HigiaHealthCode

Eina de codificació d'històries clíniques amb CIM-10

Marc Serret Monserrat

Màster Universitari en Ciència de Dades

Àrea 3: Machine Learning and Computer Vision in Healthcare and Medical Applications

Tutor/a de TF

Susana Pérez Álvarez

Professor/a responsable de l'assignatura

Laia Subirats Maté

Data Lliurament

Diumenge, 25 de maig de 2025

Universitat Oberta de Catalunya

Fitxa del Treball Final

Títol del treball:	HigiaHealthCode: Eina de codificació d'històries clíniques amb CIM-10
Nom de l'autor/a:	Marc Serret Monserrat
Nom del Tutor/a de TF:	Susana Pérez Álvarez
Nom del/de la PRA:	Laia Subirats Maté
Data de lliurament:	05/2025
Titulació o programa:	Màster Universitari en Ciència de Dades
Àrea del Treball Final:	Àrea 3: Machine Learning and Computer Vision in Healthcare and Medical Applications
Idioma del treball:	Català
Paraules clau	(DL)*Deep Learning*, (ML)*Machine Learning*, PLN (Processament de llenguatge natural)

Resum del Treball

El projecte es basa en el desenvolupament d'un sistema d'ajuda a la codificació d'altes mèdiques mitjançant tècniques de processament de llenguatge natural (PLN) i *deep learning*. L'objectiu principal és agilitzar la codificació de diagnòstics en CIM-10 a partir dels textos clínics redactats pels professionals assistencials, entrenant el model amb la codificació realitzada per experts en codificació mèdica. Aquesta eina busca reduir el temps dedicat a la codificació manual i millorar la coherència i precisió del codis assignats.
Per la implementació és farà servir una Pytorch com a eina principal per al desenvolupament dels models de *deep learning*. Els textos clínics emprats inclouen informació rellevant com la malaltia actual, evolució del pacient i altres dades clíniques recollides des de l'ingrés fins l'alta.
El sistema resultant ha de ser una eina de suport que faciliti la identificació i qualitat de la informació clínica codificada.

Abstract

The project is based on the development of a support system for medical discharge coding using natural language processing (NLP) techniques and deep learning. The main objective is to streamline the coding of diagnoses in ICD-10 from clinical texts written by healthcare professionals, training the model with coding performed by medical coding experts. This tool aims to reduce the time spent on manual coding and improve the consistency and accuracy of the assigned codes.
Pytorch will be used as the main tool for developing the deep learning models. The clinical texts used include relevant information such as the current illness, patient evolution, and other clinical data collected from admission to discharge.
The resulting system is intended to be a support tool that facilitates the identification of diagnoses and improves the quality of the coded clinical information.

Índex

1. Introducció

1.1. Context i justificació del Treball

El projecte HigaHealthCode sorgeix com a resposta a una necessitat detectada dins l'empresa on treballo actualment: la Xarxa Sanitària, Social i Docent de Santa Tecla. Aquesta entitat, que gestiona un ampli conjunt de centres sanitaris a l'àrea del Tarragonès i Baix Penedès, així com centres de serveis d'atenció intermèdia, residència i centres d'atenció primària, s'enfronta a un volum molt elevat de codificació d'altes mèdiques.

En l'actualitat, la codificació d'aquests diagnòstics, es basa en la CIM-10 (*International Classification of Diseases, Tenth Revision, Clinical Modifications*) es realitza amb l'estructura i els recursos disponibles, però el gran nombre de centres i la quantitat d'altes que generen produeix una càrrega de treball considerable. A causa de la demanda i a la necessitat de revisar més tipus d'activitats fa que la quantitat de treball hagi estat augmentant, fent molt difícil mantenir el nivell de qualitat exigit en la codificació d'altes clíniques.

En resum la justificació del projecte es basa amb els següents punts:

- **L'impacte en la gestió clínica i administrativa**: Una codificació automàtica i més precisa per millora la qualitat de la informació clínica, essencial per a la presa de decisions i la gestió hospitalària.
- **L'oportunitat de millorar processos**. La implementació d'una eina tecnològica avançada permetrà reduir els temps de processament i minimitzar error, contribuint a una gestió més eficient dels recursos.

1.2. Objectius del Treball

L'objectiu d'aquest projecte és desenvolupar un sistema d'ajuda a la codificació d'altes mèdiques basa en tècniques de processament de llenguatge natural (PLN) i *deep learning,* que permeti automatitzar la classificació de diagnòstic en CIM-10.

S'han establert els següents objectius:

1- Objectiu principal:
   - Desenvolupar un model de deep learning capaç d'automatitzar la codificació d'altes mèdiques a partir de textos clínics, millorant la precisió i l'eficiència del procés en un entorn real.

2- Objectius secundaris
   - Recollida i pre-processament de dades:
      - Extreure textos clínics d'una història clínica, garantint el compliment dels requisits de seguretat i privacitat.
      - Realitzar una neteja, normalització dels textos, així com la tokenització i vectorització utilitzant models de PLN preentrenats.
   - Desenvolupament i entrament del models:
      - Implementar i entrenar diverses arquitectures de *deep learning* (xarxes neuronals recurrents i transformadors) mitjançant Pytorch.
      - Ajustar els hiperparàmetres del model per optimitzar el rendiment, utilitzant tècniques de validació creuada per evitar el overfitting.
   - Validació i comparativa
      - Comparar els resultats obtinguts amb la codificació manual realitzada per experts, utilitzant mètriques com la precisió, el *recall* i el *F1-score*.
      - Realitzar un anàlisi d'errors per identificar àrees de millorar i validar la robustes del model.
   - Integració i avaluació pràctica.
      - Desenvolupar una API que permeti la integració del sistema dins del flux clínic de una històfria clínica.
   - Futures implementacions.
      - Analitzar possibles millores del model, processar textos en diversos idiomes i integrar-ho dins el model, o ajudar amb la codificació dels procediments (CIM10-SCP), explorar altres tecnologies emergents que puguin optimitzar aquest procès.

1.3. Impacte en sostenibilitat, ètic-social i de diversitat

- Sostenibilitat:
  La implementació d'un sistema automatitzat permetrà una optimització dels recursos humans dins l'empresa, en reduir la dependència del procés manual, els professionals dedicats a aquest àmbit podran invertir més temps en aquells casos que realment ho necessitin a més que podran dedicar més temps a formar-se, fet que de manera intrínseca farà millorar el sistema. Aquest enfocament afavorirà pràctiques més qualitatives en la gestió documental i administrativa.
- Ètic-social:
  El desenvolupament d'aquesta eina ha de complir rigorosament amb els estàndards ètics i de seguretat, per sobre de tots en l'àmbit de protecció de dades personals. Per això es garanteix el compliment del reglament general de protecció de dades (RGPD), assegurant que les dades tractades siguin tractades amb la màxima confidencialitat i seguretat, important dir que després del anàlisis de cada cas aquesta informació mai es desarà dins el model. També cal tenir en compte un dels riscos més grans, al tractar-se d'una eina de (ML), pot induir a la falsa predicció de codis diagnòstics degut a biaixos en les dades d'entrada, de manera que només servirà com una ajuda a la codificació i mai com a sistema autònom. (1)
- Diversitat:
  En el context sanitari i la zona geogràfica on ens trobem l'eina ha de reconèixer i adaptar-se a les variabilitats lingüístiques, culturals i regionals. Aquesta adaptabilitat garantirà que la solució sigui inclusiva i aplicable a tots els professionals independentment de la llengut utilitzada.

En definitiva, el projecte busca una gestió més sostenible dels recursos, un tractament ètic i segur de la informació i la promoció d'una pràctica inclusiva que té en compte la diversitat dins l'empresa.

1.4. Enfocament i mètode seguit

L'enfocament adoptat per al desenvolupament és basa en una gestió integral del projecte, ja que es tracta d'un projecte nou des de zero. Utilitzarem una metodologia àgil basada en Scrum, de manera que s'aniran realitzat entregues parcials rebent comentaris i propostes de millora per part de la tutora del treball i aplicant les modificacions en cadascuna de les iteracions.

Utilitzarem part de la metodologia Srum:

- Sprints curts.
  El projecte es dividirà en cicles curs de treball, cadascun amb objectius clars i definits. Al final de cada cicle s'avaluaran els resultats i es realitzarà l'ajust sobre la planificació.
- Revisions i retrospectives
  Cada cicle conclourà amb una revisió per tal de valorar les millores i els inconvenients que vagin apareixent.

Estratègia de recerca:

Es fonamenta en utilitzar una base sòlida en la teoria amb la finalitat de desenvolupar una aplicació completa.

- Revisió de documentació sistemàtica
  Es realitzarà una revisió de documentació contínuament durant el desenvolupament del projecte, te com a finalitat la cerca de la millor estratègia per a desenvolupar les eines basades en llenguatge natural i *deep learning.*
- S'avaluarà el model utilitzant *train-test split amb hold-out validation*, com a metodologia inicial, però amb un enfocament dinàmic i iteratius per millorar contínuament el rendiment del model. El model inclourà la reintroducció dels casos validats dins del model perquè aquest pugui aprendre progressivament i adaptar-se als nous patrons.

