Correcció automàtica en processos d'extracció de metadades mitjançant l'ús d'agents

Marc Serrano Sanz

Resum—Aquest treball presenta el disseny i la implementació d'un sistema de correcció automàtica per a processos d'extracció de metadades audiovisuals mitjançant l'ús d'agents intel·ligents. En el context d'un creixement accelerat de la intel·ligència artificial i la integració de models de llenguatge de gran escala (LLM), el projecte desenvolupa un agent capaç d'identificar i corregir errors derivats de diversos mòduls de processament: transcripció, reconeixement d'entitats (NER), paraules clau, OCR, logotips, objectes, descriptors d'escena i segmentació semàntica. A través de tècniques com el prompt chaining, la classificació supervisada i la generació automatitzada, el sistema busca garantir la qualitat de les metadades extretes i millorar l'eficiència del flux de treball audiovisual. El desenvolupament s'ha estructurat en cicles iteratius, combinant entrenament de models, interacció amb LLMs i integració modular en un agent únic.

Paraules clau—intel·ligència artificial, agents intel·ligents, metadades audiovisuals, models de llenguatge, automatització, detecció d'errors

Abstract—This project presents the design and implementation of an automatic correction system for audiovisual metadata extraction processes using intelligent agents. In the context of rapidly advancing artificial intelligence and the integration of large language models (LLMs), the project develops an agent capable of detecting and correcting errors originating from various processing modules: transcription, named entity recognition (NER), keyword extraction, OCR, logo detection, object recognition, scene descriptors, and semantic segmentation. Through techniques such as prompt chaining, supervised classification, and automatic generation, the system aims to ensure the quality of the extracted metadata and improve the efficiency of the audiovisual workflow. The development is structured in iterative cycles, combining model training, LLM interaction, and modular integration into a single agent.

Index Terms—artificial intelligence, intelligent agents, audiovisual metadata, language models, automation, error detection

1 Introducció - Context del treball

El món de la intel·ligència artificial experimenta un creixement accelerat des de l'aparició dels models extensos de llenguatge (large language models, LLM) [1]. Cada vegada més aquesta tecnologia s'ha integrat ràpidament en el nostre dia a dia fins al punt de servir com a suport fins i tot per a activitats quotidianes.

És en aquest context que apareixen els anomenats agents intel·ligents [2]. Aquests nous programes són capaços d'integrar en una mateixa execució tant l'ús de funcions definides prèviament com l'implementació a temps real de noves funcionalitats que no poden ser resoltes sense generar nou codi. Això s'aconsegueix mitjançant la integració de LLM en els bots existents fins a l'actualitat.

Per altra banda, desde l'aparició dels sistemes multimedia ha estat una tasca rellevant la extracció de dades audiovisuals en àmbits com la documentació i la gestió de la informació. La capacitat de poder fer una extracció i un processat automàtic d'aquestes dades resulta clau pels processos d'optimització i classificació de continguts. És molt important l'extracció i correcció de metadades a l'àmbit audiovisual, ja que aquestes metadades actuen com a descriptors essencials del contingut i en permeten una millor comprensió, indexació i recuperació. Una metadada incorrecta pot portar a interpretacions errònies del material, dificultar-ne la cerca o fins i tot fer que es perdi informació rellevant en processos automatitzats. Per això, assegurar la seva qualitat mitjançant mecanismes de detecció i correcció esdevé fonamental en entorns com els arxius audiovisuals.

És per això tot això que aquest treball es centra en la detecció i correcció automàtica de metadades mitjançant l'ús d'agents intel·ligents. Aquest haurà de ser capaç de, mitjançant un prompt en llenguatge natural, detectar i corregir aquelles possibles errades que hagin pogut aparèixer en la recol·lecció de metadades d'un vídeo.

D'aquesta manera, l'agent serà capaç d'actuar com un documentalista i ajudarà a garantir la qualitat de totes les dades extretes. Això s'aconseguirà incorporant

diferents entrenaments de models per a les tasques més difícils d'assolir i dotant de més llibertat a les tasques més simples.

2 Estat de l'art

Amb el ràpid progrès de les intel·ligències artificials ha aparegut el concepte d'agent. Tot i que la definició de que és un agent pot variar segons la font consultada i la línia que separa una agent d'altres tipus de programes és molt difusa, podem acordar que un agent és un programa de software que utilitza la intel·ligència artificial per assolir objectiu i completar tasques predefinides pels usuaris. Aquests agents utilitzen raonament, planificació, memòria i tenen un cert nivell d'autonomia que els permet prendre decisions, aprendre i adaptar-se. Aquest agents es poden utilitzar per assolir moltes finalitats diferents i es poden implementar en una elevada quantitat de llenguatges.

Pel que fa al python (que serà l'entorn de desenvolupament del projecte), actualment hi ha tres llibreries principals que serveixen per a la implementació d'agents autònoms. Aquestes tres són LlamaIndex, LangChain i SmolAgents.

LlamaIndex és una llibreria enfocada a la construcció de sistemes retrieval-augmented generation (RAG). La seva principal funcionalitat és la creació d'índexos sobre dades (tant estructurades com no estructurades) que permeten fer preguntes contextuals mitjançant un LLM. Tot i que útil, aquesta no és la funcionalitat principal que es vol en el projecte així que s'ha decidit descartar aquesta opció.

