UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica

Condicionant l'estil en la generació de resums abstractius de notícies

Treball Fi de Grau

Grau en Enginyeria Informàtica

AUTOR/A: Torres Bertomeu, Diego

Tutor/a: Hurtado Oliver, Lluis Felip

Cotutor/a: Segarra Soriano, Encarnación

Cotutor/a: Ahuir Esteve, Vicent

CURS ACADÈMIC: 2022/2023

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

etsinf

Escola Tècnica Superior d'Enginyeria Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Condicionant l'estil en la generació de resums abstractius de notícies

TREBALL FI DE GRAU

Grau en Enginyeria Informàtica

Autor: Diego Torres Bertomeu

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

Curs 2022-2023

Resum

La tasca de resum automàtic de textos s'ha abordat en la literatura mitjançant enfocaments extractius i abstractius. Els enfocaments extractius composen els resums seleccionant oracions o paraules directament dels documents, mentre que els enfocaments abstractius construeixen els resums reescrivint les principals oracions dels documents, més semblants als que solen generar els humans. Els sistemes principals de resum de l'estat de l'art són abstractius i estan basats en xarxes neuronals profundes (*Transformers* principalment). Hi ha un corpus de notícies periodístiques en català i en espanyol, DACSA, que proporciona una col·lecció de parells (article, resum). El corpus conté notícies procedents de diferents fonts periodístiques. En aquest treball es proposa construir models de resum abstractius per a aquest corpus, que incorporen en la fase d'ajust (*fine-tuning*) a partir d'un model de llenguatge preentrenat, a més de les dades de la tasca de resum, la informació de la font periodística associada al parell (article, resum). Aquest entrenament ens ajudarà a estudiar si hi ha diferents estils a les fonts a l'hora de redactar els resums. Així mateix, ens permetrà estudiar l'efecte d'indicar al sistema una font en concret a l'hora de resumir un article determinat, que pot coincidir o no amb la font original de l'article.

Paraules clau: resum de textos periodístics, resum abstractiu, Transformers, català, espanyol

Resumen

La tarea de resumen automático de textos se ha abordado en la literatura mediante enfoques extractivos y abstractivos. Los enfoques extractivos componen los resúmenes seleccionando oraciones o palabras directamente de los documentos, mientras que los enfoques abstractivos construyen los resúmenes reescribiendo las principales oraciones de los documentos, más similares a los que suelen generar los humanos. Los sistemas principales de resumen del estado del arte son abstractivos y están basados en redes neuronales profundas (Transformers principalmente). Hay un corpus de noticias periodísticas en catalán y en español, DACSA, que proporciona una colección de pares (artículo, resumen). El corpus contiene noticias procedentes de diferentes fuentes periodísticas. En este trabajo se propone construir modelos de resumen abstractivos para este corpus, que incorporen en la fase de ajuste (fine-tuning) a partir de un modelo de lenguaje preentrenado, además de los datos de la tarea de resumen, la información de la fuente periodística asociada al par (artículo, resumen). Este entrenamiento nos ayudará a estudiar si hay diferentes estilos en las fuentes a la hora de redactar los resúmenes. Asimismo, nos permitirá estudiar el efecto de indicar al sistema una fuente en concreto a la hora de resumir un artículo determinado, que puede coincidir o no con la fuente original del artículo.

Palabras clave: resumen de textos periodísticos, resumen abstractivo, Transformers, catalán, español

Abstract

The task of automatic text summarization has been approached in the literature through extractive and abstractive approaches. Extractive approaches compose the summaries by selecting sentences or words directly from the documents, while abstractive approaches construct the summaries by rewriting the main sentences of the documents, more similar to those generated by humans. The main state-of-the-art summary systems are abstractive and based on deep neural networks (primarily Transformers). There is a corpus of

journalistic news in Catalan and Spanish, DACSA, which provides a collection of pairs (article, summary). The corpus contains news from different journalistic sources. In this work, we propose to build abstractive summary models for this corpus, which incorporate in the fine-tuning phase from a pre-trained language model, in addition to the summary task data, the information of the journalistic source associated with the pair (article, summary). This training will help us to study if there are different styles in the sources when writing the summaries. Likewise, it will allow us to study the effect of indicating to the system a specific source when summarizing a specific article, which may or may not coincide with the original source of the article.

