Estudi de Viabilitat per a l'Optimització del Sistema de Gestió de Canvis en IT

Andreu Mir Villa

Resum— Aquest article presenta un estudi de viabilitat per a l'automatització del procés de validació de canvis en un sistema corporatiu, mitjançant tècniques d'aprenentatge automàtic. El sistema actual combina una primera validació automatitzada (RPA) amb una revisió posterior per part d'operadors humans. La proposta se centra a analitzar les dades disponibles, generar etiquetes de manera fiable i provar diferents arquitectures de modelatge per identificar les opcions més eficients. Els resultats mostren que els models basats en text lliure ofereixen un rendiment superior, i que l'absència d'etiquetes correctes és una limitació crítica. Es conclou amb la implementació d'un pipeline funcional i amb recomanacions concretes per a futurs desplegaments reals.

Paraules clau — aprenentatge automàtic, xarxes neuronals, automatització de processos, classificació binària, sistemes de gestió de canvis, embeddings.

Abstract— This paper presents a feasibility study for automating the validation process of IT change requests using machine learning techniques. The current system combines an initial RPA-based filter with a manual human review. The study explores data quality, automatic label generation, and model architecture evaluation. Results show that free-text-based models outperform others, though the lack of accurate labels remains a key limitation. A functional pipeline is delivered, along with recommendations for future deployment.

Index Terms — machine learning, neural networks, process automation, binary classification, change management systems, embeddings.

1 Introducció

AQUESTA recerca sorgeix com a resposta a una necessitat real plantejada per una empresa del sector tecnològic (ITnow), que gestiona de manera recurrent un gran volum de sol·licituds de canvi sobre la seva infraestructura informàtica. Aquestes sol·licituds —anomenades canvis— consisteixen en peticions d'usuaris per modificar qualsevol component del sistema, ja sigui de tipus software o hardware. Per a ser acceptades, aquestes peticions han de superar un procés de validació estructurat, format per dues fases clarament diferenciades i seqüencials.

La primera fase és un filtre inicial, automatitzat mitjançant un sistema de decisió determinista conegut com a RPA (Robot Process Automation), que avalua condicions simples basades en valors categòrics i descarta els canvis que no les compleixen. Si el canvi supera aquest primer control, passa a la segona fase, que consisteix en una revisió manual i supervisada per humans, on s'analitzen amb més profunditat tant els camps categòrics com els textos lliures associats al canvi. Només si es compleixen tots els requisits en ambdues fases, el canvi es considera vàlid i és acceptat.

El sistema actual de denegació, doncs, actua com una cadena de decisions encadenades, on l'automatització cobreix només una part inicial. La segona fase representa encara una càrrega important de temps i recursos per part dels operadors humans, malgrat tractar-se sovint de tasques repetitives i estructurades.

Aquest projecte es concep com un **estudi de viabilitat** per avaluar fins a quin punt és possible automatitzar aquesta segona fase (supervisada) del sistema. L'empresa mai abans havia realitzat **cap projecte sobre aquest sistema**, ni tampoc es disposa d'un inici de les dades necessàries. Els **objectius principals del treball** han estat:

1. **Determinar la viabilitat real del projecte**, establint si existeix una base tècnica i pràctica sòlida que permeti automatitzar parcialment o totalment aquesta etapa, tot mantenint els criteris de qualitat del sistema original.
2. **Descobrir els principals problemes** associats al cas d'ús, com la variabilitat en el llenguatge, la inconsistència en l'etiquetatge o les diferències entre blocs de dades, i **elaborar les solucions** més adequades per afrontar-los, tant en l'àmbit del tractament de dades com en el del disseny de models.
3. **Disposar d'una versió funcional de tipus Proof of Concept (PoC)** del sistema final, basada en el millor model disponible dins d'un pipeline complet, i acompanyada d'una **guia pràctica de mètodes**, amb recomanacions concretes sobre quin enfocament aplicar segons el context, així com alternatives adaptables a diferents situacions.

- *1637558@uab.cat*
- *Treball tutoritzat per: Ramón Baldrich Caselles (CVC)*
- *Curs 2024/25*

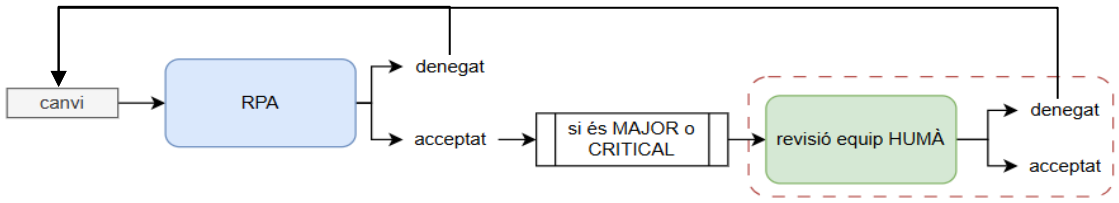

Fig. 1: Esquema del sistema de gestió de canvis IT de l'empresa. Compta amb dos processos clau. El primer és un procés automàtic determinista (RPA), que avaluar que els camps categòrics del canvi compleixin un seguit de requisits. Quan el canvi superi aquest procés, si és sensible (marcats com MAJOR o CRITICAL), aleshores passarà pel segon procés, que serà supervisat per un equip d'operaris., que mirant el text lliure i les dades categòriques acceptaran o denegaran el canvi. Aquesta serà la fase a automatitzar.

2 Punt de Partida

Aquest projecte parteix d'una situació poc habitual: **no existeixen precedents interns** ni projectes previs dins l'empresa que abordessin l'automatització del procés que es vol optimitzar. Aquest fet representa una dificultat afegida, especialment en les primeres fases del projecte, ja que no es disposa de cap referència directa ni d'experiències anteriors que serveixin com a guia o marc de comparació.

El procés de validació de canvis dins l'empresa pot ser considerat com un **exemple de procés ERP (Enterprise Resource Planning)**, específicament dins l'àmbit de la **gestió del canvi IT** [1][2]. Aquests processos tenen com a objectiu garantir que qualsevol modificació en el sistema informàtic es dugui a terme amb ordre, seguretat i eficiència. Implica coordinar els diversos agents implicats (stakeholders), mantenir una comunicació clara i constant i mesurar no només el grau de compliment formal sinó l'adopció real dels nous processos o eines.

En lloc d'optar per una plataforma ERP genèrica per a dur a terme aquest procés, **l'empresa ha desenvolupat un sistema propi i personalitzat** per a la gestió dels canvis d'infraestructura IT. Aquest sistema es basa en un **flux de decisió intern**, que combina automatització amb revisió humana, i que garanteix un control estricte sobre les modificacions que poden impactar l'operativa global.

El funcionament del sistema vist a la Fig. 1, segueix aquest flux:

1. **Un canvi entra al sistema**, generat per un usuari que sol·licita una modificació sobre algun component IT (software o hardware).
2. **El canvi passa primer per un procés automàtic** anomenat **RPA** (Robot Process Automation). Aquest sistema actua com a primer filtre, aplicant criteris simples i categòrics. Si el canvi **no compleix** els criteris definits, és **denegat automàticament** i el sol·licitant n'és notificat. En cas que **compleixi els criteris**, el canvi **passa a una segona fase** de validació.
3. Aquesta segona fase **és supervisada per humans** i només afecta als canvis que han estat **etiquetats com a MAJOR o CRITICAL**, ja que es considera que poden tenir un impacte significatiu o requereixen una anàlisi més acurada.
4. Durant aquesta revisió manual, el canvi torna a ser **avaluat i pot ser acceptat o denegat**. Si s'accepta, el canvi és executat i el procés es dona per tancat. Si es denega, es rebutja de manera definitiva i el sol·licitant rep una notificació, de la mateixa manera que en la primera fase.

Aquest sistema mixt —combinació de decisió automàtica i revisió manual— permet controlar de forma eficient la majoria de canvis, però també suposa una càrrega considerable per als humans en la segona fase. És justament aquesta part la que es pretén automatitzar mitjançant tècniques d'aprenentatge automàtic, en un entorn on fins ara no s'havia implementat cap solució similar.