LangChain ja s'apropa més a la funcionalitat que s'espera del projecte. És una eina molt més completa i amb un ecosistema extens. La seva proposta es basa en la definició de "chains" i "agents" que poden interactuar amb eines externes. D'aquesta manera, es podrien utilitzar aquestes eines externes per a la implementació dels nostres mòduls. La principal problemàtica que presenta aquesta llibreria és la seva corba d'aprenentatge. Si bé és cert que hi ha molts recursos que es poden utilitzar per entendre el funcionament d'aquesta, segueix sent força complicada en comparació amb SmolAgents.

Finalment, SmolAgents és la llibreria que s'ha decidit utilitzar per la realització d'aquest projecte. Aquesta és una eina lleugera per a la creació d'agents modulars. La principal diferència amb les altres llibreries és que és capaç de generar i executar codi. Pel que fa a l'accessibilitat, aquesta permet definir cada eina com una funció de python amb una petita descripció del que fa. La coordinació es fa mitjançant un manager agent que decideix l'ordre d'execució de les diferents eines i del codi propi generat per la LLM. SmolAgents representa per tant una solució d'alt nivell per a la implementació d'agents i resulta molt útil per aquelles persones que estan interactuant per primera vegada amb aquest nou tipus de tecnologies.

3 Objectius

L'objectiu final d'aquest treball de grau és crear un agent amb totes les eines necessàries per a ser capaç de detectar i corregir tots els errors que sorgeixin de les feines d'extracció de metadades d'un vídeo. D'aquesta manera podrà augmentar i garantir un alt nivell en la qualitat de les dades extretes d'aquests.

Per tal d'aconseguir complir amb aquest objectiu, es plantegen diversos objectius parcials que s'han d'anar complint. Aquests objectius els podem agrupar de la següent manera:

- Objectius inicials:
  - Familiaritzar-se amb l'entorn de treball i amb l'ús d'agents.
  - Identificar els mòduls més problemàtics de l'extracció de metadades i plantejar-ne la solució.
- Objectius de desenvolupament:
  - Assegurar el correcte funcionament de tots els mòduls que l'agent utilitzarà per a resoldre els problemes que se li plantegin.
- Objectius finals:
  - Integrar correctament tots els mòduls a l'agent i assegurar el correcte funcionament final d'aquest.
  - Desenvolupar una interfície gràfica des d'on poder interactuar amb l'agent.

4 Metodologia

Per a la realització d'aquest projecte és seguiran dues metodologies diferents. La primera, es basarà en el desenvolupament individual per part de l'estudiant mitjançant una metodologia "agile" que permeti la divisió d'aquest en cicles de treball curts i dinàmics, per tal d'aconseguir un progrès continu i anar assolint els objectius de manera gradual. La segona consistirà en la interacció amb el tutor mitjançant un seguit de reunions on es mostraran els resultats obtinguts de cada cicle, així com s'obtindran recomanacions i guies per al correcte desenvolupament del projecte.

5 Planificació

Per a fer la planificació d'aquest projecte s'ha decidit dividir les tasques a realitzar en diferents grups en funció dels objectius que es vagin assolint. La naturalesa modular del projecte ha facilitat molt la creació d'aquests i permet treballar en els seus diferents apartats sense que es generin conflictes. Com ha resultat d'això tenim que cada cicle de treball equivaldrà a la realització d'un dels mòduls que haurà de ser capaç d'executar l'agent. Aquests no seran els

únics, ja que inicialment es farà una tasca de familiarització amb la creació d'agents autònoms i un cop finalitzada la creació dels mòduls s'haurà de fer també una integració d'aquests dins d'un agent.

6 DESENVOLUPAMENT DELS MÒDULS

Seguidament es presentarà la solució implementada per a cada fase de la planificació. Tot i l'ordre plantejat en la planificació, l'exposició d'aquestes solucions es realitzarà agrupant aquelles fases que tenen punts en comú amb l'objectiu de facilitar la lectura. En aquest apartat es mostraran només les estratègies usades per als mòduls. La integració d'aquests en un agent es realitzarà en el següent apartat.

6.1 Identificació dels mòduls i dels seus principals problemes.

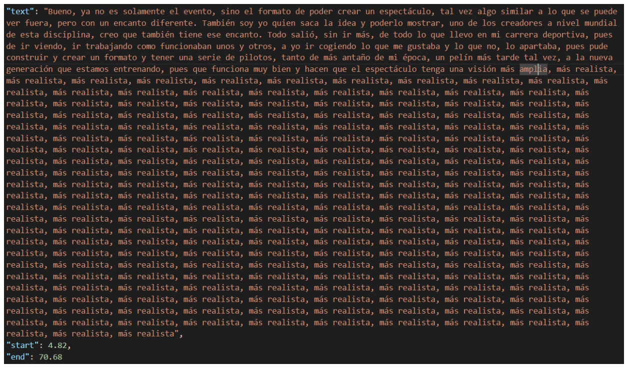
Per triar les funcionalitats que ha de tenir l'agent s'han mirat els diferents camps de metadades amb què treballa l'empresa. Aquesta té dos tipus diferents d'extraccions: les d'entrevistes i programes normals i les de documentals i programes on la rellevància la tenen les fonts visuals. Depenent del tipus d'extracció que es faci es guardaran unes categories de metadades o unes altres. Els mòduls que s'han triat per corregir han sigut els més rellevants d'entre aquests. Aquests mòduls són Transcripció, Named Entity Recognition (NER), paraules clau, segmentació semàntica, optical character recognition (OCR), logotips, objectes i descripció d'escena. Dels problemes que sorgeixen en l'obtenció de les dades de cada mòdul se'n parlarà en el seu respectiu apartat.