Key words: summary of journalistic texts, abstractive summary, Transformers, Catalan, Spanish

Índex

Índex de figures

Índex de taules

CAPÍTOL 1
Introducció

Amb la popularització d'Internet al llarg dels anys, ha crescut exponencialment la quantitat de documents que hi ha disponibles de pràcticament qualsevol àmbit que vulguem buscar. Front a aquesta quantitat abrumadora de documents que hi ha al nostre abast, no basta amb confiar en la capacitat dels buscadors per filtrar només aquells que ens puguen ser profitosos, perquè encara ens tornen un nombre de resultats que fan que siga inassumible analitzar quins tenen una informació d'utilitat en un temps raonable.

En aquestes situacions és quan entren en joc els models de resum automàtic, que permeten alliberar-nos de la càrrega d'haver de llegir documents sencers i poder llegir un text molt més breu que siga capaç de mantindre les idees principals del text original, sense que haja hagut d'haver una persona darrere encarregada de resumir-ho. Açò resulta especialment útil en àmbits com els textos de recerca, els informes mèdics o els textos periodístics.

En aquest treball ens focalitzarem en l'entrenament de models per a resum abstractiu de notícies en català. Per a la qual cosa disposem del corpus DACSA [1] que ha estat desenvolupat pel grup d'investigació ELiRF [2] i està conformat per parells de notícia-resum extrets de pàgines web de diaris tant en castellà com en català. Concretament ens centrarem en analitzar l'efecte que pot tindre sobre la qualitat i abstractivitat dels resums generats afegir-li una nova tasca de classificació al model. A més a més s'ha proposat l'entrenament d'un model bilingüe.

1.1 Motivació

Hui en dia podria afirmar que la Intel·ligència Artificial està experimentant la seua època daurada, ja no només per la quantitat d'investigadors que hi ha en les universitats i en entitats privades col·laborant per fer avenços en aquest camp, sinó perquè és una realitat que ha passat a estar present en tots els àmbits de les nostres vides. Mai s'havia parlat tant de la informàtica ni de la Intel·ligència Artificial com en aquests darrers anys i setmana sí setmana també, s'obrin els telediaris amb una notícia d'aquest tipus. Açò és positiu perquè està fent que la societat s'acoste a la IA, tinga interés per comprendre-la i està obrint debats ètics molt interessants i de vital importància per fer avançar la IA però sempre dins d'uns límits que permeten fer-la avançar al mateix ritme que la societat perquè vagen a una.

Nosaltres en aquest treball ens centrarem en concret en un camp molt interessant de la IA que és el Processament del Llenguatge Natural (PLN). Que és precisament la branca que més està en voga actualment, probablement perquè siga la que més aconsegueix arribar a sorprendre a la població o és quan ha començat a veure-la com una amenaça; perquè

podem acceptar que una màquina siga capaç de reconèxier una cara, un cotxe... Però en el moment en el que veiem que la màquina comença a adquirir capacitats que fins el moment consideràvem exclusives dels humans: la comunicació, és a dir, que no són només capaços d'entendre'ns, sinó també de generar textos amb els que poden comunicar-se amb nosaltres com a iguals. Això és el que ha passat amb ChatGPT i que ha generat tant de revol a nivell mundial i tothom ha començat a parlar-hi. De fet en aquesta gràfica 1.1 podem comprovar l'evolució en l'interés en la IA en els darrers cinc anys a través de les cerques en Google d'aquesta temàtica, justetament en el punt que està marcat, del 4 al 10 de desembre, va ser quan va eixir ChatGPT i podem veure que des d'ahí ha començat a tindre un enorme creixement l'interès de la població per aquesta tecnologia.