3. Metodologia

La metodologia seguida en aquest treball ha estat marcada per la col·laboració amb una empresa privada del sector bancari, fet que ha requerit una gran cura en el tractament de les dades. Tot i que el treball s'ha desenvolupat de manera individual, el context empresarial ha condicionat alguns aspectes clau del procés.

A nivell organitzatiu, l'empresa planifica el treball mitjançant esprints de dues setmanes, durant els quals es detallen les tasques a complir. Aquest esquema ha facilitat la gestió i el seguiment dels objectius a curt termini, mantenint un ritme constant.

Pel que fa a la gestió personal del projecte, es realitza una reunió setmanal amb el tutor del TFG. Durant aquestes sessions es revisa el treball realitzat, s'estableixen les tasques a desenvolupar i es planifica l'estratègia per a la següent setmana. Sovint, es programen tasques que s'estenen més d'una setmana, però la periodicitat de les reunions assegura un seguiment continu i permet ajustar el pla segons les necessitats.

A mesura que es tanquen grans fites o es produeixen avenços importants, es documenta el treball, tant per a l'empresa com per a la posterior elaboració dels informes de seguiment. Aquesta pràctica garanteix un registre clar i ordenat de l'evolució del projecte.

Finalment, cal destacar que el treball amb l'empresa ha condicionat el ritme d'avanç en diverses fases, especialment en l'obtenció i preparació de les dades. Donada la dimensió de l'equip i la necessitat de coordinar-se amb diversos departaments, aquest procés ha estat més lent que el que es podria assolir treballant de manera totalment individual i autònoma. Aquest factor ha marcat clarament l'evolució i el desenvolupament global del projecte.

4. DADES UTILITZADES

4.1 Obtenció de les Dades

L'obtenció de les dades ha estat un procés complex a causa de la dificultat per accedir i interpretar l'estructura del Data Warehouse i la vista de MAXIMO[1], que presenten noms i formats diferents. En total, es disposa de 110.000 registres originals. Després de filtrar només aquells canvis que passen pel sistema —que són els canvis de tipus *Normal*— la mostra es redueix a 80.000 registres.

D'aquests registres, es compta amb nombrosos camps categòrics, però per aquest treball s'han seleccionat **14 camps categòrics** principals que utilitza actualment el sistema automatitzat RPA i que es revisen en el procés. A més dels camps categòrics, es treballa amb **tres camps de text lliure** que són analitzats en el procés de supervisió manual.

Aquests textos són la descripció del canvi, el motiu del canvi i la descripció de l'impacte.

A més, alguns registres tenien el text lliure en Português, pel que s'ha usat un algorisme de **detecció i traducció automàtica** amb Google Translate API. Per garantir la privacitat, totes les dades s'han **anonimitzat** mitjançant un algorisme propi. Això ha permès treballar amb les dades fora de l'entorn intern amb garanties de confidencialitat.

4.2 Rellevància del text lliure

Tot i comptar ja amb les dades relacionades amb el canvi, no es té la variable que indica si el canvi ha estat acceptat o denegat, i es depèn de l'empresa per conèixer com obtenir aquesta. Per no aturar-se, s'opta per realitzar un primer estudi per valorar la viabilitat de usar el text lliure en models de classificació de Machine Learning (viabilitat d'emular el comportament humà). Per tal, es crea una variable objectiu a partir del camp de **descripció de l'impacte**, classificat en cinc categories d'impacte mitjançant un algoritme propi basat en **keywords**.

Per a l'estudi, es van realitzar experiments amb models de **Random Forest** [8] utilitzant embeddings **word2vec**; també es van provar altres combinacions amb **XGBoost** i embeddings basats en **TF-IDF**. L'objectiu va ser avaluar la rellevància dels camps categòrics i de text per a futures aproximacions.

Per avaluar el rendiment dels diferents blocs de dades, s'han utilitzat mètriques mitjanes, concretament l'**F1-score mitjà** i el **Recall mitjà**. L'inferència es farà sobre les combinacions dels 2 tipus de dades que tenim; Categòriques (Cat.); i Text Lliure, on es tenen dues variables; Descripció (Desc.); Motiu del Canvi (Mot.). La variable de Descripció d'impacte com s'usa per la variable objectiu, no s'avaluarà en l'experiment.

Els resultats mostren que el pitjor rendiment s'obté amb només dades categòriques. Dins el text lliure, la descripció aporta més informació que el motiu, i combinar-les millora la predicció. Afegir les dades categòriques a qualsevol combinació sempre millora el resultat. Això demostra que el text lliure conté la informació clau, però les dades categòriques enriqueixen i complementen el model. Per tant, la combinació de totes dues fonts és l'opció òptima.

[1] MAXIMO: Interfície de l'empresa per obtenir la vista de les dades dels canvis d'IT, a partir d'una capa d'abstracció sobre la base de dades.

Combinació	F1-avg	Recall-avg
Cat.. + Desc. + Mot.	0.786	0.716
Cat. + Desc.	0.754	0.678
Cat. + Mot.	0.752	0.692
Desc. + Mot.	0.747	0.677
Desc.	0.666	0.593
Mot.	0.703	0.646
Cat.	0.569	0.511

Taula 1: Comparativa dels tipus de dades en la inferència per la classificació del impacte. El text lliure és la font principal d'informació, però les dades categòriques milloren els resultats quan s'hi combinen.

4.3 Alternativa a dades reals: generació d'etiquetes

Davant la incertesa de disposar de dades prèviament etiquetades i el fet que l'empresa no pot assegurar-ne ni l'existència ni la disponibilitat en el futur, es pren la decisió de generar les etiquetes de forma automàtica, a partir de la informació històrica disponible.

Aquesta generació es basa en l'anàlisi del registre d'estats dels canvis dins el sistema MAXIMO. Quan un canvi és denegat, el procés es reinicia: el canvi original es tanca i es genera un nou registre modificat. Aquest comportament permet inferir amb certa fiabilitat l'etiqueta de cada canvi.

El criteri seguit per etiquetar és el següent:

- Si el darrer estat d'un canvi és "denegat" i aquest està tancat, s'etiqueta com a *REJECTED = True*.
- Si el canvi finalitza en estat "acceptat" i no conté cap denegació prèvia, s'etiqueta com a *REJECTED = False*.

A més de l'estat final, també s'analitza qui ha pres la decisió (sistema automàtic o revisió humana), fet que permet etiquetar també la responsabilitat de la decisió com a RPA o Human (humà). Aquesta informació és essencial per segmentar correctament els casos i estructurar el sistema de decisió proposat.

Tot aquest procés ha estat desenvolupat mitjançant un algorisme propi, capaç de navegar per la complexa estructura de la base de dades i la plataforma MAXIMO.

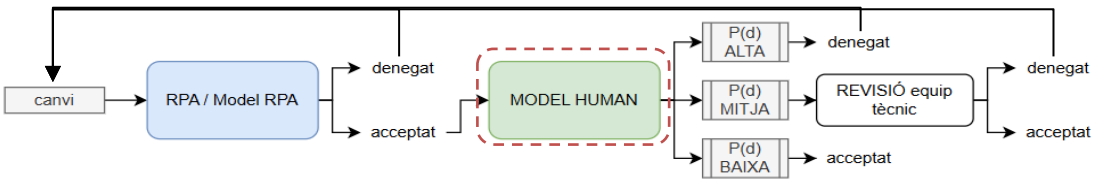

Fig. 2: Esquema de l'adaptació del sistema de gestió de canvis, que integra el sistema automàtic (RPA) com a primer filtre i model basat en Xarxes Neurals per emular el procés de decisió humà. El model retorna la probabilitat del canvi de ser denegat P(d). Aquesta arquitectura optimitza l'eficiència del procés, reduint la càrrega manual només als casos amb probabilitats mitges, on l'equip d'humans els revisarà.