6.2 Familiarització amb l'eina.

En aquest apartat es va implementar un agent bàsic seguint la guia trobada a la pàgina oficial de SmolAgents [6]. Aquest és capaç de realitzar cerques al buscador mitjançant una eina (tool) implementada en el mateix tutorial. També se li ha afegit una eina que li permet guardar el contingut trobat de la cerca en un fitxer JSON, que és el format que s'utilitzarà durant tot el treball. També s'han fet proves tant carregant els models en la gràfica local com utilitzant l'api de huggingFace.

6.3 Identificació de repeticions en transcripció.

Un dels problemes més comuns que sorgeixen del nostre mòdul de transcripció són frases on s'han generat repeticions que no tenen cap mena de sentit. Això es produeix perquè el model utilitzat per a fer la transcripció pateix algunes al·lucinacions. Els textos resultants quan es produeix aquest error tenen el següent aspecte:



"text": "Bueno, ya no es solamente el evento, sino el formato de poder crear un espectáculo, tal vez algo similar a lo que se puede ver fuera, pero con un encanto diferente. También soy yo quien saca la idea y poderlo mostrar, uno de los creadores a nivel mundial de esta disciplina, creo que también tiene ese encanto. Todo salió, sin ir más, de todo lo que llevo en mi carrera deportiva, pues de ir viendo, ir trabajando como funcionaban unos y otros, a yo ir cogiendo lo que me gustaba y lo que no, lo apartaba, pues pude construir y crear un formato y tener una serie de pilotos, tanto de más antaño de mi época, un pelín más tarde tal vez, a la nueva generación que estamos entrenando, pues que funciona muy bien y hacen que el espectáculo tenga una visión más amplia, más realista",
"start": 4.82,
"end": 70.68

Fig. 1: Imatge d'un fitxer JSON amb l'error "de repetició".

Per detectar repeticions, s´ha implementat un algorisme de compressióamb la llibreria zlib[7]. Inicialment, convertim l'string en bytes del format utf-8 i el comprimim usant l'algorisme zlib. Si el rati de compressió resulta ser molt elevat, voldrà dir que el segment de text que s'està analitzant conté moltes repeticions. Obtinguts els ratis de compressió de tots els segments de text, es comprova si el percentatge de segments amb error supera un threshold imposat de 0.15. Si aquest és el cas, s'obtenen els segments del vídeo en que es produeixen aquests errors, s'extreuen de la resta del vídeo i es tornen a passar pel mòdul de transcripció, però aquesta vegada utilitzant un preset propi de l'empresa diferent. Finalment, els resultats obtinguts s'afegeixen al fitxer de transcripció en substitució d'aquells que contenien els errors.

6.4 Altres errors en la transcripció.

Apart del problema trobat a la fase 3, s'han detectat tres errors més per corregir en aquest mòdul. Aquests tres són els següents:

- Falta de puntuació en la transcripció.
- Aparició d'una al·lucinació en forma del caràcter "... …".
- Segments d'audio no transcrits i segments on no hi ha veu que s'han incorporat en una transcripció.
- Per tal de corregir aquests tres errors, s'han generat dos procediments diferents.

6.4.1 Faltes de puntuació i al·lucinacions.

El primer d'ells consisteix en l'entrenament d'un model de detecció dels dos primers errors mencionats. Per fer això s'ha generat un dataset de 155 elements que conté dues columnes: text i label. La columna de text conté els exemples que s'han trobat dins de la base de dades de textos que continguin o bé el caràcter "... …" o una manca de puntuació significativa, així com frases correctes. Aquestes últimes han rebut el label 0 i els dos casos erronis el label 1. També s'ha creat un altre dataset semblant amb altres exemples que funcionarà com a validation i test. Un cop creat el dataset, s'ha fet un fine-tunning del model preentrenat "distilbert-base-uncased" [8]. Aquest model s'ha

entrenat amb els següents hiperparàmetres:

```
training_args = TrainingArguments(
    output_dir='models',
    num_train_epochs=10,
    logging_steps=10,
    per_device_train_batch_size=16,
    per_device_eval_batch_size=16,
    eval_strategy="epoch",
    load_best_model_at_end=True,
    metric_for_best_model="accuracy",
    save_strategy="epoch",
)
```

Fig. 2: Imatge de la definició dels hiperparàmetres.

L'estratègia escollida ha sigut la d'èpoques amb un batch size de 16. S'ha optat per aquesta estratègia degut al petit tamany del dataset. D'aquesta manera podem seguir fent l'entrenament iterant per sobre d'aquest en més d'una ocasió. La mètrica escollida ha sigut l'accuracy. Després d'haver entrenat aquest model, s'itera cada segment de text i se'n comprova si és correcte. Si la transcripció completa supera un threshold d'error del 15%, els segments que han generat l'error són extrets de la resta del vídeo i passats una altra vegada pel mòdul de transcripció utilitzant un model diferent. Finalment, aquests segments s'afegeixen al fitxer de transcripció.