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

A més de perquè siga un tema de total actualitat, la motivació d'aquest treball ve donada perquè vivim en una societat que té més accés que mai a la informació però ho volem tot reduit a un instant, volem estímuls, tenim menor capacitat de concentració a causa de les noves tecnologies. Llavors és necessari, si volem captar l'atenció dels usuaris, per exemple cap a les nostres notícies, que tinguen un estímul clar, breu i concís; a més a més del que s'ha comentat amb anterioritat, per a investigadors, metges, o gent en general que ha de llegir molts documents, és fonamental disposar d'un resum de qualitat que permeta destriar ràpidament aquells documents que són d'utilitat i aquells que no per tal d'estalviar un temps valuosíssim. En tots estos punts entren en joc intel·ligències artificials que siguen capaces de generar automàticament uns resums amb estil humà que capturen les idees fonamentals del text original, redactar-ho d'una manera comprensible i didàctica i alhora atraure l'atenció del lector. Açò podria tindre un efecte molt positiu en la societat, perquè cada vegada malgrat l'enorme quantitat de documents que estan al nostre abast, som una societat profundament desinformada, on cada vegada la gent deixa d'informar-se pels mitjans tradicionals i no se li dona tanta importància a la veracitat de les fonts i al treball periodístic de veritat, llavors estos resums automàtics poden servir per tornar a atraure al públic jove cap a fonts d'informació fiables i riguroses.

I més enllà de parlar del model simplement com a generador de resums, en el nostre cas en particular, es tracta d'un resumidor de textos en català, que té especial interés, perquè sembla que les noves tecnologies estiguen pensades per als parlants de les llengües més parlades com pot ser l'anglès, perquè hi ha més documents disponibles per poder entrenar les intel·ligències artificials, aleshores els models tendeixen a construir-se amb corpus en anglès, i al final sembla que hi haja un consens que fa que la majòria del desenvolupament aconseguit en aquesta àrea siga en l'anglès oblidant la resta de llengües. I és important que els avanços en la ciència i la tecnologia permeten millorar la vida de les persones i fer avançar les societats, però d'una manera democràctica i que no hi ha haja ciutadans ni parlants de primera i altres de segona.

1.2 Objectius

L'objectiu d'aquest projecte consisteix en generar per una banda un model que siga capaç de resumir i classificar notícies en català i per altra un model bilingüe capaç de generar resums en català i castellà a partir de notícies en català o castellà. Més concretament els objectius són:

1. **Utilitzar xarxes neuronals per a la generació de resums abstractius:** Un dels objectius clars que té este treball és entrenar un model de resum de notícies en català seguint un estil abstractiu, per a la qual cosa es disposa del corpus DACSA.

2. **Utilitzar xarxes neuronals per a la classificació de notícies:** Volem afegir a eixos models una segona tasca: que siga capaç de detectar la font a la qual pertanyen les notícies, i amb això es pretén estudiar:

   - Si realment en els textos periodístics hi ha suficients empremtes estilístiques o ideològiques com per permetre al model distingir unes fonts d'altres. I quin és el rendiment del model en aquesta tasca.
   - Detectar si hi ha marques estilístiques presents també en el resum segons la font de la qual procedisca la notícia que fa que hi haja diferències estilístiques a l'hora de generar els resums segons de quina font procedisca la notícia.
   - Si té algun efecte sobre la generació del resum quan li donem al model a banda de la pròpia notícia, la font d'algun diari (siga la font correcta o no). És a dir vol estudiar-se si donar-li la font correcta fa que millore la qualitat dels resums, si donant-li una font que no siga la correcta resumeix amb un estil diferent o fa cas del seu propi criteri? És realment capaç de resumir amb estils diferents segons la font que detecte?
   - I per últim també volem analitzar si el fet de focalitzar la seua atenció en una segona tasca: detectar la font, ha fet que siga capaç d'aprendre la forma de resum de cadascuna de les fonts i per tant genere uns resums que obtinguen una major puntuació perquè siguen més semblants als de referència. O si aquest nou model, té també millors resultats a nivell d'abstractivitat, és a dir que genera uns resums més aproximats al que seria un resum de producció humana.

3. **Utilitzar xarxes neuronals per entrenar un model bilingüe:** Per finalitzar es va voler entrenar un model que partira d'un model preentrenat només en català perquè fora capaç de resumir en català o castellà notícies originalment en català, i també de resumir en castellà notícies originalment en castellà. Aquest darrer objectiu és realment una ampliació del treball i pretén determinar si un model preentrenat inicialment en català, era capaç d'a partir d'una entrada en un idioma, generar un resum en una altra llengua que a més no havia vist mai fins a la fase d'ajust.