1.5. Planificació del Treball

HigiaHealthCode

TFM - Ciencia de dades
Marc Serret Monserrat

Inici del projecte: 16/02/2025

TASCA	PROG	INICI	FINAL
Mòdul 1 - Definició i planificació del treball final	100%	16-2-25	9-3-25
Definició del TFM: enunciat i lliurament (M1)	100%	16-2-25	9-3-25
Inicialitzar GIT	100%	16-2-25	9-3-25
Instal·lar entorn	100%	16-2-25	9-3-25
Avaluar dades de l'historia clinica	100%	16-2-25	9-3-25
Mòdul 2 - Estat de l'art o anàlisi de mercat del prc	100%	10-3-25	30-3-25
Revisió mòdul 1	100%	10-3-25	10-3-25
Cerca de models de PLN	100%	10-3-25	20-3-25
Cerca de estrategia per al DL	100%	10-3-25	20-3-25
Preparar neteja/extracció de dades	100%	21-3-25	30-3-25
Mòdul 3 - Disseny i implementació del treball	100%	31-3-25	4-5-25
Revisió mòdul 2	100%	31-3-25	31-3-25
Programació de la api	100%	1-4-25	15-4-25
Entrenament del model	100%	16-4-25	30-4-25
Avaluacio del model	100%	1-5-25	4-5-25
Mòdul 4: Redacció de la documentació del TFM	75%	4-5-25	3-6-25
Revisió mòdul 3	100%	4-5-25	4-5-25
Redacció de la memòria: lliurament preliminar (M4)	100%	4-5-25	18-5-25
Redacció de la memòria: lliurament final (M4)	100%	19-5-25	25-5-25
Presentació audiovisual del treball (M4)	0%	25-5-25	3-6-25
Mòdul 5: Defensa del projecte	0%	4-6-25	27-6-25
Lliurament de la documentació	0%	4-6-25	6-6-25
Preparar presentació defensa	0%	4-6-25	27-6-25

1.6. Breu sumari de productes obtinguts

El projecte ha generat els següents productes.

- Model *deep learning* entrenat:
  Es desenvoluparà i entrenarà un model de *deep learing* basat en arquitectures de xarxes neuronals que serà capaç de processar i analitzar textos clínics per assignar codis CIM-10 amb un alt grau de precisió.
- Validacions i avaluacions del model.
  Es realitzarà un estudi que realitzarà una validació comparant els codis generats amb els codificats per un expert, utilitzant mètriques com la precisió, el *recall* i els F1-score. A causa de la poca capacitat de classificació només s'ha pogut estudiar que el model ha après sense resultats concloents.
- Un aplicació que mitjançant una API incorpori tot els mòduls desenvolupats.
- Documentació tècnica i manuals d'usuari.
  Elaboració d'una documentació tècnica per a la implementació del programari.

1.7. Breu descripció dels altres capítols de la memòria

- Materials i mètodes:

Aquest capítol descriu de manera detallada la metodologia emprada en el desenvolupament del treball. L'enfocament s'ha centrat en tres àrees clau: la gestió i pre-processament de les variables, la implementació dels models de processament de llenguatge natural (PLN) i del disseny del model de *deep learning.*

- Resultats

Aquest apartat presenta l'anàlisi de resultats obtinguts després de l'entrenament i validació del model. S'explica com es compara els codis generats automàticament amb la codificació manual realitzada per tècnics en documentació clínica, mitjançant diverses mètriques.

- Conclusions

Finalment es resumeixen les conclusions obtingudes del projecte. A més es proposen línies futures de recerca i millores relaciones amb el ràpid avanç d'aquestes tecnologies.

2. Base teòrica i fonaments

2.1. Sistema d'Informació Sanitari i la Gestió de les dades.

En l'actualitat, els sistemes d'informació sanitari juguen un paper fonamental en la presa de decisions clíniques i en la gestió administrativa. La complexitat i el volum d'informació generada en aquests entorns requereixen de processos robustos per l'emmagatzematge, gestió i anàlisis, els quals són assolits mitjançant magatzems de dades (data warehouses, DWH) que es nodreixen mitjançant, sistemes d'extracció, transformació i carrega (ETL, de l'anglès Extract Transform, Load).

Un DWH(7) és un sistema d'emmagatzematge de dades centralitzat que permet consolidar dades que provenen de diverses fonts o sistemes, transformant-ho en un format homogeni que facilitat el seu anàlisis. Aquesta integració es fonamental ja que en l'àmbit sanitari disposa de moltes dades que provenen de sistemes heterogenis amb estructures molt diverses.

Els processos de ETL són claus en aquesta transformació, permeten netejar les dades, normalitzar-ne els format i validar-ne la qualitat per garantir la coherència del conjunt. Això és especialment rellevant en sistemes sanitaris, on una codificació errònia pot impactar directament en la presa de decisions mèdiques i la planificació de recursos.

Una altra consideració important en la gestió de dades sanitàries és la seguretat i la privacitat. Les dades clíniques contenen informació altament sensible i han de complir amb la normativa de Reglament General de Protecció de Dades (RGPD)(6). Això implica l'aplicació de mètodes d'anonimització i xifratge, i un control estricte dels accessos i la traçabilitat.

En resum, la gestió de les dades en el sector sanitari requereix d'un enfocament integral que combini tecnologies d'emmagatzematge amb processos rigorosos per a l'extracció de les dades sempre mantenint totes les mesures de seguretat per complir amb la normativa de seguretat i privacitat d'aquestes.

2.2. Processament del Llenguatge Natural

El processament de llenguatge natural (PLN) es una branca de la informàtica que s'encarrega de tractar computacionalment les llengües, combina tècniques de intel·ligència artificial, lingüística i estadística per permetre que les maquines comprenguin, analitzin i generin text en llenguatge humà.

Per entendre com funciona un sistema de PLN(Processament de Llenguatge Natural) podem definir 3 fases.

1- Pre-processament del text
   a. Tokenització: Consisteix en dividir el text original en unitats més petites, com ara paraules o frases, facilitant-ne la manipulació posterior.
   b. Normalització: Aquesta esta inclou processos com la conversió a minúscules, eliminació de signes de puntuació i altres transformacions que homogenitzen el text.
   c. Eliminar caràcters sense carrega semàntica, es realitza la supressió de paraules habituals com "el", "de", que no aporten informació significativa per la anàlisis.
   d. Lematització/stemming; Es redueixen les paraules a la seva forma base o arrel, facilitant l'agrupació de termes semànticament similars.
2- Representació vectorial: En aquesta fase, el text Preprocessat es transforma en una representació numèrica (vectors), imprescindible perquè pugui ser interpretat per models d'aprenentatge automàtic. Aquesta conversió habitualment es realitza mitjançant tècniques que tenen en compte l'entorn, basant-se amb el seu context.
3- Modelatge del llenguatge. Amb els vectors d'entrada ja disponibles, un model de ML o DL s'encarrega d'entendre el context i generar una resposta, classificació, predicció.

Un exemple d'aquest models és el BERT (*Bidirectional Encoder Representations from Transformers*)(2) que ha suposat un gran canvi dins els món del PLN. Aquest utilitza una arquitectura basada en "transformers" amb mecanismes d'autoatenció, capaços d'analitzar el context complet d'una paraula dins d'una frase, millorant notablement respecte models anteriors, *chatGPT* ha fet servir models basats en "transformers".

Malgrat els avantatges de BERT, aquest model presenta dues limitacions per al nostre projecte.

- Té una capacitat limitada per processar seqüencies llargues, amb un màxim de 512 claus, insuficients per a textos clínics extensos.

- Està entrenat amb textos generals, es ha dir wikipedia i llibres, fet que limita la seva efectivitat amb textos altament especialitzats com els clínics.

Per superar aquesta limitació, s'ha seleccionat el model Clinical Longformer(3,4), específicament dissenyat i pre-entrenat amb textos clínics reals. Aquest model ofereix:

- La capacitat d'analitzar seqüencies més llargues, de fins a 4096 claus, sens especialment adequat per a documents clínics extensos.
- Un entrenament específica amb terminologia mèdica, millorant considerablement la seva eficàcia en els nostre context.

2.3. Deep Learning

És una branca del ML, que utilitza xarxes neuronals amb múltiples capes (arquitectures profundes) per aprendre patrons complexos en grans conjunts de dades. Aquesta xarxes neuronals profundes estan formades per múltiples capes d'unitat de processament (neurones) que poden detectar estrictes complexes i no lineals en les dades, fent-les particularment eficaces per a tasques d'alt nivell com el reconeixement de llenguatge natural, la classificació d'imatges o la predicció de sèries temporals.

En el context del projecte actual, s'aplicarà el Deelp Learning mitjançant l'ús específic de PyTorch, una biblioteca de codi obert.

La implementació del Deep Learning dins del nostre projecte es basarà en la capacitat del model Clinical Longformer per generar representacions vectorial de textos clínics. Aquest vectors numèrics seran la base d'entrada per al nostre model de xarxa neuronal profunda, implementat amb PyTorch, que s'encarregarà específicament de classifica automàticament els codis diagnòstics associats als informes clínics. Aquest enfocament busca obtenir una alta precisió en la codificació dels diagnòstics, contribuint així a la millorar de l'eficiència i la qualitat en processos clínics automatitzats.