5. Plantejament pel nou sistema automatitzat

A partir de l'anàlisi del funcionament actual del sistema, es proposa una nova arquitectura que combina les dues fonts principals de decisió: el sistema automàtic (RPA) (bloc blau de la Fig. 2), i un model de IA que emuli el comportament humà (bloc verd a la Fig. 2). En el cas del primer bloc, es considera la possibilitat de usar també un model de IA que emuli el seu comportament, però es reserva aquesta decisió als responsables d'arquitectura del sistema de l'empresa (explicació més detallada al Apèndix A5). Tot i així al llarg del projecte es treballarà (com es menciona al apartat 6.1), en aquest model d'emulació del RPA.

El flux proposat és pot observar a l'esquema de Fig. 2. A continuació es detallen les fases del canvi al sistema.

1. El canvi entra al sistema com habitualment.
2. Es processa primer pel RPA, el qual actua com a filtre automàtic segons les seves regles establertes.
3. Si el canvi és aprovat pel RPA, continua cap a la següent fase.
4. En aquest punt, una capa de decisió posterior determinarà si el canvi ha de ser revisat per una persona, o pot ser directament recomanat com acceptat o denegat segons certs criteris.
5. Quan la decisió no és clara (en un rang de probabilitat intermitja establert per l'empresa), el canvi es deriva a revisió manual.

Aquest a configuració permet alleugerir la càrrega de l'equip humà, reservant la seva intervenció per als casos realment ambigus, i mantenint coherència i seguretat.

6 Models de la base

6.1 Arquitectura dels models base

Un cop generades les etiquetes i definida la variable objectiu, es poden iniciar les primeres proves de classificació. L'objectiu en aquesta fase és validar la viabilitat del sistema proposat i establir una línia base de rendiment a partir de models senzills. Cal remarcar, que el conjunt de dades és reduït, degut a que al fer balanceig de les mostres amb la variable *REJECTED*, el conjunt es veu reduït a uns 10k registres (tant en la partició de dades Humà com la de RPA la xifra és similar).

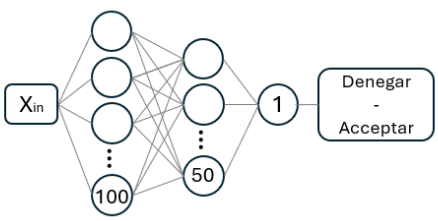

Fig. 3: Esquema del model MLP amb entrada de 64 característiques (14 bàsiques + 50 embeddings), amb capes de 100, 50 i 1 neurona, que retorna la probabilitat de denegació o acceptació del canvi.

Per a aquesta primera aproximació, es proposa utilitzar una arquitectura bàsica d'un classificador MLP (Multi-Layer Perceptron usant la llibreria de torch [3], compost per tres capes lineals de dimensions (entrada) → 100 → 50 → 1. Aquest model permet una primera exploració ràpida i controlada del comportament predictiu sobre les dades disponibles. Tots els models del projecte partiran de configuracions de MLP similars.

Tal com s'ha justificat a l'apartat 4.2, s'ha constatat que tant els camps categòrics com el text lliure aporten valor predictiu. Per això, el model treballarà combinant ambdues fonts d'informació. El text lliure es codifica mitjançant embeddings per convertir-lo en vectors numèrics; en aquest cas s'utilitza un model Word2Vec amb 50 dimensions per representar semànticament els textos.

Amb aquesta configuració, es construeixen els dos models corresponents al sistema proposat:

- **Model RPA**: entrenat i validat exclusivament amb dades en què la decisió ha estat presa pel sistema automatitzat (RPA).
- **Model Human**: entrenat i validat amb dades en què la decisió ha estat presa per un operador humà (Human).

EL RPA real, només treballa amb dades categòriques. S'ha optat per fer que el model RPA usi també les de text lliure, amb l'objectiu de tenir el millor model possible.

6.2 Resultats de classificació dels models base

Un cop definides les arquitectures inicials, es procedeix a avaluar el rendiment dels dos models base: **RPA** i **Human**. Els entrenaments s'han realitzat amb un conjunt de dades **balancejat** entre classes *REJECTED = TRUE* i *REJECTED = FALSE*, ja que al provar amb el conjunt desbalancejat no s'obtenien bons resultats (veure proves al Apèndix A2).

Model	Recall True	Recall False	Precision
m.RPA	0.786	0.921	0.891
m. Human	0.756	0.691	0.735

Taula 1: Taula comparativa del rendiment de la classificació dels dos models base, que servirà per fixar un de partida d'on millorar els models.

S'utilitzen les mètriques *Recall True* i *Recall False*, per avaluar l'esbaixament entre classes, així com el rendiment del model, pel que també s'utilitza la *Precision*. En primer lloc, s'analitzen els resultats del **model RPA**. Tot i que el RPA real no utilitza dades de text lliure, el model inclou tant **variables categòriques** com **text lliure**, per tal de maximitzar la seva capacitat predictiva. Veiem que el model RPA obté bons resultats, però **s'esperava un rendiment proper al 100%**, ja que recordem que **està emulant un procés determinista**. El resultat pot explicar-se amb el fet que no tenim tots els camps que usa el RPA, ja que l'empresa no té disponibles les vistes d'aquestes dades. A més, es detecta **un lleu esbiaix cap a la classe TRUE**, fet que es pot atribuir al desequilibri subtil present en l'origen de les dades etiquetades pel sistema real.

Pel que fa al **model Human**, el rendiment és clarament inferior —fet també esperat—, degut a la **complexitat i subjectivitat** inherent en les decisions humanes. A més, s'observa un **clar esbaixament** cap a la classe majoritària (*REJECTED = TRUE*) i també cap al conjunt d'entrenament. Aquest comportament indica que el model té dificultats per generalitzar adequadament les decisions humanes i que caldrà ajustar millor l'arquitectura.

Aquests resultats inicials són especialment útils per establir una ***baseline*** amb la que comparar els futurs models. A més ens ha permès veure el rendiment en els dos conjunts de dades (RPA i Human).

7 PROBLEMA DE L'ETIQUETA

7.1 Origen i explicació del problema

Un cop desenvolupats els primers models base i iniciats els experiments, es va detectar un **problema crític** que afecta directament l'**etiquetatge de la variable objectiu**. Aquest error no era detectable a l'inici, ja que es basava en una **explicació incorrecta** proporcionada pels responsables de l'empresa del sistema. Durant reunions posteriors es va aclarir un detall fonamental: quan un canvi és **rebutjat**, aquest **no es desa amb les dades del moment del rebuig**, sinó que es **modifica**, sobreescrivint la informació, i es torna a enviar. El sistema només conserva l'**últim estat** —normalment **acceptat**—, sense traça de les versions anteriors.

Això implica que molts dels canvis etiquetats com a **"rebutjats"**, tenen les **dades finals d'un canvi acceptat**. Com a conseqüència, les **etiquetes generades automàticament** no reflecteixen fidelment el context original del rebuig. D'altra banda, les etiquetes sobre l'**origen del canvi** (autor, RPA o humà) no es veuen afectades i es consideren **fiables**. Aquest problema no prové del nostre procés d'etiquetatge dissenyat, sinó del funcionament del sistema actual i de l'emmagatzematge de les que hi passen.

7.2 Solucions del problema

Davant d'aquesta situació, es van plantejar dues possibles solucions:

- **Solució 1: Modificar el sistema per desar els estats rebutjats**.
  Aquesta seria la solució òptima, ja que permetria recuperar les dades exactes dels canvis en el moment del rebuig. No obstant això, els responsables de l'empresa la van descartar pel cost d'emmagatzematge i manteniment que suposaria.
- **Solució 2: Captura local dels canvis rebutjats en temps real**.
  Com a alternativa viable, es proposa executar un script actiu durant mesos, que capturi i desi localment les dades dels canvis rebutjats en el moment exacte. Aquesta estratègia permetria construir un històric fiable, però cal temps per obtenir-ne un volum suficient, donat el baix nombre de canvis rebutjats per dia. A l'Apèndix A4 es fa una explicació més detallada d'aquesta solució.

