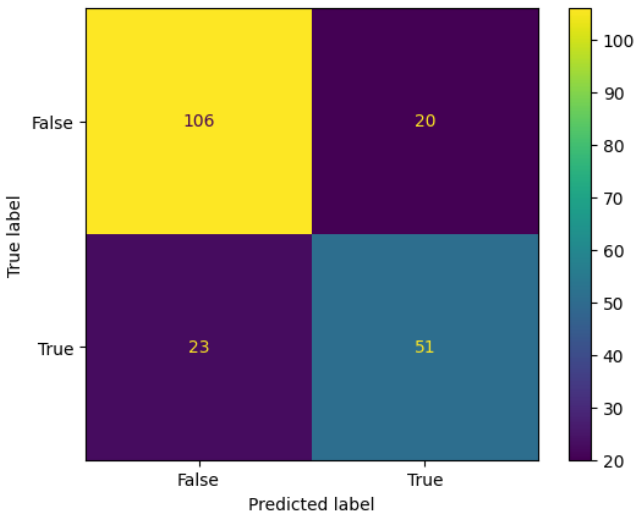


Fig. 4: Matriu de confusió de la classificació de respostes.

Per tal de visualitzar el resultat d'aquesta classificació s'ha fet servir una matriu de confusió, representada en la figura 4, que mostra una precisió de la classificació al voltant del 80%. Com que l'objectiu és permetre que el model generatiu respongui a les preguntes que el model tradicional no respon correctament, cal tenir en compte el valor dels falsos positius, és a dir, la quantitat de respos-

tes incorrectes que s'han classificat com a correctes i s'obté que s'han classificat un 15% com a falsos positius. Els valors obtinguts indiquen que el model classifica les respostes de manera acceptable, tot i això, aquesta mètrica es pot millorar augmentant la quantitat i variabilitat de les dades d'entrenament.

8.2 Avaluació de la Generació de Context

En relació amb l'avaluació del context, s'ha utilitzat la mètrica "Mean Reciprocal Rank" (MRR), tal com s'ha descrit en l'apartat 7.6 i s'ha obtingut un valor de 0.727, el que indica que, de mitjana, la resposta adient a una pregunta es troba en la primera o segona posició del context generat. Cal destacar que aquesta mètrica només té en compte el primer resultat rellevant i no els següents i, per tant, si hi ha dues respostes adients, només tindrà en compte la primera que apareix.

8.3 Avaluació de les Respostes Generades

En darrer lloc, pel que fa a l'avaluació de les respostes generades pels tres models, s'han emprat tres mètriques: la similitud de cosinus, una avaluació automatitzada mitjançant ChatGPT i una avaluació manual.

Per a obtenir el valor de la similitud de cosinus, s'ha creat un dataset amb les preguntes d'avaluació, la resposta correcta i les respostes generades pels diferents models, incloent-hi el model de xatbot actualment en ús. A continuació, s'ha calculat la similitud de cosinus entre els *embeddings* de la resposta correcta i les dels diferents models i s'ha generat un diagrama de caixes sobre aquesta mesura. Cal recordar que com més propers a zero els valors obtinguts d'aquesta mètrica, més semblants són les frases comparades i, en aquest projecte, interessa que aquests valors siguin baixos, ja que es vol que la resposta generada s'assembli el màxim possible a l'esperada.

En l'annex A4 es mostren els diferents diagrames de caixes (*boxplot*) obtinguts per a les respostes simplificades dels diversos models, i es pot observar que les mitjanes (línia vermella horitzontal) són molt similars entre tots els models, no obstant això, la distribució dels quartils varia significativament entre ells. Respecte al model actualment en ús, la seva mitjana està situada en la part superior del diagrama i no en el centre, indicant que la major part de les dades es concentren en la part inferior, però que hi ha respostes amb valors molt elevats de la similitud, que provoquen una alteració significativa de la mitjana. Pel que fa als models generatius, el model d'OpenAI obté, en general, els millors resultats amb casos en què la resposta generada és idèntica a l'esperada. Tanmateix, també ha generat respostes molt poc relacionades amb l'esperada com es pot veure en el diagrama pels valors atípics de la mètrica. D'altra banda, els models Vicuna i Koala mostren una menor variabilitat entre les respostes generades i presenten valors atípics en el rang inferior, és a dir, respostes que s'assemblen molt a la resposta correcta, però que són considerats casos aïllats o excepcionals.

