UNIVERSITAT POLITÈCNICA DE VALÈNCIA

DSIC DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dep. de Sistemes Informàtics i Computació

Generació de resums abstractius en llenguatge simplificat

Treball Fi de Màster

Màster Universitari en Intel·ligència Artificial, Reconeixement de Formes i Imatge Digital

AUTOR/A: Torres Bertomeu, Diego

Tutor/a: Hurtado Oliver, Lluis Felip

Cotutor/a: Segarra Soriano, Encarnación

Cotutor/a: Ahuir Esteve, Vicent

CURS ACADÈMIC: 2023/2024

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Departament de Sistemes Informàtics i Computació
Universitat Politècnica de València

Generació de resums abstractius en llenguatge simplificat

TREBALL FI DE MÀSTER

Màster Universitari en Intel·ligència Artificial, Reconeiximent de Formes i Imatge Digital

Autor: Diego Torres Bertomeu

Tutor: Lluis Felip Hurtado Oliver
Encarnacion Segarra Soriano
Vicent Ahuir Esteve

Curs 2023-2024

Resum

El propòsit final de la generació automàtica de resums és tornar un text considerablement més breu que l'original però mantenint les idees i els aspectes principals. Amb aquesta acció es pot col·laborar a agilitzar les tasques de tractament d'informació en àmbits tan diversos com la bibliografia mèdica, documents legals, articles periodístics, etc. Els sistemes principals de resum automàtic de l'estat de l'art són abstractius, és a dir, construeixen els resums reescrivint la informació més rellevant dels documents, i estan basats en xarxes neuronals profundes (transformers i longformers principalment). Les publicacions biomèdiques contenen les darreres investigacions sobre temes destacats relacionats amb la salut, que van des de malalties comunes fins a pandèmies globals. Sovint això pot fer que el contingut siga d'interès per a una àmplia varietat d'audiències, inclosos investigadors, professionals mèdics, periodistes i fins i tot el públic en general. No obstant això, el llenguatge altament tècnic i especialitzat que s'utilitza en aquests articles normalment dificulta que el públic no expert en comprenga el contingut. La tasca que es pretén abordar gira al voltant del resum abstractiu d'articles biomèdics, amb èmfasi a atendre audiències no expertes mitjançant la generació de resums que siguin més llegibles, que continguen més informació general i menys terminologia tècnica, cosa que es coneix com a llenguatge simplificat. En concret, tenint en compte el resum tècnic i el text principal d'un article com a entrada, l'objectiu d'aquest treball consisteix a construir un model que genere el resum en llenguatge simplificat. Es disposa de dos conjunts de dades, PLOS i eLife, del domini biomèdic, amb els triplets (article, resum tècnic, resum en llenguatge simplificat). Es proposa l'ús de models longformer a causa de la longitud de l'entrada, així com l'ús de diferents estratègies per millorar els resultats. Entre d'altres, es pot treballar amb la incorporació de coneixement extern derivat de grafs de coneixement, generació condicionada de text o tècniques d'augment de dades (Data Augmentation).

Paraules clau: resum abstractiu; transformers; longformers; llenguatge simplificat

Resumen

El propósito final de la generación automática de resúmenes es devolver un texto considerablemente más breve que el original pero manteniendo las ideas y aspectos principales. Con esta acción se puede colaborar en agilizar las tareas de tratamiento de información en ámbitos tan diversos como la bibliografía médica, documentos legales, artículos periodísticos, etc. Los principales sistemas de resumen automático del estado del arte son abstractivos, es decir, construyen los resúmenes reescribiendo la información más relevante de los documentos, y están basados en redes neuronales profundas (transformers y longformers principalmente). Las publicaciones biomédicas contienen las últimas investigaciones sobre temas destacados relacionados con la salud, que van desde enfermedades comunes a pandemias globales. A menudo, esto puede hacer que el contenido sea de interés para una amplia variedad de audiencias, incluidos investigadores, profesionales médicos, periodistas e incluso el público en general. Sin embargo, el lenguaje altamente técnico y especializado que se utiliza en estos artículos normalmente dificulta que el público no experto comprenda su contenido. La tarea que se pretende abordar gira en torno al resumen abstractivo de artículos biomédicos, con énfasis en atender audiencias no expertas mediante la generación de resúmenes que sean más legibles, que contengan más información general y menos terminología técnica, lo que se conoce como lenguaje simplificado. En concreto, teniendo en cuenta el resumen técnico y el texto principal de un artículo como entrada, el objetivo de este trabajo consiste en construir un modelo que genere el resumen en lenguaje simplificado. Se dispone de dos conjuntos de datos, PLOS y eLife, del dominio biomédico, con los tripletes (artículo, resumen técnico,

resumen en lenguaje simplificado). Se propone el uso de modelos longformer debido a la longitud de la entrada, así como al uso de diferentes estrategias para mejorar los resultados. Entre otros, puede trabajarse con la incorporación de conocimiento externo derivado de grafos de conocimiento, generación condicionada de texto o técnicas de aumento de datos (Data Augmentation).

Palabras clave: resumen abstractivo; transformers; longformers; lenguaje simplificado

Abstract

The ultimate purpose of automatic summarization is to return a text considerably shorter than the original but retaining the main ideas and points. With this action, you can help speed up information processing tasks in areas as diverse as medical bibliography, legal documents, newspaper articles, etc. The main state-of-the-art automatic summarization systems are abstractive, that is, they build the summaries by rewriting the most relevant information from the documents, and are based on deep neural networks (mainly transformers and longformers). Biomedical publications contain the latest research on prominent health-related topics, ranging from common diseases to global pandemics. This can often make the content of interest to a wide variety of audiences, including researchers, medical professionals, journalists and even the general public. However, the highly technical and specialized language used in these articles usually makes it difficult for non-expert audiences to understand their content. The task sought to be addressed revolves around abstractive summarization of biomedical articles, with an emphasis on serving non-expert audiences by generating abstracts that are more readable, contain more general information and less technical terminology, known as simplified language. Specifically, taking into account the technical abstract and the main text of an article as input, the objective of this work is to build a model that generates the summary in simplified language. There are two datasets, PLOS and eLife, from the biomedical domain, with the triplets (article, technical abstract, summary in simplified language). The use of longformer models is proposed due to the length of the input, as well as the use of different strategies to improve the results. Among others, it is possible to work with the incorporation of external knowledge derived from knowledge graphs, conditional generation of text or Data Augmentation techniques.

Key words: abstractive summarization; transformers; longformers; simplified language

Índex

Índex de figures

Índex de taules

CAPÍTOL 1
Introducció

Amb l'expansió d'Internet al llarg dels anys, la quantitat de documents disponibles en pràcticament qualsevol àmbit ha crescut enormement. Davant d'aquesta enorme quantitat de documents, no és suficient confiar en els motors de cerca per filtrar només aquells que ens poden ser útils, ja que encara ens proporcionen tants resultats que és impossible analitzar quins són realment valuosos en un temps raonable.

En aquest context, els models de resum automàtic esdevenen essencials, ja que permeten reduir la càrrega de llegir documents complets, oferint un text més curt que arreplega les idees principals de l'original, sense la necessitat què una persona faça el resum manualment. Això és especialment útil en àmbits com la recerca, els informes mèdics o el periodisme.

En aquest treball ens focalitzarem en l'entrenament de models per a resum en llenguatge simplificat d'articles biomèdics, en el marc de la competició BioLaySumm organitzada per l'ACL (https://biolaysumm.org/). És a dir que no només pretenem extraure la informació més rellevant de l'article sinó que a més aquesta siga verídica i entenible pel públic general (planer). Amb les diferents aproximacions ens centrarem en analitzar com afecta a la qualitat dels resums el fet de donar-li més context al model, fins i tot ampliant amb informació externa (RAG); condicionar la generació del resum o aplicar un model de regressió per triar el millor resum per cada mostra d'entre diversos candidats.

1.1 Motivació

Hui en dia podria afirmar-se que l'Aprenentatge Automàtic està experimentant la seua època de major esplendor, ja no només per la quantitat de grups de recerca que hi ha en les universitats i en les empreses privades col·laborant per fer avenços en aquest camp, sinó perquè és una eina que ha passat a estar present en el dia a dia de les persones. Mai s'havia parlat tant de la informàtica ni de la Intel·ligència Artificial com en els últims anys, cada dia les notícies estan plenes d'articles al voltant d'aquest tema, alguna nova aplicació, un nou descobriment, etc. Això és positiu perquè la societat s'està acostant a la IA, mostrant interès per comprendre-la, i generant uns debats ètics necessaris per al seu desenvolupament equilibrat amb la societat.

En aquest treball, ens centrem en el Processament del Llenguatge Natural (PLN), una branca de la IA que actualment gaudeix de gran popularitat, possiblement perquè impressiona més la població o perquè es percep com una amenaça. És una cosa acceptada que una màquina pugui reconèixer una cara o un cotxe, però quan aquesta comença a adquirir habilitats com la comunicació, que abans consideràvem exclusives dels humans, i pot generar textos per comunicar-se com a iguals, genera un gran impacte. Això s'ha

vist amb ChatGPT, que ha causat un gran rebombori mundial. Com es pot observar en la gràfica 1.1 l'interès per la IA ha crescut notablement des del llançament de ChatGPT.

Figura 1.1: Evolució de les cerques en Google del terme "Intel·ligència Artificial" en els darrrers cinc anys

A més, en una societat amb accés sense precedents a la informació però amb una capacitat d'atenció reduïda per les noves tecnologies, necessitem mètodes per captar l'atenció del públic, com en el cas dels articles científics o biomèdics. Això requereix estímuls clars, breus, concisos i a un nivell comprensible. També per als investigadors, metges o qualsevol persona que necessita revisar molts documents, és fonamental comptar amb resums de qualitat que permeten identificar ràpidament quins documents són útils, estalviant així temps valuós. En aquest context, els models que generen resums automàtics amb estil humà són claus. Han de captar les idees principals del text original, redactar-les de manera comprensible i didàctica, i atraure l'atenció del lector. Això podria tenir un impacte molt positiu, ja que, malgrat l'abundància de documents, la nostra societat està cada vegada més desinformada, i sovint es recorre a fonts ràpides i fàcils en lloc de les rigoroses. Aquests models podrien ajudar a dirigir especialment el públic més jove cap a fonts d'informació més fiables.

1.2 Objectius

En aquesta secció detallarem els distints objectius que es plantegen assolir amb aquest projecte:

1. **Generació de resums abstractius amb llenguatge simplificat:** L'objectiu fonamental i més gran d'aquest projecte és en definitiva ser capaços de crear models que donat un article biomèdic siguen capaços no només de resumir-lo amb les seus pròpies paraules, sinó que a més aquest resum siga comprensible per un públic no expert. De manera que este tipus de text siga més accesible per al públic en general i es democratitze el seu ús.

2. **Ús de Longformers:** Els textos biomèdics proporcionats són de naturalesa complexa i extensa. La qual cosa implica que necessitem emprar un tipus de model com els Longformers que estiguen preparats per gestionar textos d'entrada amb una longitud superior als tradicionals 512 tokens. Tot i així també es requerirà fer una selecció amb consciència de quines parts de tot el text proporcionem al model.

3. **Aplicació de tècniques Retrieval-Augmented Generation (RAG):** Una de les línies que els organitzadors de la competició BioLaySumm proposen com a prometedora,

és l'ús de RAG. I és que en este tipus de contextos a on els models han de treballar en un domini tan específic i en especial quan han de dedicar-se a una tasca generativa, el coneixement assolit pels models durant el preentrenament moltes vegades no és suficient i és necessari incrementar el context que els arriba.

4. **Generació condicionada:** Un dels altres objectius del treball, i que també es planteja com una línia prometedora, és condicionar la generació. En concret, pretenem condicionar-la a partir de donar-li certa informació en el decoder i a través d'indicacions com si es tractara d'un LLM.

1.3 Estructura de la memòria

La memòria d'aquest projecte està conformada per un total de sis capítols, els quals anem a descriure a continuació breument per donar una visió general:

- **Capítol 1, Introducció.** El primer capítol pretén descriure l'àmbit en què s'ubica el treball, justificar la seua existència i explicar els objectius que es pretenen aconseguir amb ell.

- **Capítol 2, Estat de l'art.** L'objectiu d'aquest capítol és proporcionar un marc teòric que situe el lector, ajudant-lo a comprendre conceptes fonamentals per a entendre el projecte desenvolupat i els termes utilitzats en la memòria. A més, s'explica l'evolució i els treballs més recents en el processament del llenguatge natural, amb un enfocament focalitzat en la generació automàtica de resums.

- **Capítol 3, Metodologies i sistemes utilitzats.** En el tercer capítol s'explica la tasca proposada en la competició, es descriu el corpus emprat al llarg del projecte, es discuteixen les diferents alternativas per representar els textos, s'expliquen les mètriques oficials de la competició emprades per realitzar l'avaluació dels models, així com l'arquitectura dels models transformers i en particular Longformers que s'utilitzen com a punt de partida.

- **Capítol 4, Eines utilitzades.** En el quart capítol es descriuen les distintes eines tant software com hardware que s'han emprat i que han permés el desenvolupament d'aquest projecte.

- **Capítol 5, Experimentació i resultats.** En aquest capítol ens centrem en el treball que s'ha desenvolupat durant la realització del treball fi de màster, exposant amb detall el procediment que s'ha anat seguint, problemes que han anat apareixent, decisions que s'han anat prenent i una discussió dels resultats obtinguts per cadascun dels models.

- **Capítol 6, Conclusions.** En l'últim capítol tanquem amb les conclusions on recapitulem què és el que s'ha assolit amb aquest treball, a quines dificultats ens hem enfrontat i el treball futur que es podria abordar per ampliar aquest treball.

1.4 Context i col·laboracions

La realització d'aquest projecte s'ha realitzat amb el gaudiment d'una beca formativa de col·laboració de tipus A oferida per l'Institut Valencià d'Intel·ligència Artificial (VRAIN

[1]). Amb la finalitat de formar a l'alumnat de màster i acostar-lo al món de la recerca en l'àmbit de la intel·ligència artificial durant els seus estudis de postgrau.

En concret s'ha col·laborat amb el grup ELiRF [2] (Enginyeria del Llenguatge i Reconeixement de Formes). Que està especialitzat en la recerca en l'àrea del Processament del Llenguatge Natural (PLN). El projecte ha constat de la participació en la competició BioLaySumm 2024[3] organitzada per l'ACL i la consegüent experimentació que s'ha fet per a aquest TFM.

També cal agrair que a banda de la beca de col·laboració proporcionada pel VRAIN, vaig ser beneficiari d'una beca per estudiar el màster concedida pel valgrAI[4] (Valencian Graduate School and Research Network of Artificial Intelligence). Que és una fundació sense ànim de lucre financiada per la Generalitat Valenciana i conformada per les cinc universitats públiques valencianes.

1.5 Vinculació amb els estudis cursats

Per a la realització d'aquest treball han sigut necessaris una sèrie de coneixements adquirits al llarg de tot el màster així com els que ja havíem obtingut durant la carrera i la realització del TFG que es va desenvolupar amb el mateix grup de recerca. Però en especial d'aquells assolits en la branca de Tecnologies del Llenguatge i precisament impartides per membres d'ELiRF: Lingüística Computacional (LC), Traducció Automàtica (TA) i Aplicacions de la Lingüística Computacional (ALC). Aquestes assignaturas ens han permès conèixer de primera mà les tècniques, models i tecnologies que es gastaven històricament i aquells que millor funcionen en l'actualitat en les diverses àrees que envolten el Processament del Llenguatge Natural. No obstant, la resta d'assignatures del màster també han sigut molt útils per aportar una sèrie de coneixements fonamentals al voltant del *Machine Learning* així com proporcionar major fluïdesa a l'hora de desenvolupar el programari necessari per realitzar els entrenaments i avaluacions dels models.

[1] https://vrain.upv.es/
[2] https://xarrador.dsic.upv.es/
[3] https://biolaysumm.org/
[4] https://valgrai.eu/

CAPÍTOL 2
Estat de l'art

2.1 Processament del Llenguatge Natural

El Processament del Llenguatge Natural és una àrea d'estudi que comprèn la Lingüística Aplicada i la Intel·ligència Artificial. La seua finalitat és aconseguir que els ordinadors puguen "entendre" el llenguatge humà i així facilitar la interacció persona-màquina. El PLN pot classificar-se en dos blocs [1]: *Natural Language Understanding* (NLU) que es centre en arribar a entendre el text i extraure coneixement d'ell. Mentre que el *Natural Language Generation* (NLG) s'enfronta a una tasca més complexa: generar un text que siga coherent a partir d'una entrada que pot ser textual o no.

L'estat de l'art del PLN ha avançat molt ràpidament en els últims anys gràcies a la gran quantitat de dades disponibles, la potència computacional i als avanços en algorismes d'aprenentatge profund. Algunes de les principals innovacions i tècniques que són l'avantguarda del PLN: [2]

- **Models preentrenats**: Hi ha models lingüístics preentrenats en una o vàries llengües amb una quantitat massiva de dades, com BERT, que poden agafar-se com a punt de partida per a especialitzar-los en alguna tasca particular. L'entrenament de models lingüístics preentrenats s'ha demostrat que permet millorar els resultats obtinguts quan s'entrenen models especialitzats en una determinada tasca partint d'un model ja preentrenat [3], a banda de l'estalvi de temps i recursos que suposen.

- **Transferència d'aprenentatge**: És una tècnica que permet adaptar models entrenats en una tasca perquè aprenguen una altra. Aquesta forma de treballar s'ha demostrat que millora l'eficiència dels models de PLN [4].

- **Mecanismes d'atenció [5]**: Han permès als models de PLN centrar-se en parts concretes d'una frase o document, la qual cosa ha aconseguit millorar la seua precisió. En aquest punt ens centrarem quan expliquem els *Transformers* que tot i que no van ser els primers en parlar dels mecanismes d'atenció sí que van aconseguir explotar aquesta tècnica per obtindre uns resultats molt bons.

- **Aprenentatge multitasca [6]**: És possible entrenar un sol model per resoldre diferents tasques simultàniament, ajudant a millorar la seua eficàcia i reduir la quantitat de dades d'entrenament necessàries respecte d'haver entrenat models separats per cada tasca.

Algunes de les aplicacions fonamentals en les quals s'aprofita el PLN són [1, 7]:

- **Resum automàtic**: El model rep un text i genera un altre d'eixida que resumeix el contingut fonamental del text d'entrada. Pot seguir una tècnica extractiva o abstractiva.
- **IA conversacional [8]:** Implica construir models que són capaços d'entendre i generar un diàleg que simula l'humà.
- **Traducció automàtica**: Donat un text en una llengua genera com a eixida eixe mateix text en un altre idioma.
- **Reconeixement d'entitat nomenada (NER)**: Detecta paraules que identifiquen persones, llocs, empreses, etc.
- **Etiquetat de part del discurs (POS)**: Porta a terme un etiquetat morfosintàctic de les paraules que formen part del discurs, és a dir, determinar si és un substantiu, un pronom, un verb, etc.