Tot i que aquesta segona opció no resol el problema a curt termini, és l'única via raonable per obtenir dades verídiques de cara al futur. Arrel d'aquest problema, els experiments posteriors es centraran en avaluar diversos mètodes i models amb les dades actuals, assumint-ne les limitacions. A més, **els resultats obtinguts fins ara han estat coherents**, probablement perquè els models capten relacions internes entre el text lliure i les etiquetes. Això fa que els experiments segueixin sent útils per estudiar la viabilitat i comparar metodologies.

8 EXPERIMENTS PER LES PRIMERES MILLORES

8.1 Modificacions aplicades

Aquest apartat recull els primers experiments realitzats per millorar el rendiment del model **Human**, que partia de resultats baixos i presentava problemes importants com **overfitting en entrenament**, **esbiaix entre classes** i una elevada **sensibilitat a la partició de dades**. Tots els experiments s'han dut a terme utilitzant embeddings Word2Vec de 50 dimensions i mantenint la mateixa estructura d'avaluació.

Es detallen a continuació les modificacions aplicades, els motius de cada canvi i les millores observades:

- **Ràtio de partició 35–65**: es dona més pes al conjunt de test per obtenir una avaluació més realista de la generalització del model. Aquesta simple modificació redueix lleugerament l'overfitting sobre el conjunt d'entrenament.
- **Inicialització de pesos**: aplicada per reduir la variabilitat entre execucions que dificultava extreure conclusions. El rendiment mig es manté similar, però les diferències entre experiments disminueixen notablement.
- **Scheduler de *learning rate***: permet un descens progressiu del ritme d'aprenentatge per evitar mínims locals. La millora és modesta, però contribueix a una optimització més estable.
- **Ampliació a MLP de 3 capes**: es dona al model més capacitat per capturar relacions complexes. Això estabilitza els resultats i redueix l'overfitting tant sobre entrenament com entre classes.
- **Regularització (L2 i *Weighted Loss*)**: s'aplica per compensar l'esbiaix entre classes. S'obté un model menys sobre ajustat, però els resultats continuen mostrant certa inestabilitat entre execucions.
- **Dropout i Batch Normalization**: combinats per

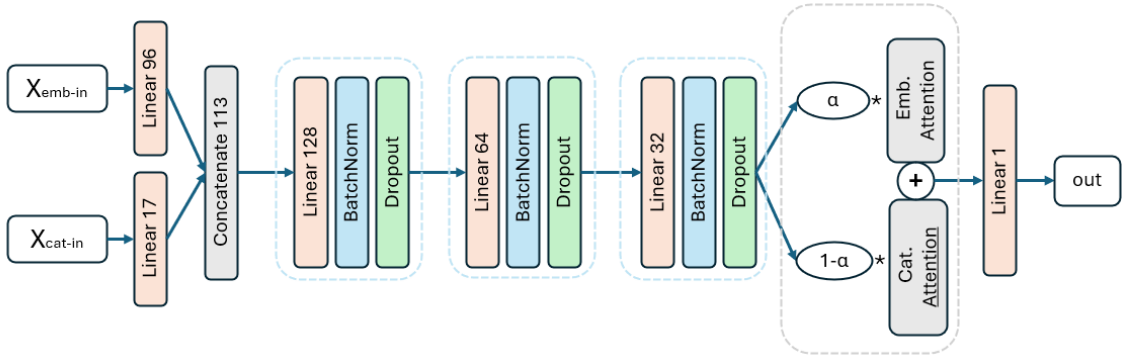

Fig. 4: Arquitectura MLP amb mecanisme d'atenció que combina representacions textuals i categòriques mitjançant una projecció conjunta, capes MLP i dues branques d'atenció paral·leles. A més d' una fusió amb pes aprenentable que integra ambdues fonts abans de la sortida binària.

controlar l'overfitting i millorar la regularització interna. Aquesta configuració aconsegueix un rendiment robust i consistent, sense inestabilitats.

- **Aplicació d'atenció**: es prova un model que incorpora mecanismes d'atenció tant sobre embeddings com sobre variables categòriques. Aquesta arquitectura millora notablement el *Recall* True i manté un bon equilibri amb el *Recall* False, sent el millor model obtingut amb *Recall True* ≈ 0.807, *Recall F* ≈ 0.756 i *Precision* ≈ 0.762.

Per tant, com s'observa a la taula de resultats Taula A.1 del Apèndix A.1, el millor model ha estat el que aplica Attention, explicat al següent apartat.

En tots els casos, existeix una clara sensibilitat del model a la partició de dades, degut a la poca quantitat d'aquestes. **Per poder avaluar realment l'efecte de cada millora i eliminar la inestabilitat deguda a la partició**, s'ha acabat aplicant un **model d'assemblatge amb bagging**. Aquest consisteix en la generació de 20 models independents, cadascun entrenat sobre una partició diferent de les dades. La predicció final es decideix per votació entre aquests estimadors. A diferència d'un enfocament *k-fold*, amb bagging es pot obtenir un **model final entrenat i usable**, més generalitzat. Aquesta tècnica fa desaparèixer la sensibilitat a la partició i permet extreure conclusions fiables dels experiments, tot i que incrementa el temps d'entrenament.

8.2 Arquitectura del Model d'Attention

El model amb atenció que ofereix el millor comportament tracta per separat les dades d'embedding i les categòriques.. L'estructura, observable a la Fig. 4, segueix el següent flux:

1. **Projecció inicial**: les entrades embeddings es projecten a 96 dimensions, i les variables categòriques a 17. S'uneixen i es passen a un espai de 128 dimensions.
2. **Bloc MLP**: tres capes *fully connected* amb *ReLU*, normalització i *dropout*, que redueixen fins a un vector de 32 dimensions.
3. **Atenció múltiple**:

- Una atenció *multihead* s'aplica sobre els embeddings, capturant relacions internes de la representació textual.
- En paral·lel, s'aplica una atenció simple sobre les variables categòriques, aprenent la seva importància relativa.

4. **Fusió amb pes aprenentable** α: el model aprèn automàticament fins a quin punt confiar en cada tipus d'informació. El vector final combina l'atenció d'ambdues fonts.
5. **Sortida**: un *fully connected* i una sigmoide proporcionen la predicció binària final.

Aquest disseny permet al model captar millor les interaccions rellevants en el conjunt de dades i generalitzar millor, convertint-se en la base per a futures millores.

9. Experiments amb embeddings

9.1 Experiments realitzats

Després d'explorar exhaustivament canvis estructurals al model (vegeu apartat anterior), es constata que el marge de millora a nivell d'arquitectura és molt limitat. Per tant, s'identifica com a línia principal de treball la millora dels **embeddings** del text lliure, que, tal com ja es va comentar a l'apartat 4.2, tenen un impacte rellevant en el model.

Aquestes primeres millores passaran per provar diferents embedders, usant com a classificador el model amb Attention, funcionant dins del model d'assemblatge amb bagging. Seguidament es detallen aquests primers embedders usats.

- **Word2Vec (W2V) [4]**: és el mètode base utilitzat fins ara. Funciona generant vectors per a cada paraula segons el seu context dins del corpus (CBOW o Skip-gram) i produeix un espai semàntic compacte i eficient.
- **Doc2Vec (D2V) [5]**: derivat de W2V però a nivell de frase o document. Es calcula la mitjana dels vectors de paraula juntament amb una representació pròpia per a cada frase, generant un embedding únic per cada text complet.
- **MiniLM [6]**: model Senten ce Transformer lleuger, que produeix embeddings a nivell de frase de 384 dimensions. Molt potent semànticament, però amb un espai de característiques elevat.
- **DistilBERT [7]**: una versió reduïda de BERT (Sentence Transformer), amb embeddings de 512 dimensions. És robust i potent, però encara més alt en dimensió que MiniLM.