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.** Este capítol té per objectiu descriure l'àmbit en el que s'ubica el treball, la justificació de la seua existència i els objectius que es pretenen aconseguir amb ell.

- **Capítol 2, Estat de l'art.** La finalitat del segon capítol és fonamentalment oferir un marc teòric per tal de situar al lector, perquè entenga alguns conceptes teòrics essencials per a comprendre el projecte que s'està desenvolupant i els termes que s'utilitzen al llarg de la memòria, així com explicar l'evolució i els treballs més recents en el processament del llenguatge natural com un camp específic dins de l'aprenentatge automàtic i més en concret la generació automàtica de resums.
- **Capítol 3, Metodologies i sistemes utilitzats.** En el tercer capítol es descriuen els diferents corpus emprats al llarg d'aquest projecte, es discuteixen els diferents sistemes de representació de textos, s'expliquen les mètriques triades per a realitzar l'avaluació dels models, així com l'arquitectura dels models BART que s'utilitzen com a punt de partida.
- **Capítol 4, Eines utilitzades.** En aquest capítol es descriuen les distintes eines software i hardware que s'han emprat i que han possibilitat la realització d'aquest projecte.
- **Capítol 5, Experimentació i resultats.** En el cinquè capítol ens centrem en el treball que s'ha desenvolupat durant la realització del treball fi de grau, 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.** Finalment tanquem amb un capítol de conclusions on recapitulem què és el que s'ha aconseguit amb aquest treball, a quines dificultats ens hem enfrontat i en quines línies es podria continuar ampliant aquest treball.

1.4 Vinculació amb els estudis cursats

Per a la realització d'aquest treball han sigut necessaris una sèrie de coneixements adquirits al llarg de tota la carrera, però en especial d'aquells assolits durant l'especialització en la branca de Computació, que podria afirmar-se que és aquella que guarda una major vinculació amb la Intel·ligència Artificial. Han sigut necessaris els coneixements apresos al llarg de totes les assignatures de programació en general: IIP, PRG i EDA. Dins de la branca cal destacar percepció (PER) i aprenentatge automàtic (APR) perquè són aquelles en les quals vam aprendre diferents algorismes i tècniques de Intel·ligència Artificial, en APR en concret vam aprendre sobre xarxes neuronals; estos coneixementos han resultat fonamentals per poder entendre el funcionament i els conceptes sobre els quals subjauen els Transformers, amb els quals s'han entrenat els nostres models. I possiblement l'assignatura que ha tingut major relació amb aquest treball, ha sigut Sistemes d'emmagatzemament i recuperació d'informació (SAR), perquè en ella se van explicar tècniques per representar, recuperar i tractar informació, així com conceptes vinculats amb el PLN, i de fet és impartida pels mateixos docents que tutoritzen este TFG.

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 el podem dividir en dos blocs [3]:

- **Natural Language Understanding (NLU)**: Centrat en aconseguir comprendre un text i extraure informació a partir d'ell, per exemple per analitzar el sentiments dels tuits. Es tracta d'un procés complex perquè com comentarem més endavant en aquesta secció, quan expliquem el mecanisme d'atenció dels Transformers per entendre una paraula no es prou amb entendre-la aïllada, ni només amb la informació de les paraules anteriors, sinó que és necessari prestar atenció a certes parts del discurs que tindran una major influència sobre ella que altres.
- **Natural Language Generation (NLG)**: En este cas ja no es tracta de comprendre el text, sinó de generar-lo, formen part del procés decidir el que es vol transmetre, organitzar-lo i triar les paraules adequades per expressar-lo. Aquest procés és mes complex, especialment perquè normalment va lligat a l'anterior, primer rep un text d'entrada que ha de comprendre i a partir d'ahí generar un text d'eixida, per exemple un resum o una traducció.

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: [4]

- **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 [5], 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 [6].
- **Mecanismes d'atenció [7]**: 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 [8]**: 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 [3, 9]:

- **Resum automàtic**: El model rep com a entrada un text i genera un text d'eixida que resumeix el que ha rebut. Pot seguir una tècnica extractiva (crea el resum a partir de paraules que conformen el text) o abstractiva (utilitza les seues "pròpies" paraules).
- **Traducció automàtica**: Donat un text en una llengua genera com a eixida eixe mateix text en un altre idioma.
- **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.
- **Reconeixement d'entitat nomenada (NER)**: Detecta paraules que identifiquen persones, llocs, empreses, etc.
- **IA conversacional [10]:** Implica construir models que són capaços d'entendre i generar un diàleg que simula l'humà.