I ara passarem a fer un recorregut evolutiu de les tècniques i arquitectures que han anat emprant-se en el PLN [1]. Encara que la recerca en esta matèria va començar ja en els anys 40 en traducció automàtica, parlarem d'un passat més recent, a partir de quan se van introduir les xarxes neuronals recurrents, perquè tot i que les convolucionals també es van aplicar en tasques de PLN no van tindre el mateix èxit que van tindre en l'àmbit de la Visió per Computador.

Les *Recursive Neural Network* (RNN) [9] van aparèixer amb l'objectiu de cobrir la necessitat de tindre un major coneixement del context. El funcionament d'aquestes xarxes consisteix en què a cada pas reben una paraula de la seqüència com entrada així com l'estat ocult del pas anterior i amb eixa informació generen el nou estat ocult. Este estat ocult és la base d'aquesta arquitectura, perquè és el que representa el context, la manera de recordar allò que s'ha llegit fins el moment. Però també presenten una sèrie de problemes: [10]

- Com acabem d'explicar, per poder analitzar una paraula s'han d'haver processat totes les anteriors, és a dir, és una tasca essencialment seqüencial que impossibilita la paral·lelització.
- Per a cadenes llargues se pot anar perdent la informació dins d'este estat ocult mentre avança en la seqüència, és el que se coneix com esvaïment del gradient, perquè no és capaç de recordar informació de paraules prou allunyades que realment poden ser d'elevada rellevància per molt que estiguen separades. Com ocorre per exemple en les oracions subordinades:

  *El llibre que em vas deixar va **agradar-me** molt.*

  Amb una alta probabilitat en el moment que arribe a **"agradar-me"** ja no sap a quin objecte directe s'està referint. A vegades també pot passar l'efecte contrari, és a dir que en lloc de tendir a zero el gradient, hi ha una explosió del gradient, és a dir que se fa incontrolablement més gran.
- Un altre problema, que s'ha intentat solucionar amb arquitectures bidireccionals, és que el model original només té en compte les paraules prèvies, però per tal d'entendre el context poden ser igualment (o més) importants les paraules consegüents.

Precisament el problema de l'esvaïment del gradient que ocasionava al cap i a la fi que la xarxa no fóra capaç de recordar informació de seqüències llargues, va portar a l'evolució

d'estes xarxes cap a les *Long short-term memory* (LSTM) [11]. Se tracta d'una arquitectura basada en RNN però que introdueix nous comportaments a través de tres portes i un *cell state*. L'eixida que proporciona LSTM en cada moment depèn de 3 components, els dos que ja estaven presents en RNN: l'estat ocult que s'haja generat en l'estat anterior i una paraula de la seqüència d'entrada; i a banda d'això introdueix el *cell state* que és bàsicament la memòria a llarg termini que té la xarxa en un moment donat. D'altra banda incorpora 3 portes: porta d'oblit (*forget gate*), d'entrada(*input gate*) i d'eixida (*output gate*): [12]

- El primer pas consisteix en passar per la porta d'oblit, on l'estat ocult anterior i la paraula d'entrada entren a una xarxa neuronal que torna un vector de components entre 0 i 1 que reflexen com de rellevant és la paraula associada a eixa component respecta a la d'entrada. El resultat multiplica al cell state de tal manera que en multiplicar per nombres propers a 0 "oblidarà" o almenys perdran pes aquelles parts que considere irrellevants. Per tant esta porta és la base, perquè és la que decideix què recordar i què oblidar.

- El següent pas s'encarregarà de decidir quina nova informació afegir al *cell state*. Això es fa mitjançant l'estat ocult anterior i la paraula d'entrada, passant-les per dos xarxes neuronals. La "xarxa de nova memòria" que genera un vector de pesos en l'interval [-1,1] que indiquen en quina quantitat s'ha de modificar cadascuna de les components del *cell state*, és important que hi haja nombres negatius per tal de reduir-ne l'impacte. L'altra xarxa per la qual passa és la porta d'entrada que actua com a filtre per identificar quines de les components del "nou vector de memòria" interessa realment mantindre, per això genera un vector amb valors [0,1]. Finalment se multipliquen ambdós vectors i se sumen al *cell state* que ja teníem.

- El tercer i últim pas es basa en calcular el nou estat ocult, per a això es filtra el nou *cell state* que s'ha generat per una xarxa tanh que torna un vector amb valors en l'interval [-1,1] i això juntament amb la paraula d'entrada i l'estat ocult previ entra en la porta d'eixida, que és una xarxa neuronal semblant a la d'oblit que trau només aquella informació que siga vertaderament útil (representada mitjançant un vector de pesos) i en multiplicar els dos vectors queda com a resultat el nou estat ocult.

Aquest seria un procés iteratiu que s'executaria per cadascun dels elements que composen la seqüència d'entrada i finalment s'aplicaria una capa linial que convertiria l'últim estat ocult en una eixida entenible. El LSTM va tindre una evolució, la *Gated Recurrent Unit* (GRU) que obtenia uns resultats semblants però reduint la complexitat [13], és a dir, havia d'aprendre una menor quantitat de pesos. Això és gràcies a combinar el *cell state* i l'estat ocult en un a soles, i la porta d'entrada i d'oblit també en una conjunta, la d'actualització. Per tant quedaran només dos portes, la de reseteig, que decideix quina informació de l'anterior continua sent necessària i la d'actualització que determina a partir de l'eixida de la porta anterior quin ha de ser el nou estat ocult.

Finalment arribem a l'arquitectura que és estat de l'art, que va revolucionar el món del PLN, els *Transformers*. Per poder entendre el funcionament d'aquesta nova arquitectura és precís entendre el concepte fonamental en el qual es basa: l'atenció. Fins el moment el que s'estava fent servir era un model *encoder-decoder* en el qual primer la cadena d'entrada passava pel codificador que s'encarregava d'anar tokenitzant la cadena d'entrada, extraent una representació adequada i una vegada havia acabat, entrava en el descodificador i per exemple s'encarrega de generar un resum o una traducció; el problema d'açò és que segons en quin moment de la generació se trobe, pot ser més útil una part de l'entrada o altra. A partir d'aquesta intuïció és quan entra en joc el concepte d'atenció. D'aquesta manera es tindrà en compte la part de l'entrada que més influència

tinga en cada moment, independentment de la seua posició, però a canvi s'introdueix un cost computacional major perquè s'ha d'aprendre a ponderar quins tokens [1] són més "rellevants". Amb esta idea passem a explicar en què consisteixen els diferents mecanismes d'atenció [14, 15]:

- **Atenció en seq2seq**: Les arquitectures seq2seq són aquelles en les quals no es necessita només entendre una seqüència d'entrada, sinó també generar un text d'eixida, per exemple en la traducció o resum automàtic. Cadascun dels tokens del descodificador s'han de fixar en els tokens del codificador per decidir quins d'aquests necessiten una major atenció.

- ***Self-attention***: És un mecanisme com l'anterior, però ara s'aprèn la importància que tenen per cada paraula la resta de paraules del text del qual en formen part. La qual cosa permetrà que una paraula siga entesa en el context en el qual es troba, característica fonamental per poder desambiguar i permetre que una paraula polisèmica siga entesa correctament, com en estes dos oracions en les quals apareix la paraula *ratolí* però amb una acepció diferent:

  El gat es va menjar al **ratolí**.
  M'he comprat un nou **ratolí** per a l'ordinador.

- ***Multi-head attention***: L'atenció pot executar-se diverses vegades en paral·lel per crear una atenció *multi-head*, les eixides independents que se generen són concatenades. La motivació per utilitzar este mecanisme és que permet que cada cap d'atenció es focalitze de manera diferent en les diverses parts de la seqüència, per exemple entre les dependències a llarg i a curt termini.

A continuació expliquem els fonaments darrere dels *Transformers*. Es tracta d'una arquitectura dissenyada per investigadors de Google en el 2017 inicialment plantejada per a tasques de traducció automàtica i que ha revolucionat el món de l'aprenentatge automàtic i en especial del PLN i ha desbancat a les arquitectures abans esmentades, en la secció de Justificació de les decisions explicarem les raons. Els *Transformers* estan construïts mitjançant N codificadors enllaçats els uns als altres i N descodificadors també enllaçats; no utilitza cap tipus de recurrència o convolució, simplement aplica el mecanisme d'atenció en cadascun dels codificadors i descodificadors. Per tant l'atenció se converteix en la pedra angular d'aquesta arquitectura. Cosa gens sorprenent si tenim en compte el títol del paper on se va presentar aquesta nova tecnologia: *"Attention is all you need"* [15]. El seu funcionament el podem dividir en 4 passes:

- **Primer pas: Afegir codis posicionals als *word-embeddings***
  Primer que res tant l'entrada com eixida que se li passa al Transformer, cal que passe per una transformació a *word-embeddings*, que són vectors de longitud fixa on cada posició es correspondria amb un atribut i en quin percentatge el posseeix. Després, com en els Transformers desapareixen la recurrència i les convolucions de les arquitectures prèvies, hem d'inserir d'alguna manera la informació posicional, perquè sàpiga quin és l'ordre original de la seqüència. Això es fa mitjançant els codis posicionals (*positional encodings*) que permeten emmagatzemar la informació sobre la seua posició en els mateixos *embeddings* en lloc d'anar analitzant la cadena paraula a paraula permetent així trencar amb la dependència que tenien les anteriors arquitectures. En el cas que a nosaltres ens ocupa que són els models seq2seq, tenen una particularitat, i és que els tokens estan desplaçats una posició a la dreta

[1] Un token és la unitat mínima en la qual descomposem els textos per al seu posterior processament.

i comencen amb un token especial, "begining of sentence": <bos>. És per això que cal afegir els tokens posicionals abans de començar a passar-li'ls al primer codificador i descodificador.

- **Segon pas: Codificació**
  Com hem dit el codificador està composat per un conjunt de N capes idèntiques (6 en el paper original) connectades enter elles i cadascuna d'estes capes està composada al seu torn per 2 subcapes: una primera que aplica el mecanisme de la multi-head self-attention sobre els embeddings d'entrada, per tal de determinar l'anteció que mereix cada token, este resultat és normalitzat i passa a la següent subcapa que és una xarxa neuronal de tipus feedforward. En ambdós casos se gasten connexions residuals i normalització. I el resultat d'estos codificadors serà l'entrada del següent codificador i l'últim d'ells li ho passarà al descodificador.

- **Tercer pas: Descodificació**
  El descodificador està composat igualment per N capes idèntiques (també 6 en el paper original) connectades entre elles; en este cas a més de les dos subcapes de les quals ja hem parlat en l'encoder, s'insereix una tercera, encarregada d'aplicar l'atenció *multi-head* sobre l'eixida del descodificador. Veiem en detall què fan cadascun d'estos components:

  1. Els embeddings d'eixida, sobre els qual s'han afegit ja els codis posicionals i s'ha aplicat un desplaçament a dretes, se passen a una atenció *multi-head*, però amb una particularitat, i és que s'emmascaren totes aquelles posicions consegüents. Açò juntament amb el desplaçament a dretes garanteix que quan se fa una predicció en un moment donat només se tenen en compte les eixides produïdes amb anterioritat.
  2. La informació se passa a una altra atenció *multi-head* sense emmascarar que permetrà a cada posició del descodificador tindre atenció sobre la seqüència d'entrada.
  3. El resultat és finalment passat a una xarxa *feedforward*.

  D'una manera semblant a la del codfiicador, totes les parts fan servir les connexions residuals seguides d'una normalització. El resultat del descodificador serà l'entrada del següent descodificador i en l'últim cas serà ja la del classificador.

- **Quart pas. Classificador**
  En esta part simplement cal utilitzar una transformació linial i una softmax per tal de convertir l'última eixida del descodificador en un vector de probabilitats (de la mida del vocabulari) de quin hauria de ser el pròxim token generat.

La grandesa que van tindre els *Transformers* va ser no només aconseguir traure uns millors resultats front al que fins el moment eren els models estat de l'art [15], sinó a més permetre entrenar models a una major velocitat gràcies a que superaven les limitacions dels models previs que estaven molt restringits a la seua naturalesa linial; mentre que esta nova arquitectura obria pas a la paral·lelització, la qual cosa també permetia una major democratització perquè qualsevol puguera entrenar els seus models. El major problema de l'arquitectura *Transformer* és la necessitat de gran quantitat de mostres per al seu entranament, però açò més o menys se pot pal·liar gràcies a l'ús dels models preentrenats, que aprofiten la transferència de l'aprenentatge (*transfer-learning*) [4].

Per tot el que hem vist podem concloure que el PLN, és un camp d'estudi molt actiu i que està constantment en avanç. I el seu estat de l'art es caracteritza per tindre uns models d'una precisió molt elevada que són capaços tant d'entendre com de generar textos amb

una fluïdesa i naturalitat remarcables, que tenen el potencial de revolucionar una gran part de sectors. Tant l'arquitectura Transformer com de Longformer s'explicaran amb més detall en el Capítol 3.

2.2 Generació automàtica de resums

Els treballs vinculats amb el PLN no s'han conformat simplement amb aconseguir que les intel·ligències artificials siguen capaces de processar i comprendre un text, sinó que van més enllà i volen que també tinguen la capacitat de generar text per elles mateixes, en concret una àrea d'investigació molt activa és precisament la que treballem en aquest projecte: la generació automàtica de resums.

Quan parlem de generació de resums podem considerar diferents classificacions: [16]

- Abstractiu vs extractiu: Un resum extractiu és aquell que està constituït exclusivament per paraules o seqüències de paraules presents en el text d'entrada. Mentre que si s'utilitzen paraules que no hi estaven presents, es tracta d'un resum abstractiu. Hi ha mètriques que permeten avaluar com d'abstractiu és un resum, com discutirem en la secció de mètriques.
- Nombre de textos d'entrada: Els textos generats poden ser el resum d'un únic text o bé un resum general d'un conjunt de textos (multi-document summarizaton).
- Segons l'objectiu: Pot buscar-se resumir el text en general, o estar focalitzat a un context, una temàtica... proporcionats (*query-focused*), la qual està experimentant un creixement de la seua rellevància [17].

Les primeres aproximacions a la generació de resum automàtic van ser extractives i consistien en puntuar les millors frases mitjançant algun sistema com Lead-K que selecciona les K primeres frases d'un text, perquè se suposa que aquestes serien les frases més rellevants, i en certs àmbits com per exemple els textos periodístics aquesta intuïció resultava prou efectiva. Però el problema d'este mecanisme és que degut a la impossibilitat de combinar informació important que està repartida pel text, s'originen resums amb falta de cohesió i coherència (pensem per exemple en les anàfores o sobretot en les catàfores que fan referència a una entitat que encara no ha aparegut en el text), llavors si volem aconseguir uns resums que s'aproximen a la qualitat dels humans, no tenim prou amb el resum extractiu.

En contraposició tenim els models abstractius, que són els resums que busquem aconseguir perquè són els més semblants als de producció humana, perquè la nostra tendència no és extraure frases sense més, sinó reformular-les amb les nostres pròpies paraules de manera que posem de manifest les idees principals. No obstant, suposen un gran repte perquè quan els humans resumim portem a terme un procés en el qual apliquem una gran quantitat de coneixement que hem anat aprenent al llarg dels temps sobre el comportament del llenguatge i dels temes concrets que tracta el document en qüestió, per la qual cosa transmetre aquest coneixement a les màquines no és una tasca gens fàcil [18] i a més implica que el model haja de tindre una major comprensió del text. Un exemple d'este tipus de resums és el cas de PEGASUS [19] que ha obtingut uns resultats molt bons en certes tasques i a més és capaç d'adaptar-se sense complicacions a conjunts de dades que no havia vist prèviament.

Des del treball de Rush et al. (2015) [20] que aplicava xarxes neuronals de traducció per a generar resums abstractius, els resumidors abstractius automàtics han estat construits en la seua gran majoria mitjançant mètodes que utilitzen xarxes neuronals. Segui-

dament explicarem algunes de les idees claus que van aparèixer a partir de la investigació de resums abstractius mitjançant xarxes neuronals:

L'esquema Codificador-Descodificador

Actualment la majoria de resumidors abstractius es basen en el model seq2seq, que fa servir una arquitectura codificador-descodificador. El codificador s'encarrega de construir cada frase com una llista de vectors de longitud fixa (*word-embeddings*), que capturen cada paraula i el seu context . I després el descodificador genera un resum a partir d'eixos vectors codificats. L'entrenament té lloc mitjançant parells de document-resum per maximitzar la probabilitat d'un resum correcte:

Codificador

Té una finalitat semblant a la d'extracció d'informació en les aproximacions clàssiques, perquè ambdues se centren en capturar la informació rellevant per a posteriorment generar un resum de qualitat. Engloba dos passes fonamentals:

1. **Preprocessament de dades**: Per preprocessar les frases d'entrada molts models utilitzen unes representacions basades en paraules (tot i que per algunes llengües com el xinès una representació basada en caràcters pot ser més adient [21]). La majoria dels models empren vectors de paraules ja preentrenats mitjançant grans corpus, com és el cas de word2vec [22] o GloVe [23], però altres opten per aprendre els *word-embeddings* durant el mateix entrenament. Com codificar documents molt llargs pot convertir-se en una tasca molt complexa, una manera de la qual s'ha afrontat aquest repte per tal de mantindre tot el context, és comprimir-lo mitjançant mètodes extractius que seleccionen les frases més representatives del document [24].

2. **Selecció del codificador**: Amb l'objectiu de generar models que aconseguisquen un millor aprenentatge de la representació abstractiva del text d'entrada i controlar el fluxe d'informació que circula entre el codificador i el descodificador algunes investigacions s'han centrat en seleccionar i dissenyar els seus propis codificadors. Típicament per construir el codificador s'han emprat CNNs [25], que després van ser substituits pels RNNs [26, 27] per la seua incapacitat per processar seqüències llargues, no obstant encara no aconseguien resoldre eixe problema del tot, per la qual cosa van ser substituides per les LSTM [11] o en alguns casos per GRUs que necessiten menys paràmetres i per tant són més ràpides d'entrenar obtenint uns resultats equivalents [13].

Descodificador

Pel descodificador s'opta normalment per una implementació mitjançant RNNs. En cadascun dels pasos, el RNN rep com a entrada dos vectors, un que representa la seqüència que ha generat ell mateix fins el moment i el vector que ha generat el codificador a partir de la seqüència d'entrada, i torna un vector de la mida del vocabulari que es converteix en un vector de probabilitats mitjançant una capa softmax i o bé se genera la paraula més probable, o el que és més comú: agafar les k més probables on k és el tamany del *beam* [20].

Millores aplicables a l'esquema Codificador-Descodificador

- **Atenció**: Hi ha algunes frases o paraules que són més importants que altres al llarg de tot el document d'entrada, i aquelles més importants són les que apareixeran en el resum amb una major probabilitat. Per tal de poder identficiar-les podem aprofitar els mecanismes d'atenció.