El problema principal detectat és que el lèxic del text lliure que tenim és molt concret i els textos són curts. Això fa que un espai de característiques massa gran provoqui sobre ajustament i dificultats en la representació del text.

L'ideal seria un Embedder que amb poques característiques pugui representar text molt concret.

Per a tal es proposen dues idees:

1. **Sentence Transformer (ST) + Encoder**: S'afegeix un *autoencoder* (AE) entrenat sobre tot el text lliure del dataset. Aquest AE s'encarrega de reduir l'espai de característiques sortint del ST (com MiniLM o DistilBERT) a una dimensió personalitzable, mantenint la màxima informació rellevant. Això permet adaptar el embedding a les necessitats concretes del model.
2. **Finetuning (FT) del ST**: Es refina el model MiniLM utilitzant les dades anonimitzades del domini propi. Això permet adaptar el model al lèxic real de l'entorn. Encara que el conjunt de dades és limitat, aquest enfocament serà especialment útil a futur, quan es disposi de corpus més extensos.

Aquestes dues estratègies es combinen en nous experiments. Provarem el Encoder tant en el MiniLM com el DistilBERT, i el finetuning sobre el MiniLM amb Encoder.

A més, cal remarcar que, després de diversos tests, s'ha determinat que una **dimensió d'espai de característiques de 60** (explicació del experiment al Apèndix A3) és la més adequada per a l'Encoder. Aquest paràmetre podria canviar amb dades noves, per tant, caldrà tornar a entrenar l'AE i recalibrar el sistema si s'amplia el corpus.

9.2 Resultats obtinguts

Observem a continuació la taula de resultats dels diferents experiments d'embeddings. Els experiments s'han realitzat usant el model descrit a la Fig. 4, i usant el sistema de *bagging* ensambler, amb 20 models estimadors.

Embedder	Recall True	Recall False	Precision
W2V	*0.807*	*0.756*	*0.782*
D2V	*0.771*	*0.768*	*0.762*
MiniLM	*0.789*	*0.747*	*0.768*
DistlBERT	*0.778*	*0.752*	*0.762*
MiniLM + Enc	*0.806*	*0.782*	*0.794*
DistlBERT + Enc	*0.801*	*0.770*	*0.787*
MiniLM fine-tuned + Enc	*0.791*	*0.788*	*0.792*

Taula 2: Comparativa del rendiment del model amb els diferents embedders, usant el millor model amb bagging ensamble. En verd es marquen els millors models. S'opta pel darrer, ja que permet molt marge de millora amb el fine-tuning.

Els resultats mostren clarament l'impacte del tipus d'embedding sobre el rendiment global i l'equilibri entre classes. A continuació es resumeixen les principals observacions extretes de les proves:

- **W2V**: dona un prou bon resultat, però bastant esbiaixat per la falta de representació que ofereix.
- **D2V**: ofereix un rendiment relativament més baix, i no aporta millores significatives respecte W2V.
- **MiniLM**: bones prestacions però pitjor rendiment que W2V.
- **MiniLM + Encoder**: gran millora, redueix l'esbiaix i augmenta lleugerament la precisió.
- **MiniLM finetuned + Encoder**: el **millor model obtingut**, amb un balanç molt òptim entre *recall* True i False, i una precisió elevada. Tot i tenir resultats similars al MiniLM + Encoder, redueix encara més l'esbiaix entre classes.

Observant els resultats, podem concloure que afegint un Encoder millora significativament els models ST, permetent treballar amb representacions més compactes i eficients. A més, el **finetuning** permet adaptar el model al vocabulari específic, minimitzant el sobre ajustament.

El millor model és el que **combina Encoder + Finetuning**, ja que ofereix **bons resultats amb mínim esbiaix**. Amb dades reals futures i corpus més grans, aquest model serà probablement el més robust i escalable.

10. Experiments amb un Model Combinat

10.1 Idea del Model Global

L'objectiu d'aquest apartat és dissenyar un model capaç d'integrar diferents embeddings d'un mateix text per tal de captar millor la seva semàntica i millorar la capacitat predictiva general. La motivació és clara: cada embedding captura una perspectiva diferent del llenguatge —per exemple, Word2Vec reflecteix relacions sintàctiques i ocurrència de paraules, mentre que BERT o MiniLM capturen context i significat semàntic amb molta més profunditat. Combinant-los, es pretén aprofitar el millor de cada un.

Aquest enfocament, vist a la Fig. 5, es planteja com un **model combinat**, en el qual es combinen les sortides de diversos **models individuals**. Cada model individual té l'arquitectura del model d'Attention, però usa un Embedder diferent. Cadascun rep com a entrada el seu embedding corresponent (Word2Vec, Doc2Vec, MiniLM, MiniLM + Encoder, MiniLM finetuned, MiniLM + Encoder + Finetuning), juntament amb les variables categòriques.

Seguidament, els resultats individuals, es "fusionen" per donar una sortida única.

Sobre aquesta arquitectura, s'han treballat dues maneres d'entrenar el model, amb diferents opcions de "Fusió".

- En una primera versió, **s'entrena el model combinat usant els models individuals preentrenats**, i fusionant les sortides individuals amb una mitja ponderada.

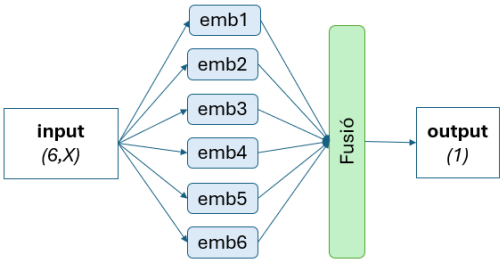

Fig. 5: Esquema del model combinat multi-input, amb una primera capa de 6 models amb embeddings diferents, seguida d'una capa de fusió que combina les sortides individuals i genera un output únic.

- En la segona versió, es proposa una millora fonamental: **entrenar alhora el model combinat i els models individuals**. Això es fa mitjançant una los pel model combinat, i una los per cada model individual.

Aquesta segona versió, com veurem al següent apartat, és molt més robusta, pel que es treballen varis mètodes de fusió de les sortides individuals:

1. **Fusió per mitjana simple**: Aquest mètode calcula la sortida de cada model individual (cada un generant un vector de 32 dimensions) i simplement en fa la mitjana element a element. Aquesta mitjana representa la combinació de totes les fonts d'informació, assumint que totes tenen la mateixa importància. Finalment, la mitjana es passa per una capa d'output que genera la predicció final. Aquesta és la fusió més senzilla i directa, sense aprendre pesos específics per cada model.
2. **Convolució horitzontal**: En aquest cas, els vectors resultants de cada model s'apilen i es tracten com canals en una dimensió seqüencial. Sobre aquest conjunt de vectors s'aplica una convolució 1D que fusiona les diferents sortides en un sol vector, capturant interaccions entre els models en la dimensió dels canals (models). Aquesta capa convolucional aprèn a ponderar i combinar automàticament les diferents sortides segons la seva rellevància conjunta. El resultat es passa després per una capa d'output per obtenir la predicció final.
3. **Convolució interna**: Aquest mètode primer genera una predicció per cada model individual amb la capa d'output, obtenint una sortida escalar per model. Aquests escalar s'apilen i s'organitzen com una seqüència que després es passa per una convolució 1D de baixes dimensions per aprendre combinacions i dependències entre aquestes prediccions individuals. Això permet captar relacions complexes entre les prediccions abans de fusionar-les en un únic valor final.
4. **Sub-CNN**: La fusió Sub-CNN utilitza una xarxa convolucional 1D jeràrquica que processa els vectors d'entrada combinant les seves característiques. Aplica successivament convolucions i funcions d'activació ReLU per extreure patrons complexos, seguida d'un pooling adaptatiu per reduir la dimensió i finalment una última convolució que genera un únic vector resum que serveix per a la predicció final. Aquesta estructura permet captar relacions internes i fusionar la informació de manera eficaç.