6.4.2 Segments no transcrits o mal acotats.

El segon procediment consisteix a trobar aquells segments del vídeo que no s'han transcrit i, amb menys importància, aquells que estan mal acotats. Per fer això s'ha realitzat un preprocessat dels vídeos per generar un dataset amb què entrenar un model. El primer que s'ha fet és, utilitzant un model de Silero-VAD [9], s'han obtingut els segments del vídeo en els que hi ha una persona parlant. Seguidament, s'ha realitzat una codificació binària en funció dels instants del vídeo que hi ha persones parlant i s'ha obtingut un array amb tants elements com duració té el vídeo. Posteriorment, s'ha fet el mateix amb els segments donats per la transcripció obtinguda del mòdul. Un cop tenim els dos mapes binaris que indiquen en quins instants de temps es detecta una veu es genera el dataset de la següent manera:

- Es comprova si en l'instant i els dos mapes no coincideixen (0 no es detecta veu, 1 es detecta veu). En cas afirmatiu, si l'1 el té el mapa generat per la detecció de veu, se suma 1 a la seva variable; sinó se li suma a la variable de la transcripció.
- Un cop obtinguts tots els errors generem un diccionari que conté un valor amb l'error total, un amb els errors de la transcripció (errors deguts a segments no transcrits) i un amb els errors de la detecció de veu (errors en l'acotació temporal dels segments de la transcripció). Aquests valors són normalitzats.

Agafant diversos exemples de vídeos on sabem que hi ha errors, es genera un dataset de 35 columnes on els vídeos que tenen errors reben el label 1 i els correctes el label 0. Per fer l'entrenament, s'utilitza el model de random forest classifier de la llibreria Scikit-Learn [5], els hiperparàmetres d'aquest s'obtenen usant la funció GridSearchCV del mateix Scikit-Learn. Un cop entrenat el model, realitzem el preprocessat abans anomenat amb els vídeos dels quals busquem els errors. Gràcies al model, detectem si el vídeo cau en la categoria d'erroni o no i gràcies al mapa binari podem obtenir aquells segments que no s'han transcrit del vídeo i aquells amb els timestamps mal acotats. Finalment, els segments trobats són passats un altre cop al mòdul amb un model diferent i corregits posteriorment al fitxer de transcripció.

6.5 Correcció de NERs.

Un dels principals problemes en la detecció d'entitats amb NER (Named Entity Recognition) és l'aparició de falsos positius, especialment degut a la gran quantitat de text que s'ha de processar per classificar correctament les entitats. Per afrontar aquest repte, s'ha plantejat l'ús d'un model de llenguatge gran (LLM), concretament el Ministral-8B-Instruct, com a mecanisme de validació de les etiquetes detectades. L'estratègia proposada es basa en el concepte de prompt chaining. Inicialment, per cada entitat detectada (NER), es genera un primer prompt que inclou: la paraula identificada, la categoria assignada i el segment de text on apareix. Aquest prompt es passa al LLM, que retorna un 1 si considera que la classificació és correcta o un 0 si la considera incorrecta.

A partir d'aquesta primera resposta, només en els casos considerats erronis (0), es desencadena una segona fase. En aquesta, s'envia un nou prompt al model per generar una justificació raonada que expliqui per què l'etiquetatge inicial pot ser incorrecte. Un cop obtinguda aquesta justificació, es construeix un tercer prompt que incorpora tota la informació anterior (entitat, categoria, context i raonament), i se li torna a demanar al model que avaluï novament si la classificació és correcta.

Finalment, si després d'aquest procés la resposta continua sent 0, es considera que es tracta efectivament d'un fals positiu i la instància es descarta del fitxer de NERs.

6.6 Correcció de les paraules claus.

Per al mòdul de paraules clau, l'objectiu principal és assegurar que les paraules extretes del text siguin realment rellevants en el seu context. Per validar aquesta rellevància, s'ha aplicat una estratègia molt

similar a la utilitzada en la validació de les entitats NER. En aquest cas es fan servir dos prompts.

El primer prompt es construeix amb la paraula clau detectada i el segment de text del qual s'ha extret. El model de llenguatge (LLM) rep aquesta informació i avalua la rellevància de la paraula dins del seu context, retornant un valor numèric entre 1 i 10. Si la puntuació obtinguda és superior a 5, la paraula clau es considera vàlida i s'incorpora al nou fitxer de paraules clau. En cas contrari, és descartada.

Seguidament, s'ha afegit un pas addicional per substituir les paraules descartades. Aquesta extensió consisteix en generar noves paraules clau per als segments corresponents, mitjançant un segon prompt orientat a la generació automàtica. Per tal d'aconseguir obtenir uns bons resultats amb el segon prompt s'ha utilitzat la llibreria pydantic[12]. Aquesta llibreria ens permet forçar al LLM a retornar una estructura concreta amb la resposta. Gràcies a això, podem demanar al LLM que ens respongui amb una estructura JSON que contè la nova paraula clau. Aquest pas, tot i que pugui no semblar molt rellevant, millora considerablement les noves paraules claus proposades.

Així doncs, el mòdul de paraules clau actua actualment com un mecanisme de filtratge, centrant-se en la detecció i eliminació de falsos positius així com també en la generació automàtica en aquells casos on els resultats no eren prou bons.

6.7 Correcció de l'extracció d'objectes.

En els vídeos on la transcripció no aporta informació rellevant, es pretén extreure altres elements contextuals, com ara els objectes presents a l'escena. El mòdul encarregat d'aquesta tasca genera un llistat amb els objectes detectats, els instants temporals en què apareixen i el nivell de confiança associat a cada predicció. L'objectiu d'aquest mòdul és filtrar els falsos positius i eliminar aquells objectes que no siguin significatius dins del context del vídeo.