  La idea fonamental darrere de l'atenció és alimentar el descodificador amb un vector d'entrada (conegut com vector de context) que codifica les frases importants [28], la qual cosa permetrà calcular un pes per a cadascun dels elements en cada pas i així poder aprofitar la informació processada per tal de poder separar la informació rellevant d'aquella que no és important. Segons si utilitzem una atenció a nivell global (de frase) o local (de paraula), el model resultant tindrà l'habilitat d'extraure informació rellevant a diferents nivells.

- **Distracció/Cobertura**: Tot i que l'atenció ens permet identificar i focalitzar-nos en aquelles frases més importants, no està exempta de problemes. S'ha observat que l'atenció pot centrar-se extremadament en el mateix contingut, de manera que el resum generat acaba sent molt redundant. Ahí és on entra en joc la distracció, també anomenada cobertura, [29] que evita centrar-se en el mateix continugt reduint la probabilitat/pes del contingut repetit en el vector de context, en el d'atenció i en la descodificació sobretot, encara que també pot aplicar-se durant l'entrenament.

- **Mecanismes de còpia**: Aquelles paraules que siguen molt freqüents tendiran a ser identificades com importants pel mecanisme d'atenció. Per contraposició, el model no tindrà l'habilitat per generar aquelles paraules rares o que esten fora del vocabulari. Per solucionar aquest problema es van proposar mecanismes de còpia mitjançant xarxes neuronals amb punters [30], que permet al model copiar directament en l'eixida elements presents en l'entrada, permetent que la xarxa també se centre en aquelles paraules rares o fora del vocabulari. Per implementar-ho, s'"equipa" al descodificador amb un interruptor que determina quan utilitzar el punter i quan el generador. Malgrat que haja resultat un mecanisme útil per generar resums llegibles, comporta un problema evident: que si copia directament paraules de l'entrada, s'assemblaran més a les aproximacions extractives que a les abstractives, especialment quan el descodificador sobreutilitza el punter. Per tant s'hauria de controlar el punt fins al qual aquest punter és utilitzat pel descodificador.

- ***Reinforcement learning***: El model Codificador-Descodificador compta amb dos debilitats: que la xarxa s'entrena maximitzant una mesura per a la semblança dels resums generats als de referència, com pot ser ROUGE, que no és necessàriament equivalent a minimitzar la pèrdua. I en segon lloc, mentre que el descodificador ha estat entrenat mitjançant resums de referència (generats per humans), a l'hora de la descodificació d'una paraula, el que té en compte és el resum que el propi model ha generat en l'últim pas, la qual cosa pot afectar notablement al seu rendiment per l'*exposure bias* [31]. Per afrontar aquests problemes s'ha aplicat el *Reinforcement Learning* (RL) [24], que permet resoldre un problema d'optimització, en el cas de la generació de seqüències, en lloc de minimitzar la *loss* podem maximitzar una recompensa basada en la mètrica d'avaluació del resum que desitgem, o fins i tot fer servir una funció objectiu híbrida que combine ambdues [32]. La qual cosa permet prendre decisions a nivell global (de frase) en lloc de local (de paraula) durant la generació.

Hui en dia la Intel·ligència Artificial i en concret el PLN tenen un gran popularitat que en bona mesura ha vingut donada gràcies al desenvolupament de noves tècniques d'a-

prenentatge molt exitoses. L'aprenentatge profund (*deep learning*) ha guanyat una gran importància en haver demostrat que pot aprofitar-se per millorar els resultats d'algunes tasques, en especial com s'ha comentat prèviament, els resultats obtinguts pels *Transformers* han resultat ser molt positius en el camp del PLN. Alguns sistemes importants que utilitzen aquests models són BART, BERT o PEGASUS.

Després d'haver fet un recorregut pels diferents mètodes i tècniques que s'han emprat històricament i actualment en el resum abstractiu, anem a explicar el model BART, donat que en el preentrenament del model presentat es segueix una estratègia com la d'aquest.

BART (*Bidirectional and Auto-Regressive Transformer*) [33] és un model lingüístic preentrenat desenvolupat per Facebook, que està basat fonamentalment en les arquitectures de BERT i GPT-2; una de les seues característiques fonamentals és la bidireccionalitat, és a dir que el text és processat de principi a fi però també del final al principi, la qual cosa permet capturar una major quantitat d'informació contextual; i és auto-regressiu, això és, genera tokens d'eixida d'un en un, condicionat tant a l'entrada com als tokens que ha generat prèviament. Es pot aprofitar per fer fine-tuning per a tasques de tot tipus: Etiquetat de seqüències, de tokens, traducció automàtica i generació de textos. Però per la seua naturalesa està especialment recomanat per a sequence-to-sequence. La idea bàsica d'aquest model és que els textos d'entrada passen primer per un codificador que de manera aleatòria els corromp, és a dir se'ls aplica soroll mitjançant alguna de les tècniques que comentarem. Sobre eixe text corromput, el model intenta reconstruir-ho, llavors amb les eixides que genere el descodificador, se comparen amb l'entrada original (sense corrompre) i s'optimitzen els paràmetres mitjançant l'entropia creuada. Algunes de les possibles tècniques per introduir soroll en les mostres d'entrada són les que podem veure gràficament en la Figura 2.1:

- *Token masking*: De forma aleatòria alguns tokens són substituis pel token especial [MASK], aquesta tècnica es feia servir ja en el model de BERT [34].
- *Text-Infilling*: S'emmascara tota una seqüència de longitud variable segons una distribució de Poisson i la feina del model és descobrir quina era la longitud d'eixe espai en blanc (identificat també amb un token [MASK]).
- *Token deletion*: S'eliminen certs tokens i el model haurà de decidir en quines posicions falten tokens.
- *Sentence permutation*: Es reordenen aleatòriament les diferents oracions que composen el text.
- *Document rotation*: El text es desplaça de manera que comence amb un token seleccionat aleatòriament i la tasca a la qual s'enfronta el model és trobar quin era el token d'inci original.

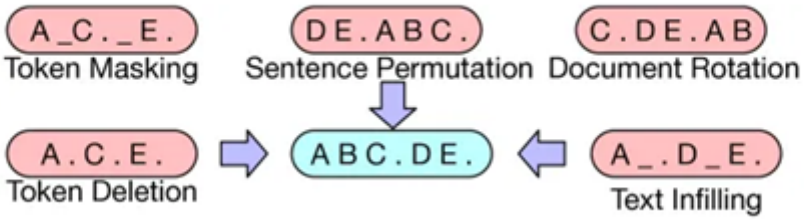


Figura 2.1: Exemples de tècniques per introduir soroll sobre les mostres d'entrada [33]

Totes aquestes tècniques per afegir soroll permeten al model preparar-se per treballar amb textos d'entrada que no són perfectes, tenen algun inconvenient, en definitiva el model resultant està més preparat per enfrontar-se a la realitat. Aquesta preparació ha resultat ser molt efectiva, ha obtingut uns resultats d'estat de l'art en tasques de classificació i ha millorat els resultats anteriors en nombroses tasques de generació de textos [33]. A més també s'ha pogut comprovar com afegir estes tasques durant el preentrenament milloren l'abstractivitat, que és un dels objectius que tenim en aquest treball.

2.3 Representació dels textos

Aquest és un aspecte fonamental, perquè la manera en la qual representem els textos és la manera mitjançant la qual aconseguim fer arribar la informació a la nostra Intel·ligència Artificial, perquè l'hem de representar de manera numèrica, donat que els models treballen amb nombres en coma flotant i cal convertir les cadenes de text a una seqüència de nombres de la mateixa dimensionalitat per a totes les mostres. A continuació explicarem diferents estratègies de representació i parlarem en concret dels *word-embeddings* que és el sistema que finalment utilitza la nostra arquitectura.

2.3.1. One-Hot

Aquesta representació consisteix en representar cadascuna de les paraules d'una frase donada com un vector de zeros i uns de dimensió $|V|$, on $|V|$ és la talla del vocabulari. De manera que la representació d'un text serà una matriu on cada columna és cadascuna de les paraules del text en qüestió i cada fila es correspondrà amb una paraula del vocabulari i així tindrem un 0 o un 1 segons si coincideixen el valor de la fila i la columna o no. Per la qual cosa per cada paraula del text hi haurà un sol 1 i la resta estarà a zeros. L'avantatge d'aquesta representació és la seua senzillesa, però és molt ineficient a nivell espacial, perquè s'està emmagatzemant una matriu de dimensionalitat molt elevada, que serà molt dispersa, és a dir que emmagatzemarà poca informació en comparació a l'espai que està ocupant, perquè necessitarem N vectors de $|V|$ components i cada vector tindrà com a molt un 1 i no aporta cap informació semàntica.

2.3.2. Bossa de paraules

La bossa de paraules, *bag-of-words* (BOW) en anglès és una manera de representació de textos que consisteix en construir un vocabulari de mida $|V|$ amb totes les paraules que poden aparèixer en el document. I després per cadascuna de les frases d'entrada s'aplica un procés de vectorització, és a dir, cada frase se convertirà en un vector de dimensió $|V|$ on cada posició indicarà el nombre de vegades que eixa paraula ha aparegut en la frase. Es tracta d'un model simple i ràpid de calcular. Però només permet considerar les freqüències d'aparició, es perd per complet tota informació posicional i igual que l'anterior tampoc aporta informació semàntica. Aquesta representació sol emprar-se quan tenim diversos documents per conèixer la freqüència de les paraules en documents, mitjançant *Term frequency*(tf) i *Inverse Document Frequency* (idf). Podem trobar un exemple de representació de text mitjançant la bossa de paraules en la Figura 2.2.

2.3.3. Word-Embeddings

Finalment anem a parlar del mecanisme per a representar textos que es fa servir en els Transformers i en general que més tendeix a utilitzar-se en PLN. Perquè si bé les anteriors

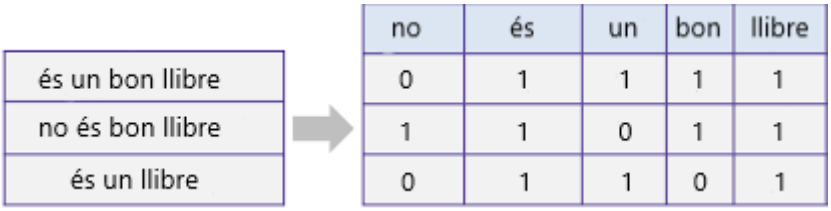


Figura 2.2: Exemple de representació mitjançant Bossa de paraules

representacions eren acceptables per a certes tasques com generació de text o classificació, per a altres com anàlisi de sentiment o traducció que precisen d'un major enteniment del context no són suficients perquè no guarden informació semàntica. Precisament ahí és on juguen un paper interessant els *word-embeddings*. Consisteix en representar les paraules en lloc de com un vector discret, com un vector de longitud reduïda (típicament de 50 a 300) i cada posició es correspondria amb un atribut. Es calculen mitjançant una xarxa neuronal, primer incialitzant els valors de manera aleatòria i se van aprenent durant l'entrenament i és ella la que decidirà quins són els atributs a utilitzar. És difícil interpretar què és cadascun dels atributs, però si veiem una representació visual de l'espai on estan distribuïdes les paraules, podrem observar com els vectors conserven informació d'aspectes del llenguatge i guarden la relació existent entre les diferents paraules. En la Figura 2.3 pot apreciar-se com paraules que estan relacionades per gènere, per capitalitat o conjugacions verbals són representades pròximes en l'espai vectorial.

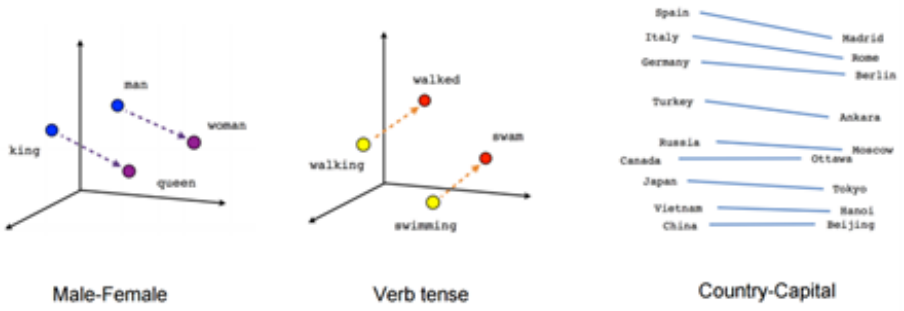


Figura 2.3: Representació de la proximitat en l'espai vectorial entre *word-embeddings* de paraules relacionades (Font: TensorFlow.org)

En aquesta figura pot observar-se com les diferents representacions mitjançant *word-embeddings* han estat capaces de capturar certes relacions semàntiques, com per exemple de gènere, temps verbal o un estat i la seua capital. Aquesta relació no és només visual, sinó que pot observar-se també a nivell aritmètic:

$$vec["príncep"] - vec["home"] + vec["dona"] \simeq vec["princesa"]$$

Una mateixa paraula pot tindre un significat totalment diferent segons el context en el qual se trobe (polisèmia). Els *word-embeddings* es divideixen entre contextuals i incontextuals [35].

- **Incontextuals**: Per a este cas no es va a tindre en compte en quin context es troba una paraula a l'hora de generar el seu embedding, açò és, que per cada paraula del vocabulari hi haurà un sol vector associat. El mètode més comú per generar este tipus d'embeddings és l'algorisme Word2Vec [22], que ofereix dos models: *Continuous Bag-Of-Words* (CBOW) i skip-gram, aquest últim és el que sol funcionar millor, i el que fa és per a les frases que conformen el corpus, tracta d'utilitzar cada paraula

per predir quines paraules seran veïnes perquè l'objectiu de l'entrenament d'aquest model és que per a una paraula donada ens done la probabilitat de que cadascuna de les paraules del vocabulari siga veïna d'ella. Aquesta alternativa és una bona aproximació per a tasques de PLN com la classificació o anàlisi de sentiments, que són aplicacions en les quals el significat global es pot comprendre a partir dels significats individuals, però per a tasques on el context sí que té major rellevància com traducció automàtica o resposta a preguntes, aquestes representacions tenen una major limitació.

- **Contextuals**: Apareixen precisament per superar la limitació de les representacions anteriors, perquè se presenta la necessitat de capturar la informació contextual per resoldre certes tasques del PLN. Els resultats demostren que amb este tipus de *word-embeddings* se milloraven els resultats que s'havien aconseguit fins el moment amb els incontextuals [36]. En aquest cas cada paraula, una paraula del vocabulari serà transformada en un vector de pesos o altre segons el context en el qual s'haja trobat. BERT és un exemple de model que empra embeddings contextuals i a diferència d'altres models, no atén simplement a les paraules que li precedeixen, sinó que per construir el context, ho fa de forma bidireccional.

Positional-Encoding

En el cas particular de *Transformers*, la representació de les paraules no es queda simplement en aplicar-los el *word-embedding*, sinó que a més se li afig una informació posicional, perquè com ja s'ha explicat en aquest document, els Transformers analitzen frases senceres, no van paraula per paraula, per tant d'alguna manera han de conservar la informació sobre quina relació de posició guarden les diferents paraules. De manera que quan se converteix una paraula a *word-embedding*, abans se li afig la posició que ocupa, dins de la frase de la qual forma part, de manera que una mateixa paraula que ocupe diferents posicions en la frase tindrà representacions lleugerament diferents. De manera que el resultat d'una frase serà una matriu, perquè cada paraula es converteix en tot un vector.

2.4 Retrieval-Augmented Generation (RAG)

El Retrieval Augmented Generation (RAG) és una tècnica innovadora en el camp del processament del llenguatge natural que combina la generació de textos amb la recuperació d'informació. En el context de la generació automàtica de resums simplificats, aquesta tècnica pot ser molt útil per millorar la qualitat i la rellevància dels resums produïts. Els grans models de llenguatge (LLMs), tenen emmagatzemada una gran quantitat de coneixement en els seus paràmetres, la qual cosa ha permés assolir resultats d'estat de l'art en moltes tasques, però en canvi, hi ha certs àmbits a on no aconsegueixen funcionar tan bé, perquè són problemes més intensius en quant al coneixement del domini. I s'ha demostrat que aplicar tècniques de RAG són molt útils per a tasques de NLP [37], millorant els resultats d'estat de l'art en certes tasques com *Question Answering* i per a tasques generatives s'obtenen textos més específics, diversos i factuals. RAG no només està present en tasques generatives, sinó també ajuda en tasques de comprensió (NLU) [38].

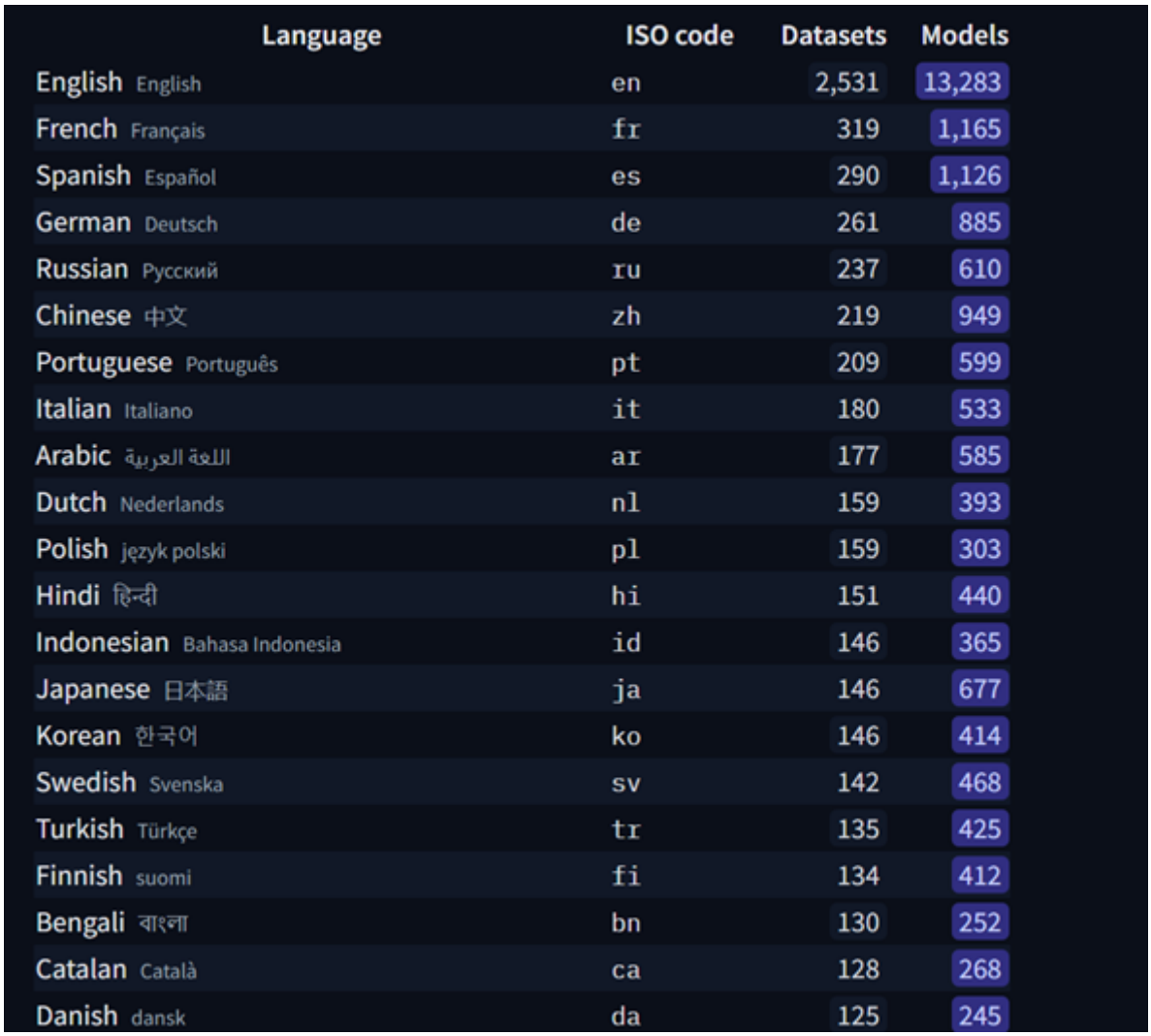
El concepte de RAG consisteix en combinar dos components: un de recuperació d'informació i un de generació de text. De manera que el primer component s'encarregarà de recuperar fragments de text rellevants que serviran com a context addicional per a la generació del text. La forma de recuperar la informació és molt diversa, pot ser un altre model que donat un text d'entrada siga capaç de generar context relatiu a ell, pot tractar-se d'un graf de coneixement, una base de dades sobre la qual fem consultes, etc. En qual-

sevol de les seues formes, la preparació d'aquest component és fonamental, i requereix d'un procés d'enginyeria prèvia. Els textos seleccionats pel primer component serviran per enriquir l'entrada al model de generació, com a context addicional. Per exemple en tasques de resum automàtic permeten generar textos més coherents i rellevants. A més donen molta flexibilitat perquè permeten que els models s'adapten amb major facilitat a dominis en els quals no s'havien especialitzat durant l'entrenament.