I ara passarem a fer un recorregut evolutiu de les tècniques i arquitectures que han anat emprant-se en el PLN [3]. 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. El primer que es va utilitzar van ser les *Convolutional Neuronal Networks* (CNN): són un tipus de xarxa que funciona molt bé per a imatges i en general tipus de dades que estiguen estructurades. La idea bàsica que hi ha darrere és que es disposen diferents capes de manera que les primeres seran capaces de detectar formes més senzilles com formes geomètriques i conforme anem avançant seran capaces de reconèixer estructures més complexes com cares, objectes, flors, etc. En la Figura 2.1 podem veure quin aspecte tindria una arquitectura típica d'una xarxa CNN.

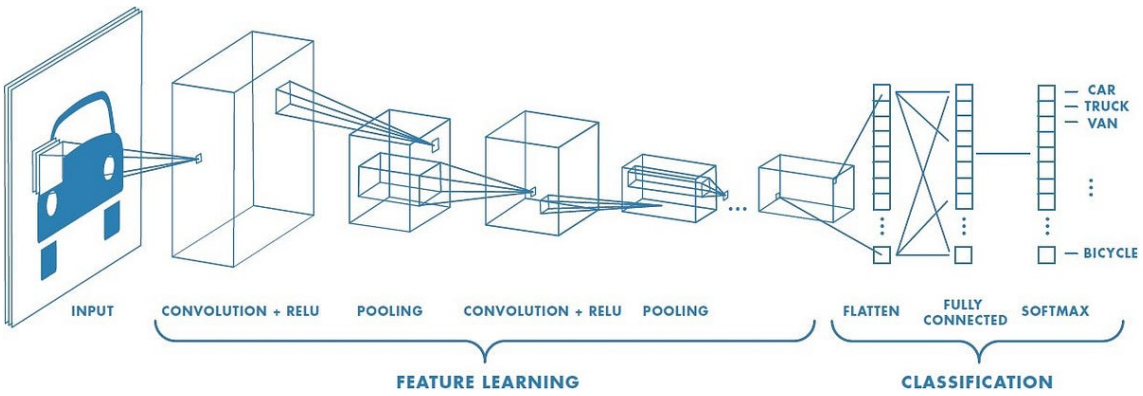

Figura 2.1: Exemple d'arquitectura d'una xarxa CNN

La seua funcionalitat bàsica se pot resumir en l'ús de 4 tipus de capes: [11]

- **Capa d'entrada**: Conté els valors dels píxels de la imatge proporcionada com entrada

- **Capa convolucional**: Consisteix en aplicar un filtre, que tindrà dimensionalitat NxN inferior a la matriu d'entrada, al llarg de totes les submatrius que se puguen formar de NxN sobre la matriu original, mitjançant un producte escalar com pot observar-se en la Figura 2.2. Llavors aplicant estos filtres reduïm la dimensionalitat, però ho fem en amplada i alçada de la matriu, però augmenta en profunditat, perquè no s'aplica un sol filtre, sinó que s'aplica un kernel: conjunt de filtres. Llavors si apliquem un kernel de 32 filtres obtindrem una profunditat de 32 matrius. Cadascun d'estos filtres són la base de les CNN, perquè precisament són els que permeten a la xarxa detectar patrons i són els valors d'eixos kernels els que ha d'aprendre durant l'entrenament.