Per assolir aquest propòsit, s'ha plantejat una estratègia similar a la utilitzada en la fase 5. Es fa servir el model de visió "Qwen/Qwen2.5-VL-7B-Instruct-AWQ" [11], combinat amb una cadena de prompts. En primer lloc, per a cada objecte detectat, el model rep el fotograma corresponent juntament amb la pregunta: "Is there any 'element' in this image? Answer only with Yes or No." Si la resposta és negativa, l'objecte es descarta automàticament.

En cas que la resposta sigui afirmativa, es formula un segon prompt en què se sol·licita al model una puntuació de 0 a 10 sobre la rellevància de l'objecte dins de la imatge. Si la valoració és inferior a 5, l'objecte també és eliminat. Aquesta doble validació permet corregir prediccions errònies i millorar la qualitat final del llistat d'objectes.

Fig. 3: Exemple d'imatge on el model ha detectat l'objecte "bird". Aquesta instància s'ha guardat.

Fig. 4: Exemple d'imatge on el model ha detectat l'objecte "bird". Aquesta instància s'ha descartat.

6.8 Correcció d'OCRs i de logotips.

El primer pas se centra en la validació dels logotips, seguint una estratègia molt similar a la del mòdul d'objectes. La diferència principal és que, en aquest cas, no ens interessa la rellevància del logotip dins de l'escena, sinó simplement verificar si és present o no. Per això, n'hi ha prou amb la primera fase de la cadena de prompts.

Concretament, es genera un prompt per al model de visió amb la pregunta: "Is there the logo of 'logo' in this image? Answer only with Yes or No." Si la resposta és negativa, el logotip es descarta del document; si és afirmativa, es manté. Aquesta validació esdevé especialment rellevant, ja que el mòdul de detecció de logotips presenta dificultats tècniques i, actualment, no produeix resultats satisfactoris. Com a solució temporal, s'ha optat per implementar aquest sistema de filtratge amb el model de visió per reduir la presència de falsos positius.

Tot seguit, abordem la problemàtica associada a l'extracció de textos OCR. Per aquest mòdul, també es fa ús del mateix model de visió i llenguatge, aplicant de nou una estratègia de prompt chaining. En primer lloc, es passa al model la imatge i el text inicialment extret, juntament amb un prompt que pregunta si l'extracció d'aquest text és correcta. Si la resposta és afirmativa, la instància es considera vàlida i s'inclou al fitxer de metadades.

En cas que el model consideri incorrecta l'extracció, s'inicia una segona fase de validació. Primer, es formula un prompt que pregunta al model si en aquell fotograma hi ha algun OCR present, amb una resposta tancada ("Yes" o "No"). Si la resposta és "No", la instància es descarta. Si és "Yes", es demana al model que extregui el text visible, amb la instrucció de retornar exclusivament el contingut textual.

Finalment, es genera un tercer prompt on es demana al model que valori la rellevància d'aquest nou text extret dins del context de la imatge, assignant una puntuació del 0 al 10. Si la valoració és igual o superior a 5, el text es considera rellevant i s'inclou al fitxer final. En cas contrari, es descarta.

Amb aquesta estratègia, el sistema no només corregeix possibles errors d'extracció, sinó que també filtra textos irrellevants, com ara aquells que poden aparèixer al fons d'un plató o escenari i que no aporten informació útil a l'escena analitzada.

6.9 Correcció dels descriptors d'escena.

Els descriptors d'escena són conjunts d'una o dues paraules que tenen com a finalitat descriure el contingut visual del vídeo en un moment determinat. Com que es tracta d'un altre mòdul basat en la informació visual, s'aplica l'estratègia de prompt chaining utilitzada prèviament en altres mòduls, aprofitant el bon rendiment del mateix model de llenguatge multimodal emprat anteriorment.

En aquest cas, el procés consta de dues etapes. En primer lloc, es genera un prompt en què es pregunta al model si el descriptor proposat pel mòdul es correspon realment amb el contingut del fotograma. La resposta ha de ser tancada, amb un "Sí" o un "No". Si la resposta és afirmativa, la instància es considera correcta i es guarda.

En canvi, si la resposta és negativa, s'activa un segon prompt on se li demana al model que proporcioni un nou descriptor —format per una o dues paraules— que defineixi de manera més precisa el contingut del frame analitzat.

Aquest mecanisme permet corregir descriptors que podrien resultar imprecisos, ambigus o directament erronis, assegurant així una millor qualitat en l'etiquetatge visual de les escenes.

6.10 Correcció de la segmentació semàntica.

Els fitxers amb la segmentació semàntica contenen un petit resum del que s'expressa en uns certs instants de temps, que venen acotats pels seus corresponents timestamps. Per resoldre aquest mòdul s'utilitza un altre cop l'estratègia de prompting. Inicialment, es demana, passant-li al LLM els segments de text i el seu corresponent resum, que valori si aquest és encertat o no. Per millorar els resultats en aquesta primera consulta utilitzem la llibreria abans mencionada pydantic. Gràcies a aquesta llibreria, obligem a la LLM a que ens retorni el booleà en funció de si el resum és correcte i també un raonament sobre el perquè és correcte o no. Tot i que després no utilitzem aquest raonament, el fet d'obligar a la LLM a fer-lo la força a dedicar més esforços a arribar a la solució de la pregunta. Aquesta estratègia és coneix com a *chain of thought* [15]. Un cop aplicada aquesta estratègia, es valora el resultat obtingut de l'anterior prompt. En cas d'obtenir un resultat negatiu, se li demana en un segon prompt a la LLM que generi ella mateixa el resum adient als segments que li han passat. En cas contrari, s'accepta el resum que teniem fins ara i es passa a la següent iteració.