2.5 Corpus

Actualment, com hem esmentat en la discussió sobre l'estat de l'art del Processament del Llenguatge Natural (PLN), les tècniques més utilitzades es basen en xarxes neuronals, que requereixen grans volums de dades per aprendre els pesos de les seues connexions. Aquest requisit de dades és encara més accentuat en el cas dels *Transformers*, les xarxes neuronals més populars en l'actualitat, ja que tenen una estructura molt gran amb un nombre massiu de paràmetres que cal entrenar. A més, és important assenyalar que per entrenar aquest tipus de models no només es necessita el corpus per al fine-tuning, que especialitza el model en una tasca concreta com el resum automàtic; abans d'això, cal pre-entrenar el model amb un volum de dades encara més gran. I més en aquest entrenament en el qual partim d'un model que ja va necessitar dades per ser preentrenat: Longformer "base", després s'ha continuat preentrenat amb dades específiques del domini i per últim s'ha fet el fine-tuning amb dades específiques del domini i la tasca. Per tant, una part crucial del procés és recopilar i preparar un conjunt de dades de qualitat que el model puga emprar. La part positiva és que tenim un immens repositori de textos lliurement accessibles a la World Wide Web (WWW), que abasten tot tipus de temes; la part negativa és que aquests textos no es poden utilitzar directament des de les pàgines web. És necessari fer *web scraping* per extreure els arxius desitjats d'una pàgina i netejar-los, per exemple, eliminant el format HTML. No obstant hui en dia tenim disponibles una gran quantitat de conjunts de dades ja preparats per a ser utilitzats en entrenaments en diferents tasques especialitzades. Un problema és que la major part dels conjunts de dades disponibles estan, naturalment, en les llengües més parlades, ja que hi ha més recursos d'on extreure informació. Això crea un cercle viciós: com que hi ha molts documents en anglès, els models i datasets es desenvolupen en anglès perquè més recursos suposen millors resultats. Aleshores, com hi ha més eines disponibles en anglès, se segueixen creant recursos en anglès. Això perjudica les llengües minoritàries, que es veuen minoritzades en l'àmbit tecnològic, provocant una desigualtat en els drets lingüístics i les oportunitats per als seus parlants.

Per exemple en la Figura 2.4 els conjunts de dades i models que HuggingFace ofereix per entrenar models. Es pot observar clarament la diferència entre l'anglès i altres llengües. És rellevant esmentar HuggingFace no només perquè és la tecnologia que utilitzem en el nostre treball, sinó també perquè ha crescut molt en els darrers anys fins crear una comunitat molt gran i molt sòlida en l'àmbit de l'aprenentatge automàtic, especialment per als models basats en Transformers. HuggingFace proporciona una llibreria que simplifica notablement l'entrenament d'aquests models, fet que atrau molts usuaris, tant experimentats com novells que comencen a explorar aquest camp.



Language	ISO code	Datasets	Models
English English	en	2,531	13,283
French Français	fr	319	1,165
Spanish Español	es	290	1,126
German Deutsch	de	261	885
Russian Русский	ru	237	610
Chinese 中文	zh	219	949
Portuguese Português	pt	209	599
Italian Italiano	it	180	533
Arabic اللغة العربية	ar	177	585
Dutch Nederlands	nl	159	393
Polish język polski	pl	159	303
Hindi हिन्दी	hi	151	440
Indonesian Bahasa Indonesia	id	146	365
Japanese 日本語	ja	146	677
Korean 한국어	ko	146	414
Swedish Svenska	sv	142	468
Turkish Türkçe	tr	135	425
Finnish suomi	fi	134	412
Bengali বাংলা	bn	130	252
Catalan Català	ca	128	268
Danish dansk	da	125	245

Figura 2.4: Taula de conjunts de dades i models de HuggingFace en diferents llengües

CAPÍTOL 3
Metodologies i sistemes utilitzats

En aquest capítol explicarem la tasca de la competició, comentarem els diferents mecanismes i sistemes que hi ha darrere de la preparació de corpus, entrenament de models i avaluació de resultats realitzats per portar a terme aquest projecte.

3.1 Descripció de la tasca

En l'edició del 2024, BioLaySumm planteja una tasca que és crear un resum planer a partir d'un article de recerca biomèdica i el seu resum tècnic (*abstract*). L'organització proporciona un dataset biomèdic [39] que conté articles biomèdics procedents de dos fonts diferents: *eLife Sciences*[1] i *Public Library of Science* (PLOS)[2]. Cada mostra conté el text de l'article, el resum tècnic i el resum planer de referència. En l'Apèndix B podem trobar un exemple d'un article d'eLife i del resum planer oferit com a referència.

Per mesurar el rendiment dels sistemes, la competició proporciona una sèrie de mètriques per avaluar tres aspectes diferentes: *Rellevància*, *Llegibilitat* i *Factualitat*. Per a la ***Rellevància*** es van seleccionar ROUGE-{1,2,L} [40] i BERTScore [41]. Per a la ***Llegibilitat*** Flesch-Kincaid Grade Level (FKGL) [42], Dale-Chall Readability Score (DCRS) [43], Coleman-Liau Index (CLI) [44] i LENS [45]. I per a la ***Factualitat*** es va triar l'AlignScore [46]. Aquestes mètriques s'explicaran amb més detall en aquest capítol.

3.2 Descripció del corpus

En aquesta secció explicarem quin és el contingut i estructura del corpus emprat per a l'entrenament, validació i avaluació dels models. Abans de poder començar a fer els entrenaments, calia fer un estudi d'algunes característiques del corpus i realitzar un cert preprocés del mateix, que explicarem en l'apartat següent. En la Taula 3.1 podem trobar la distribució de les mostres dels dos corpus en cadascuna de les particions proporcionades. Es pot observar que les mostres estan fortament desbalancejades en favor de PLOS, la qual cosa pot plantejar un repte si es desenvolupa un únic model que resumisca articles d'ambdues fonts.

[1] https://elifesciences.org/
[2] https://plos.org

Font	# Train	# Val	# Test
eLife	4346 (91.9%)	241 (5.1%)	142 (3.0%)
PLOS	24773 (94.3%)	1376 (5.2%)	142 (0.5%)

Taula 3.1: Distribució de les mostras del corpus acompanyat del percentatge que representa la partició en cada mostra.

3.2.1. Preprocés i estadístiques del corpus

Ara passarem a explicar el preprocés que es va aplicar sobre el corpus per poder deixar-lo totalment preparat per poder fer tots els entrenaments que explicarem en futurs capítols.

Selecció de seccions

Com ja hem comentat en diverses ocasions en aquesta memòria, teníem un problema amb la longitud dels textos que havíem de processar, i l'ús de Longformers no era suficient per pal·liar-ho, llavors calia fer una selecció de quines eren les seccions que havíem de proporcionar-li al model per a generar el resum amb llenguatge simplificat. Vam decidir que el resum tècnic o abstract era la peça fonamental de tot l'article, perquè contenia un resum de tot el que contava i a més precisament el nostre objectiu era aconseguir una versió simplificada d'eixe resum, per tant l'havíem d'incloure necessàriament. De la resta de seccions que conformen un article científic vam pensar que la informació que més ens podia interessar eren la introducció i la discussió perquè són les parts que es centren en els punts principals de l'article, i a més gasten, en general, un llenguatge menys tècnic perquè tenen una funció més divulgativa que l'explicació de la metodologia, els experiments, etc.

No obstant, açò va presentar un nou problema i era que cada article tenia una estructura diferent, no tots tenien dues seccions explícitament anomenades Introducció i Discussió. Aleshores després de fer una anàlisi exhaustiva de quins noms de secció apareixien i amb quina freqüència vam fer una tria de termes que podíem acceptar com a secció d'Introducció i altres com a secció de Discussió. A continuació es detallen els finalment seleccionats, es seleccionarà la secció de l'article que continga alguna de les paraules llistades:

- Introducció:
  - *Introduction*
  - *Background*
  - *Motivation*
  - *Recent Developments*
  - *Summary*
- Discussió:
  - *Discussion*
  - *Discusion* (apareixia en ambdós formats)
  - *Conclusion*
  - *Result*
  - *Future*
  - *Concluding*

En la Taula 3.2 podem veure quants tokens ocupava l'article complet i quant ocupaven cadascuna de les seccions, era fonamental tindre esta informació per poder decidir quines seccions podien cabre en l'encoder i el decoder i quines combinacions podíem provar. Ací podem trobar una diferenciació molt clara entre ambdós fonts i és que tant l'article com els resums tècnics i especialment els planers, són prou més llargs en eLife que en PLOS. En el cas del resum planer són més o menys el doble de llargs.

Font	Secció	Train	Validació	Test
eLife	Introducció	1296.91	1299.59	1344.85
eLife	Discussió	2301.15	2219.30	2328.42
eLife	Resum tècnic	209.53	208.77	301.55
eLife	Resum planer	480.86	491.00	-
eLife	Article complet	12729.47	12522.27	11082.54
PLOS	Introducció	1095.30	1098.86	1127.52
PLOS	Discussió	2077.82	2120.82	1967.06
PLOS	Resum tècnic	337.51	341.33	349.61
PLOS	Resum planer	245.26	244.86	-
PLOS	Article complet	8371.27	8351.34	8555.39

Taula 3.2: Mitjana de longitud de les diferents seccions en cadascuna de les particions i fonts.

RAG: Extracció d'entitats biomèdiques

Una de les línies per les que vam apostar des del principi per abordar aquesta tasca era augmentar el context que se li proporcionava al model mitjançant RAG. Tot i que teníem un problema d'espai, creíem que podia ser molt beneficiós per al model que li proporcionàrem informació específica de l'àmbit biomèdic. En concret, la nostra idea era extraure les entitats biomèdiques que apareixen en l'abstract i proporcionar-li una descripció de les mateixes, d'aquesta manera estem indicant-li al model en quins punts s'ha de centrar i a més estem introduint descripcions en llenguatge planer, per tal que el model reba també eixe tipus de vocabulari i finalment poder ajudar a la llegibilitat dels textos generats. Perquè si no li proporcionem este tipus de descripcions, estem confiant en que durant les fases de preentrenament haja assolit suficient coneixement de l'àmbit com per poder descriure ara termes i conceptes tècniques d'una manera simplificada.

Per portar a terme aquest ampliament de context el que vam fer va ser preparar una *pipeline*, que tot i que ho vam fer per preprocés, estaria disponible per fer RAG en temps real sobre mostres que li anaren arribant. A la *pipeline* li arriba el resum tècnic i el primer que fa és trobar les entitats biomèdiques presents, per a la qual cosa gastem el model BERN2 [47] que està especialitzat en la tasca de *Name Entity Recognition* (NER) en l'àmbit biomèdic. Amb això aconseguíem que cadascuna de les mostres disposara d'un llistat d'entitats, amb el text original, l'entitat a la qual pertany, i el seu identificador únic de *Medical Subject Headings* (MeSH) [3] si en té associat. Una vegada teníem l'entitat reconeguda, féiem una consulta a l'API de Wikipedia per obtindre la descripció del terme, i en cas d'ambigüitat [4], si l'entitat disposava d'un MeSHId, aleshores buscàvem la descripció que ofereix MeSH per a eixe terme. Però les dades que s'oferien en la web de MeSH, no estaven totalment processades com per a poder-les fer servir, sinó que era un fitxer ASCII, llavors vam haver de processar-lo per convertir-lo en una mena de base de dades i poder fer eixes consultes.

[3] MeSH és un vocabulari de termes biomèdics vinculats amb la salut creat per la *National Library of Medicine* https://www.nlm.nih.gov/mesh/meshhome.html

[4] L'ambigüitat podia donar-se perquè un mateix terme estiguera relacionat en diferents àmbits i aleshores l'API no era capaç de ressoldre-ho

En la Taula 3.3 podem comprovar l'elevat nombre d'entitats que es troben en l'*abstract* de cadascuna de les mostres, si tenim en compte que de cadascuna de les entitats tenim pensat afegir la descripció i que contem amb un espai limitat com ja s'ha explicat, es tracta d'una quantitat desmesurada d'informació. Llavors el que vam fer va ser quedar-nos amb aquelles que realment foren rellevants, això és les que aparegueren tant el resum tècnic com en el planer, perquè realment les que ens interessa que el model conega són les que han d'aparèixer en el resum planer perquè són les que ha de ser capaç d'explicar. I aleshores això va donar lloc a un nombre d'entitats que ja podíem assumir, un poc menys de la meitat, com podem veure en la Taula 3.4. No obstant, en un escenari real com per exemple el de test, no contem amb el resum planer, llavors per fer eixa intersecció caldria entrenar un model que seleccionara les entitats realment rellevants del resum tècnic, que es deixa plantejat com a treball futur. En contradicció amb el que havíem comentat anteriorment de que els resums planers d'eLife eren prous més llargs que els de PLOS, en els de PLOS trobem aproximadament el doble d'entitats que en eLife.

Font	Partició	Mitjana entitats / mostra
eLife	Train	15.31
eLife	Val.	14.99
PLOS	Train	23.45
PLOS	Val.	23.53

Taula 3.3: Mitjana d'entitats en l'*abstract* de cada mostra en les diferents fonts i particions.

Font	Partició	Mitjana entitats / mostra
eLife	Train	6.12
eLife	Val.	4.91
PLOS	Train	11.12
PLOS	Val.	11.19

Taula 3.4: Mitjana d'entitats en l'*abstract* interesectades amb el resum planer de cada mostra en les diferents fonts i particions.

Com teníem limitacions d'espai també ens interessava conèixer quant d'espai ocupava en total afegir la descripció de cadascuna de les entitats rellevants, en la Taula 3.5 trobem la mitjana del que ocupaven les tres primeres frases de la descripció de les entitats de cada mostra. Vam agafar tres perquè consideràvem que era la quantitat de frases suficient com per aportar informació rellevant però sense saturar en excés l'encoder, sabent que per les longituds que estàvem veient que ocupaven ja les seccions d'interés per elles mateixes, afegint esta informació el resultat anava a ser que en moltes mostres s'haguera de truncar l'entrada de l'encoder, de manera que si féiem descripcions molt llargues el model veuria la descripció completa d'unes poques entitats en lloc de verue un poc de moltes entitats.

Font	Partició	Longitud mitjana descripció
eLife	Train	273.87
eLife	Val.	277.05
PLOS	Train	425.72
PLOS	Val.	433.52

Taula 3.5: Mitjana de tokens que ocupen les tres primeres frases de la descripció de les entitats de cadascuna de les mostres en les diferents fonts i particions.

3.3 Mètriques d'avaluació

Per a avaluar el rendiment dels nostres models de resum hem fet servir mètriques ben conegudes. Es pretrenien avaluar 3 aspectes: rellevància, llegibilitat i factualitat.

- **Rellevància:** Aquest aspecte pretèn avaluar quant s'aproximen els resums generats pels diversos models front als de referència. Per a això gastem mètriques com ROUGE [40] i BERTScore [41], que són ampliament acceptades per la comunitat com bones mesures per avaluar la qualitat de resum del model tot i que també han sigut criticades i s'estan investigant altres mesures que siguen més precises com Pyramid o METEOR.
- **Llegibilitat:** Donat que el nostre objectiu no és només generar resums dels articles biomèdics sinó que aquests siguen accesibles al públic general i per tant gasten un llenguatge més planer amb menys terminologia tècnica. Per a avaluar aquest aspecte es gasten Flesch-Kincaid Grade Level (FKGL), Dale-Chall Readability Score (DCRS), Coleman-Liau Index (CLI) i LENS[45]. Totes elles menys LENS s'han de minimitzar perquè represent el nivell que cal tindre per poder comprendre el text, de manera que com menor siga més accesible al públic general.
- **Factualitat:** Aquest aspecte s'avalua per tal de controlar les al·lucinacions del model, i que la informació generada siga consistent amb la d'entrada. Per això es gasta AlignScore [46].

A continuació expliquem amb més detall cadascuna de les mètriques:

3.3.1. ROUGE

Es tracta d'una mètrica per avaluar la rellevància d'un text, en concret torna un valor entre 0 i 1 que expressa com de semblants són dos textos, en el nostre cas: el resum que ha generat el model i el que tenim de referència. Dins de la mètrica de ROUGE, existeixen diferents variants que explicarem a continuació, però bàsicament hi ha dos tipus, ROUGE-L que busca la cadena coincident més llarga per realitzar les puntuacions i ROUGE-N que mesura el solapament entre n-grames. Per al càlcul de totes aquestes mètriques ens hem recolzat en la llibreria evaluate que expliquem en la secció d'eines software emprades.

ROUGE-1

És un cas particular de ROUGE-N en el que es calcula la superposició paraula per paraula. Per al ROUGE realment poden calcular-se tant el recall, com la precisió i el F-score, mètriques també ben conegudes. El recall seria el nombre de paraules del resum de referència que poden trobar-se en el generat, la fórmula seria:

$$Recall = \frac{\sum_{t \in S_r} Count(S_g, t)}{|S_r|}$$

On S_r i S_g són els resums de referència i generat respectivament, S_r el vocabulari del resum de referència i $|S_r|$ el nombre de paraules que conté. La funció $Count(text, t)$ conta el nombre de vegades que coincideixen les aparaicions del token t en $text$.