Com s'ha esmentat anteriorment, també s'han realitzat proves amb les respostes desenvolupades i els resultats obtinguts, mostrats en l'annex A5, tot i ser millors, no difereixen significativament dels de les respostes simplificades. El model d'OpenAI aconsegueix reduir la mitjana de la similitud cosinus en un 4,45%, passant de 0,135 a 0,129, mentre que els models Vicuna i Koala incrementen la seva mitjana en un 3,42% i 8,27% respectivament.

Addicionalment, s'ha avaluat automàticament les respostes desenvolupades, ja que aquestes obtenien millors resultats que les respostes simplificades, generades pels diferents models mitjançant ChatGPT. La descripció d'aquesta avaluació i els paràmetres es poden trobar en l'apartat 7.6. En la taula 2 es mostra la mitjana de l'avaluació tant per a totes les respostes com per a les correctes i incorrectes individualment. Com s'esperava a partir dels resultats obtinguts amb la similitud cosinus, el model d'OpenAI, Curie, és el que obté millors resultats, no obstant això, s'observa que el promig de les respostes incorrectes és molt baix, el que indica que aquestes contenen errors greus.

TAULA 2
Resultats avaluació de les respostes amb ChatGPT

Avaluació	Curie	Vicuna 13B	Koala 13B
Promig	5,35	4,89	4,18
Promig (Correctes)	7,38	6,96	6,04
Promig (Incorrectes)	1,89	1,35	1,02

Per a identificar perquè és tan baixa l'avaluació de les respostes incorrectes s'ha realitzat una classificació i anàlisi manual de les respostes generades, tant simplificades com desenvolupades, pel model d'OpenAI, ja que és el que ha obtingut millors resultats. S'ha assolit que, per a les respostes simplificades, del total de 200 preguntes, el model ha generat una resposta adient per a 90, el que representa només un 45% del total i, per tant, la millora del model generatiu no és significativa. Per tal d'identificar la causa d'aquest baix rendiment s'ha fet una classificació segons el tipus de resposta i s'ha detectat que el model funciona pitjor quan la resposta a generar ha de contenir opcions per a l'usuari, per exemple, "Todas las tarjetas tienen un límite establecido [...]. Si quieres más información de la tarjeta Y di 1, si quieres más información sobre la tarjeta Z di 2.", mentre que el rendiment per a les respostes que consten únicament de text és molt superior. En l'annex A6 i A7, es pot observar que per a les respostes amb opcions, el model genera erròniament el 72,4% (89 respostes) mentre que per a les basades en text es redueix al 27,3% (21 respostes), en canvi, si ens centrem en els resultats de les respostes desenvolupades, s'han generat 126 respostes correctes, el que representa un 63% del total en comparació amb el 45% de les respostes sim-

plificades. Aquest augment del 40% respecte a les respostes simples, concorda amb les observacions anteriors, que indiquen que el model funciona millor quan no treballa amb respostes que contenen opcions. Això és degut al fet que, durant la generació de la resposta, el model interpreta les opcions com un llistat de passos a seguir en comptes d'opcions i, per tant, és important reduir al mínim les opcions presents en les respostes o estructurar-les de manera que el model les pugui comprendre adequadament.

Pel que fa a les respostes correctes, el model demostra un rendiment notable, ja que és capaç de determinar la intenció de la pregunta de l'usuari i generar una resposta coherent d'acord amb les directrius de l'entrenament. A més a més, l'ús de models generatius permet adaptar el contingut de la resposta a la pregunta de l'usuari, combinant diverses respostes individuals de les dades d'entrenament i filtrant la informació no rellevant. D'aquesta manera, les respostes són més precises i ajustades a la qüestió plantejada, el que resulta en un estalvi de temps de comunicació i una major satisfacció per part dels usuaris.

Fig. 5: Gràfic de barres de la classificació d'errors en les respostes.