3. Materials i mètodes

Aquest capítol descriu de manera detallada el conjunt d'eines, tecnologies i estratègies metodològiques utilitzades per al desenvolupament del projecte. El treball s'ha estructurat seguint una arquitectura modular, que va des de la gestió de les dades i la seva extracció de la historia clínica, fins a la seva transformació i anàlisi mitjançant un model pre entrenat. A més s'ha desenvolupat una API amb la finalitat de garantir la integració amb sistemes clínics en existents.

En els següents apartats s'exposen les tecnologies utilitzades, les fonts de dades, els processos per pre-processament, l'arquitectura del model, les estratègies d'entrenament i validació, i la seva integració operativa en entorns reals.

3.1. Tecnologies utilitzades

Per a la fase inicial d'extracció i preparació de dades, s'ha fet ús de l'eina Spoon del paquet de Pentaho Data Integratia, una eina d'ETL visual que ha permès construir fluxos de dades de manera modular. Mitjançant Spoon, s'ha automatitzat l'obtenció de les dades dels diversos sistemes, les dades clíniques s'han agafat d'un servidor SQL, mentre que les dades de la codificació s'han extret de la codificació del Conjunt Mínim Basic de Dades (CMBD) que es troben en un altre sistema, aquesta segmentació ha requerit transformacions específiques per garantir la compatibilitat i la integritat de la informació abans d'incorporar-la al Date Warehouse.

A continuació un cop s'han obtingut les dades, el desenvolupament de l'eina s'ha centrat en les tecnologies següents.

- Spoon Pentaho, data integration: S'ha utilitzat per integrar diverses fonts de dades, la historia clínica amb les dades codificades del CMBD, així com fer el traspàs de la informació entre el sistema productiu i l'entorn on s'ha desenvolupat l'aplicació.
- PostgreSQL: S'utilitza com a base de dades relacional principal per a l'emmagatzematge i consulta de dades històriques. Es seu ús esta justificat per la seva estabilitat, suport per a consultes complexes i integració amb altres eines analítiques. El DWH conté taules optimitzada amb informació clínica estructurada i no estructurada, ja pre processades per a l'estudi, incloent una columna per diferenciar els diversos conjunts.

- Python: És el llenguatge principal emprat per a la construcció del sistema. Permet la integració fluida de biblioteques especialitzades en tractament de dades, processament de text i aprenentatge profund. Les biblioteques claus són:
  - Pandas i NumPy per a la manipulació de dades
  - BeautifulSoup per la manipulació de textos
  - Scikit-learn per a transformacions i mètriques d'avaluació
  - Transformers de Huggins Face per accedir al model Clinical Longformer.
  - PyTorch(9) com a fons per a la implantació del model de deep learning, optimitzat per entrenament.
- FastAPI(API RESTful)(8): El sistema utilitza un interfície REST dissenyada amb Fast API, que permet consultes en temps real per part del sistema d'història clínica. Aquesta API està preparada per acceptar dades en format JSON, processar-les mitjançant el model i retornar les prediccions de codis CIM-10.
- GIT: El control de versions es duu a terme mitjançant Git, assegurant traçabilitat i replicabilitat del codi font i facilitant el treball incremental amb diverses etapes de millorar del model.

3.2. Font de dades

Les dades utilitzades en el projectes constitueixen un actiu fonamental per al desenvolupament i entrenament del sistema de codificació automàtica. Aquestes dades poden provenir de dues fonts principals. El Data Warehouse corporatiu i els sistemes operatius connectats mitjançant l'API per la interacció a temps real.

3.2.1. Data Warehouse (DWH).

El DWH(7), construït en PostgreSQL, integra la informació provinent de diversos sistemes assistencials de la Xarxa Sanitaria Social i Docent de Santa Tecla. Conté tant dades clíniques estructurades com textos lliures en forma de camps no estructures, extrets directament de l'historia clínica de l'organització (Higia HC).

Les dades han estat organitzades en una taula específica optimitzada per a l'entrenament, validació i prova del model. A continuació es detallen els camps d'aquesta:

- Dades estructurades (variables categòriques): Aquestes variables s'utilitzen per enriquir els *embeddings* (representacions vectorials) del model i millorar el context de la predicció.
    - *Edat*: edat del pacient (numèrica entre 0 i 120)
    - *Genere*: Gènere del pacient (0 – Homes, 1 – Dones)
    - *C_alta*: Circumstància d'alta(1 – Domiciliaria , 2 – Trasllat a un hospital d'aguts, 3 – Trasllat a un sociosanitari , 4 – Trasllat a una residencia , 5 – Alta voluntària, 6 – Defunció , 7 – Evasió , 8 – Hospitalització a domicili.
    - *Periode*: any d'activitat, en format yyyy
    - *Servei*: codis identificadors del servei hospitalària (exemple 10101: Medicina interna)
- Dades no estructurades: Aquestes variables són tots camps lliures on els diversos professionals sanitaris han introduït dades durant l'ingrés del pacient.
    - *motiuingres:* motiu de la consulta o ingrés del pacient.
    - *malaltiaactual:* descripció de la malaltia del pacient.
    - *exploracio:* resultats de l'exploració física i clínica del pacient.
    - *provescomplemetnariesing:* proves complementaries realitzades en l'ingrés del pacient.

  - *provescomplementaries:* proves realitzades durant l'estada del pacient.
  - *evolució:* evolució clínica durant l'ingrés .
  - *antecedents*: antecedents mèdics i quirúrgics rellevants.
  - *cursclinic*: registre detalla i seqüencial de l'estada. Inclou l'evolució i seguiment dels pacient durant tot l'ingrés, com per exemple la presa de constants, evolució de les analítiques i l'evolució del pacient al llarg del temps.
- Variable objectiu:
  - *dx_revisat:* codis CIM-10 assignats pel servei de codificació. Aquesta és la variable objectiu del model.
- Altres camps de control (no utilitzats per al model però si com a variables funcionals.)
  - *cas*: identificador únic del cas
  - *us_estatentrenament*: estat de processament del registre (1 realitzat, 0 pendent)
  - *dx_prediccio*: camps de predicció generats pel model
  - *us_dataentrenament*: Variable temporal per saber en quin moment s'ha utilitzat el registre
  - *us_registre*: Variable aleatoritzada per dividir el conjunt de dades entre entrenament , validació o predicció.

Aquest conjunt ha estat prèviament aleatoritzat i dividit mitjançant ETL, un 80% del model s'ha utilitzat per entrenament, un 10% per validació i un 10% per predicció, per garantir la separació adequada, la aleatorització s'ha fet per any i servei. Per garantir una mostra uniforme de les dades.

3.2.2. Entrades en temps real via API.

Per a la fase operativa del projecte, el sistema haurà d'estar preparat per acceptar consultes externes a través una API RESTful. Aquesta interfície permet rebre dades en format JSON, estructurades de manera idèntica a les utilitzades en fase d'entrenament , per garantir la compatibilitat i coherència dels resultats.

D'aquesta manera l'aplicació es podria cridar des de diversos mòduls, serviria per realitzar la codificació inicial del professional de manera automàtica fent que els codificadors només hagin de comprovar la validesa, juntament amb el percentatge de validesa de cadascun dels codis. La API no disposa d'entorn gràfic, mitjançant crides POST als punts d'entrada s'executen les rutines corresponents, pel que fa a la base de dades només cridar la ruta, començarà a entrenar amb els casos que tingui pendents, mentre que la ruta amb JSON, requereix que la crida porti un fitxer amb les dades per tal de poder fer l'entrenament.

Un exemple d'un fitxer JSON seria el següent.

```
cas_clinic = {
    "cas": "CASE_001",
    "motiuingres": "Pacient de 65 anys, ingressa per hipertensió arterial descontrolada.",
    "malaltiaactual": "El pacient presenta hipertensió arterial de llarga evolució, amb valors de 180/100 mmHg.",
    "exploracio": "A l'exploració física, es detecta hipertensió arterial i es recomana tractament.",
    "provescomplementariesing": "Es realitzen proves complementàries per avaluar l'estat de l'hipertensió arterial.",
    "provescomplementaries": "Electrocardiograma normal. Analítica amb creatinina 1.2 mg/dL.",
    "evolucio": "L'evolució del pacient és favorable, amb millora de l'hipertensió arterial.",
    "antecedents": "El pacient té antecedents d'hipertensió arterial.",
    "cursclinic": "El curs clínic és estable, amb control de l'hipertensió arterial.",
    "dx_revisat": "I10|I11.9|I12.9",  # Codis CIE-10 per hipertensió
    "edat": 65,
    "genere": 1,
    "c_alta": 1,
    "periode": 2024,
    "servei": 10101
}
```

Il·lustració 1: Exemple JSON

Es important tenir en compte les dues entrades acaben executant els mateixos processos per tant han de tenir la mateixa estructura. Si una variable es buida el sistema ignora aquella variable i no la te en compte en la generació dels embeddings.

Aquesta dualitat de fonts (històrica per entrenament i temps real) permet validar el model en condicions de producció i facilitat la seva integració dins de qualsevols historia clínica o sistema de codificació extern.

3.3. Pre-processament i transformació dels textos clínics.

És una fase crítica del projecte, ja que té com objectiu preparar la informació no estructurada per tal que pugui ser interpretada per els models. Per tal de netejar aquestes aplicarem els següents processos.