6.11 Resultats dels mòduls.

Després d'implementar els mòduls independents que utilitzarà l'agent s'ha pogut comprovar el correcte funcionament de cadascun d'aquests. Els mòduls més destacables pels seus resultats han sigut els relacionats amb el model de visió. Aquest ha estat capaç de corregir la gran majoria dels errors que es produïen en l'extracció de les metadades relacionades i d'aportar nova informació valuosa quan ha sigut necessari. Un exemple d'això és el test que s'ha realitzat en el mòdul de correcció d'objectes. En aquest test s'ha pogut observar que s'ha descartat 161 instàncies d'un total de 1081. D'aquestes, s'ha observat que un 90% eren falsos positius. Això ha millorat significativament les dades que s'entreguen finalment al client. També s'ha utilitzat el mateix video per testejar el funcionament del mòdul de OCR. En aquest cas s'ha pogut observar que s'han eliminat 109 de les 125 instàncies d'OCR que s'havia detectat. Un cop revisades aquestes, s'ha pogut determinar que un 95% de les instàncies eliminades eren o al·lucinacions del programa o OCRs que no resultaven rellevants per l'escena.

Pel que fa als mòduls de text, s'ha fet un test del mòdul de correcció de paraules clau i s'ha trobat que el mòdul ha canviat la paraula clau en un 68.7% dels casos. Revisant manualment les noves paraules clau obtingudes, s'ha arribat a la conclusió de que les noves paraules proposades eren millor que les anteriors en un 75% dels casos.

També cal destacar tant per resultats com per la quantitat de temps invertit el mòdul de correcció de les transcripcions. Al ser la base sobre la qual s'extreuen moltes altres metadades, ha calgut invertir-hi la major part del temps del desenvolupament dels mòduls tant a preparar conjunts de dades com a fer les proves pels diferents models que s'han usat. Aquest mòdul s'està executant actualment en segon pla per sobre de tots els vídeos que es reben i està reduint de mitja en un 50% els errors d'aquests tipus que es detecten després de l'aplicació d'aquest mòdul.

7 Integració dels mòduls a l'agent

Amb els mòduls ja desenvolupats, ja es pot procedir a la part de la integració de tots ells dins d'un agent que

serà l'encarregat de gestionar-los. Aquesta integració s'ha dut a terme mitjançant la creació d'un agent principal, que és capaç de gestionar les crides a cada mòdul de forma autònoma i organitzada, seguint un flux definit.

Per assolir aquest objectiu, s'han implementat diverses tools específiques. Cada tool gestiona la crida a un dels mòduls concrets prèviament explicats. Això permet que l'agent pugui interactuar amb qualsevol component del sistema mitjançant un nom i una descripció funcional, sense necessitat de conèixer-ne els detalls interns.

Ha estat també necessari crear tools auxiliars encarregades de preparar l'entorn d'execució. En concret, aquestes eines tenen la funció de copiar automàticament els vídeos i els fitxers de metadades corresponents des del servidor d'origen fins a la màquina on s'executa l'agent.

L'agent actua, per tant, com un orquestrador intel·ligent, amb la capacitat de decidir quins mòduls s'han d'executar i en quin ordre, segons la consulta que se li hagi formulat. Aquesta arquitectura dona una alta escalabilitat a l'agent, permetent així afegir-li noves funcionalitats afegint noves tools sense necessitat de tocar res del que ja està implementat. Com s'ha explicat abans, l'agent s'ha implementat mitjançant la llibreria SmolAgents que proporciona un entorn de treball molt còmode pel tipus d'agent que es pretenia construir.

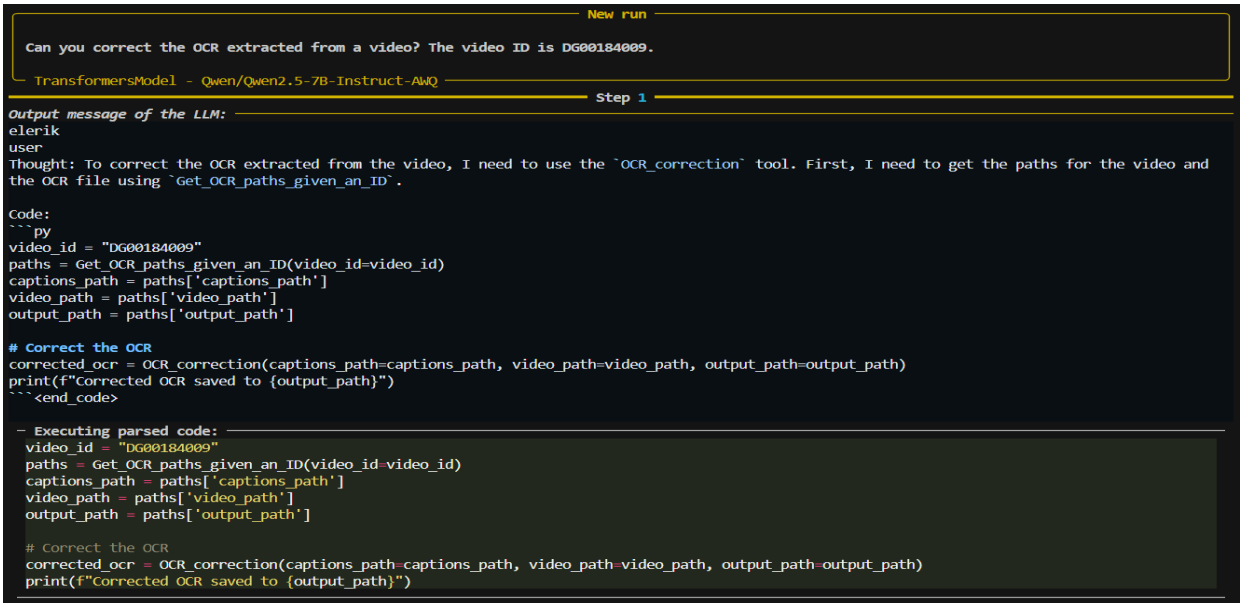


Fig. 5: Exemple d'execució d'una query per part de l'agent.