Addicionalment, s'ha realitzat una classificació dels errors en la generació de les respostes per tal d'obtenir una anàlisi més exhaustiva de com prevenir-los. Aquesta classificació es presenta en un gràfic de barres a la figura 5, on la tonalitat de blau representa la divisió segons si són respostes simplificades o desenvolupades i, les definicions de les categories establertes es poden trobar en l'annex A8 (Taula 3). Es pot observar que les categories de coherència i comprensió presenten el nombre més gran d'errors, però que aquest es redueix quan s'utilitzen respostes desenvolupades, tot i que manté una proporció similar respecte al total. Aquests errors es poden reduir fent ús de models més complexos o augmentant la quantitat de dades d'entrenament i la seva variabilitat així com proporcionant una formulació més extensa de la pregunta. Les categories restants representen una proporció petita dels errors, però inclou una categoria que requereix atenció, que és la categoria de dades inventades (Fake Data), que fa referència a respostes que incorporen informació fictícia o inexistent en les dades d'entrenament. Errors d'aquests tipus poden tenir repercussions molt greus per a l'empresa, com possibles multes o sancions. Alguns exemples d'aquestes respostes es mostren en l'annex A9 (Taula 4), on en la primera resposta el model inventa un número de telèfon i la segona i tercera indiquen a l'usuari que accedeixi a una pàgina web que no existeix.

Amb l'objectiu d'identificar l'arrel del problema (*root cause*) en la generació de dades falses, s'ha realitzat una cerca exhaustiva de patrons en les preguntes i en el context de les respostes amb dades inventades. Inicialment, s'ha analitzat del contingut de la pregunta per identificar si la intenció principal era demanar pel tipus de dades falses directament, com un número de telèfon o una pàgina web, no obstant, cap de les preguntes analitzades mostrava aquesta intenció, sinó que l'objectiu era obtenir informació relativa a un servei o producte. A més a més, s'ha observat que en aquelles respostes en què el model inventa dades, no hi havia cap referència similar en el context generat ni en les dades d'entrenament, indicant que aquest error no pot atribuir-se a una confusió en la transformació del context en la resposta. Finalment, s'ha constatat que totes les preguntes amb respostes que contenien dades inventades havien estat respostes correctament en altres registres, descartant així que el problema esdevingui pel format de la pregunta en si. Per tant, s'ha deduït que aquest error prové directament del model ja que, malgrat reduir significativament la probabilitat d'al·lucinacions en la generació de text mitjançant l'aplicament d'*embeddings*, encara és possible que es produeixin. Com a conseqüència, aquests successos no es poden evitar controlant les preguntes o el context i, per tant, s'ha de controlar de manera posterior a la generació de respostes.

Una possible manera de limitar o censurar aquestes respostes és mitjançant la detecció de patrons en la resposta, com poden ser números de telèfon o pàgines web, i no permetre al xatbot enviar aquesta resposta a l'usuari i avisar a un operador. Una opció automàtica per a detectar aquests errors és l'ús de dos models generatius, com està explicat en l'apartat 11 de desenvolupaments futurs. Finalment, una alternativa és derivar automàticament les preguntes que causen més problemes al model generatiu a un operador. Encara que per a dur a terme aquest procés, és imprescindible conèixer quines són aquestes preguntes i, com s'ha exposat en paràgrafs anteriors, aquesta

tasca és molt complexa, ja que no existeix un tipus específic de pregunta que causa problemes i, en conseqüència, l'opció de derivació automàtica segons la pregunta no és viable en aquest context.

9 Costos i Limitacions

Respecte a les limitacions de l'API d'OpenAI [15], l'empresa té establerts dos límits: peticions per minuts (RPM) o tokens per minut (TPM). Els límits de la modalitat de pagament pel model "Curie" és de 3.500 RPM i 8.750.000 TPM per a *embedding* i 3.500 RPM i 2.250.00 TPM per a generació de xat. Per a determinar si les limitacions de peticions poden afectar el funcionament habitual del xatbot, s'ha realitzat una estimació de la quantitat de peticions simultànies a partir del mateix dataset que pel càlcul del cost i s'ha obtingut que de mitjana, es generen 6.053 peticions per dia el que equivaldria a un pic màxim de 2.119 peticions simultànies (35% del total). Per tant, no hi hauria cap problema per al funcionament normal, ja que el límit establert per l'API especifica 3.500 peticions per minut. Addicionalment, diàriament es gestionen 1.560.102 tokens i, per tant, seria molt difícil arribar al límit de 2.250.000 tokens per minut.