La precisió és el nombre de paraules que s'han generat i són rellevants, és a dir, que apareixen en el resum de referència, és com la fórmula anterior però el denominador el governa el resum generat.

$$Precisió = \frac{\sum_{t \epsilon S_r} Count(S_g, t)}{|S_g|}$$

El F-score és una mètrica que permet combinar els dos valors anteriors, fins i tot es pot fer de manera ponderada mitjançant el valor β.

$$F_\beta\text{-}score = \frac{(1+\beta^2)R(S_g,S_r)P(S_g,S_r)}{R(S_g,S_r)+\beta^2 P(S_g,S_r)}$$

On $R(S_g, S_r)$ i $P(S_g, S_r)$ tornen respectivament el valor del Recall i Precisió sobre el resum generat i el de referència. Per a les nostres avaluacions hem fet servir la la mètrica F-score, amb $\beta = 1$ (F-1), perquè és la que permet tindre una visió més general amb un únic valor.

ROUGE-2

Té el mateix comportament que l'anterior amb l'única diferència que ara es fan els càlculs mitjançant bigrames, és a dir que el vocabulari que es té en compte és el de tots els parells de paraules consecutives possibles que se poden generar. Aquesta mesura és rellevant perquè està demostrat que té una gran vinculació amb la llegibilitat del text, tot i que en aquest treball la llegibilitat es mesura amb altres mètriques.

ROUGE-L

Es considera que de totes, és la més fiable per comparar la semblança entre els dos textos proporcionats. És diferent de les dos anteriors perquè realitza el còmput a partir de la cadena coincident de major longitud.

3.3.2. BERTScore

És una mesura que no aporta realment informació per ella mateixa, en canvi sí que aporta informació a l'hora de comparar dos resums generats i veure quin s'assembla més al de referència. El funcionament consisteix en calcular per a cada token del resum generat la seua semblança amb els tokens del resum de referència, per a això es gasten *embeddings* contextuals que han sigut preentrenats amb BERT (d'ahí el nom de la mesura). Per tant permeten avaluar la distància semàntica, superant els problemes i limitacions d'aquelles mètriques que es basen en n-grames.

3.3.3. Flesch-Kincaid Grade Level (FKGL)

Es tracta d'una mètrica àmpliament utilitzada en el camp de l'educació per mesurar la complexitat d'un text, per tal de valorar el nivell educatiu necessari per poder llegir-lo. S'obté una puntuació que equival al sistema d'avaluació dels Estats Units d'Amèrica, donat que va ser desenvolupada en un programa finançant per la Marina d'EUA. Es calcula mitjançant la següent fórmula:

$$FKGL = 0.39 \cdot \left(\frac{\#\text{paraules}}{\#\text{frases}}\right) + 11.8 \cdot \left(\frac{\#\text{síl·labes}}{\#\text{paraules}}\right) - 15.59$$

3.3.4. Dale-Chall Readability Score (DCRS)

En la mateix línia que FKGL, DCRS mesura la dificultat de comprensió d'un text per al lector. Utilitza una llista de 3000 paraules que podrien entendre estudiants de quart grau

d'Estats Units d'Amèrica, i considera complicada qualsevol paraula que estiga fora d'eixe conjunt. Es calcula mitjançant la següent fórmula:

$$DCRS = 0.1579 \cdot \left(\frac{\text{\#paraules difícils}}{\text{\#paraules}} \cdot 100\right) + 0.0496 \cdot \left(\frac{\text{\#paraules}}{\text{\#frases}}\right)$$

3.3.5. Coleman-Liau Index (CLI)

Novament aquesta mètrica aproxima el grau d'estudis respecte del sistema del EUA necessari per comprendre un text. Aquest introdueix una millora respecte del FKGL perquè en lloc de calcular-ho mitjançant síl·labes ho fa mitjançant caràcters, perquè els ordinadors són més precisos d'eixa manera. Es calcula mitjançant la següent fórmula:

$$CLI = 0.0588 \cdot L - 0.296 \cdot S - 15.8$$

A on la L és el nombre mitjà de paraules per cada 100 paraules i S la mitjana de frases per cada 100 paraules.

3.3.6. LENS

Aquesta és la primera mètrica automàtica supervisada aprensible per avaluar la simplificació de text, demostra una major correlació amb els juis dels humans que les seues alternatives. Per calcular-la, donat un text d'entrada c, l'eixida del sistema corresponent s, i un conjunt de n referències: $R = \{r_1, r_2, ..., r_n\}$, LENS produeix una puntuació $z_{max} = \max_{1 \leq i \leq n}(z_i)$ que maximitza les puntuacions de qualitat z_i de s en relació a cada referència r_i. El model llavors codifica tots els textos en vectors $(\boldsymbol{c, s, r_i})$ utilitzant un encoder basat en Transformers, en el paper mencionen que el que millor funciona és RoBERTa [48]. I ho combinen en una reperesentació intermitja:
$H = [s; r_i; s \odot c; s \odot r_i; |s \neg c|; |s \neg r_i|]$ a on [;] indicat concatenació i $\odot$ és el producte *pointwise*. I eixa H és l'entrada a una xarxa *Feed Forward* per predir eixe valor z_i. Per tant veiem que és un tant complexa de calcular, i de fet ha consumit prou de temps per la intervenció de xarxes neuronals artificials.

3.3.7. AlignScore

Aquesta és una de les mètriques proposades per avaluar la consistència factual entre dos textos, amb l'objectiu de penalitzar les contradiccions, al·lucinaciones, etc. I a diferència de les seues alternatives, pretèn ser una funció de domini general. El càlcul es fa mitjançant un model que gasta, novament, RoBERTa com a model preentrenat i aplica 7 tasques de preentrenament típiques de NLP per a proporcionar comprensió lingüística. Aborda el problema d'este una aproximació d'aliniament, per entrenar un model que estime l'aliniament entre dos textos:

$$AlignScore(o, l) = \underset{j}{mean}\{\max_{i}\{alignment(o'_i, l'_j)\}\}$$

A on $\boldsymbol{o}$ és el context, $\boldsymbol{l}$ és el resum en este cas, $\boldsymbol{\{o'_i\}}$ és el conjunt de chunks en els que es divideix el context, $\boldsymbol{\{l'_j\}}$ el conjunt de frases del resum i $\boldsymbol{alignment(\cdot)}$ és la probabilitat de que el model prediga que estan aliniats ambdós textos. Aquesta no presenta la complexitat de l'anterior, en canvi com també hi ha xarxes neruonals involucrades també ha implicat un alt consum de recursos.

3.3.8. Puntuació final

Com hem vist, estan presents moltes mètriques en l'avaluació, aleshores per tal de poder comparar els models entre ells necessitàvem agrupar totes aquestes mesures en una sola. Per això vam decidir que per a cada aspecte trauríem la mitjana d'entre totes les seues mètriques de manera que tots tres quedaren en l'interval [0,1]. En el cas de la rellevància només calia fer la mitjana aritmètica entre ROUGE-n i BERTScore, la factualitat només la mesurava una mètrica sobre la qual no calia aplicar cap transformació, però sobre la llegibilitat calia aplicar l'equació 3.2 per tal d'aconseguir dues coses: que foren valors normalitzats, entre 0 i 1, i a més a més que correlaren positivament amb la qualitat dels resums. Es pot observar que s'aplica un llindar superior de 20 sobre FKGL, DCRS i CLI mitjançant 3.1 perquè 20 és un valor de llegibilitat suficientment alt per al tipus de textos amb els quals estem treballant. Finalment, per computar la puntuació total de cada model es calcula la mitjana armònica dels tres aspectes.

$$CC_f^z(x) = \frac{z - f(x)|_{[0,z]}}{z} \tag{3.1}$$

$$\begin{aligned} Llegibilitat(x) = (& \\ & CC^{20}_{FKGL}(x) + \\ & CC^{20}_{DCRS}(x) + \\ & CC^{20}_{CLI}(x) + \\ & \frac{LENS(x)}{100} \\ ) \cdot & \frac{1}{4} \end{aligned} \tag{3.2}$$

3.4 Arquitectura Transformer

En la secció d'estat de l'art ja s'han explicat les diferents arquitectures de xarxes neuronals que s'han emprat en els darrers anys per entrenar models de PLN. La nostra aposta ha sigut per la de *Transformers*, que des que van aparèixer en el 2017 van revolucionar totalment aquesta àrea de treball. Es tracta d'una eina que obté molt bons resultats i requereix un temps d'entrenament inferior a les seues competidores.

Ara passarem a veure amb més detall els avantatges i desavantatges. A banda dels bons resultats [15] un altre avantatge destacable d'aquesta arquitectura és que no és seqüencial com CNN o RNN, és a dir que les frases són processades com una unitat en lloc de paraula per paraula la qual cosa permet tindre una visió més contextual del text i a més habilita la paral·lelització d'aquests models per tal d'accelerar els entrenaments. La gran innovació que van introduir va ser l'ús de la *self-attention*, que serveix per saber quines paraules són rellevants dins de la frase i quines paraules són rellevants per a altres paraules. Gràcies a açò va aconseguir-se que estos models pogueren tindre una major comprensió dels textos humans. Un altre avantatge que també hem comentat prèviament és el *transfer-learning*, que en estos models està molt present com comentarem en el següent apartat. Sí que és cert que presenta un inconvenient i és la immensa quantitat de documents que necessita per entrenar-se, però avui en dia això no és un gran problema gràcies a la *World Wide Web* i la gran quantitat de models preentrenats que existeixen. Tot i que sí que pot ser més problemàtic per a llengües més minoritàries que tenen menys documents disponibles, però això s'ha demostrat que no és un problema tan gran, perquè

encara així s'aconsegueixen resultats d'estat de l'art o pròxims a ell també amb llengües com el català amb menor quantitat de documents [49].

3.4.1. Arquitectura Longformer

Si bé l'atenció era un dels punts clau dels Transformers que ha propulsat el seu èxit, també és cert que és una operació molt cara de realitzar (quadràtica amb la longitud de la cadena). Eixa operació és precisament el coll de botella que presenta aquesta arquitectura, la qual cosa ha fet que la major part dels models entrenats es limitaren als 512 tokens d'entrada i eixida que es van plantejar al paper original. Si bé, això en certs contextes pot ser suficient, en altres àmbits com per exemple al que ens enfrontem en aquest treball es queden molt curts, llavors com a resposta a aquestes necessitats van aparèixer arquitectures com els Longformers [50].

El Longformer escala linialment amb la longitud de la seqüència, facilitant el procés de documents que tinguen milers de tokens. Per aconseguir això combina finestres d'atenció local amb atencions globals que han de ser especificades per l'usuari final per poder adaptar-se a la tasca final. Les finestrs d'atenció local ajuden a construir una representació contextual mentre que la global és la que permet construir una representació apropiada de tota la seqüència per portar a terme la predicció. Aquesta arquitectura va trencar amb la tendència que hi havia en el moment, en què es desenvolupaven arquitectures molt complexes per adaptar l'arquitectura a cadascuna de les tasques específiques: *Question Answering*, *Document Classification*... Mentre que el que es planteja en este cas és una arquitectura simple de propòsit general.

En la Figura 3.1 podem veure la comparativa que presenten al paper original de l'evolució del cost temporal i espacial de les seues arquitectures front a les de Transformer a mesura que s'incrementa la longitud de les seqüències. I amb l'arquitectura *Longformer-chunk* es demostra un decrement del cost temporal i sobretot espacial.

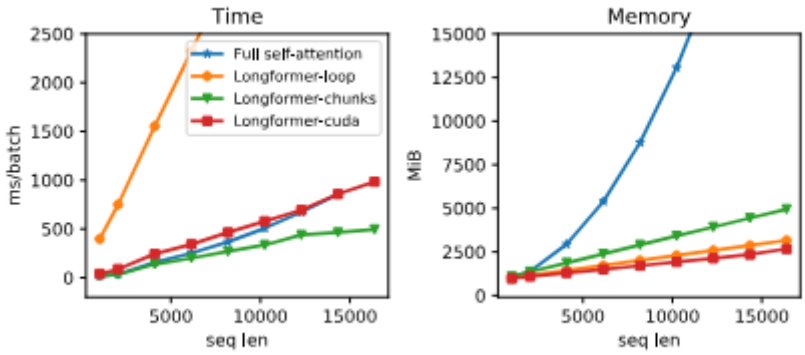


Figura 3.1: Comparativa de l'evolució del cost temporal i espacial en el càlcul de l'atenció segons la longitud de la seqüència per a diferents arquitecures

Per a les tasques seq2seq, ofereixen una variant: LED (Longformer-Encoder-Decoder) que és la que ens interessa a nosaltres perquè fem una tasca de resum automàtic. El que proposen és una arquitectura que gaste l'atenció local+global que ja ha demostrat ser més eficient en l'encoder, de manera que el cost serà linial en l'entrada, però per al decoder mantenen la *full self-attention* estàndard. Per alleugerir l'elevat cost de preentrenar aquest tipus de models el que van fer va ser inicialitzar els paràmetres de LED amb els de BART i mantindre el mateix nombre i mida de les capes. El funcionament de BART l'explicarem en la següent secció.

3.5 Models preentrenats

Com ja hem comentat en l'estat de l'art, en xarxes neuronals existeix una tècnica que és el *transfer-learning* molt útil perquè permet reaprofitar el que ja han après altres models prèviament entrenats. Aquesta tècnica cobra un paper especialment rellevant en el cas dels Transformers, en els quals s'utilitza en gairebé tots els models. La idea és preentrenar un model per a aconseguir un model amb coneixement lingüístic, i llavors podem continuar l'entrenament per especialitzar-lo (*fine-tuning*) en alguna tasca concreta com el resum en el nostre cas, estalviant així una gran quantitat de recursos, perquè precisament el preentrenament és un procés molt costós. En el nostre cas per a tots els entrenaments vam partir d'un model LED preentrenat pel grup d'investigació ELiRF que ha dirigit aquest TFM.

Per a aquesta tasca, s'ha utilitzat un Longformer encoder-decoder (LED) [50] ja que havíem de resumir textos llargs, com el cas dels articles científics. D'aquesta manera podem augmentar la quantitat d'informació disponible al costat del codificador. Hem utilitzat com a punt de partida model base de LED d'AI2[5], disponible públicament al repositori de HuggingFace[51] i es va entrenar contínuament amb dades del domini.

Per a la fase de preentrenament, es va seguir la metodologia d'entrenament utilitzada en el treball *News Abstractive Summarization* (NAS) [49]. La metodologia combina múltiples tasques de preentrenament per incorporar coneixements lingüístics en la fase de preentrenament potenciar l'abstractivitat dels resums elaborats. La incorporació d'aquestes tasques hauria d'ajudar el model a transferir coneixements específics de la tasca de resum i augmentar el rendiment del model després de la fase de *fine-tuning*, tal com va ocórrer al treball NAS original.

Les dades utilitzades per a la fase de preentrenament continu, van ser seleccionades específicament per adaptar el model al domini de recerca biomèdica. Per a això es van recol·lectar textos de diverses fonts: *abstracts* (resums tècnics) de PubMed [52] (17M de mostres), articles de PubMed, articles del dataset `scientific_papers` [6] [53] (240K). I també els articles i resums tècnics proporcionats en la partició d'entrenament de la competició (eLife + PLOS) (29K).

Degut a limitacions d'infraestructura, es va limitar l'entrada del codificador a 4096 tokens. Per la qual cosa, per aprofitar al màxim les dades disponibles, els textos van ser separats en línies gastant una finestra de no més de 4000 paraules. I es van generar submostres que contingueren almenys una nova línia i les finestres es plenaven amb tantes paraules com fora possible. El nombre total de mostres va ser 59M.

Treballar amb LongFormers, com s'ha explicat anteriorment, requereix seleccionar quins tokens van a rebre atenció global addicional a la local. En el treball original [50], els autors recomanaven que el token `[CLS]` continguera atenció global. No obstant, es va hipotetitzar, que afegint més atencions al llarg de l'entrada podia incrementar el rendiment. Per això es van afegir tokens especials (`<sent>`) cada N frases, que tingueren atenció global. El nombre de frases no era una constant, sinó que es col·locava al final de cada N frases amb una longitud total de k paraules. Es van realitzar una sèrie d'experimentacions preliminars que van determinar que els millors resultats s'obtenien amb $k = 20$ paraules de separació.

Aquest preentrenament va ser realitzat en el cluster del VRAIN [7] durant 3 èpoques. Aquest cluster està dotat amb 8 gràfiques NVIDIA A40 amb 48GB de VRAM i va requerir

[5] https://huggingface.co/allenai/led-base-16384
[6] http://tiny.cc/54x2yz/scientific_papers
[7] Institut al qual pertany el grup que ha dirigit aquest treball. https://vrain.upv.es/

d'un mes de preentrenament. Els hiperparàmetres seleccionats van ser: 128 mostres per dispositiu, acumulació de gradient de 4 passes, un learning rate de 5^{-5}, *gradient-checking* i AdamW amb quantificació a 8-bits com a optimitzador.

CAPÍTOL 4
Eines utilitzades

En aquest capítol parlarem de les distintes eines de software i hardware que hem emprat i que han permès portar a terme aquest projecte.

4.1 Software

A continuació, explicarem els diferents recursos de software que hem decidit utilitzar per a aquest treball, juntament amb les raons que han motivat la seua elecció. Començarem amb una visió general del llenguatge Python i després detallarem les diverses llibreries que han estat essencials per dur a terme aquest projecte amb èxit.

4.1.1. Python

Python és un llenguatge interpretat, d'alt nivell i orientat a objectes, que tot i ser més antic que Java, s'ha mantingut relativament discret durant bona part de la seua existència. No obstant això, en els darrers anys, s'ha convertit en un dels llenguatges més utilitzats i amb més projecció de creixement. Això és degut al fet que és un llenguatge bastant senzill per a principiants i molt llegible. Hem optat per utilitzar Python en aquest projecte per diverses raons. És àmpliament preferit per desenvolupadors de ciència de dades i Intel·ligència Artificial, fet que ha generat una gran comunitat de suport i nombrosos fòrums de discussió i llibreries. Aquest últim punt ha sigut crucial, ja que hi ha una àmplia varietat de llibreries disponibles per a tot tipus de tasques, i especialment per a IA, com ara TensorFlow, scikit-learn, Keras, etc. Entre aquestes es troba la llibreria Transformers de Huggingface, que hem utilitzat i que detallarem més endavant. Encara que Python podria semblar una elecció poc adequada perquè no és tan ràpid com altres llenguatges, gràcies a l'ús de certes llibreries implementades en C o Rust i optimitzades per al tractament de grans volums de dades, com la paral·lelització, es converteix en un llenguatge tan potent com els altres.