10.2 Resultats Obtinguts

La **primera versió provada**, basada en fusió per mitjana simple, tot i funcionar, té un gran inconvenient. Com que disposem d'un conjunt de dades limitat, el model combinat s'entrena i valida amb el mateix conjunt que els models individuals que el formen. Això fa que molts registres ja hagin estat vistos pels models individuals durant el seu entrenament, generant un **overfitting** (observable a la Fig. 6) en el conjunt d'entrenament que limita la generalització del model global.

Per això, es va optar per la **segona versió**, que evita aquest overfitting, com s'observa a la Fig. 7. Aquesta arquitectura és capaç de aprendre ela millor manera de combinar els vectors dels models individuals, per donar la classificació final. Seguidament es mostra el rendiment dels 4 mètodes provats.

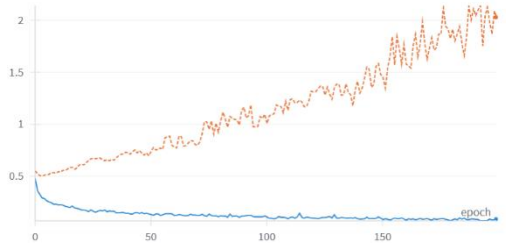

Fig. 6: Corba de Loss per 200 epochs de la 1a versió del model combinat. En taronja es mostra Loss del test, i en blau la del Train.

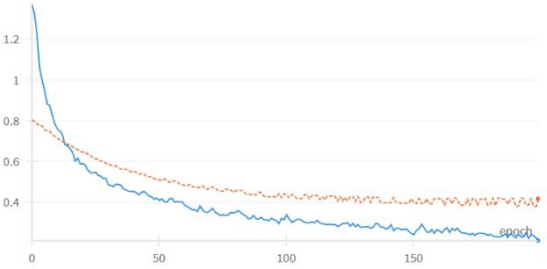

Fig. 7: Corba de Loss per 200 epochs de la 2a versió del model combinat. En taronja es mostra Loss del test, i en blau la del Train.

Mètode	Recall True	Recall False	Precision
1	*0.778*	*0.801*	*0.782*
2	*0.802*	*0.781*	*0.795*
3	*0.801*	*0.776*	*0.793*
4	0.812	0.810	0.805

Taula 3: Taula comparativa dels experiments amb el model combinat multi-input amb les 4 maneres de fusionar els vectors dels 6 models individuals en un únic valor.

Observem a la Taula 4, que el millor model és el que implementa el mètode 4, la Sub-CNN. Es considera millor model ja que dona resultats amb esbiaix de classes nul, i compta amb les millors mètriques fins al moment.

S'ha valorat la possibilitat de seguir optimitzant aquesta

CNN per explorar possibles millores addicionals, però es considera que dins del context actual —condicionats pel problema de l'etiqueta descrit al apartat 7.1— no seria eficient dedicar més recursos en aquest punt. El model ja compleix amb escreix els requisits de robustesa i rendiment.

L'ús d'aquest model en la versió final ja implementada al sistema, serà condicionat a la decisió de l'empresa d'assumir el cost extra d'entrenament que implica el model. Per aquest motiu, i per mantenir més simplicitat en el model, els experiments fets a continuació, utilitzen el millor model individual (apartat 9.2), en comptes del model combinat.

11. Experiment de Creuament de Dades

Per tal de validar la **robustesa i capacitat de generalització** del model desenvolupat, s'ha realitzat un experiment de **creuament de dades** utilitzant els dos conjunts disponibles: **Human RPA**. Fins ara, els models s'havien entrenat i validat per separat sobre cadascun dels blocs. L'objectiu d'aquest experiment és comprovar si el model pot mantenir un bon rendiment en un conjunt determinat, fins i tot quan ha estat entrenat amb dades provinents d'un altre origen.

En concret, s'ha analitzat com es comporta el model quan s'entrena amb la combinació **Human+ RPA**, però es testa només sobre un d'aquests conjunts per separat. S'han considerat dos escenaris:

1. Entrenament amb Human + RPA, test amb Human. (HR-R)
2. Entrenament amb Human + RPA, test amb RPA. (HR-H)

Conjunt	Recall True	Recall False	Precision
HR-H	*0.763*	*0.623*	*0.697*
HR-R	*0.886*	*0.873*	*0.879*
Human	*0.791*	*0.788*	*0.792*
RPA	*0.935*	*0.946*	*0.937*

Taula 4: Resultats del experiment de creuament de dades, usant el millor model individual (apartat 9.2). També es mostra el rendiment del model amb els conjunts de dades Human i RPA individuals.

Els resultats mostren dues situacions diferenciades:

- En el cas de **RPA**, el model presenta un rendiment **molt alt i consistent**, gairebé idèntic al que s'obtenia amb entrenament exclusiu sobre dades de RPA. Això demostra que **aquest conjunt és molt robust** i que el model és capaç d'integrar informació externa sense perdre capacitat predictiva. El rendiment elevat reforça també la qualitat i coherència de les dades de RPA.
- En el cas de **Human**, el rendiment es manté en valors acceptables, però amb **una lleugera davallada**, especialment en el recall de la classe negativa. Aquest comportament apunta a una **menor robustesa** sobre aquest conjunt, possiblement deguda a problemes d'etiquetatge ja mencionats anteriorment. Tot i això, **els resultats segueixen sent positius**, i es preveu que amb dades reals més ben definides, el model **també mostri un bon comportament**, tal com anticipen els bons resultats obtinguts en la resta d'experiments.

En conjunt, aquest experiment reforça la confiança en el model: **es demostra que és especialment robust en el cas del dades RPA** i que **té potencial per generalitzar bé en altres contextos**.

12. Nou estudi de rellevància del text lliure

12.1 Estudi amb nova variable objectiu

Després de completar els experiments per millorar i validar el model, continua existint una limitació important: la qualitat de l'etiquetatge, especialment pel que fa al camp *REJECTED*. Per tal de realitzar un últim anàlisi significatiu, es replanteja l'estudi de rellevància del text lliure, però ara amb dues diferències fonamentals:

1. **Usant el millor model** desenvolupat.
2. **Utilitzem una nova variable objectiu**, independent de la variable objectiu corrupta *REJECTED*.

Aquesta nova variable és l'**escenari** en què s'ha aplicat el canvi (pot ser "Producció Caixa" o altres). Aquesta no està afectada per errors d'etiquetatge i ofereix una alternativa més fiable per avaluar la utilitat del text lliure.

L'experiment es realitza amb tot el conjunt de dades disponible, **sense fer particions entre RPA i Human**, ja que aquesta partició va lligada amb la variable *REJECTED*, i la partició no aporta valor en el cas d'aquest experiment.

S'han entrenat tres models amb diferents combinacions de dades; un amb text lliure i categòriques (Tot), un segon només amb text lliure, i el tercer només categòriques.

Combinació	Recall True	Recall False	Precision
Tot	*0.631*	*0.849*	*0.753*
Text Lliure	*0.640*	*0.845*	*0.754*
Categòriques	*0.331*	*0.902*	*0.667*

Taula 5: Comparativa de rendiment de les diferents combinacions dades per a classificar el canvi escenari de "Producció de Caixa" o en "Altres", basat en les mètriques de Recall i en Precisió.

Els resultats mostren que les dades categòriques **redueixen el rendiment del model amb Text lliure i categòriques.**

El model que utilitza **només text lliure supera clarament** el que utilitza només variables categòriques i lleument el que combina ambdues.

Això indica que el text lliure conté informació rellevant i suficient per descriure l'escenari d'un canvi, i possiblement, el canvi en si.

12.2 Estudi amb millor model

El segon estudi s'ha dissenyat per completar l'anàlisi anterior, però en aquest cas **mantenint la variable objectiu de *REJECTED*** i emprant la **partició de dades Human**, ja que en aquest cas, com usem l'etiqueta de denegació, si que té valor usar la partició. Usem la partició Human ja que l'objectiu és veure la rellevància del text lliure en un model que emuli el comportament humà.