En relació amb els costos, el model "Curie" té un cost per cada mil tokens de 0,0020 dòlars, mentre que pel model "text-embedding-ada-002" és de 0,0004 dòlars. S'ha estimat que, de mitjana, es gestionen diàriament 33.029 tokens per *embeddings* i 1.560.102 tokens del tipus generació de xat i s'obté que la part *d'embedding* tindria un cost diari aproximat de 0,013 $ i la part de xat de 3,12 $ que equivaldria a un cost mensual de 0,40 $ i 97,72 $ respectivament o un cost per consulta promig de 0.0005 $.

En referència als models Vicuna i Koala, els quals són de codi obert, es pot fer una execució local sense cap restricció o limitació establerta per una API, essent els límits només els imposats pel maquinari i l'amplada de banda disponible. El model Vicuna 7B requereix 14 GB i 30 GB de GPU i CPU respectivament, mentre que el model Vicuna 13B requereix 28 GB i 60 GB. Respecte al model Koala, els requisits no estan disponibles públicament, però són similars als del model Vicuna, ja que ambdós estan basats en el mateix model de llenguatge LLaMA i tenen una estructura similar.

10 Desenvolupaments futurs

Una de les millores que es podria implementar per a millorar la qualitat de les respostes o minimitzar els errors generats és l'ús de models més sofisticats i avançats, els quals tenen la capacitat de gestionar converses més complexes i generar respostes més precises amb una quantitat total d'errors reduïda. Malgrat això, l'ús d'aquests models comporta un cost substancialment més elevat i requereix un major temps d'execució i poder computacional. Per tant, és necessari realitzar un estudi per tal d'avaluar si a aquest canvi és beneficial per a l'empresa.

Una altra millora que cal considerar està relacionada amb les dades d'entrenament. Com s'ha esmentat prèviament, la transformació de dades a *embeddings* té un cost molt baix, i això permet utilitzar una gran quantitat de dades i, a més a més, es poden afegir noves dades posteriorment de manera molt fàcil el que permet adaptar el model a les noves necessitats.

Finalment, un mètode per a controlar les respostes errònies és l'ús d'una combinació de dos models generatius, on el primer seria responsable de gestionar tot el procés de generar la resposta i el segon s'encarregaria de detectar la validesa d'aquesta i, en cas que la resposta generada sigui considerada no vàlida, es tornaria a generar o requeriria la intervenció d'un operador. Aquest enfocament permetria un control semiautomàtic del flux de les respostes i milloraria la seva qualitat, no obstant això, també implicaria un augment en la complexitat i el cost d'implementació.

11 Conclusions

Els resultats obtinguts durant aquest treball suggereixen que és viable complementar, o fins i tot substituir, els xatbots basats en regles amb models generatius amb l'objectiu de millorar la qualitat de les respostes, però mantenint la intervenció d'operadors humans com a requisit indispensable.

Pel que fa a les respostes correctes o vàlides, l'increment en la generació d'aquestes, mitjançant l'ús de models generatius en comparació amb els models tradicionals, juntament amb el baix cost d'execució associat, constitueixen arguments significatius per a la consideració de la seva implementació. A més a més, les respostes correctes exhibeixen un nivell de precisió més elevat, fins i tot, arribant a combinar elements de diverses respostes per a adaptar-se a les consultes dels clients. Finalment, la facilitat d'incorporar noves dades permet adaptar de manera simple el xatbot a noves directrius en comparació amb altres mètodes. La combinació d'aquests aspectes es traduiria en una millora en la qualitat del servei i la satisfacció del client i una reducció en el temps promig de comunicació.

No obstant això, cal tenir en compte els desafiaments inherents als models generatius en relació amb la generació de respostes incorrectes, ja que aquestes poden tenir repercussions greus per a l'empresa. Pel que fa als errors de coherència o comprensió, aquests es poden abordar mitjançant l'augment de la quantitat, qualitat i variabilitat de les dades d'entrenament, així com l'ús de models més sofisticats. En canvi, la generació de dades fictícies no es pot resoldre directament, és a dir, no es pot preveure, però sí que es pot gestionar posteriorment amb diferents

aproximacions com poden ser la cerca de patrons en la resposta o utilitzar una combinació de models generatius com s'ha esmentat en l'apartat 10. D'aquesta manera, es reduiria la intervenció dels operadors només a les consultes que ho requereixin i aquelles en què el model no pugui respondre adequadament.