- Normalització del text a unicode: Convertim el text a codificació utf-8
- Eliminem marques HTML: Utilitzarem una llibreria de *Python* anomenada *BeautifulSoup* per eliminar possibles etiquetes HTML incrustades en els textos clínics.
- Conversió a minúscules: Unifiquem el text per evitar distorsions durant la tokenització.
- Eliminació de puntuació i caràcters especials no informatius, mitjançant expressions regulars, es filtren símbols i signes que no aporten valor semàntic.
- Eliminació d'espais redundants: Es redueixen múltiples espais consecutius i s'eliminen els espais en blanc per l'esquerra i la dreta.
- Eliminació de paraules que no aporten contingut semàntic. Utilitzo una llibreria que aporta totes aquestes paraules del castellà i una llista manual de les de català.

Aquest processament està integrat dins la API de manera que independentment de l'origen les dades es tractaran utilitzant la mateixa lògica. Un exemple del que faria el procés anterior seria el il·lustrat en la següent imatge.

```
Text original:

<p><strong>Motiu d'ingrés:</strong> Pacient de 65 anys, ingressa per <em>hiperte
nsió arterial</em> descontrolada.</p>
<p><strong>Malaltia actual:</strong> El pacient presenta <em>hipertensió arteria
l</em> de llarga evolució, amb valors de 180/100 mmHg.</p>
<p><strong>Exploració:</strong> A l'exploració física, es detecta <em>hipertensi
ó arterial</em> i es recomana tractament.</p>
<p><strong>Proves complementàries:</strong> Es realitzen proves complementàries
per avaluar l'estat de l'<em>hipertensió arterial</em>.</p>
<p><strong>Evolució:</strong> L'evolució del pacient és favorable, amb millora d
e l'<em>hipertensió arterial</em>.</p>
<p><strong>Antecedents:</strong> El pacient té antecedents d'<em>hipertensió art
erial</em>.</p>
<p><strong>Curs clínic:</strong> El curs clínic és estable, amb control de l'<em
>hipertensió arterial</em>.</p>


Text processat:
motiu ingrés pacient 65 ingressa hipertensió arterial descontrolada malaltia act
ual pacient presenta hipertensió arterial llarga evolució valors 180 100 mmhg ex
ploració exploració física detecta hipertensió arterial recomana tractament prov
es complementàries realitzen proves complementàries avaluar estat hipertensió ar
terial evolució evolució pacient és favorable millora hipertensió arterial antec
edents pacient té antecedents hipertensió arterial curs clínic curs clínic és es
table control hipertensió arterial
```

Il·lustració 2: Exemple de tractament de dades

3.4. Arquitectura del model.

El model utilitzat ha estat dissenyat específicament per abordar la classificació múltiple de codis CIM-10 a partir de informes cínics en llenguatge natural. La seva arquitectura es basa en un model pre-entrenat "Clinical Longformer", personalitzat mitjançant un model de d'entrenament profund, que és invocat des d'un mòdul principal, que es el que gestiona la carrega, validació i predicció.

Les característiques tècniques clau són les següents:

1. Model base – Long Former(3): Com ja s'ha dit s'ha escollit per la seva capacitat de processar seqüencies de fins a 4096 claus, no confonguem les claus amb els codis finals, les claus són les unitats bàsiques en què es divideix el text d'entrada per tal de ser processat per el model.
2. Capçalera dual de classificació: Es tant important codificar els codis com l'ordre d'aquests.
   - Classificació de codis: genera una predicció multi etiqueta, ja que cada cas pot tenir fins a 15 codis diagnòstics.
   - Ordre dels codis: Classifica els codis segons la seva rellevància clínica. Aquesta funcionalitat s'utilitza per prioritza les prediccions, donant més pes al codis més importants.
3. Variables categòriques estructurades: A banda de la informació no estructurada, cada cas inclou informació estructurada que pot ser rellevant per la codificació d'aquestes altes. Un exemple de la importància d'aquests es que hi ha diversos codis que estan limitats per edat, es ha dir una persona de 90 anys no pot tenir diabetis gestacional.
4. Funció de pèrdua i optimització:
   - S'utilitza *BCEWithLogitsLoss(11),* forma part de la llibreria de Pytorch, es una funció idònia per a la classificació multi-etiqueta amb sortides independents per cada codi, integra la *sigmoide* (per convertir les sortides contínues del model en probabilitats entre 0 i 1) i la *cross-entropy* (per calcular la pèrdua entre la predicció i les etiquetes reals) en una sola operació, millorant l'estabilitat numèrica i el rendiment en la classificació multi etiqueta.
   - Es calculen pesos per classe segons la seva freqüència en el conjunt d'entrenament, per compensar la presència desigual de codis, aquesta

ponderació es molt important per tal de millorar la sensibilitat del model envers diagnòstics menys freqüents.

- Només utilitzarà els models prescrits en algun moment, el catàleg ens permet fins a 98000 codis diferents, no obstant en un entorn real rarament s'utilitzen tots.
- Es fa servir el classificador AdamW(10), una versió millorada de l'optimitzador Adam que separa explícitament la regularització per caiguda de pes. Aquest paràmetre penalitza els pesos grans durant l'entrenament, afavorint solucions més simples i generalitzades, es una mesura per prevenir el sobre entrenament.
- Com a planificador de decreixement de taxa d'aprenentatge utilitzarem *StepLr,* cada època aquest disminueix la taxa per fer un model més estable.
- També com a mesura d'optimització s'ha implementat una eina de parada, si no es detectat una millora significativa en les mètriques de validació en mes de 5 èpoques el sistema pot interrompre per evitar el sobre entrenament i reduir els temps de càlcul.
- S'ha implementat la compatibilitat amb la tecnologia *cuda* de nvidia.

5. Entrenament incremental cas a cas: Aquesta estratègia permet re entrenar el model amb cada nou cas validat, fet que afavoreix l'aprenentatge continu. Cada cas es procés individualment durant 5 èpoques per millorar la seva incorporació al model sense comprometre la seva estabilitat global.
6. Persistència i control de versió: El sistema desa l'estat del model, l'optimitzador, el planificador d'aprenentatge i la llista de codis predit mitjançant arxius. Aquest gestió garanteix la capacitat de reprendre l'entrenament, fer auditoria del rendiment i reprendre l'entrenament en cas de fer modificacions.

3.5. Entrenament, validació i mètriques.

L'estratègia adoptada es basa en una aproximació incremental i adaptativa El model no s'entrenarà només amb un conjunt de dades massiu i estàtic, sinó que incorporarà nous casos clínics validats de manera contínua. Això permet mantenir una actualització dinàmica del coneixement del sistema i adaptar-se a l'evolució dels patrons clínics i dels criteris de codificació.

Entrenament incremental:

- Cada nou cas clínic validat és utilitzat per entrenar el model durant un cicle curt de màxim 5 èpoques, permetent una ràpida assimilació d'informació sense comprometre el rendiment global.
- Es fa servir una rutina de parada cada 3 èpoques per detectar si el model millorar significativament durant l'entrenament. En cas contrari, s'atura anticipadament per evitar sobre ajustaments i accelerar el procés.
- El sistema actualitzarà progressivament el conjunt de codis coneguts, ampliant les seves capacitats de predicció a mesura que apareixen noves etiquetes.

Validació continua:

- El sistema permet validar la capacitat predictiva generant prediccions per als codis més probables, tot i que es defineix un llindar de confiança superior al 90% amb la finalitat de mostrar només prediccions robustes.
- A banda de comprovar que els codis siguin correctes també es valida l'ordre amb el que apareixen.

Mètriques utilitzades:

Per avaluar el comportament del model és registren les següents mètriques.

- Precisió(*Precision*): per mesurar la proporció de codis predits correctament. Es valorarà que la predicció tingui un alt percentatge de confiança + 90%, indica la fiabilitat del model.
- Sensibilitat(*Recall*): Per detectar fins a quin punt el model identifica tots els codis reals.
- F1-Score: com a compromís entre precisió i sensibilitat, especialment útil en entorns amb codis molt desequilibrats.
- Exactitud (*Accuracy*): Útil per mesurar el grau d'encert global del model, no obstant no te en compte l'ordre

- Order Accuracy i Kendall-Tau: Avaluant la qualitat de l'ordre dels codis predits en relació amb l'ordre clínic establert pel professional, es una mètrica que te en compte quantes parelles de codis estan correctament ordenades i retorna un valor entre -1 i 1, on -1 estan correctament ordenats però en ordre invers , mentre que 1 es correctament ordenat amb l'orientació correcta.
- Pèruda (Loss): Es mostren separadament la pèrdua de classificació i la d'ordre, donant visibilitat de la contribució de cada component.