4.1.2. NLTK

Natural Language Toolkit és un conjunt de recursos que faciliten el processament del llenguatge natural amb Python. Ofereix recursos com llistes de paraules, corpus, i models lingüístics, a més d'altres eines. En el nostre projecte, hem utilitzat NLTK específicament per a subdividir els textos generats pel model en oracions, amb la funció *sent-tokenize* o en paraules amb *word-tokenize*.

4.1.3. NumPy

És una llibreria que ajuda a tractar vectors i matrius. L'hem emprat per facilitar les operacions amb aquests tipus de dades i compatibilitat amb la resta de llibreries.

4.1.4. Scikit-Learn

És una llibreria centrada en aprenentatge automàtic que ofereix eines útils tant en l'entrenament com en l'avaluació dels resultats.

4.1.5. evaluate

Aquesta llibreria s'ha emprat calcular mètriques com BERTscore i ROUGE.

4.1.6. HuggingFace

Una vegada decidit que entrenaríem el nostre model utilitzant l'arquitectura Transformers i Python com a llenguatge de programació, encara calia seleccionar quina llibreria usaríem per dur a terme l'entrenament. HuggingFace ha desenvolupat diverses llibreries especialitzades en l'entrenament i avaluació de models Transformers, fet que ens va portar a escollir-les. Ens referim específicament a dues llibreries:

- **Datasets**: Possibilita gestionar eficientment conjunts de dades de grans dimensions, la qual cosa ens interessa especialment en el nostre cas donat que per a entrenar xarxes neuronals i en particular models Transformers es necessiten un gran nombre de dades. Entre altres avantatges, aquesta llibreria permet guardar les dades en memòria cau, evitant així haver de carregar-les i processar-les contínuament. També suporta el multiprocessament, distribuint les tasques de processament de dades per accelerar el procés. Un problema comú amb conjunts de dades molt grans és que poden no cabre a la memòria RAM. Per solucionar-ho, la llibreria utilitza un mecanisme de mapeig de memòria mitjançant l'estructura *Apache Arrow*.

- **Transformers**: Aquesta és la llibreria fonamental en la qual es sustenta aquest treball, perquè és mitjançant la qual s'ha portat a terme l'entrenament i també el tokenitzat de textos.

Aquestes llibreries simplifiquen considerablement el desenvolupament del codi d'entrenament, però no només aporten comoditat al programador. També estan optimitzades per processar les dades tant en la fase de preprocessament del conjunt de dades com durant l'entrenament. A més, integren optimitzadors com Optuna, que realitzen una cerca prèvia per determinar els millors hiperparàmetres. Si durant l'entrenament es produeix alguna interrupció de la màquina o un altre tipus de problema (una situació que hem experimentat en diverses ocasions durant aquest treball), aquestes llibreries permeten continuar l'entrenament des de l'últim checkpoint guardat, en lloc de començar des de zero, preservant així el progrés acumulat fins al moment. Aquesta funcionalitat és especialment valuosa, atès el temps i els recursos que consumeix l'entrenament d'aquests models. Tot i que presenten la limitació de no permetre modificar l'arquitectura interna dels Transformers, per als objectius d'aquest projecte, les funcionalitats proporcionades per la llibreria han estat més que suficients.

4.1.7. deepspeed

Aquesta llibreria està dissenyada per accelerar els entrenaments en PyTorch, optimitzant l'ús de la memòria i la potència per entrenar models grans de manera distribuïda, aprofitant al màxim el paral·lelisme del hardware. Aquesta característica és especialment beneficiosa per aprofitar un dels grans avantatges de l'arquitectura Transformers, que és la seua capacitat de paral·lelització. La llibreria està integrada per ser utilitzada conjuntament amb la llibreria Transformers de HuggingFace.

4.2 Hardware

Tot i que aquest treball és un projecte de software i d'IA, és certament rellevant parlar de la part del hardware perquè els avanços que s'han fet en aquest camp són fonamentals per possibilitar realment l'entrenament i democratitzar l'accés a la IA, perquè més enllà de que l'entrenament puga ser paral·lelitzat o no i les optimitzacions a nivell de software, les xarxes neuronals o la ciència de dades es basen en realitzar una quantitat substancial d'operacions bàsiques independents. Totes aquestes operaciones es poden accelerar mitjançant operacions matricials. Ahí és on juguen un paper especial les targetes gràfiques, perquè estan especialitzades precisament per al processament d'operacions amb matrius. La diferència entre entrenar un model amb una màquina amb i sense GPU és abismal. Hui en dia Google també ha tret un nou tipus d'unitat de processament que són les TPU [54] que estan encara més especialitzades per a entrenaments d'IA i per tant això permetria accelerar el procés encara més, tot i que no hem pogut utilitzar aquesta tecnologia en el nostre projecte. En la gràfica de la Figura 4.1 podem veure la diferència entre emprar GPUs vs CPUs, i el clar benefici que suposa a nivell temporal utilitzar GPUs per entrenar xarxes neuronals. Aquesta gràfica tot i ser del 2013 continua vigent avui en dia perquè s'ha seguit la mateixa tendència.

Nosaltres en concret hem treballat en la màquina Tardis proporcionada pel grup d'investigació ELiRF, que té les següent característiques tècniques:

- **Processador**: Intel(R) Core(TM) i9-10940X
- **Memòria RAM**: 128 GB DDR4
- **Targetes gràfiques**: 2 targetes gràfiques RTX 3090 amb 24GB de VRAM, encara que per als entrenaments només podíem emprar una de les dos, per a altre tipus d'execucions com els scripts de test sí que vam poder aprofitar les dues gràfiques.
- **Sistema Operatiu**: Ubuntu 22.04.1 LTS
- **Entorn Python**: conda 23.1.0

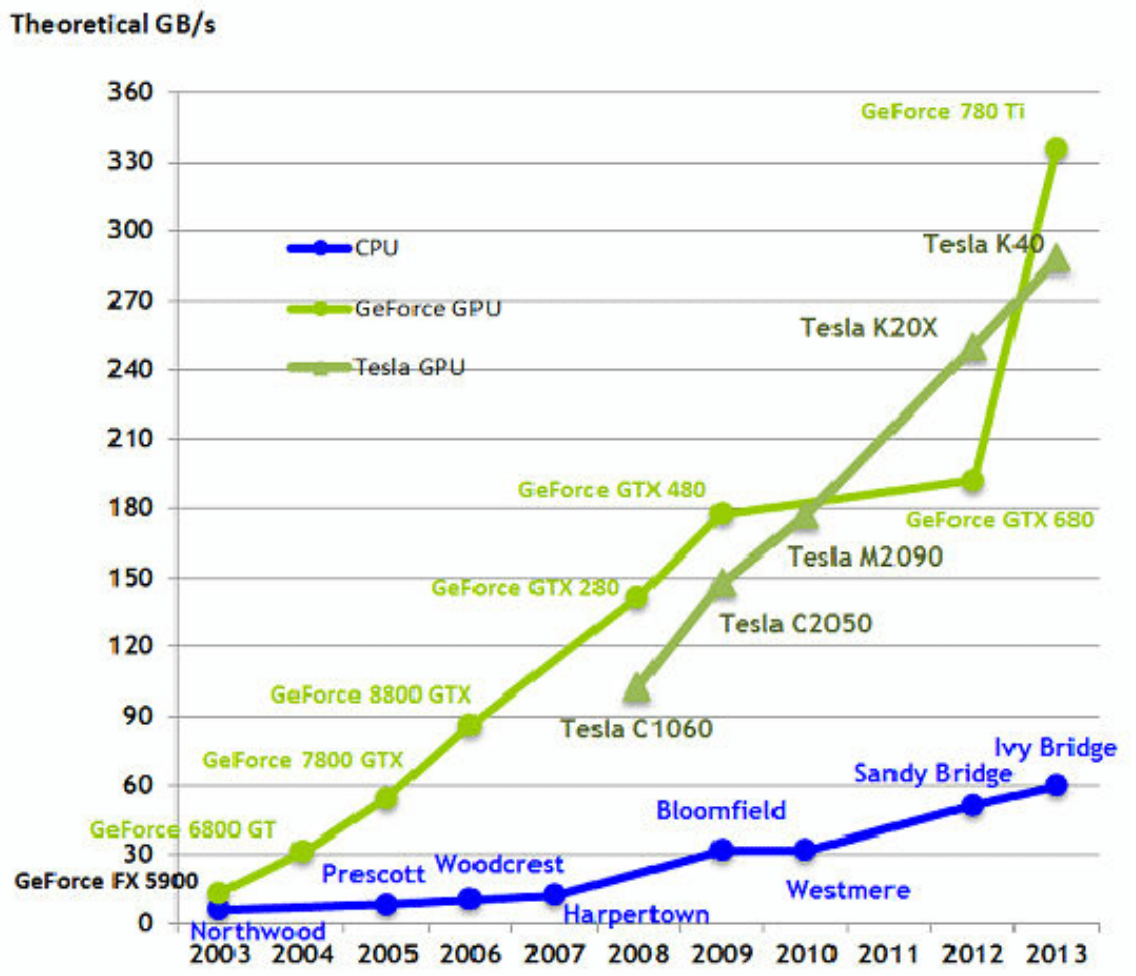


Figura 4.1: Comparativa ús GPU vs CPU per a entrenament de models d'IA [55]

CAPÍTOL 5
Experimentació i Resultats

En aquesta capítol contarem tot el procés que hem seguit al llarg del desenvolupament del projecte, per preparar els models i avaluar-los posteriorment. I també plantejarem les hipòtesis de les quals partíem i que justifiquen cadascun dels models que hem entrenat. Així com una anàlisi dels resultats que discutirem i ens portaran a unes conclusions que confirmaran o desmentiran eixes hipòtesis inicials. Tot i que en la competició es plantejaven dos corpus: eLife i PLOS, per limitacions temporals i de recursos, en aquesta memòria comentem només les experimentacions realitzades sobre eLife.

Vam fer una sèrie d'experimentacions "preliminars" aprofitant els primers entrenaments per tal de seleccionar els millors paràmetres de generació. En concret volíem saber quin nombre mínim i màxim de tokens devia generar el model. Vam provar amb el mínim i màxim d'entre totes les mostres d'entrenament, la mitjana $\pm$ desviació estàndard i el 1r i 3r quartil. Donat que emprar els quartils era l'opció amb la qual obteníem millors resultats i per estalviar temps en les avaluacions de cada model es va decidir que els models generaren els resums amb eixes longituds: [423, 556] tokens.

En les següents seccions analitzarem primerament cadascun dels models entrenats i els seus resultats en validació i seguidament estudiarem els obtinguts en la partició de test pels models més prometedors.

5.1 Model base

5.1.1. Experimentació

El primer que vam fer va ser entrenar un model *baseline* perquè volíem tindre una referència amb la qual poder anar comparant els nostres resultats. Per això vam seleccionar BioBART [56], que era un model que s'ajustava prou bé al domini, donat que havia estat preentrenat amb una gran quantitat de textos biomèdics que són la matèria amb la qual volem treballar. També era apropriat perquè la resta d'entrenaments els fem partint d'un model que s'ha preentrenat amb tècniques inspirades en aquest model.

Per fer *fine-tuning* de BioBART, donat que teníem l'espai limitat a tan sols 512 tokens i qualsevol de les seccions de l'article ocupava més, vam decidir que li donaríem com a entrada simplement l'*abstract* de l'article i a partir d'això havia de generar el resum amb llenguatge simplificat.

5.1.2. Anàlisi de resultats

En la Taula 5.1 podem veure els resultas que ha obtingut aquest model en la partició de validació per als 3 aspectes a avaluar així com per la mitjana armònica entre els tres aspectes. Aquests resultats ens serviran de *baseline* per poder tindre una referència amb que poder comparar el rendiment dels següents models. Amb la informació que tenim pel moment, es pot destacar que té una factualitat prou baixeta mentre que per la resta de criteris obté uns resultats més positius. Pel que fa a la llegibilitat, FKGL i CLI indicarien que els resums generats requereixen un nivell educatiu equivalent al batxillerat. Mentre que DCRS indicaria un nivell universitari, la qual cosa posa de manifest que hi haurà certa discrepància entre les mètriques de llegibilitat, tot i que les conclusions que es poden extraure de cadascuna d'elles és més o menys la mateixa.

Model		Rellevància				Llegibilitat				Factualitat
	Puntuació	**R-1↑**	**R-2↑**	**R-L↑**	**BERTs↑**	**FKGL↓**	**DCRS↓**	**CLI↓**	**LENS↑**	**AlignScore↑**
BioBART	0.5089	0.5005	0.1252	0.4705	0.8573	10.65	9.58	12.25	63.38	0.5384

Taula 5.1: Taula de resultats per a la partició de validació per al model base: BioBART. R = ROUGE F1, BERTs = BERTScore, FKGL = Flesch-Kincaid Grade Level, DCRS = Dale-Chall Readability Score, CLI = Coleman-Liau Index. La fletxa al costat del nom de la métrica indica si és directa (↑) o inversament (↓) proporcional al rendiment del model.

5.2 Models Simplificadors

5.2.1. Experimentació

Una vegada teníem els resultats del *baseline*, vam dividir els entrenaments en dos "famílies" de models: Simplificadors i Resumidors. En primera instància, els models simplificadors, sorgeixen de la idea d'aproximar la tasca des d'una perspectiva més senzilla, en lloc d'haver de resumir tot un article científic i damunt en llenguatge planer, donat que realment el nostre objectiu és obtindre una mena d'*abstract* que no siga tècnic, podem replantejar el problema com si fora una reescriptura del text amb un llenguatge més simple. Però ara la nostra idea era a banda d'entrenar un model base que només rebera el resum tècnic com havíem fet amb BioBART, entrenar altres models en els quals anàrem incrementant eixe context amb informació addicional per veure si d'eixa manera s'aconseguia millorar el rendiment del model. Llavors calia tindre més espai que 512 tokens, per això vam començar a emprar el nostre model LED preentrenat que disposava de 4096 i 1024 tokens en encoder i decoder respectivament. A continuació descriurem cadascun dels experiments que s'han realitzat en aquesta sèrie de models que anomenem Simplificadors.

Simplificador

El primer model d'aquesta família consistia en replicar el *baseline*, és a dir, proporcionant-li al model només l'abstract perquè genere el resum planer. Gràcies a això podem comparar-los amb el nostre primer model de referència i veure quin era el rendiment del nostre model preentrenat. I de pas també serveix per veure una comparativa entre emprar un model transformer "tradicional" com BioBART front a un com Longformer que planteja nous mecanismes d'atenció.

Simplifcador$_{DEC}$

Una de les vies que es plantejava com a prometedores pels organitzadors de la competició era la generació condicionada. Aleshores un dels nostres obejctius era precisament condicionar aquesta eixida proporcionant-li certa informació en el decoder perquè també li puga prestar atenció durant la generació. En aquest cas vam decidir que a banda de passar-li l'*abstract* en l'encoder també ho faríem a l'entrada del decoder.

Simplifcador$_{RAG-k}$

Un altre dels objectius d'aquest treball era portar a terme tasques de RAG, per tal de poder incrementar el context que li proporcionem el model perquè genere el resum de l'article. Aquesta és precismanet una de les raons per les quals hem seleccionat l'arquitectura LED per als nostres models. En concret per incrementar el context en aquests models, com hem explicat en capítols anteriors, realitzem un procés d'extracció de les entitats biomèdiques que apareixen en l'abstract, les intersectem amb les que apareixen en el resum planer i afegim la descripció de cadascun d'aquests termes per tal que el model tinga accés a un vocabulari menys tecnificat i puga disposar del coneixement per reformular aquests termes en el seu resum. De manera que aquest model ha sigut entrenat posant en l'encoder tant l'abstract com les entitats biomèdiques que apareixien en ell juntament amb les seues descripcions. Hem seguit dos estratègies a l'hora de separar la informació de l'abstract i la descripció de les entitats: un simple token que envolte cadascuna de les parts diferenciades ($< TSUM > Abstract < /TSUM >< NER > Entitatsdescrites < /NER >$) i una frase a mode de prompting com es faria en un LLM, en concret, després de l'abstract, com a presentació de les entitats (*"Below are the descriptions of each of the biomedical entities that must appear in the summary."*). La k pot prendre dos valors: 3 i N. N significa que agafem totes les frases de la descripció de cada terme, i 3 és per al cas en què ens quedem només amb les 3 primeres frases, perquè tot i haver incrementat el nombre de tokens disponibles aquest continua sent limitat i amb 3 frases consideràvem que podíem tindre suficient informació de cadascun.

Simplifcador$_{NA}$

Ja hem comentat que els models Longformers tenen mecanismes d'atenció global i local per finestres, aleshores hem d'especificar en quines posicions es posa atenció global. Per poder avaluar l'efecte que realment està tenint sobre el rendiment del model eixe mecanisme d'atenció, en aquest model vam decidir provar una versió *No attention* (NA) que és una modificació del Simplificador original en la qual només posem un token d'atenció en el primer token: CLS. L'atenció global sobre el primer token sí que la mantenim perquè així ho recomanaven en el paper original els autors de Longformer [50].

5.2.2. Anàlisi de resultats

En aquest apartat comentarem els resultats obtinguts per la família de models Simplificadors que es troben en la Taula 5.2, a més ara ja podem realitzar una anàlisi comparativa respecte al model de referència.

Simplificador

El primer que vam fer va ser replicar el model baseline però gastant com a model preentrenat el LED que hem descrit anteriorment. Aquest és el model que millor ha funcionat

de la seua família i obté un rendiment comparable al del baseline, tot i que es queda una miqueta per baix. Fent una anàlisi més exhaustiva, ha empitjorat en les mètriques de rellevància (excepte en ROUGE-2 que ha pujat una miqueta) i en totes les de llegibilitat, mentre que la factualitat l'ha pujat un parell de punts.

Simplifcador$_{DEC}$

Per a aquest experiment provem a condicionar la generació proporcionat-li l'abstract també a l'entrada del decoder, i el que obtenim és una baixada molt gran del rendiment: 9 punts de la puntuació final. Aquesta baixada ve justificada per la rellevància i la llegibilitat, especialment destacable la baixada de 46 punts en LENS. Mentre que anant a contracorrent amb tota la resta de mètriques, la factualitat s'incrementa molt considerablement: 21 punts. La justificació d'este comportament pot trobar-se en que durant la generació ara es para tant atenció creuada com autoatenció a un text molt tècnic. Llavors continua generant amb l'estil d'eixe text fent-lo menys llegibile, a més el "distrau" del seu objectiu final que és aconseguir un resum planer, i per això baixa la rellevància; però com està tenint més en compte que en el model anterior un fragment de l'article original, es manté més coherent a aquest i per tant puja la factualitat.