Addicionalment, és possible aprofitar els models en altres tasques com ara la classificació automàtica de respostes per part del model generatiu per a obtenir una anàlisi de les consultes més freqüents al xatbot, identificant aquelles amb major incidència d'errors o aquelles que generen més interaccions amb un operador. Una altra aplicació és l'ús dels embeddings per a l'anàlisi dels clústers en les preguntes formulades o, fins i tot, modificar el procés per a obtenir les respostes eliminant la generació d'aquestes i seleccionant la primera resposta del context generat i validant-la mitjançant el model generatiu. Això representaria una millora dels resultats respecte al model tradicional i evitaria els problemes relacionats amb la generació de respostes, però serien necessàries dades d'entrenament amb un alt nivell de qualitat i variabilitat.

Cal remarcar que els resultats obtinguts en aquest projecte reflecteixen el rendiment dels models en registres de transcripcions telefòniques, els quals, sovint, consisteixen en preguntes breus amb menys context en comparació amb les consultes escrites. Per tant, l'aplicació d'aquests mateixos models en altres àmbits, com ara xats en viu a través de pàgines web o aplicacions, o en comunicacions per correu electrònic, podria assolir millor resultats.

En resum, és viable utilitzar models generatius per a millorar el rendiment dels models tradicionals, però s'ha de tenir en compte els riscos associats a l'ús d'aquests models en entorns comercials.

Agraïments

Desitjo expressar el meu sincer agraïment al meu tutor de la universitat, l'Antonio Espinosa, i al Daniel Franco per la seva orientació, suport i dedicació durant el desenvolupament d'aquest treball. Vull donar les gràcies també al tutor de l'empresa col·laboradora, el Carlos Ortet, per la seva guia i col·laboració en aquest projecte, que ha enriquit amb la seva experiència pràctica. A més, no puc deixar de reconèixer el suport incondicional de la meva mare que ha estat un pilar fonamental, no només durant aquest projecte sinó durant tota la carrera.

Bibliografia

[1] IBM: Watson, dossier d'aplicacions, eines i solucions d'IBM per a empreses (2020) [Consultat: 10 de març de 2023]. https://www.ibm.com/es-es/watson

[2] CaixaBank: Noa, l'assistent virtual de CaixaBank Now (2016) [Consultat: 10 de març de 2023] Disponible a Internet: https://www.caixabank.es/particular/bancadigital/noacaixabank now.html

[3] OpenAI: Resum dels models disponibles a l'API (2023) [Consultat: 04 de març de 2023]. Disponible a Internet: https://platform.open ai.com/docs/models/

[4] OpenAI: API (2023) [Consultat: 10 de març de 2023]. Disponible a Internet: https://platform.openai.com/

[5] Koala: Model conversacional per a la recerca acadèmica (2023) [Consultat: 15 d'abril de 2023]. Disponible a Internet: https://bair.berkeley.edu/blog/2023/04/03/koala/

[6] Vicuna: Model de xatbot "Open-Source" (2023) [Consultat: 15 d'abril de 2023]. Disponible a Internet: https://vicuna.lmsys.org

[7] LLaMA: Model de llenguatge fonamental obert i eficient (2023) [Consultat: 15 d'abril de 2023]. Disponible a internet: https://arxiv.org/abs/2302.13971

[8] Bank of America: Funcionament del xatbot Erica (2022) [Consultat: 10 de març de 2023]. Disponible a Internet: https://promotions.bankofamerica.com/digitalbanking/mobileb anking/erica

[9] UPS: Infogràfic xatbot UPS Bot (2023) [Consultat: 10 de març de 2023]. Disponible a Internet: https://www.chatbotguide.org/ups-bot

[10] OpenAI: Xatbot ChatGPT (2023) [Consultat: 10 de març de 2023]. Disponible a Internet: https://chat.openai.com/chat