Aquestes mètriques s'usen tant per monitorar l'evolució del model com per predir quan revisar, reajusta o reforçar el seu entrenament amb noves dades. Aquestes dades queden registrades per cada cas en un taula de la base de dades que es la que s'utilitzarà per analitzar els resultats. A continuació, a la il·lustració 3, es poden observar les mesures generades en un cas. Es tracta d'un entrenament fictici durant el desenvolupament de l'eina.

```
INFO:app.ml.engine:Conjunt de codis entrenats carregat correctament. Total: 5
INFO:app.ml.engine:Codis disponibles per validació: ['E119', 'I680', 'I69', 'O89', 'O90']
Input ids are automatically padded from 11 to 512 to be a multiple of `config.attention_window`: 512
INFO:app.ml.engine:=== VALIDACIÓ CAS: test5 ===
INFO:app.ml.engine:→ Codis reals (2):
INFO:app.ml.engine:   1. I680
INFO:app.ml.engine:   2. E119
INFO:app.ml.engine:→ Codis predits (>90% confiança):
INFO:app.ml.engine:   1. I69 (100.0%)
INFO:app.ml.engine:→ Top 5 codis més probables (entrenats):
INFO:app.ml.engine:   1. I69 (100.0%)
INFO:app.ml.engine:   2. O90 (0.0%)
INFO:app.ml.engine:   3. E119 (0.0%)
INFO:app.ml.engine:   4. I680 (0.0%)
INFO:app.ml.engine:   5. O89 (0.0%)
INFO:app.ml.engine:→ Ordre real:        I680 → E119
INFO:app.ml.engine:→ Ordre predit:      I680 → E119
INFO:app.ml.engine:→ Kendall-Tau:       1.00 → L'ordre predit és molt similar al real
INFO:app.ml.engine:→ Mètriques de Classificació:
INFO:app.ml.engine:   • Accuracy:  0.00
INFO:app.ml.engine:   • Precision: 0.00
INFO:app.ml.engine:   • Recall:    0.00
INFO:app.ml.engine:   • F1 Score:  0.00
INFO:app.ml.engine:→ Mètriques d'Ordre:
INFO:app.ml.engine:   • Order Accuracy: 1.00
INFO:app.ml.engine:   • Kendall-Tau:    1.00
INFO:app.ml.engine:→ Pèrdues:
INFO:app.ml.engine:   • Code Loss:  0.0003
INFO:app.ml.engine:   • Order Loss: 8.9108
```

Il·lustració 3. Exemple de validació en un conjunt de proves

3.6. Consideracions de seguretat i anonimització.

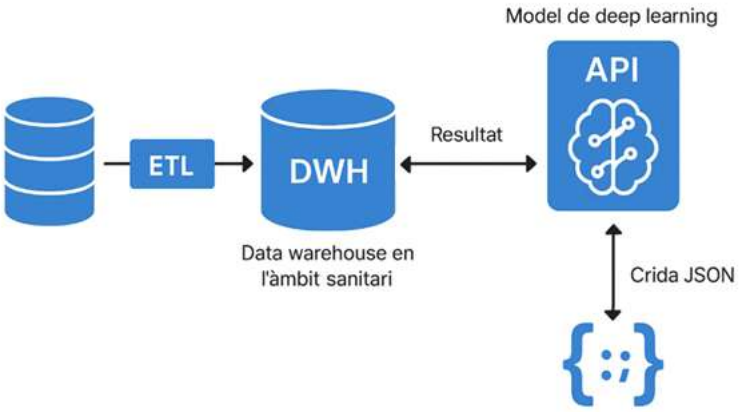
Donat que aquest projecte es desenvolupa en l'àmbit clínic i fa ús de dades sensibles, s'ha posat especial atenció a garantir el compliment de la RGPD(6). Totes les dades utilitzades en el procés d'entrenament i validació han estat prèviament anonimitzades per evitar qualsevol risc de re identificació dels pacients. Només s'ha mantingut el codi del cas per identificar el cas que tractarem i inclús així, no s'utilitza per fer l'entrenament i no queda desat en cap lloc del programa.

L'aplicació es desplegarà com una API, per el que es poden integrar mesures de protecció i autenticació mitjançant claus i/o ip.

Aquesta capa d'integració assegura que el model pot ser utilitzat en entorns clínics reals, mantenint el compromís amb la seguretat.

4. Desenvolupament del model

El desenvolupament del projecte s'ha dut a terme mitjançant un enfocament modular i estructurat, amb l'objectiu d'abordar la complexitat pròpia de l'entorn clínic real i assegurar una elevada capacitat d'adaptació.

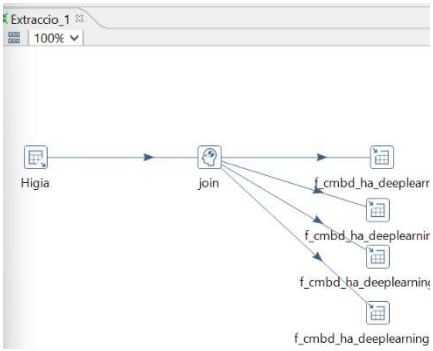


Il·lustració 4: Estructura modular del projecte

Les dades tot i que poden venir mitjançant JSON, per al desenvolupament del projecte s'ha fet servir la versió mitjançant base de dades. La estructura mostra només quin seria el flux utilitzant durant el desenvolupament.

4.1. Gestió de dades: ETL i Data Warehouse.

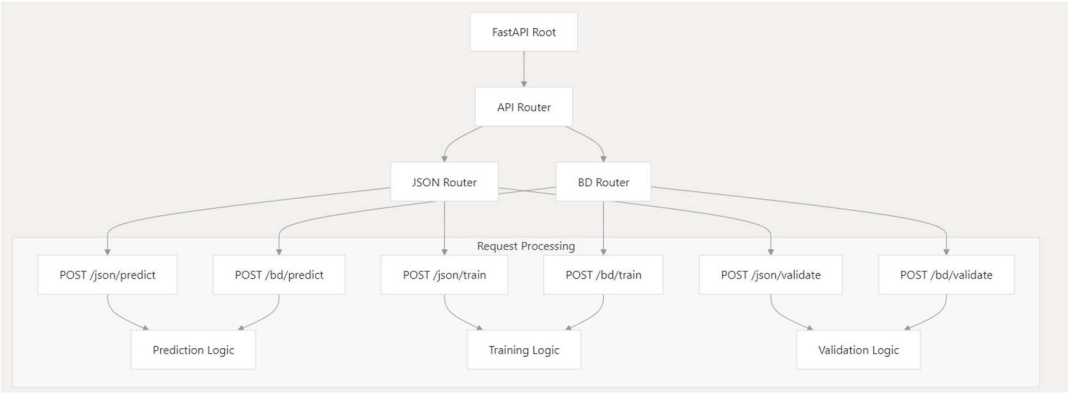
Es va optar per utilitzar Spoon de Pentaho Data Integration per crear els fluxos de ETL. Aquesta eina permet extreure les dades de sistema de producció (Higia) on tenim les dades de treball i les inserim dins del DWH



Il·lustració 5: ETL d'extracció de dades del HIS al DWH

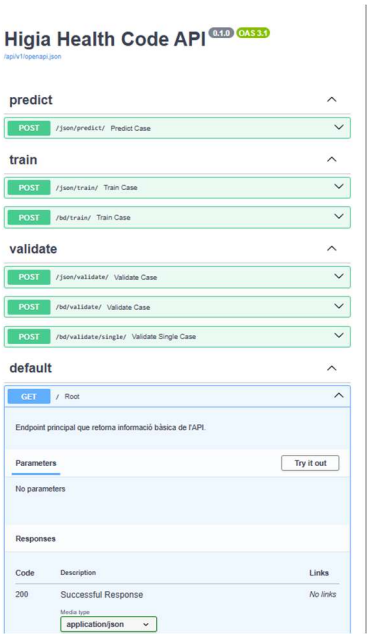
4.2. Disseny de la API i entorn d'execució

La API es la plataforma sobre la que s'ha implementat la solució. S'ha utilitzat *FastAPI* per la facilitat a l'hora d'implementar una API així com el mòdul de documentació que facilita la tasca d'implementació.



Il·lustració 6:Estructura de la API desenvolupada.

També adjunto una imatge de la documentació que s'ha mencionat anteriorment. Que es on es poden observar tots els punts d'accés de l'aplicació.



Il·lustració 7: Menú de documentació de l'API

4.3. Pre-processament dels textos clínics

S'ha desenvolupat dins la mateixa API un mòdul per fer el tractament de texts, ja que l'intenció es que sigui una eina que completa, que siguí capaç de rebre textos en format html que és el format en que es desa la informació a la base de dades. Per fer aquest procés s'han utilitzat diverses llibreries de Python, la primera s'anomena *BeautifulSoup(13)*, en un principi es una llibreria especialitzada per a realitzar *scrapping* (Extracció de dades de pagines web), però s'ha utilitzat la part de la llibreria referent a la neteja de textos.

```
    soup = BeautifulSoup(text, 'html.parser')
    text = soup.get_text()

    # Eliminar puntuació i caràcters especials no informatius
    text = re.sub(r'[^\w\s]', ' ', text)

    # Eliminar espais múltiples
    text = re.sub(r'\s+', ' ', text)
```

Il·lustració 8: Procés de neteja dels textos.

La segona llibreria que s'ha utilitzat es *ntlk (14)* es una llibreria de Python especialitzada en el tractament de textos. En aquest cas en concret s'ha fet servir amb la finalitat d'eliminar paraules de contingut buit. Degut a que molts del professionals que utilitzen el programa són de parla catalana s'ha hagut de fer un petit mòdul addicional amb aquelles paraules que s'han considerat que tenen un contingut buit.

```python
def remove_stopwords(self, text: str) -> str:
    """
    Elimina les paraules sense càrrega semàntica.

    Args:
        text: Text del qual eliminar les stopwords

    Returns:
        Text sense stopwords
    """
    words = text.split()
    return ' '.join([word for word in words if word not in self.stopwords])
```

Il·lustració 9: Eliminació dels elements sense carrega semàntica.