8 Implementació d'una interfície gràfica

Un cop es s'ha tingut l'agent funcional s'ha decidit generar una interfície gràfica per tal de que el client pugui utilitzar-lo d'una manera molt més còmoda i intuitiva. Aquesta interfície s'ha desenvolupat seguint una estructura web API. Pel desenvolupament de la web s'han utilitzat els llenguatges HTML, CSS i JS. En el cas de la API, aquesta s'ha implementat mitjançant la llibreria de python FastAPI [17]. La interfície té el següent aspecte:

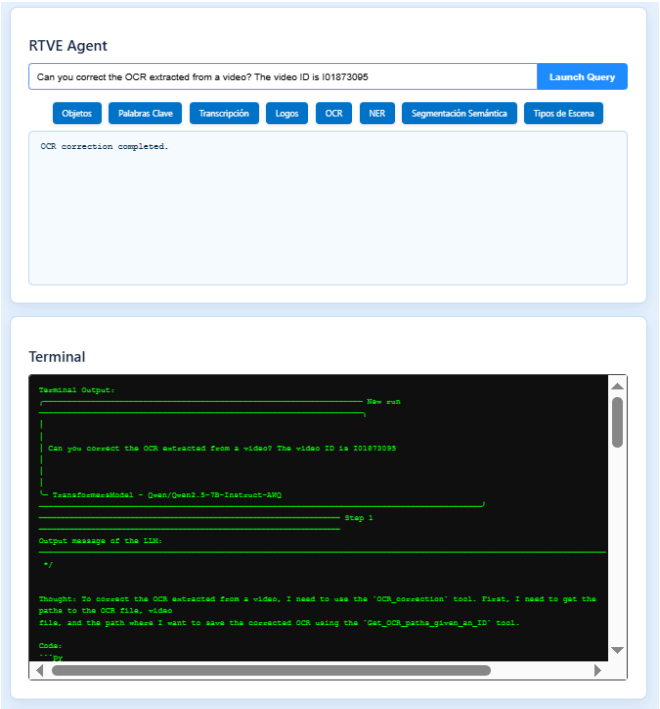


Fig. 6: Part superior de la interfície.

Fig. 7. Part inferior de la interfície.

En aquesta interfície s'hi poden trobar tres apartats diferents. En el primer calaix s'hi mostra el resultat obtingut de la consulta així com una recomanació sobre com formular els prompts per tal d'aconseguir un millor resultat. El següent apartat anomenat "Terminal" mostra totes les instruccions generades per l'agent que han acabat generant l'output. Això s'ha implementat utilitzant la llibreria io combinada amb una redirecció de la sortida estàndard (stdout), que és la que apareix al terminal. Aquest output redirigit es guarda en un fitxer el contingut del qual és llegit pel Front End de la interfície. Finalment, l'última caixa anomenada "Comparación" fa aparèixer uns botons que indiquen els instants de temps en què s'ha generat un canvi. En ser pressionats, en les dues caixes apareixen la versió anterior i la nova de la instància que s'està revisant.

9 Estructura i tecnologies utilitzades

Per a la implementació del projecte s'ha fet servir el llenguatge de programació Python, donada la seva versatilitat i el seu ampli ecosistema de llibreries orientades al tractament de dades, a la intel·ligència artificial i a la integració modular. El desenvolupament s'ha dut a terme dins l'entorn Visual Studio Code (VS Code), que ha permès una organització eficient del codi, l'ús d'entorns virtuals i la integració amb sistemes de control de versions.

Pel que fa a les llibreries utilitzades, la base del projecte se sosté en la llibreria SmolAgents, que facilita la creació d'agents autònoms en entorns Python. Aquesta s'ha complementat amb altres llibreries com Transformers, per a la interacció amb models de llenguatge preentrenats, i Scikit-learn, per a processos de validació i anàlisi. També s'han utilitzat llibreries de suport com json, per a l'estructuració i emmagatzematge de les metadades, zlib, per a la compressió de dades, i subprocess, que tindrà un paper rellevant en la integració final dels mòduls com a processos independents dins de l'agent principal.
Quant als models emprats, el sistema s'ha construït entorn de models de llenguatge multimodals, entre els quals destaquen Ministral-8B-Instruct i Qwen2.5-VL-7B-Instruct-AWQ, que han estat seleccionats per la seva eficiència i capacitat d'anàlisi tant textual com visual. Aquests models s'han aplicat a diferents mòduls del sistema com ara NER, OCR, paraules clau o descriptors visuals.