[11] Word2Vec: Estimació eficient de representacions de paraules en un espai vectorial (2013) [Consultat: 20 d'abril de 2023]. Disponible a Internet: https://arxiv.org/abs/1301.3781

[12] Doc2Vec: Distributed Representations of Sentences and Documents [Consultat: 20 d'abril de 2023]. Disponible a Internet: https://arxiv.org/abs/1405.4053

[13] Tiktoken: llibreria de codificació de parelles de bytes (2023) [Consultat: 20 d'abril de 2023]. Disponible a Internet: https://github.com/openai/tiktoken

[14] FastChat: plataforma oberta d'entrenament i avaluació de xatbots (2023) [Consultat: 20 d'abril de 2023]. Disponible a Internet: https://github.com/lm-sys/FastChat

[15] Cost API: Límits i cost de d'OpenAI (2023) [Consultat: 20 d'abril de 2023]. Disponible a Internet: https://platform. openai.com /docs/guides/rate-limits/overview

APÈNDIX

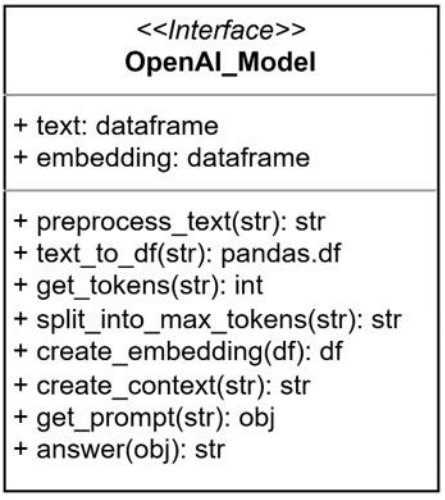
A1. PLANIFICACIÓ PROJECTE

TAULA 1
PLANIFICACIÓ DEL DESENVOLUPAMENT DEL PROJECTE

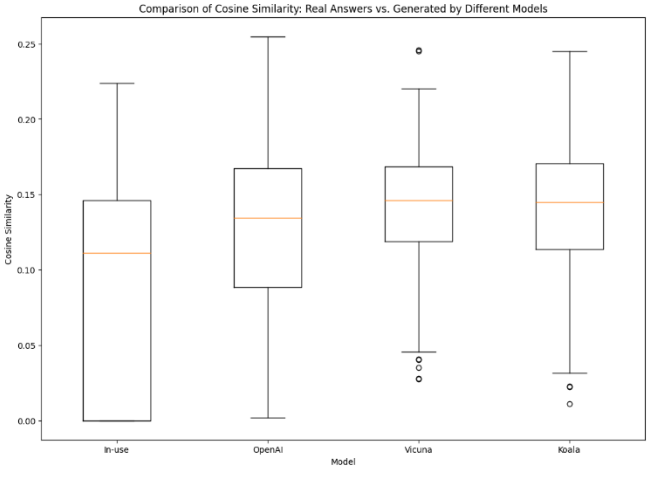
Setmana 1	Identificar objectius i fites del projecte.
Setmana 2	Establir planificació del projecte
Setmana 3	Anàlisi i realització de proves amb l'API d'OpenAI
Setmana 4	Anàlisi i realització de proves dels diferents models GPT-3 d'OpenAI
Setmana 5 i 6	Recopilació i preprocessament de les dades d'entrenament.
Iteracions	Entrenament del model i avaluació del rendiment
	Integració i testeig del model per interactuar amb l'usuari.
	Recopilació i anàlisi de mètriques per identificar millores
	Realització comparativa amb models similars
Setmana 14	Avaluació de l'èxit del projecte i identificar àrees de millora.
Setmana 15	Desenvolupament de la memòria final.
Setmana 16	Presentació dels resultats projecte