Per tal de finalitzar amb el tractament dels textos, s'ha passat a codificació UTF-8, s'han transformat tots els caràcters a minúscules i s'han eliminat espais en blanc no necessaris.

4.4. Selecció del model base: Clinical Longformer

Donada la longitud dels textos a estudiar es va descartar el us de BERT i es va decidir centrar en la implementació d'un model de transformers basat en Clinical Longformer, ja que aquest permet processar seqüencies molt més llargues (fins a 4096 tokens). Aquesta decisió m'ha permès poder mantenir la informació completa dels textos clínics sense truncaments innecessaris.

Es important comentar que tot i que el model esta al núvol, la eina només descarrega el model el primer cop que s'executa per tal de descarregar-lo. Les següents execucions només utilitza el model en local d'aquesta manera s'elimina la dependència d'estar connectat a la xarxa externa.

```python
config_path = os.path.join(LOCAL_LONGFORMER_PATH, "config.json")
if not os.path.exists(LOCAL_LONGFORMER_PATH) or not os.path.exists(config_path):
    logger.info(f"Model no trobat localment. Descarregant des de Hugging Face ({MODEL_ID})...")

    if os.path.exists(LOCAL_LONGFORMER_PATH):
        shutil.rmtree(LOCAL_LONGFORMER_PATH)

    # Descarregar amb el nou mètode recomanat
    tokenizer = LongformerTokenizer.from_pretrained(
        MODEL_ID,
        local_files_only=False,
        force_download=True
    )
    base_model = LongformerModel.from_pretrained(
        MODEL_ID,
        local_files_only=False,
        force_download=True
    )

    os.makedirs(LOCAL_LONGFORMER_PATH, exist_ok=True)
    tokenizer.save_pretrained(LOCAL_LONGFORMER_PATH)
    base_model.save_pretrained(LOCAL_LONGFORMER_PATH)
    logger.info("Model descarregat i desat localment")
else:
    logger.info("Model ja existeix localment, no es descarregarà")
```

Il·lustració 10: Sistema per validar l'existència del model en local.

El codi anterior realitza una validació, en cas que el model no estigui al directori corresponent descarrega el model de la xarxa, pas que només s'hauria d'executar en la primera execució de l'aplicació. El funcionament normal es que el model es carregui des del model existent.

4.5. Arquitectura del model propi

El model final desenvolupat, s'anomena CIE10Classifier, es basa en una arquitectura híbrida que combina representacions vectorials de textos llargs amb informació categòrica clínica estructurada. Les característiques del model són les següents.

- Codificador de text utilitzant el model *Longformer.*
- Classificador de codis: Xarxa neuronal de dues capes que transforma el vector de text combinat en una predicció multi etiqueta (Codis CIM-10). Es fa servir la funció d'activació (ReLU) i una capa de descartament aleatori (dropout) per evitar l'ajust excessiu (sobre aprenentatge).
- Classificador d'ordre: Estructura paral·lela que aprèn a predir la rellevància relativa de cada codi assignat, millorant la qualitat de la codificació.
- Representacions de variables categòriques: S'incorporen representacions específiques per edat, gènere, tipus d'alta, servei mèdic i període d'activitat, amb la finalitat de capturar característiques rellevants que afecten a les codificacions. Per exemple no tindria mai sentit un codi que comencí per O en un home. Ja que les O són codis de diagnòstics associats a l'obstetrícia.
- Projecció conjunta: Les representacions categòriques es combinen i es projecten a l'espai del text, habilitant que el model integri informació narrativa i estructurada de manera coherent.

Les decisions preses en el desenvolupament del model han estat:

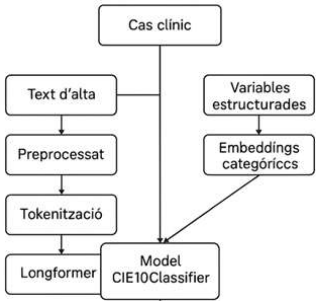
- L'elecció del model de transformers, s'ha optat per l'ús d'aquest per la seva capacitat de gestionar textos llargs.
- La classificació dels codis i l'ordre, el primer codi sol representar el diagnòstic principal, el següents afecten a la complexitat de l'alta així com al comorbiditats.
- La integració de variables categòriques, s'ha utilitzat variables estructurades com són el gènere, edat , tipus d'alta, ja que aporten informació complementaria als texts.
- Ús del *BCEWithLogitsLoss*. Permet gestionar la classificació de múltiples etiquetes
- Aprenentatge incremental: Aprenentatge progressiu cas a cas, de manera que el model es pot adaptar contínuament als nous patrons.

4.6. Estratègia d'entrenament incremental

L'estratègia d'entrenament adoptada es fonamenta en l'aprenentatge incremental cas a cas. Aquest enfocament ha estat escollit per permetre que el model s'adapti de manera dinàmica als canvis en els models de codificació. Així s'evita la necessitat de re-entrenaments massius que podrien resultar costosos en temps i recursos.

Les característiques són les següents:

- Entrenament per a cada cas, validat per el professional de codificació s'utilitzaran per actualitzar el model durant un cicle d'entrenament de x èpoques, aquestes èpoques es poden configurar dins la API, en el nostre cas serà de 5 èpoques per cada cas.
- Ús d'aturada anticipada: Si no es detecta una millora significativa en la pèrdua(los rate) de validació durant x èpoques consecutives, el procés d'entrenament s'atura per evitar el sobre aprenentatge. En el nostre cas 2 èpoques.
- Optimitzador i funció de pèrdua: Es va servir l'optimitzador *AdamW* amb una taxa d'aprenentatge adaptativa gestionada per un planificador der reducció gradual (StepLR). La funció de pèrdua principal és *BCEWithLogitsLoss,* adaptada per a problemes de classificació múltiple.
- Actualització contínua del conjunt de codis; Només es permet predir aquells codis que han estat entrenats almenys un cop. Això resol una problemàtica detectada durant el desenvolupament, el catàleg complet de codis supera els 86000 codis, la qual cosa causava prediccions absurdes o inconsistents. Limitant aquest codis es millora la precisió i fiabilitat.
- Després de cada sessió d'entrenament, es desa l'estat del model, l'optimitzador, el planificador de taxa d'aprenentatge, el catàleg de codis entrenat i el binaritzador d'etiquetes (MultiLabelBinarizer(15)). Això garanteix que el sistema pugui continuar l'entrenament en el futur sense perdre el coneixement adquirit.



Il·lustració 11: Esquema d'implantació del model d'aprenentatge de DL

Aquest plantejament d'aprenentatge incremental assegura que el model pugui evolucionar conjuntament amb la pràctica clínica real, permetent una codificació cada vegada més precisa i adaptada als nous criteris.

4.7. Desenvolupament del motor de l'aplicació.

El motor de l'aplicació s'ha implementat tot en el mòdul engine.py, centralitza totes les operacions necessàries per a la gestió del model, l'entrenament incremental i la realització de prediccions en temps real, Aquest motor integra el preprocessament de textos, les crides al model i l'actualització contínua del coneixement adquirit.

Les funcions principals del motor són :

- Inicialització del models: Carrega el model i el tokenitzadors Longformes des de fitxers locals. Si no es troben disponibles, es descarreguen automàticament i es guarden per a futures sessions.
- Gestió del dispositiu de càlcul: Detecta si es disposa de GPU i adapta el model al dispositiu disponible (cuda o cpu) per optimitzar el rendiment.
- Lectura i preparació de les dades: Gestiona la preparació dels textos i de les variables categòriques.
- Entrenament incremental: Gestiona l'actualització del model cas a cas, integrant l'estratègia d'aprenentatge incremental ja descrita, incloent aturada anticipada i ajust dinàmic de la taxa d'aprenentatge.
- Validació: Permet validar un cas, calculant mètriques de classificació (precisió, sensibilitat, F1) i mètriques d'ordre (exactitud de l'ordre i Kendall-Tau). Aquest procés assegura que el model no només assigni correctament els codis, sinó que també te en compte el seu ordre.
- Predicció en temps real: Permet generar prediccions per a nous casos clínics, limitant la sortida als codis entrenats prèviament i aplicant un llindar de confiança del 90% per filtra les prediccions amb menys fiabilitat.
- Persistència i seguretat: Després de cada entrenament o actualització, el motor desa l'estat complet del model. A més, incorpora validacions rigoroses per detectar inconsistències en les dades o en l'estat del model.

Aquest motor constitueix l'eix central de l'aplicació , assegurant que totes les etapes des de la recepció del text clínic fins a la generació de codis diagnòstics es duguin a terme de manera segura, eficient i adaptable al context.

4.8. Conclusions sobre el desenvolupament.

El desenvolupament del projecte ha requerit d'una combinació de solucions innovadores adaptades a un entorn real. Les decisions preses durant el disseny i la implementació han estat clau per garantir l'eficàcia, l'escalabilitat i la seguretat del sistema. Tot i que s'ha desenvolupat tot en base a la base de dades del DWH, també s'ha implementat la modalitat que els punts de la API puguin ser cridats mitjançant fitxers JSON, d'aquesta manera facilita la integració en diversos entorns.