El format triat per a l'intercanvi d'informació entre mòduls i l'emmagatzematge de resultats és JSON, ja que proporciona una estructura clara, lleugera i fàcilment interoperable entre eines i tecnologies.
Finalment, pel que fa a la infraestructura de processament, s'ha fet servir una GPU NVIDIA GeForce RTX 4060 Ti amb 16 GB de VRAM, que ha estat suficient per entrenar i executar els models durant la fase de desenvolupament. Tot i així, es preveu que, per a la integració completa del sistema, podria ser necessària una targeta gràfica amb més capacitat. En cas que aquest requeriment es confirmi, s'indicarà explícitament a la versió final del document.

10 Conclusions

Gràcies a aquest projecte he pogut augmentar de manera significativa els meus coneixements sobre la implementació d'agents autònoms, aplicant-los directament a l'entorn laboral. La versatilitat que tenen permet que siguin utilitzats en molts àmbits diferents i puguin assolir qualsevol tipus de tasques que se'ls hi proposi. En el cas del meu treball, he pogut generar una eina útil que servirà per millorar considerablement totes aquelles dades que s'estan extraient. Aquesta eina també té una gran escalabilitat de tal manera que si es detecten nous errors a corregir o es volen afegir nous

mòduls de correcció es podrà utilitzar aquesta mateixa eina. A part dels agents, també s'ha evidenciat que els LLM poden ser utilitzats no només com a generadors de contingut, sinó també com a eines d'avaluació, validació i millora de resultats obtinguts per altres models.

D'altra banda, la metodologia iterativa i modular que s'ha realitzat ha resultat molt satisfactòria per a aquest treball. La possibilitat de desenvolupar, provar i corregir cada eina de manera independent ha facilitat tant la detecció d'errors com l'optimització dels recursos. Igualment, la integració posterior ha permès una coordinació eficient del conjunt del sistema.

Pel que fa als resultats obtinguts per l'aplicació de l'agent, com s'ha mencionat abans, aquests han sigut significativament positius. Gràcies a l'agent s'ha pogut reduïr els errors en tots els mòduls en que s'ha aplicat, destacant el 95% d'eliminació d'al·lucinacions gràcies al mòdul d'OCR i el 75% de millores en la generació de paraules clau.

Les possibles millores que es podrien fer d'aquest projecte van molt enfocades a l'optimització dels processos. Carregar LLMs resulta molt costós temporalment i l'execució de molts prompts sobre aquests fa que hi hagi mòduls que tardin un temps considerable en entregar una resposta.

Bibliografia

[1] «¿Qué es un LLM (modelo de lenguaje de gran tamaño)?», Amazon Web Services (AWS). [Online] Disponible a: https://aws.amazon.com/es/what-is/large-language-model/ [Accedit: 20-feb2025]

[2] «What is an intelligent agent?», TechTarget. [Online] Disponible a: https://www.techtarget.com/searchenterpriseai/definition/agent-intelligent-agent [Accedit: 25-feb2025]

[3] «SmolAgents» [Online] Disponible a: https://huggingface.co/docs/smolagents/index [Accedit: 25-feb2023]

[4] «Transformers», HuggingFace. [Online] Disponible a: https://huggingface.co/docs/transformers/index [Accedit: 6-mar2025]

[5] «Scikit-Learn», scikit-learn. [Online] Disponible a: https://scikit-learn.org/stable/ [Accedit: 1-mar2023]

[6] «Agents - Guided tour» HuggingFace. [Online] Disponible a: https://huggingface.co/docs/smolagents/guided_tour [Accedit: 7-mar2025]

[7] «zlib — Compression compatible with gzip», Python [Online] Disponible a: https://docs.python.org/3/library/zlib.html [Accedit: 11-mar2025]

[8] «DistilBERT base model (uncased)», HuggingFace [Online] Disponible a: https://huggingface.co/distilbert/distilbert-base-uncased [Accedit: 14-mar2025]

[9] «Silero-VAD», GitHub [Online] Disponible a: https://github.com/snakers4/silero-vad [Accedit: 20-mar2025]

[10] «The Quantized Ministral 8B Instruct 2410 Model», HuggingFace [Online] Disponible a: https://huggingface.co/shuyuej/Ministral-8B-Instruct-2410-GPTQ [Accedit: 24-mar2025]

[11] «Qwen2.5-VL-7B-Instruct-AWQ», HuggingFace [Online] Disponible a: http://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct-AWQ [Accedit 30-mar2025]

[12] «Pydantic», Pydantic [Online] Disponible a: https://docs.pydantic.dev/latest/ [Accedit 20-maig2025]

[13] «¿Qué es un agente de IA?», Google Cloud [Online] Disponible a: https://cloud.google.com/discover/what-are-ai-agents?hl=es [Accedit 10-jun2025]

[14] «llamaindex vs langchain vs hugging face smolagent a comprehensive comparison», Towards AI [Online] Disponible a: https://towardsai.net/p/machine-learning/llamaindex-vs-langchain-vs-hugging-face-smolagent-a-comprehensive-comparison [Accedit: 24-maig2025]

[15] «Chain-of-Thought Prompting», Prompt Engineering Guide [Online] Disponible a: https://www.promptingguide.ai/techniques/cot [Accedit: 24-maig2025]

[16] «Flask», Flask [Online] Disponible a: https://flask.palletsprojects.com/en/stable/ [Accedit: 7-jun2025]

[17] «FastAPI», FastAPI [Online] Disponible a: https://fastapi.tiangolo.com/ [Accedit: 7-jun2025]

- *E-mail de contacte: marcserrano0410@gmail.com*

- *Menció realitzada: Computació*
- *Treball tutoritzat per: Carlos García Calvo*
- *Tutor d'empresa: Francesc Tarrés*
- *Curs 2024/25*