A2. EXEMPLE DEL FUNCIONAMENT DEL XATBOT ERIKA

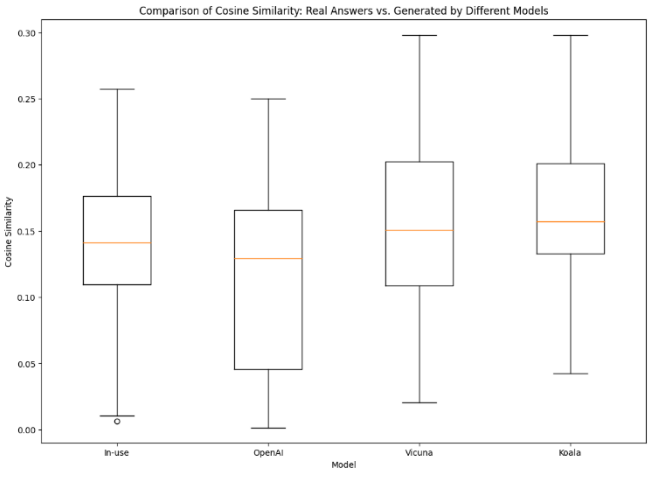
A3. DIAGRAMA UML DE LA CLASSE DEL MODEL D'OPENAI



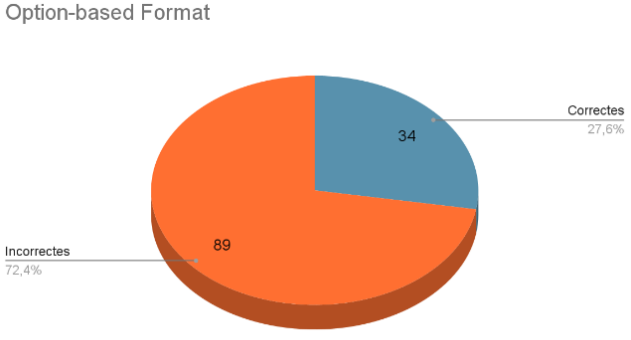
A4. DIAGRAMA DE CAIXA DE LA SIMILITUD COSINUS ENTRE LES RESPOSTES REALS I LES RESPOSTES SIMPLIFICADES GENERADES PER CADA MODEL



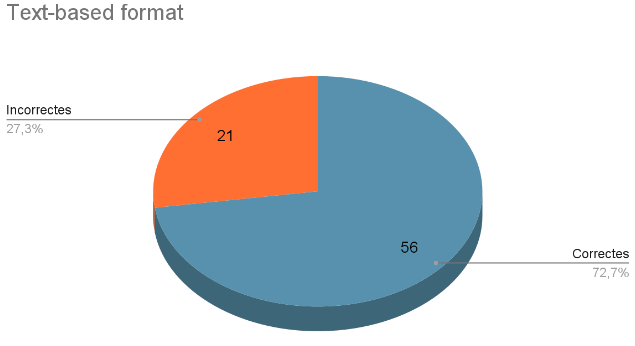
A5. Diagrama de caixa de la similitud cosinus entre les respostes reals i les respostes desenvolupades generades per cada model



A6. Diagrama de sectors de la distribució de validesa de les respostes que contenen opcions



A7. Diagrama de sectors de la distribució de validesa de les respostes que contenen text



A8. Definició de categories d'error

TAULA 3
Definició de les categories d'error de les respostes

Categoria	Explicació
Coherència (*Consistency*)	Inclou les respostes correctes, però que contenen errors ortogràfics o semàntics.
Comprensió (*Understanding*)	Són les respostes incorrectes que el model ha generat perquè no ha entès la intenció de la pregunta correctament.
Valor numèric (*Numeric value*)	Respostes en les quals hi ha algun valor numèric incorrecte, com pot ser el preu d'un servei.
Dades falses (*Fake data*)	Engloba les respostes que contenen informació inventada o que no existeix en les dades d'entrenament.
Estructural (*Formatting*)	Respostes que tenen un format o estructura incorrecte com poden ser salts de línia.
Altres (*Other*)	Errors que no es poden classificar en les anteriors.

A9. Exemples de respostes generades

TAULA 4
EXEMPLES DE RESPOSTES AMB DADES INVENTADES

Si quieres información sobre el seguro de [...], puedes llamar al 1-800-922-9696.
Si quieres información sobre cómo puedes domiciliar tu nómina, puedes dirigirte a https://www.inss.gob.es/es/informacion-y-servicios/empleo/nominacion-nominadores.html.
Si olvidaste tu contraseña, puedes recuperarla en el siguiente enlace: https://www.institucionfinanciera.com/es/informacion-general/noticias/o-lvido-de-la-contraseña-de-la-aplicacion.html