5. Anàlisis de resultats

L'execució del procés d'entrenament i validació ha seguit correctament tots els passos establerts. El sistema ha funcionat com estava previst, aplicant estratègies d'optimització, parada anticipada, i persistència de l'estat del model entre sessions per evitar pèrdues en cas de fallides.

Tot i el correcte funcionament tècnic els resultats obtinguts mostren una manca de confiança predictiva flagrant, no ha estat capaç d'encertar cap codi amb el grau de confiança que s'havia establert (90%).

El codi que ha obtingut un grau de confiança més elevat en un cas ha estat del 0.53% i es tracta del codi I10 corresponent a Hipertensió essencial.

Tanmateix, això no implica necessàriament un error de modelització. Al analitzar les sortides completes, que inclouen els top-15 codis amb més probabilitat (encara que no superin el llindar), es detecten indicis de generalització. El model tendeix a assignar més probabilitats als codis que són més freqüents, cosa que indica que ha après patrons estadístics bàsics de distribució

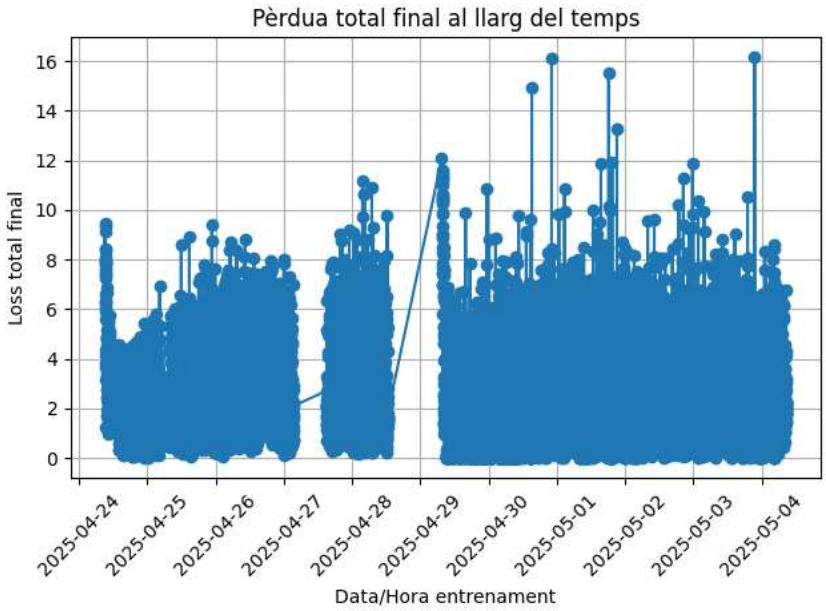
Aquest comportament es pot observar clarament en la imatge següent.

id	experimento_id	caso_id	codis_reals	codis_predits_top15	probs_predits_top15
113	eval_20250505_085317	113	[13]	[15]	[15]
			K7040	I10	9.133753E-4
			K7030	E785	2.8349363E-4
			D62	E119	2.340874E-4
			K3189	N179	1.5232891E-4
			E119	Z87891	1.3239765E-4
			I259	J449	1.0149899E-4
			I509	D509	9.1416114E-5
			Z955	O480	8.602386E-5
			K766	Z7901	7.539186E-5
			N1832	J9600	7.400712E-5
			D631	I259	6.611718E-5
			Z8673	N401	6.457299E-5
			Z8616	K5730	6.1854706E-5
				I5031	6.0248287E-5
				E7800	5.8641406E-5
143	eval_20250505_085633	143	[12]	[15]	[15]
			J9600	I10	0.0021598088
			I509	E785	6.1185966E-4
			J1008	N179	2.4172528E-4
			J13	E119	1.7431747E-4
			N179	D509	1.5746643E-4
			K560	I4891	1.4452863E-4
			E119	Z87891	1.2359807E-4
			I4891	Z8673	1.2156797E-4
			I10	N189	1.20434925E-4
			E785	I509	1.0462417E-4
			Z743	M810	1.0429585E-4
			Z880	J9600	1.0238339E-4
				K5730	8.526321E-5
				F78A9	8.4885476E-5
				I4821	8.459352E-5

Il·lustració 12: Exemple de predicció

- En el cas 112, el codi **I10** no forma part dels codis reals però obté una probabilitat de $9.13x10^{-4}$
- En canvi, en el cas 143, on **I10** sí que forma part del conjunt de codis reals, el model assigna una probabilitat superior: 0.00215.

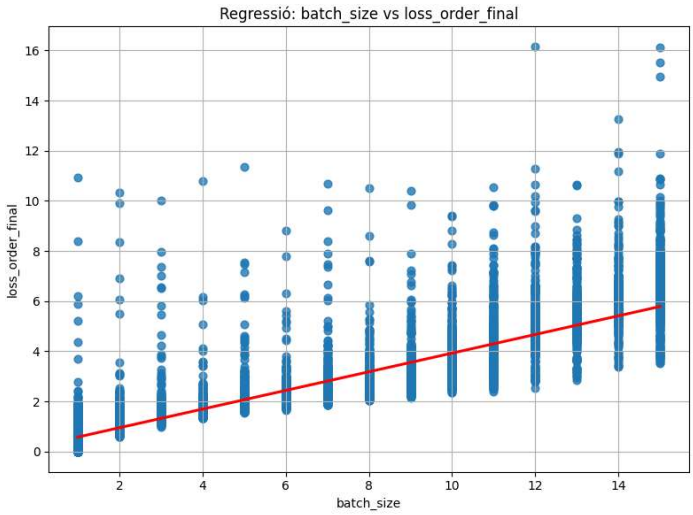
Aquest increment, tot i ser petit, indica que el model reacciona lleugerament millor davant codis realment presents, fet que suggereix un cert grau d'aprenentatge.
Ja que no podem avaluar les mètriques referents al grau de predicció l'anàlisi s'ha centrat en examinar el comportament del model durant l'entrenament, especialment pel que fa a l'evolució de la pèrdua (loss)



Il·lustració 13: Gràfic de les pèrdues durant l'entrenament

Els principals punts que podem extreure del entrenament són:

- Entrenament parcialment correcte: El model mostra una reducció clara de la pèrdua durant les primeres sessions, indicant que és capaç d'aprendre patrons bàsics a partir dels primers casos, fins al 26-04, tanmateix a partir del 27-04 la pèrdua es troba molt irregular, fet que indica que l'eficàcia de l'aprenentatge disminueix amb el temps, probablement a la repetició, saturació o sobre entrenament dels casos,
- Mecanisme de seguretat: El model es desa darrere de cada entrenament, la qual cosa assegura la recuperació de l'estat entrenat en cas de fallides. Aquest mecanisme ha funcionat correctament i ha permès preservar la continuïtat del sistema al llarg del període analitzat.



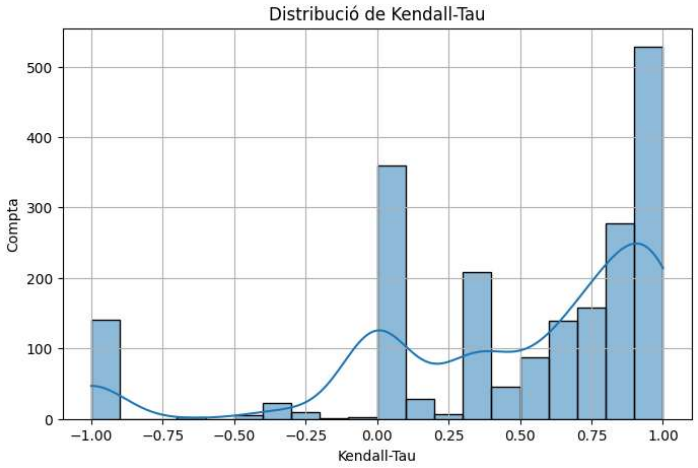
Il·lustració 14: Mida del lot contra pèrdua final

Amb el gràfic anterior podem dir que a mesura que augmenta la grandària del lot, la pèrdua final també incrementa. Això indica que el model presenta més dificultats per aprendre quan treballa amb lots grans en comparació amb lots petits. Aquest ha estat el primer indicador que l'estudi no ha anat tan bé com s'esperava inicialment, i posa de manifest la importància de revisar i ajustar els hiperparàmetres per optimitzar el rendiment del model.

En l'apartat de l'ordre, tot i que durant l'entrenament es veia que la pèrdua era més elevada com, només tenia en compte els codis correctes si que s'ha pogut comprovar que el model ha après a establir un cert ordre entre les prediccions.
La distribució de Kendall-Tau mostra un patró interessant:

- Hi ha una concentració molt alta al voltant de l'1, indicant que en molts casos el model es capaç d'ordenar els codis de manera molt similar a l'ordre real.
- L'altre bloc interesant es entre 0.5 i 0.9, es troben en rangs de coherència parcial, on l'ordre no es perfecte però conserva una lògica amb el real.
- Hi ha una concentració al voltant del 0, aquests casos no mostres cap coherència amb l'ordre real, fet que pot ser degut a casos poc representatius o que no segueixen cap patró en concret en l'odre, de manera sistemàtica poden estar en primera o en ultima posició.
- El pic en -1, serien els casos que el model ordena a la inversa de l'ordre correcte, pot ser degut a casos fora de normal o patrons erronis en la codificació.