S'aplica novament el millor model individual, i

s'entrenen amb les mateixes combinacions de dades del subapartat anterior.

Combinació	Recall True	Recall False	Precision
Tot	*0.812*	*0.810*	*0.805*
Text Lliure	*0.787*	*0.743*	*0.765*
Categòriques	*0.650*	*0.752*	*0.707*

Taula 6: Comparativa de rendiment de les diferents combinacions dades per a classificar el canvi en acceptat/denegat, basat en les mètriques de Recall i en Precisió.

Observem que el model combinat de text lliure i categòric, és el que millor funciona, contrari als resultats obtinguts al subapartat anterior. Això confirma la rellevància del text lliure, i també la de les categòriques.

Amb aquests dos estudis podem concloure que el **text lliure és altament rellevant i predictiu**. Permet construir un model que **emet decisions similars a les dels humans**. Per tant, **és viable emprar models de xarxes neurals basats en text lliure i ajudats per dades categòriques per emular el comportament humà.**

13. Comparativa final dels models

Un cop ja donades per finalitzades les proves per aconseguir millores del rendiment dels models, veig adequat visualitzar una taula amb els increments aconseguits entre models, partint del model base vist a l'apartat 6.2. L'increment és el percentatge de millora respecte l'anterior. Per calcular-lo es pren la mitjana de les mètriques vistes fins al moment. S'inclou també el increment final del model base fins el darrer model.

Model	Mitja Metrics	Increment
1. Base	0.732	-
2. MLP Attention	0.772	*5.6%*
3. MLP + Enc + FT	0.782	*1.3%*
4. Model Global	0.809	*3.45%*
		10.52%

Taula 7: Comparativa dels diversos models treballats, mostrant l'increment del rendiment calculat com la mitjana de les mètriques usades al llarg del projecte.

Observem que el darrer model (descrit a l'apartat 10.1), obté una millora de més del 10% respecte el base. Aquest increment és també l'esperat en cas de realitzar els mateixos experiments ja amb dades correctament etiquetades. Això afirma la rellevància dels experiments.

13. Conclusions

Aquest projecte ha demostrat que **és tècnicament viable desenvolupar un sistema basat en xarxes neuronals** capaç de predir amb precisió si una sol·licitud de canvi IT serà acceptada o denegada. La clau d'aquest èxit ha estat l'aprofitament dels **camps de text lliure**, que han demostrat tenir un pes fonamental en la presa de decisions, molt per sobre de les variables categòriques tradicionals.

Tanmateix, un dels descobriments més rellevants ha estat la **detecció d'una limitació crítica del sistema actual**: la variable objectiu, no es desa al sistema actual, i l'obtinguda manualment, no reflecteix correctament l'estat real del canvi. Aquesta problemàtica ha estat tractada a fons a l'apartat 7, on s'ha proposat una **solució viable i aplicable** per capturar en temps real l'estat correcte dels canvis rebutjats. Aquesta proposta ja ha permès a l'empresa identificar una mancança important i ha motivat la dedicació de recursos per implementar una solució robusta.

Tot i la dificultat que suposa treballar amb etiquetes corruptes, el projecte ha aconseguit desenvolupar un **model final que supera clarament els primers models base**, i s'espera un molt bon rendiment al aplicar-lo a les dades correctes.

Cal remarcar, però, que **no ha estat possible fer una comparativa directa amb el pipeline corporatiu actual**, atès que aquest no es pot executar sobre dades històriques externes. Aquesta limitació impedeix una comparació exhaustiva, però **els resultats obtinguts amb el model desenvolupat deixen entreveure una millora significativa** en la capacitat predictiva del sistema.

Diverses fites del projecte **ja han aportat valor directe a l'empresa**. L'idea del model amb *fine-tuning* i *Encoder* personalitzat s'està aplicant en altres iniciatives internes, i el codi dels experiments de text lliure s'ha reutilitzat amb noves dades d'altres projectes. A més, la **detecció i solució del problema d'etiquetatge** ha revelat una limitació estructural i ha activat accions correctores concretes.

En definitiva, aquest projecte no només ha estat un **exercici d'aprenentatge tècnic i científic rigorós**, sinó també una **aportació real i significativa** al funcionament actual i futur de l'organització. És un pas ferm cap a la modernització i l'automatització intel·ligent dels processos de gestió del canvi IT.

Referències

[1] Invgate, "Organizational Change Management," Invgate ITSM, 2025. [Online]. Available: https://invgate.com/es/itsm/change-management/organizational-change-management. [Accessed: 23-Mar-2025].

[2] Calvert, Cheryl. "A change-management model for the implementation and upgrade of ERP systems." (2006).

[3] Collobert, Ronan, Samy Bengio, and Johnny Mariéthoz. "Torch: a modular machine learning software library." (2002).

[4] Rong, Xin. "word2vec parameter learning explained." *arXiv preprint arXiv:1411.2738* (2014).Activeloop,

[5] Lau, Jey Han, and Timothy Baldwin. "An empirical evaluation of doc2vec with practical insights into document embedding generation." *arXiv preprint arXiv:1607.05368* (2016).

[6] Peña, Fabian C., and Steffen Herbold. "Evaluating the Performance and Efficiency of Sentence-BERT for Code Comment Classification." *2025 IEEE/ACM International Workshop on Natural Language-Based Software Engineering (NLBSE)*. IEEE, 2025.

[7] Sanh, Victor, et al. "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." *arXiv preprint arXiv:1910.01108* (2019).

[8] Farooqi, M. Mashab, Hasan Ali Khattak, and Muhammad Imran. "Data quality techniques in the internet of things: Random forest regression." *2018 14th International Conference on Emerging Technologies (ICET)*. IEEE, 2018.

APÈNDIX

A1. TAULA DE RESULTATS DE LES PRIMERES MILLORES

Seguidament es mostra la taula de resultats dels canvis detallats a l'apartat 8.1.

Canvi	Recall True	Recall False	Precision
Model Base	0.754	0.691	0.735
Split ratio (35 – 65)	*0.7*	*0.691*	*0.735*
Init weights	*0.762*	*0.701*	*0.738*
LR scheduler	*0.765*	*0.712*	*0.745*
SGD optimizer	*0.657*	0.683	*0.722*
Weighted Loss	*0.765*	0.726	*0.738*
L2 regularization	*0.771*	0.752	*0.742*
MLP drop-out & batch norm	*0.779*	0.759	*0.751*
MLP amb Attention	0.781	0.762	0.774

Taula A.1: Comparativa mitjançant les mètriques de Recall per classes i de Precision els diferents canvis realitzats sobre el model base.

A2. EXPERIMENTS BALANCEIG DE DADES

Com s'ha comentat a l'apartat 4.1 i 6.1, tot i tenir un volum elevat de dades inicialment (fins a 110k), aquest s'ha vist reduït després d'aplicar certs filtres de dades que integra el sistema de gestió de canvis de l'empresa. I s'ha reduït encara més al balancejar el conjunt amb a les etiquetes generades. Aquest balanceig ve justificat per un mal rendiment del model en el conjunt sense balancejar. A continuació a la Taula A.2 s'observa la diferència.

Model	Recall T - F	Precision
Sense Balanceig	*0.69 – 0.81*	*0.80*
Amb Balanceig	*0.75 – 0.69*	*0.81*

Taula A.2: Comparativa de rendiment dels models base usant conjunt de dades balancejat cap a la classe minoritària versus sense balancejar.

Observem com efectivament amb Balanceig el esbiaix es redueix significativament, i obtenim resultats més estables dins el que cap.
Es van voler seguir amb les proves, per veure si es podia treure partit al fer de tenir mostres inutilitzades. Per a ferho, va aplicar-se una Loss Ponderada per classes (veure formula (1)), de la llibreria torch, i va jugar-se a afegir o treure mostres. Aquestes però no tenien efectes en el rendiment, i seguíem obtenint resultats similars o pitjors al del data set desbalancejat.
Tanmateix aquesta conclusió s'aplica únicament al nostre context, on tenim la variable objectiu corrupta. Per tant, en un entorn real, es possible que jugant amb la ponderació de la Loss, es pugui obtenir millors resultats.