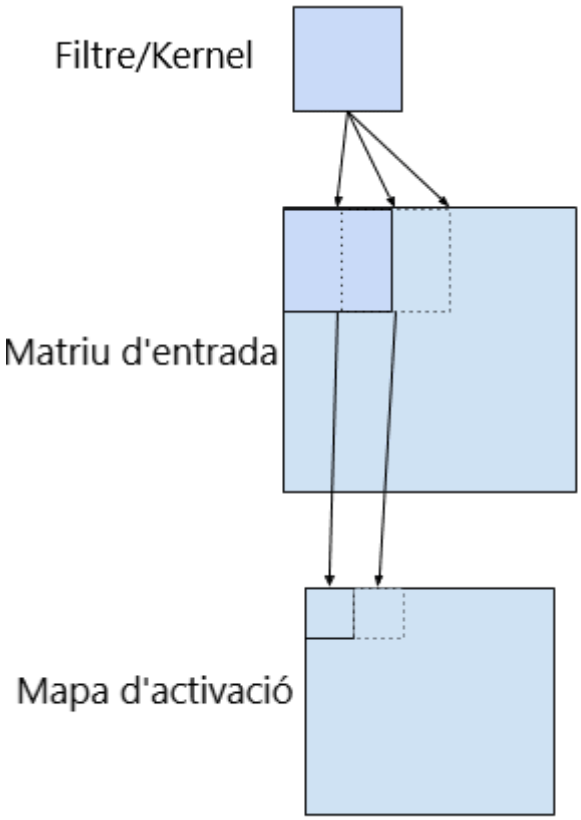

Figura 2.2: Aplicació d'un filtre sobre una matriu

- **Capa de pooling**: Esta és la vertaderament encarregada de reduir la dimensionalitat de la representació, de manera que redueix el nombre de pesos a aprendre. S'aplica sobre el resultat de la capa anterior, per exemple si ha obtingut un mapa d'activació de mida WxWxD i apliquem una matriu de pooling de FxS, eixe mapa d'activació se quedarà amb una mida W'xW'xD on W' = $\frac{W-F}{S}$ + 1. Els dos tipus principals de pooling són max-pooling [Figura 2.3] i average-pooling.

- **Capes totalment connectades**: Són les encarregades de portar a terme una classificació, després d'haver aplicat un conjunt de capes de filtratge + pooling. En general es recomana l'ús de ReLU, però també hi ha altres funcions no linials que poden aplicar-se com una Sigmoid o una Tanh.

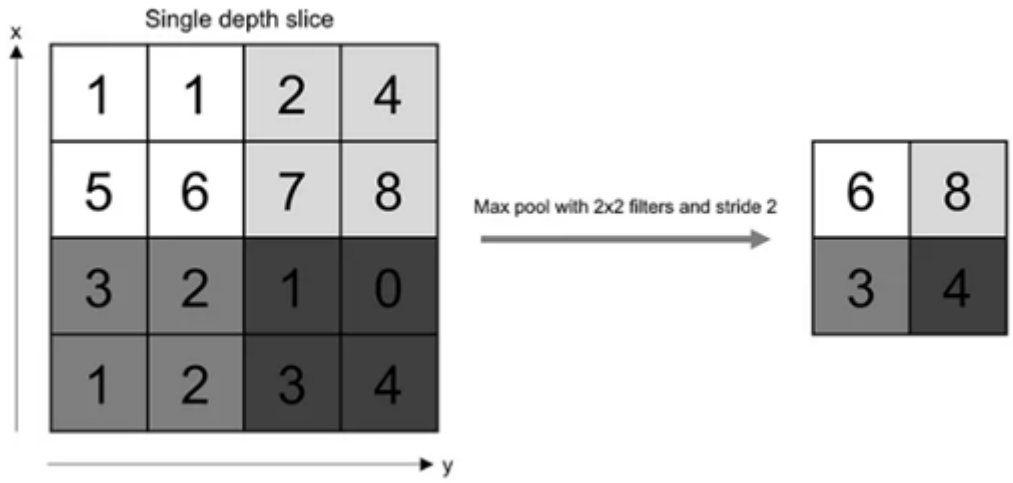

Figura 2.3: Matriu resultant d'una capa de max-pooling(Font: O'Reilly Media)

Per aplicar les CNN al PLN la idea seria continuar fent servir una matriu però ara cadascuna de les files seria la representació d'una paraula mitjançant algun sistema de codificació com *word-embedding* [12] que veurem més endavant. Llavors els filtres que s'apliquen tindran la mateixa amplada que la matriu original, i serà l'altura el que variarà segons la mida de la finestra de paraules que vulguem considerar (n-grames). Típicament serà de 2 o 5 perquè a partir de 5-grames comença a tindre un cost computacional excessiu. I encara que semble poc intuïtiu i molt limitat perquè només considerar 5 paraules veïnes, segons quina llengua pot no ser suficient per entendre el context perquè les paraules que donen sentit a una paraula estiguen molt allunyades d'aquesta, en la pràctica han obtingut resultats prou bons [13] i el gran avantatge és la seua rapidesa.