Aquesta distribució apunta a que malgrat la manca de precisió en la selecció dels codis, el model ha interioritzat certes regles d'ordre, quan els codis són reconeguts. D'aquesta manera es pot justificar l'ús de Kendall-Tau com a mètrica per a l'ordre.



Il·lustració 15: Distribució de l'ordre de Kendall-Tau

6. Punts de millora

Un cop analitzat els resultat del projecte, i malgrat que el model no ha aconseguit aprendre de manera òptima, es poden plantejar hipòtesis clares sobre els punts que han fallat. La mitjana de pèrdua obtinguda per la capa de classificació de codis és de 0.0016 mentre que la de ordre es molt més elevada, amb un valor mitja de 2.2656

- La capa de classificació de codis, presenta una pèrdua relativament baixa, la qual cosa indica que el model és capaç d'aprendre patrons dels entrenament. A partir d'això, s'ha observat que el model tendeix a generalitzar en excés, afavorint codis que apareixen amb més freqüència. Això provoca un biaix cap als codis majoritaris i redueix la sensibilitat en codis menys presentats.
- La capa d'ordre ofereix un rendiment més alt de cara als resultats pel que fa a la gestió de l'ordre relatiu, al utilitzar només els codis correctes per tal de calcular el ordre fa que hi hagi menys opcions i acabi encertant mes que en la predicció de codis, però amb la pèrdua observada podem dir que es tracta d'un model molt inestable que impacta negativament en el sistema.

Ajustos en l'entrenament i optimitzacions en la part del motor de l'aplicació.

- Tal i com s'ha observat amb lots més petits, l'aprenentatge tenia menys pèrdues.
- Ajustar la taxa d'aprenentatge i utilitzar una tècnica d'escalfament progressiu.
    - Durant les primeres èpoques, s'utilitza una taxa d'entrenament molt petita i es va incrementat gradualment fins arribar al valor objectiu, un cop assolit el punt objectiu aquest torna a baixar fins a 0.
        - Evitar inestabilitat inicial
        - Permet que el model ajusti els pesos poc a poc
        - Millora la convergència en arquitectures grans com es el cas del *longformer*
        - Es sol utilitzar amb optimitzadors com el AdamW.
- Entrenament escalonat codis i ordre

  Entrenar inicialment només la capa de codis, congelar la capa d'ordre. Un cop assolits bons resultats en la classificació, desbloquejar la capa d'ordre i realitzar l'entrenament de la capa d'ordre aplicant una ponderació a cadascuna de les pèrdues.

Modificacions de l'arquitectura del model

- Utilització de capes diverses per a les entrades de text
  Implementar capes de classificació separades per cadascun dels camps de text clínic. Cada camp tindrà la seva pròpia capçalera d'atenció i classificació, i les sortides es combinaran en una etapa posterior amb una capa de fusió. Això permetria:
  - Capturar la informació específica i diferencial de cada camp.
  - Reduir el soroll generat per ajuntar textos diversos.
  - Millorar el descobriment de xarxes ocultes entre les diverses capes.

- Revisió de les capes categòriques, durant el desenvolupament es va reduir la projecció categòrica a 128 dimensions per limitar la complexitat i el temps d'entrenament. Tanmateix, per assegurar una interacció correcta amb el arrays de text, seria recomanable projecta també els arrays categòrics a 768 dimensions.

- Millorar la funció de pèrdua, considerar buscar una funció més especifica per l'algoritme d'ordenació per complementar la CrossEntropyLoss que faig servir actualment.

Amb aquestes actualitzacions crec que el model seria més equilibrat, i capaç de millorar la precisió de classificació de codis i ordre d'aquests. Tot i que no se si arribaria a un percentatge d'acceptació del 90%, ja que es un valor molt elevat.

7. Conclusions

El desenvolupament del projecte ha permès construir un sistema innovador per a la codificació automàtica d'altes mèdiques basat en tècniques de processament de llenguatge natural i entrenament profund. Aquest sistema s'ha desenvolupat amb el context d'una historia clínica en concret, tot i que es podria adaptar a qualsevol historia amb una petit desenvolupament. I demostra que avui en dia es poden abordar problemes complexos amb aquest tipus d'eines, encara que han sorgit problemes amb més temps crec que es possible obtenir millor resultats.

Conclusions principals del treball:

- El model ha aconseguit reduir la pèrdua durant l'entrenament, indicant capacitat d'aprenentatge sobre els patrons clínics, especialment en la classificació de codis.
- No obstant, els resultat obtinguts estan lluny del nivell necessari per tal que sigui una eina utilitzable en aquest moment.
- La metodologia incremental ha estat útil per incorporar nous casa i mantenir el sistema dinàmic, però també ha evidenciat la necessitat de reforçar la gestió de codis rars i l'ajust dels codis minoritaris.

Conclusió sobre l'assoliment d'objectius:

- S'ha assolit l'objectiu de desenvolupar el model, la creació d'una API funcional i la integració amb un sistema real.
- No s'ha assolit la meta d'obtenir una eina útil en aquest moment per a la codificació de altes clíniques.
- La planificació prevista s'ha seguit en la seva majoria, s'han tingut que fer ajustos metodològics que han fet tornar a punts anteriors, i la dificultat d'accedir a les dades des del principi ha fet que s'hagin pogut realitzar poques proves amb dades reals abans de realitzar l'entrenament.

Avaluació dels impactes prevists

- Sostenibilitat: Si l'eina funciones realment podria optimitzar l'ús de recursos humans, alliberant temps dels codificadors per a realitzar tasques més complexes o realitzar validacions mes acurades.

- Ètic-social: El compliment de la RGPD ha estat rigorós, assegurant que no es guardessin dades personals al model i mantenint la confidencialitat, a més a més que el model treballa en una xarxa aïllada sense fer us d'internet.
- Diversitat: El model s'ha adaptat per a català , castellà i angles, tot i que el *longformer* accepta moltes altres llengües.

Avaluació dels impactes imprevistos

- La manca d'accés a les eines optimes i les dades des d'un principi ha limitat la possibilitat de realitzar proves més exhaustives de cara al desenvolupament de l'eina desviant els esforços cap a garantir la viabilitat tècnica de l'entrenament. Aquest desafiament s'ha resolt centrant el projecte en construir una arquitectura flexible i escalable.

Línies de treball futurs

- Implementar totes les millores esmenades al punt 6 del projecte.
- Integrar la codificació dels procediments.
- Integrar el pes del GRD (grups relacionats per el diagnòstic)

8. Glossari

BERT - Bidirectional Encoder Representations from Transformers
DL – Deep Learning
DWH – Data Warehouse.
ML – Machine Learning.
PLN – Processament de llenguatge natural
RGPD – Reglament General de Protecció de Dades.

9. Bibliografia

1. Deep Learning Deep ethics: Ètica per a l'ús de la intel·ligència artificial en medicina | Institut Borja de Bioètica [Internet]. [citat 18 de març de 2025]. Disponible en: https://www.iborjabioetica.url.edu/ca/blog-de-bioetica-debat/deep-learning-deep-ethics-etica-lus-de-la-intelligencia-artificial-en-medicina

2.Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

3.Beltagy I, Peters ME, Cohan A. Longformer: The Long-Document Transformer [Internet]. arXiv; 2020 [citat 29 de març de 2025]. Disponible en: http://arxiv.org/abs/2004.05150

4.Tinn R, Cheng H, Gu Y, Usuyama N, Liu X, Naumann T, et al. Fine-Tuning Large Neural Language Models for Biomedical Natural Language Processing [Internet]. arXiv; 2021 [citat 29 de març de 2025]. Disponible en: http://arxiv.org/abs/2112.07869

5.Kim M, Jung Y, Jung D, Hur C. Investigating the Congruence of Crowdsourced Information With Official Government Data: The Case of Pediatric Clinics. J Med Internet Res. 3 de febrero de 2014;16(2):e29.

6. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data

Protection Regulation) (Text with EEA relevance) [citat 8 de abril de 2025] Disponible en : http://data.europa.eu/eli/reg/2016/679/oj

7. W. H. Inmon, “Building the Data Warehouse,” Wiley Publishing, Inc., Hoboken, 2005.

8. Sebastián Ramírez. FastAPI Documentation. [Online]. Disponible a: https://fastapi.tiangolo.com/

9. Documentació oficial de pytorn [Online]. Disponible a https://pytorch.org/docs/stable/index.html

10. Ilya Loshchilov, Frank Hutter “Decoupled Weight Decay Regularization” [Internet]. arXiv; 2017 [citat 27 de abril de 2025]. Disponible en: https://arxiv.org/pdf/1711.05101

11. BCEWithLogitsLossh [citat 27 de abril de 2025 [Online]. Disponible a: https://pytorch.org/docs/stable/generated/torch.nn.BCEWithLogitsLoss.html

12. M. G. KENDALL, A NEW MEASURE OF RANK CORRELATION, *Biometrika*, Volume 30, Issue 1-2, June 1938, Pages 81–93, https://doi.org/10.1093/biomet/30.1-2.81

13. BeautifulSoup [Online]. Disponible a: https://pypi.org/project/beautifulsoup4/

14. NTLK (Naturla Language Tookit [Online]. Disponible a https://www.nltk.org

15. MultiLabelBinarizer [Online] Disponible a https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MultiLabelBinarizer.html

10. Annexos

Git-Hub - https://github.com/mserretm/HigiaHealthCode