Simplifcador$_{RAG-k}$

Es van aplicar tècniques de RAG amb l'objectiu de proporcionar-li més context al model i que disposant de més informació generara uns resums més acurats i més llegibles. Dels diferents tipus de RAG que hem provat, ha funcionat millor mantindre tota la descripció de cada terme en lloc de tallar-lo a les 3 primeres frases i també ha funcionat millor el fet d'emprar prompting i donar-li una explicació de la separació que s'estava produint entre l'abstract i les entitats descrites, en lloc d'emprar simplement un token separador que no aportava cap detall al model. No obstant, aquests models no han aconseguit l'objectiu perseguit, donat que tots ells han funcionat pitjor que el Simplifcador base, el millor que hem pogut obtindre ha sigut un 0.5028 (mig punt menys que el base) amb Simplificador$_{RAG-N-prompt}$. Fer servir RAG empitjora totes les mètriques respecte al Simplifcaodr base, però novament la factualitat s'aconsegueix millorar, de fet qualsevol canvi que fem sobre l'encoder o el decoder aconsegueixen millorar la factualitat respecte al Simplificador base i al baseline. En els models que gasten l'aproximació amb token separador també ha aconseguit millorar la FKGL, mentre que la resta de mètriques de llegibilitat baixen, la qual cosa mostra que hi ha encara més discrepància de la que havíem comentat inicialment entre les mètriques de llegibilitat perquè fins i tot pot millorar una mentre baixen la resta.

Simplifcador$_{NA}$

En aquest model el que pretenïem no era millorar els resultats, sino ser capaços de quantificar la importància d'emprar els tokens d'atenció global. I el que hem comprovat és que efectivament si no gastàvem més que un token d'atenció global al principi, s'empitjoraven els resultats a excepció de la factualitat. No obstant la diferència és bastant subtil, tan sols 3 dècimes en la puntuació final, perquè en aquest cas particular tots els resums tenen menys de 1024 tokens per tant caben perfectament en la finestra local i llavors encara que no gastem atenció global, es pot construir l'*embedding* de cada paraula tenint en compte tot el context. Malgrat no ser necessari sí que és beneficiós el fet d'haver posat un major nombre de tokens d'atenció distribuits al llarg de l'abstract front a només posar-ne un.

Model		Rellevància				Llegibilitat				Factualitat
	Puntuació	R-1↑	R-2↑	R-L↑	BERTs↑	FKGL↓	DCRS↓	CLI↓	LENS↑	AlignScore↑
Simplificador	**0.5078**	**0.499**	**0.1281**	**0.4673**	**0.8535**	11.61	**9.66**	**12.44**	**62.07**	0.5596
Simplificador$_{DEC}$	0.4173	0.4139	0.0814	0.3846	0.8281	14.3	11.44	15.05	16.72	**0.769**
Simplificador$_{RAG-N-prompt}$	0.5051	0.4875	0.1192	0.4583	0.8491	11.83	9.87	12.79	55.65	0.6001
Simplificador$_{RAG-N-token}$	0.5002	0.4853	0.1187	0.4567	0.8492	11.2	9.92	12.74	56.88	0.5648
Simplificador$_{RAG-3-prompt}$	0.5028	0.4872	0.1177	0.4563	0.8491	11.89	9.92	12.83	54.78	0.5988
Simplificador$_{RAG-3-token}$	0.4968	0.4827	0.1173	0.4531	0.8483	**11.19**	9.97	12.86	55.47	0.5629
Simplificador$_{NA}$	0.5041	0.4941	0.1272	0.4647	0.8504	11.52	9.91	12.81	58.35	0.5719

Taula 5.2: Taula de resultats per a la partició de validació de la família de models "Simplificadors". R = ROUGE F1, BERTs = BERTScore, FKGL = Flesch-Kincaid Grade Level, DCRS = Dale-Chall Readability Score, CLI = Coleman-Liau Index. La fletxa al costat del nom de la métrica indica si és directa (↑) o inversament (↓) proporcional al rendiment del model.

5.3 Models Resumidors

5.3.1. Experimentació

L'altra família de models que presentem són els "Resumidors" que són models en els quals portem a terme la tasca de resumir l'article per se. En este cas fem una selecció de les seccions que consideràvem que podien aportar una informació que fora de major interés per al nostre objectiu. Com ja hem explicat anteriorment vam triar les seccions d'Introducció i Discussions perquè serien en les que s'explicaren de manera més succinta allò que es detalla en l'article. A més a més serien les que emprarien un llenguatge més simplificat i comprensible pel públic general.

Resumidor I

Aquest model rep l'Abstract, la Introducció i les Discussions, en eixe ordre. Perquè vam considerar que l'abstract podia ser el que continguera la informació més important, i a més, com hem explicat en la família de Simplificadors, l'objectiu final és aconseguir una mena d'abstract però amb llenguatge planer, llavors no només el vam afegir sinó que el vam col·locar el primer, per seguir l'estructura natural d'un article però sobretot perquè com les tres seccions superan els 4096 tokens alguna part de la informació s'anava a perdre, aleshores d'aquesta manera asseguràvem que no es perguera la informació que fins el moment consideràvem de major rellevància. També cal destacar, que cadascuna de les seccions estava envoltada per tokens especials: $< TSUM > Abstract < /TSUM >< INTRO > Introduccio < /INTRO >< DISC > Discussio < /DISC >$. I per tal de què cadascuna de les seccions queden més separades a ulls del model, a l'hora de col·locar les atencions globals, seguirem el procés que ja s'ha detallat de col·locar a l'inici d'una frase després d'una seqüència de frases que en total tinguen almenys acumularen 20 paraules, però al principi de cada secció també es col·locarà un nou token d'atenció i es resetejara el conteig.

Resumidor I$_{DEC}$

Seguint la mateixa línia que el Simplificaodr, vam replicar el Resumidor I però ara proporcionant-li l'abstract a l'inici del decoder.

Resumidor I$_{NA}$

De la mateixa manera que havíem fet en la família dels models Simplificadors, en els Resumidors també volíem fer un estudi de què succeia si deixàvem d'aprofitar els meca-

nismes d'atenció global, perquè a més en aquest tipus de model podia tindre un major efecte donat que ara hi ha una major quantitat d'informació en l'encoder i s'ocupen diverses finestres, per tant sí que es podia beneficiar més de la transmissió d'informació entre finestres locals que en el cas dels Simplificadors.

Resumidor II

Donat que en totes les proves que havíem fet fins el moment havia estat present l'abstract en l'entrada perquè la intuició ens deia que era la part més important, vam voler provar un model en el qual no rebera aquesta secció, simplement li aplegara la Introducció i Discussions.

Resumidor II$_{DEC}$

Al model anterior també li hem afegit l'abstract com a entrada del decoder, per tal de mantindre la consistència amb la resta de models en els quals s'han provat les versions amb i sense informació en el decoder. Però especialment perquè ja havíem vist en altres casos que afegint aquesta informació es milloraven substancialment els resultats de la factualitat, i podia ser que en aquest cas en què no es veiera l'abstract en l'encoder, proporcionar-li-ho com a informació nova en el decoder podia millorar uns quants punts a diferència d'en la resta de casos.

5.3.2. Anàlisi de resultats

En aquest apartat comentarem els resultats obtinguts per la família de models Resumidors que es troben en la Taula 5.3.

Resumidor I

Ja amb la primera versió del primer model Resumidor podem veure que obtenim millors resultats dels que havíem obtingut amb qualsevol dels models Simplificadors, de manera que era el millor model fins el moment. La superació als Simplificadors ve donada exclusivament per la factualitat, en la resta de mesures ha empitjorat. És lògic que la factualitat haja millorat donat que abans al model només li proporcionàvem l'abstract, aleshores a penes tenia informació de la que apareguera en l'article, mentre que en els models Resumidors li donem més text original de l'article a través de la Introducció i les Discussions. Si ho comparem amb BioBART, la puntuació final també ha sigut millor però de les mètriques només l'ha superat en la ROUGE-2 i AlignScore, la resta també les empitjora.

Resumidor I$_{NA}$

Els resultats obtinguts demostren que la baixada en puntuació total és exactament la mateixa que en els Simplificadors: 0.3, és a dir que en qualsevol cas el fet de no fer servir els mecanismes d'atenció global disponibles suposen una pèrdua de qualitat en els resums generats, però no és molt significativa. La baixada respecte del Resumidor I es produeix en quasi totes les mètriques a excepció d'aquelles que ja hem vist en altres ocasions que pugen quan la resta baixen: ROUGE-2, FKGL i CLI a on sí que millora lleugerament.

Resumidor I_{DEC}

Les conclusions que podem extraure per a la versió del Resumidor I afegint l'abstract en el decoder són les mateixes que havíem extret en la família dels Simplificadors, i és que empitjora totes les mètriques, en especial LENS però a canvi puja uns 11 punts en AlignScore.

Resumidor II

Aquest model va sorprendre, perquè es va entrenar sense esperar grans resultats d'ell i en canvi va ser el que van obtindre millors resultats, és el millor model de la família dels Resumidors però a més obté els millors resultats en comparació a tots els models entrenats per a ROUGE-1, ROUGE-2, ROUGE-L, DCRS i CLI (empata amb BioBART que era el millor fins el moment). Tot això ho fa a canvi d'empitjorar la factualitat respecte de Resumidor I, la qual cosa té sentit perquè hem deixat de proporcionar-li l'abstract, que en experiments anteriors havíem pogut constatar que ensenyant-li'l més vegades: en encoder i decoder, millorava significativament l'AlignScore. No obstant, la seua factualitat segueix sent millor que la dels Simplificadors i el baseline. La conclusió general que es pot extreure amb aquest model, és que tot i que havíem apostat per la importància de l'abstract, aquest és fonamental per a la factualitat, però en canvi per a la rellevància i la llegibilitat confón al model.

Resumidor II $_{DEC}$

Finalment al Resumidor II tampoc li ha beneficiat l'ús de l'abstract en el decoder, sí que permet millorar al voltant d'11 punts la factualitat però en la puntuació final pateix una baixada de rendiment fins i tot més exagerada que en la resta de casos, donat que en aquest cas ha sigut de 6.4 punts.

Model		Rellevància				Llegibilitat				Factualitat
	Puntuació	**R-1↑**	**R-2↑**	**R-L↑**	**BERTs↑**	**FKGL↓**	**DCRS↓**	**CLI↓**	**LENS↑**	**AlignScore↑**
Resumidor I	0.5136	0.4981	0.1259	0.4659	0.8533	12.22	9.93	12.85	57.26	0.6297
Resumidor I_{DEC}	0.4911	0.4893	0.1204	0.4586	0.8461	12.98	10.44	14.22	38.54	**0.7359**
Resumidor I_{NA}	0.5106	0.495	0.1286	0.4642	0.8513	11.62	9.98	12.67	55.44	0.609
Resumidor II	**0.5223**	**0.5069**	**0.1317**	**0.4760**	**0.8552**	**11.63**	**9.54**	**12.25**	**61.7**	0.6032
Resumidor II_{DEC}	0.4663	0.4652	0.1044	0.4346	0.842	13.76	10.83	14.61	35.8	0.7111

Taula 5.3: Taula de resultats per a la partició de validació de la família de models "Resumidors". R = ROUGE F1, BERTs = BERTScore, FKGL = Flesch-Kincaid Grade Level, DCRS = Dale-Chall Readability Score, CLI = Coleman-Liau Index. La fletxa al costat del nom de la métrica indica si és directa (↑) o inversament (↓) proporcional al rendiment del model.

5.4 Resultats test

5.4.1. Experimentació

En aquesta darrera secció del capítol d'Experimentació i Resultats comentarem els resultats que han obtingut per a la partició oficial d'avaluació de la competició aquells models que semblaven més prometedors en la fase de validació.

Per al test, a banda d'agafar aquells models que millor funcionaren, també vam introduir un model de Rankeig. Donat que els models de *Machine Learning* tenen una gran variabilitat i en el cas dels models generatius encara més, creíem que simplement quedarnos amb la primera mostra que presentara un model podia ser un poc pobre i podíem no

estar explotant els models al màxim. De manera que es va entrenar un model que seleccionara per a cada mostra quin era el millor resum d'entre un conjunt. Aquest model va ser entrenat per Vicent Ahuir, doctorand del grup d'investigació a càrrec d'aquest treball. En concret, es tracta d'un model de regressió que donat un resum genera tres eixides: les puntuacions estimades per cadascun dels tres aspectes, dels quals apliquem la mitjana armònica per triar el millor model, és a dir que és un model que intenta aproximar el valor de la Puntuació que hem plantejat en cadascuna de les Taules d'aquest document. Per fer aquest entrenament es disposava de poca varietat de dades, per la qual cosa es van emprar LLMs per fer *Data Augmentation* mitjançant 5 tasques: reescriure i completar tant el resum tècnic com el simplificat així com completar el simplificat amb informació del tècnic. Per això es van emprar 4 LLMs estat de l'art: Vicuna 13b [57], Alpaca 13b [58], OpenChat 7.5b [59], i Llama2 13b [60]. En concret, es tracta del model anomenat Rankeig, per al qual vam generar amb el model Simplificador 10 resums per cada mostra i vam deixar que fora el model de rankeig el que seleccionara per cadascuna quin era el millor resum.

5.4.2. Anàlisi de resultats

En la Taula 5.4 podem trobar els resultats obtinguts en la partició de test d'aquells models que en validació es va observar que eren més prometedors. El que podem observar és novament el mateix que havíem extret de la partició de validació: BioBART funciona lleugerament milor que el model Simplificador i el Resumdior II és el que millor funciona. El fet d'haver afegit el model de Rankeig, a penes ha millorat els resultats que ja obtenia Simplificador agafant simplement la primera mostra generada, però en l'àmbit d'una competició sempre és interessant incrementar les puntuacions tant com siga possible.

Model		**Rellevància**				**Llegibilitat**				**Factualitat**
	Puntuació	**R-1↑**	**R-2↑**	**R-L↑**	**BERTs↑**	**FKGL↓**	**DCRS↓**	**CLI↓**	**LENS↑**	**AlignScore↑**
BioBART	0.5143	**0.506**	0.1247	**0.4746**	**0.8594**	**10.68**	**9.54**	**12.33**	**66.35**	0.5458
Simplificador	0.5111	0.4921	0.1195	0.4388	0.8519	12.51	9.84	13.0	57.56	**0.6432**
Rankeig	0.5114	0.4929	0.1198	0.4401	0.852	12.49	9.82	12.97	57.53	0.6417
Resumidor II	**0.52**	0.504	**0.1264**	0.4714	0.8562	11.85	9.59	12.56	62.61	0.606

Taula 5.4: Taula de resultats per a la partició de test. R = ROUGE F1, BERTs = BERTScore, FKGL = Flesch-Kincaid Grade Level, DCRS = Dale-Chall Readability Score, CLI = Coleman-Liau Index. La fletxa al costat del nom de la métrica indica si és directa (↑) o inversament (↓) proporcional al rendiment del model.

CAPÍTOL 6
Conclusions

Al llarg d'aquesta memòria s'ha fet un repàs detallat de l'evolució i situació actual del PLN, més en concret de la generació automàtica de resums, s'han explicat les distintes mètriques utilitzades per avaluar els sistemes de resum automàtic, explicat l'arquitectura en la qual estan basats els nostres models i s'han descrit els corpus amb els quals han estat entrenats. A més a més, s'ha explicat minuciosament el procés que es va seguir durant els entrenaments i els resultats que van obtindre's.

En el treball s'han presentat un conjunt de models que podem distribuir en dos famílies: Resumidor i Simplificador. El primer s'encarregava de resumir l'article en un abstract amb un llenguatge simplificat i el segon reescrivia l'abstract d'una manera simplificada. Tots els models s'han *fine-tunejat* a partir d'un model Longformer que havia continuat preentrenant-se amb documents d'àmbit biomèdic. D'aquesta manera podíem fer front a l'extensa longitud dels documents amb els que treballàvem i cobríem un dels objectius d'aquest projecte. Dins dels models presentats s'han seguit aproximacions aliniades amb els objectius plantejats, per una banda per intentar condicionar la generació mitjançant la introducció de l'abstract en l'entrada del decoder, la qual cosa incrementava substancialment la factualitat però reduia també significativament la puntuació final del model. D'altra banda s'han fet servir mecanismes de RAG per ampliar el context que se li proporcionava al model i no ha permès millorar els resultats en cap dels aspectes, no obstant ens ha permès extraure una informació prou rellevant i és que funcionava millor separar els apartats de l'encoder mitjançant llenguatge natural (prompting) front a l'ús d'un simple token especial sense semàntica. També s'ha experimentat amb els mecanismes d'atenció global que ofereixen aquests models Longformers demostrant que és beneficiós introduir-los per tal què el model no es centrara només en la finestra local a l'hora de construir els embeddings sinó que tinguera en compte altres tokens ubicats en altres finestres per tindre un major context. Finalment també s'ha presentat un model de rankeig que permet seleccionar d'entre diversos resums generats per un model el que s'espera que tinga una major puntuació.

Seguidament es presenten les principals dificultats que han anat apareixent durant la realització d'aquest projecte, com els hem enfrontat a elles i les possibles línies de treball que es poden realitzar-se partint d'aquest treball.

6.1 Reptes i solucions

En aquesta secció exposarem algunes de les dificultats i reptes més destacats que hem anat abordant al llarg de la realització d'aquest projecte i les solucions que hem plantejat:

- **Temps d'execució:** Un dels problemes al que hem de fer front cada vegada que entrenem un model d'Aprenentatge Automàtica, en especial quan es tracta d'arquitectures amb tants paràmetres com *Transformers*, perquè això implica necessitar una major quantitat de mostres d'entrenament. El fet d'emprar més mostres d'entrenament es tradueix bàsicament en un major consum energètic i temporal. No obstant, com ja hem explicat prèviament al Capítol 4 estos temps s'han pogut reduir perquè traem profit dels avenços en el hardware i de llibreries com *deepspeed* que permeten traure-li el màxim profit a les targetes gràfiques. Encara que fins ara hem parlat només dels temps d'execució dels entrenaments, tampoc és trivial el temps dedicat a les avaluacions, especialment si tenim en compte que algunes de les mètriques emprades gasten models d'intel·ligència artificial.

- **Caigudes de les màquines:** Un altre problema que vam afrontar, però que estava fora del nostre control, va ser que les màquines es van desconnectar en diverses ocasions per manteniment, talls de llum, etc. Això no només ens va impedir treballar durant els dies que les màquines estaven inactives, sinó que també va provocar la pèrdua de part del progrés en els entrenaments i validacions en curs. Tot i que la llibreria de *HuggingFace* permet reprendre l'entrenament des d'un checkpoint guardat, evitant la pèrdua total del progrés i haver de començar de nou, les desconnexions de Tardis van afectar l'entrenament. Això es deu al fet que els checkpoints es guardaven a cada època, de manera que el progrés dins d'una època es perdia. Això també va impactar en les avaluacions, ja que alguns *scripts* necessitaven molt de temps per executar-se.

- **Entrenar un *Longformer*:** Aquest punt realment no ha sigut un problema, sinó que es tracta més bé d'un repte, que ha sigut haver d'enfrontar-se a entrenar un nou tipus de model, que si ve manté l'arquitectura de Transformers té una sèrie de peculiaritats com longituds distintes d'encoder i decoder o finestres d'atenció local que tenen en compte atencions globals i llavors s'ha de gestionar en quines posiciones col·locar eixos tokens d'atenció global tan valuosos, tasca que no és genys menyspreable.