A3. EXPERIMENTS PEL TAMANYS DEL EMBEDDING

Com bé s'ha comentat a l'apartat 9.2, per triar el millor tamany per al embedding, s'han dut a terme varis experiments amb diferents mides. Seguidament, i de la mateix manera que fins ara, es mostra la taula amb els experiments realitzats. Les proves ja s'han fet usant el model MLP amb Attention i amb del Encoder amb el MiniLM. Ja que en el moment de realitzar aquests experiments ja s'havien fet les primeres proves amb aquest model. Igual que en la resta de proves, s'ha emprat bagging ensamble de 20 estimadors per resultats més fiables.

Tamany	Recall True	Recall False	Precision
3	*0.754*	*0.719*	*0.729*
4	*0.765*	*0.732*	*0.732*
6	*0.762*	*0.743*	*0.735*
8	*0.768*	*0.748*	*0.740*
10	*0.771*	0.751	*0.743*
15	*0.775*	0.754	*0.750*
20	*0.777*	0.757	*0.761*
40	*0.780*	0.758	*0.770*
60	0.781	0.762	0.774
80	0.779	0.751	0.771
100	0.776	0.749	0.752

Taula A3: Comparativa dels diferents tamanys d'embedding usant el model de MiniLM amb Encoder. Permet descobrir quin és el millor tamany d'embedding

Observem a la Taula A.3, que el millor ha estat el de 60, però tampoc amb molta diferència amb els veïns propers. Això és estrany ja que el lèxic com ja s'ha comentat al apartat 4, és curt, de entre 5 i 15 paraules. Pel que s'esperava que donés millor rendiment un tamany similar al nombre de paraules, com succeeix en la majoria de casos de embedders.
Igual que a l'apartat anterior, aquest resultat pot ser degut al problema de l'etiqueta corrupta. És per això que serà necessari repetir aquest experiment un cop es tinguin les dades reals.

A4. EXPLICACIÓ DE LA SOLUCIÓ AL PROBLEMA DE L'ETIQUETA

Davant la limitació estructural del sistema, que no conserva traça dels canvis denegats una vegada modificats, s'ha dissenyat una estratègia per capturar en temps real els canvis rebutjats, abans que les seves dades siguin sobreescrites. La solució plantejada es basa en l'execució d'un script iteratiu que monitoritza la base de dades durant un període prolongat de temps, amb l'objectiu d'acumular un volum suficient de dades etiquetades per entrenar models robustos i fiables, tal i com es menciona a l'apartat 7.2.
Captura de canvis denegats (*REJECTED = True)*
Per a identificar els casos denegats, el sistema seguirà la següent lògica:

1. Inicialització del monitoratge: L'script recull tots els identificadors (ID) de canvis actius que es troben en estat no tancat i desa una còpia temporal

de les seves dades actuals, incloent-hi un hash global per **detectar possibles modificacions posteriors.**

2. Seguiment del registre d'estats: L'script observa contínuament les entrades del registre d'estats. Quan apareix una nova entrada per a un dels canvis monitoritzats, es verifica si el nou estat és ***"REASON REJECTED".***
3. Verificació de la integritat de dades:
   - **Si les dades del canvi no han estat modificades** (hash inalterat), es consideren vàlides i es guarden etiquetades com a REJECTED = True, amb l'autor extret del registre d'estats.
   - **Si les dades ja han estat modificades**, s'utilitza la còpia temporal feta al pas 1 per recuperar l'estat anterior al rebuig. Aquesta versió es guarda com a exemple de canvi denegat, també amb l'autor corresponent.

Aquest sistema permet capturar el moment exacte en què el canvi ha estat denegat, preservant el context i les dades originals que han provocat la decisió.
Per a identificar els casos acceptats, es defineix un flux alternatiu:

1. Es detecten canvis que han passat de l'estat *IN-PRG* (en progrés) a *CLOSED*.
2. Es recuperen les dades més recents del canvi en el moment de tancament, que es guarden amb l'etiqueta *REJECTED = False* i l'autor extret del darrer registre d'estats.

El sistema estima que pot etiquetar aproximadament **2.000 canvis mensuals**, dels quals **uns 1.200** corresponen a canvis de tipus *Normal* (els únics que passen per tot el sistema). S'estima que **almenys 800** d'aquests seran acceptats, i la resta, denegats.
Com que l'objectiu és construir un conjunt **balancejat**, es proposa aturar l'script un cop s'assoleixi el nombre necessari de canvis denegats (REJECTED = True). Si s'estima una mitjana de **300–400 canvis denegats mensuals**, i es busca obtenir **10.000 registres totals** amb **5.000 denegats**, el procés complet duraria aproximadament **12 mesos**. Aquesta és una estimació inicial que s'haurà de revisar un cop l'script estigui actiu i es conegui el ritme real de generació de canvis.
Validació de l'script
Atesa la naturalesa **temporal i asincrònica** de les dades reals, és molt difícil verificar el funcionament correcte de l'script directament sobre el sistema productiu. Per aquest motiu, s'ha optat per una **fase prèvia de validació en entorns sintètics**, generant manualment conjunts de dades que simulen els fluxos esperats de canvi, amb estats i modificacions controlades.
Aquest enfocament permet comprovar si l'script és capaç de:

- Detectar correctament els estats de rebuig.
- Comprovar la integritat de les dades abans de ser modificades.
- Recuperar la versió correcta del canvi a etiquetar.

Només si es confirma el comportament correcte amb dades simulades, es procedirà al desplegament en entorn real.
Consideracions finals
És important remarcar que el **primer desplegament** de l'script pot no ser completament funcional, ja que hi ha múltiples variables desconegudes, com ara:

- El nombre real de canvis diaris (acceptats i denegats).
- El temps mitjà entre la creació del canvi i la resolució (acceptació o rebuig).
- La variabilitat en el comportament dels usuaris i processos.

Aquestes incerteses podrien afectar el rendiment de l'script, especialment durant les primeres setmanes. És per això que es recomana monitoritzar activament els primers resultats, analitzar el ritme de recollida i revaluar la seva continuïtat abans d'arribar als 12 mesos de monitoratge previstos.

A5. Opcions pel RPA del nou sistema automatitzat

Tal i com es comenta a l'apartat 5, i com es veu al bloc blau de la Fig. 2, es plantegen dues opcions per tal de dur a terme el procés de decisió determinista en el nou sistema automatitzat.
La primera passaria per utilitzar el mateix RPA que s'usa en el sistema actual. Com a beneficis es té que no caldria entrenar cap model, ni dedicar costos de còmput ni temps per a generar aquest. A més, t'assegures que funcioni al 100% de precisió. Per contra, si es volgués usar el model Human (bloc verd a Fig. 2) en el sistema on ara està integrat el RPA, degut a la complexitat de l'arquitectura i del funcionament, implicaria redissenyar el flux i realitzar varies modificacions a baix nivell, les quals l'empresa per temes de complexitat es possible que no vulgui assumir.
D'altre banda, la segona opció, seria usar un model de IA també per emular el comportament del RPA. D'aquesta manera el cost d'integració d'ambdós models al sistema seria molt més senzill, ja que directament es redissenyaria nun sistema que realitzes crides a aquests models quan fos necessari. Per contra, mai aconseguirà emular al 100% el comportament del RPA actual, tot i que podria intentar ferse més determinista, integrant algunes condicions que permetessin replicar el comportament del RPA. A més, igual que el cas del model Human, tindria un cost de manteniment addicional, que seria el reentrenament del model cada cert temps, per tal d'actualitzar-lo amb les dades més noves.
Sigui com sigui aquesta és una decisió que cau en mans dels responsables de l'arquitectura de la gestió de Canvis de IT de l'empresa, i que per tant, el màxim que es pot fer en aquest projecte, és plantejar les opcions, tal i com s'ha fet.