- **Mida del model:** Vinculat al punt anterior, els models Longformers ocupen prou d'espai i a més necessiten més memòria perquè les entrades que reben són de major longitud que un Transformer estàndard. Això ha generat problemes que en certs casos han pogut solucionar-se baixant la mida del batch, però en altres casos en els que incrementàvem notablement la quantitat de posicions a on es parava atenció global, no va ser prou amb això sinó que va caldre aplicar tècniques com *gradient checkpointing*, emprar AdamW amb quantificació a 8 bits, i com amb tot això el *batch_size* encara estava limitat a 1, vam aplicar acumulació del gradient per "simular" un batch més gran. També cal destacar que treballar amb Longformers implica fer una gestió de l'atenció global que no és trivial.

- **Varietat de mètriques:** Un repte al qual d'havíem de fer front per participar en aquesta competició, és que s'avaluaven diversos aspectes, cadascun amb vàries mètriques havent un total de 10, aleshores això era bastant problemàtic perquè s'havia de trobar alguna manera de combinar les puntuacions de cada mètrica per poder comparar els resultats dels diferents models. Però hi ha mètriques que eren directament proporcionals al rendiment, altres inversamnet proporcionals, i no totes es movien en els mateixos intervals de valors. A més a més, com hem vist al llarg de la memòria, hi ha un model millor o pitjor segons la mètrica que tries, perquè rellevància i legibilitat sí que pareixien més coherents, però la factualitat moltes vegades s'incrementava molt significativament quan baixaven els altres dos aspcetes. I fins i tot amb mètriques dins del mateix aspecte a avaluar s'ha observat que quan una

s'incrementava una altra baixava, sense mantindre sempre coherència entre elles, la qual cosa dificultava l'anàlisi que es podia fer.

6.2 Treball futur

Encara que s'han realitzat bastants experimentacions al llarg d'aquest projecte, el temps disponible era limitat, llavors ara comentarem algunes línies en les quals es podria haver continuat avançant i que podrien enriquir la investigació feta.

- **Experimentar amb noves formes de RAG:** Tot i que l'aproximació que s'ha seguit de RAG amb la descripció de les entitats presents en el resum tècnic i planer no ha permès millorar els resultats, es considera que és una via molt prometedora en aquest tipus d'àmbit que queda demostrat tant per la literatura com pels resultats en la competició de l'any passat. Per tant, aquest treball podria extendre's mitjançant l'aplicació d'altres tècniques de RAG com per exemple els grafs de coneixement.
- **Generació condicionada:** Una de les propostes d'aquest treball era condicionar la generació com ja es va fer en el TFG [61]. Aquesta ha sigut explorada a través d'afegir l'abstract en l'entrada del decoder, però seria interessant condicionar altres aspectes com per exemple el nivel d'abstractivitat o el grau de longitud del resum.
- **Continuar explorant les atencions del Longformer:** Un dels punts clau d'aquest treball ha sigut l'ús de models Longformer que a permeten controlar en quines parts del text col·locar atenció global, més enllà de l'atenció local que es para en cada finestra de 1024 tokens. En els nostres models vam emprar l'aproximació de col·locar un a l'inici de cada frase després d'unes quantes frases que acumularen almenys vint paraules i també al començament de cada nova secció. I també s'ha estudiat el que succeïa en llevar tots els tokens d'atenció a excepció del primer, però aquesta podria fer-se un estudi més detallat de com es beneficia el model d'afegir més o menys tokens d'atenció i d'afegir-los en certs punts estratègics que poden no ser els que nosaltres hem proposat.

Bibliografia

[1] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia Tools and Applications*, vol. 82, 07 2022.

[2] F. Yvon, "Recent advances in deep learning for nlp," *LISN — CNRS and Université Paris Saclay Data Science "Summer" School Palaiseau*.

[3] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[4] S. Ruder, M. E. Peters, S. Swayamdipta, and T. Wolf, "Transfer learning in natural language processing," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, (Minneapolis, Minnesota), pp. 15–18, Association for Computational Linguistics, June 2019.

[5] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.

[6] S. Ruder, J. Bingel, I. Augenstein, and A. Søgaard, "Latent multi-task architecture learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 4822–4829, 2019.

[7] "Natural Language Processing Tasks." Consultat en https://towardsdatascience.com/natural-language-processing-tasks-3278907702f3.

[8] Z. Xue, R. Li, and M. Li, "Recent progress in conversational ai," 2022.

[9] K. M.Tarwani and S. Edem, "Survey on recurrent neural network in natural language processing," *International Journal of Engineering Trends and Technology*, vol. 48, pp. 301–304, 06 2017.

[10] "Comprenent els problemes amb RNN." Consultat en https://www.analyticsvidhya.com/blog/2021/07/lets-understand-the-problems-with-recurrent-neural-networks/.

[11] J. S. Sepp Hochreiter, "Long short-term memory," 1997.

[12] "Explicació de les LSTM." Consultat en https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9.

[13] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[14] D. Hu, "An introductory survey on attention mechanisms in NLP problems," *CoRR*, vol. abs/1811.05544, 2018.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[16] H. Saggion, "Automatic summarization: An overview," vol. 13, pp. 63–81, 06 2008.

[17] I. Mani, "Recent developments in text summarization," in *Proceedings of the Tenth International Conference on Information and Knowledge Management*, CIKM '01, (New York, NY, USA), p. 529–531, Association for Computing Machinery, 2001.

[18] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. Trippe, J. Gutiérrez, and K. Kochut, "Text summarization techniques: A brief survey," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 8, pp. 397–405, 07 2017.

[19] J. Zhang, Y. Zhao, M. Saleh, and P. J. Liu, "Pegasus: Pre-training with extracted gap-sentences for abstractive summarization," in *Proceedings of the 37th International Conference on Machine Learning*, ICML'20, JMLR.org, 2020.

[20] A. M. Rush, S. Chopra, and J. Weston, "A neural attention model for abstractive sentence summarization," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 379–389, Association for Computational Linguistics, Sept. 2015.

[21] C. Chang, C. Huang, and J. Y. Hsu, "A hybrid word-character model for abstractive summarization," *CoRR*, vol. abs/1802.09968, 2018.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Proceedings of Workshop at ICLR*, vol. 2013, 01 2013.

[23] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," pp. 1532–1543, Oct. 2014.

[24] Y.-C. Chen and M. Bansal, "Fast abstractive summarization with reinforce-selected sentence rewriting," pp. 675–686, July 2018.

[25] J. Wu, "Introduction to convolutional neural networks," *National Key Lab for Novel Software Technology. Nanjing University. China*, vol. 5, no. 23, p. 495, 2017.

[26] S. Chopra, M. Auli, and A. M. Rush, "Abstractive sentence summarization with attentive recurrent neural networks," pp. 93–98, June 2016.

[27] R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulçehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," pp. 280–290, Aug. 2016.

[28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[29] P. Nema, M. M. Khapra, A. Laha, and B. Ravindran, "Diversity driven attention model for query-based abstractive summarization," pp. 1063–1072, July 2017.

[30] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," 2017.

[31] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," 2016.

[32] R. Paulus, C. Xiong, and R. Socher, "A deep reinforced model for abstractive summarization," 2017.

[33] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 7871–7880, Association for Computational Linguistics, July 2020.

[34] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, (Minneapolis, Minnesota), pp. 4171–4186, Association for Computational Linguistics, June 2019.

[35] Q. Liu, M. J. Kusner, and P. Blunsom, "A survey on contextual embeddings," *CoRR*, vol. abs/2003.07278, 2020.

[36] P. Gupta and M. Jaggi, "Obtaining better static word embeddings using contextual embedding models," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, (Online), pp. 5241–5253, Association for Computational Linguistics, Aug. 2021.

[37] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, (Red Hook, NY, USA), Curran Associates Inc., 2020.

[38] Y. Hu and Y. Lu, "Rag and rau: A survey on retrieval-augmented language model in natural language processing," 2024.

[39] T. Goldsack, Z. Zhang, C. Lin, and C. Scarton, "Making science simple: Corpora for the lay summarisation of scientific literature," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, (Abu Dhabi, United Arab Emirates), pp. 10589–10604, Association for Computational Linguistics, Dec. 2022.

[40] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, (Barcelona, Spain), pp. 74–81, Association for Computational Linguistics, July 2004.

[41] T. Zhang*, V. Kishore*, F. Wu*, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *International Conference on Learning Representations*, 2020.

[42] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers, and B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel," *Research Branch report*, vol. 8, p. 75, 1975.

[43] E. Dale and J. S. Chall, "A formula for predicting readability," *Educational research bulletin*, vol. 27, no. 1, pp. 11–28, 1948.

[44] M. Coleman and T. L. Liau, "A computer readability formula designed for machine scoring," *Journal of applied psychology*, vol. 60, no. 2, p. 283, 1975.

[45] M. Maddela, Y. Dou, D. Heineman, and W. Xu, "LENS: A learnable evaluation metric for text simplification," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 16383–16408, Association for Computational Linguistics, July 2023.

[46] Y. Zha, Y. Yang, R. Li, and Z. Hu, "AlignScore: Evaluating factual consistency with a unified alignment function," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (A. Rogers, J. Boyd-Graber, and N. Okazaki, eds.), (Toronto, Canada), pp. 11328–11348, Association for Computational Linguistics, July 2023.

[47] M. Sung, M. Jeong, Y. Choi, D. Kim, J. Lee, and J. Kang, "Bern2: an advanced neural biomedical namedentity recognition and normalization tool," 2022.

[48] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[49] V. Ahuir, L. F. Hurtado, J. A. Gonzalez, and E. Segarra, "NASca and NASes: Two Monolingual Pre-Trained Models for Abstractive Summarization in Catalan and Spanish," *Applied Sciences*, vol. 11, no. 21, 2021.

[50] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[51] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, (Online), pp. 38–45, Association for Computational Linguistics, Oct. 2020.

[52] National Center for Biotechnology Information (NCBI), "Pubmed: A resource by the national center for biotechnology information." https://pubmed.ncbi.nlm.nih.gov/, 2024.

[53] A. Cohan, F. Dernoncourt, D. S. Kim, T. Bui, S. Kim, W. Chang, and N. Goharian, "A discourse-aware attention model for abstractive summarization of long documents," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018.

[54] N. P. Jouppi, C. Young, N. Patil, D. Patterson, G. Agrawal, R. Bajwa, S. Bates, S. Bhatia, N. Boden, A. Borchers, *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proceedings of the 44th annual international symposium on computer architecture*, pp. 1–12, 2017.

[55] V. Thambawita, R. Ragel, and D. Elkaduwe, "To use or not to use: Graphics processing units for pattern matching algorithms," 12 2014.

[56] H. Yuan, Z. Yuan, R. Gan, J. Zhang, Y. Xie, and S. Yu, "BioBART: Pretraining and evaluation of a biomedical generative language model," in *Proceedings of the 21st Workshop on Biomedical Language Processing* (D. Demner-Fushman, K. B. Cohen, S. Ananiadou, and J. Tsujii, eds.), (Dublin, Ireland), pp. 97–109, Association for Computational Linguistics, May 2022.

[57] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," March 2023.

[58] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford alpaca: An instruction-following llama model." https://github.com/tatsu-lab/stanford_alpaca, 2023.

[59] G. Wang, S. Cheng, X. Zhan, X. Li, S. Song, and Y. Liu, "Openchat: Advancing open-source language models with mixed-quality data," *arXiv preprint arXiv:2309.11235*, 2023.

[60] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open foundation and fine-tuned chat models," 2023.

[61] D. Torres Bertomeu, "Condicionant l'estil en la generació de resums abstractius de notícies," *Universitat Politècnica de València.*, 2023.

APÈNDIX A

Objectius de desenvolupament sostenible

En 2015 l'ONU va aprovar una agenda d'objectius de desenvolupament sostenible (ODS) perquè els diferents països adoptaren mesures per millorar la societat. Eixa agenda pot classificar-se en la llista de 17 objectius fonamentals representats en la Figura A.1 que involucra àmbits com l'educació de qualitat, l'eficiència energètica i industrial o protecció del medi ambient. A continuació presentem la vinculació que té el treball desenvolupat amb alguns dels punts dels ODS:

Figura A.1: Llista dels 17 objectius de desenvolupament sostenible plantejats per l'ONU a l'agenda per al 2030

- **Reducció de les desigualts** (Objectiu 10): Com es plantejava precisament en el punt de la motivació, aquest treball té una particularitat i és que entrena un model per al resum de textos simplificats. De manera que estem democratitzant l'accés a la informació en un àmbit d'interés públic com és la salut i la medicina, per tal que tots aquells que tinguen interés en la matèria o vulguen aprofitar la immensa quantitat de coneixement que hi ha en tots els articles biomèdics que tenim disponibles i que en bona mesura s'han finançat amb fons públics, puguen fer-ho, sense que siga necessari ser experts en la matèria i amb independència del seu nivell educatiu. Perquè hi ha molta gent que per qüestions econòmiques no ha pogut permetre's

una educació superior, i això no hauria de ser una raó per discriminar-los i no permetre'ls l'accés a certes àrees de coneixement.

- **Consum i producció responsables** (Objectiu 12) i **Acció climàtica** (Objectiu 13): Tot i que puga semblar contradictori aquest objectiu perquè l'entrenament d'intel·ligències artificils resulta en un consum ingent de recursos, especialment energètics, l'ús de tècniques com el *transfer-learning* que permet que entrenem models a partir de models preentrenats, que en PLN és la fase més costosa en la qual se fa un entrenament més genèric del model, se li ensenya una llengua, etc. Com podem realitzar aquest preentrenament una sola vegada i després entrenar múltiples models especialitzats en diferents tasques a partir d'aquest, la part més costosa de l'entrenament s'ha produit una sola vegada i s'ha pogut aprofitar per molts projectes. De fet és la manera de la qual se sol treballar amb *Transformers* i la que seguirem en concret al llarg d'aquest projecte.

- **Educació de qualitat** (Objectiu 4): Hi ha molts països en els quals encara hui en dia hi ha una manca d'accés a la informació per la manca de recursos econòmics disponibles per a invertir en educació. Els avanços en el camp del PLN contribueixen a pal·liar aquest problema, perquè faciliten l'accés a la informació. Per exemple amb el resum automàtic de textos biomèdics, que permeten millorar la recerca d'informació abreujant textos de manera que capten de manera correcta la temàtica i idees principals del text original.

Objetius de Desenvolupament Sostenible	**Alt**	**Mitjà**	**Baix**	**No procedeix**
ODS 1. **Fi de la pobresa.**				X
ODS 2. **Fam zero.**				X
ODS 3. **Salut i benestar.**		X		
ODS 4. **Educació de qualitat.**	X			
ODS 5. **Igualtat de gènere.**				X
ODS 6. **Aigua neta i sanejament.**				X
ODS 7. **Energia assequible i no contaminant.**				X
ODS 8. **Treball digne y creixement econòmic.**		X		
ODS 9. **Indústria, innovació i infraestructures.**		X		
ODS 10. **Reducció de les desigualtats.**	X			
ODS 11. **Ciutats i comunitats sostenibles.**				X
ODS 12. **Producció i consum responsables.**	X			
ODS 13. **Acció pel clima.**	X			
ODS 14. **Vida submarina.**				X
ODS 15. **Vida d'ecosistemes terrestres.**				X
ODS 16. **Pau, justícia i institucions sólides.**				X
ODS 17. **Aliances per aconseguir objetius.**				X

APÈNDIX B

Exemple d'article i resum amb llenguatge simplificat

A continuació podem trobar un exemple d'un article d'eLife per poder visualitzar el llenguatge que es gasta en l'article original i quin format tenen els resums de referència. Per extensió mostrem només les primeres frases de les seccions de l'article que hem fet servir per als entrenaments: Abstract tècnic, Introducció i Discussions.

Abstract tècnic The virus SARS-CoV-2 can exploit biological vulnerabilities (e.g. host proteins) in susceptible hosts that predispose to the development of severe COVID-19. To identify host proteins that may contribute to the risk of severe COVID-19, we undertook proteome-wide genetic colocalisation tests, and polygenic (pan) and cisMendelian randomisation analyses leveraging publicly available protein and COVID-19 datasets [...]

Introducció At the current time, the coronavirus disease 2019 (COVID-19) pandemic is implicated in the deaths of more than 4 million people worldwide (Dong et al., 2020). Although effective vaccines have been developed to substantially reduce mortality and morbidity due to severe COVID-19, the emergence of mutated strains of the SARS-CoV-2 virus has challenged the effectiveness of existing vaccines and raised the urgency of identifying alternate therapeutic pathways to target the virus (Tegally, 2020; Erik et al., 2020 ; Collier et al., 2021). Nevertheless, it is likely that the mutated strains of SARS-CoV-2 will continue to exploit the same vulnerable host biology to bind onto and infect cells and, in susceptible individuals, evade immune defences and promote the excessive host inflammatory response that is characteristic of severe COVID-19 (Gordon et al., 2020a). Therefore, the identification of host proteins that play roles in COVID-19 susceptibility and severity remains crucial to the development of therapeutics as host protein mechanisms are independent of genomic mutations in the virus. An improved understanding of these therapeutically relevant virus-host pathways may also be important in combating viruses beyond SARS-CoV-2 (Perrin-Cocon et al., 2020) [...]

Discussions Our systematic proteome-wide MR and genetic colocalisation analysis supported several previously proposed proteins and suggested additional clinically actionable targets for COVID-19 (Table 1). Of particular note, we provided pan- and cis-MR evidence with strong genetic colocalisation support for the ABO signal for most COVID-19 phenotypes. Although the ABO protein itself is not clinically actionable, the ABO signal was linked to plasma concentrations of several clinically tractable targets. We demonstrated that the CD209 protein we had found to have the strongest association with this ABO signal has a direct interaction with the SARS-CoV-2 spike protein, providing further evidence for a plausible mechanism. Our analyses also supported the role of soluble IL-6R in hospitalised COVID-19, with evidence from pan- and cis-MR analyses but limited evidence of genetic colocalisation with hospitalised COVID-19 but supported by the recent COVID-19 clinical trials of tocilizumab (which is partially mimicked by the IL-6R instrument used in the present study). Using a proteome-wide 'colocalisation-first' approach, we recapitulated previously reported targets (e.g. OAS1) and uncovered additional novel proteins that may play causal roles in COVID-19 susceptibility (THBS3), or severity (FAS) [...]

Resum planer Individuals who become infected with the virus that causes COVID-19 can experience a wide variety of symptoms. These can range from no symptoms or minor symptoms to severe illness and death. Key demographic factors, such as age, gender and race, are known to affect how susceptible an individual is to infection. However, molecular factors, such as unique gene mutations and gene expression levels can also have a major impact on patient responses by affecting the levels of proteins in the body [...]