Precisament eixa necessitat que comentàvem de conèixer el context per poder entendre el significat de les paraules ens porten a una nova arquitectura, les *Recursive Neural Network* (RNN) [14]. 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: [15]

- 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) [16]. 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*): [17]

- 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 [18], 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 [19, 20]:

- **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 *banc* però amb una acepció diferent:

  Ens vam seure en el *banc* que acabaven de pintar.
  Va entrar al *banc* per actualitzar la llibreta.

- ***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"* [20]. 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

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

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 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 [20], 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*) [6].

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.

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: [21]

- 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 [22].

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 és ahí on se trobaran les frases més rellevants, i en general per a textos periodístics aquesta intuïció ha resultat prou efectiva, o també seleccionant frases que contingueren paraules del títol... 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.

D'ahí van aparèixer uns models híbrids en els quals a banda de la fase extractiva s'incorporen textos resultants d'una generació abstractiva, que en suma obtenen uns resums més concisos i amb millors resultats que els purament extractius. Per a aquesta aproximació hi ha dos tendències que presenten els millors resultats en aquesta àrea: *Reinforcement Learning model* (RL) [23] i *Inconsistency Loss model* (IL) [24].

El model RL es basa en una estratègia *extraction-then-abstraction*. En el qual existeixen dos móduls separats que s'enllacen mitjançant un entrenament de reforç. Aquest entrenament es fa mitjançant documents etiquetats[2] de manera que el módul extractiu

[2]En l'àmbit de la generació de resums quan parlem d'etiqueta es tracta d'un resum de referència que típicament ha generat una persona.

s'encarrega de recuperar la frase del document que tinga major semblança [3], aquesta frase s'estiqueta positiva i la resta negatives, permetetent reformular aquesta tasca com un problema de classificació). I una vegada s'ha acabat la fase extractiva entra en joc el mòdul abstractiu, encarregat de comprimir el fragment perquè s'assemble més al resum de referència del text en qüestió.

D'altra banda, el model IL segueix una estratègia *extraction-with-abstraction*, la qual cosa li permet aconseguir una major llegibilitat[4]. L'estructura de l'entrenament és molt semblant a la dels models RL, però aquesta tècnica posa el focus en combinar una atenció [5] a l'oració en la part extractiva i una atenció a la paraula en l'abstractiva. Per finalitzar, per tal de garantir que les dues atencions siguen coherents es gasta la *inconsistency loss*, que assegura que les frases que tinguen molta atenció, també la tinguen a nivell de paraula [25].

I després tenim els models abstractius, que segueixen mantenint una primera fase en la qual se seleccionen les part més importants del text original i el resum final s'obté a base de parafrasejar o reformular eixes frases. Aquests 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 queden 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 [26] 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 [27] 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.

Tot seguit passarem a fer un recorregut per l'evolució dels resums automàtics [28], en concret dels abstractius, primerament comentarem algunes de les tècniques que es feien servir abans de l'aparició de les xarxes neuronals, perquè aquest és un camp en el que es porta investigant des dels anys 50 i després ja passarem a comentar les tècniques més modernes amb xarxes.

Durant l'època anterior a les xarxes neuronals, per millorar realment els resultats que obtenien els resums extractius s'aplicava un procés que consistia en 3 subtasques:

- **Extracció d'informació**: Durant aquesta fase s'extrau la informació més important del text d'entrada, emprant el que es coneix com *query-based extraction* [29] i filtrant aquells continguts que tinguen una baixa probabilitat d'aparèixer al resum. O una altra aproximació es feia extraent *Information Items* (INITs) [30] que se definia com una tripleta subjecte-verb-objecte ubicada en l'espai i el temps. Aquesta extracció es veia beneficiada quan feia resums d'un domini específic perquè podia aprofitar el coneixement sobre eixe domini per guiar l'extracció. Però en casos com una reunió d'empresa en la qual es discuteix sobre temes molt diversos no pot aplicar-se un sol domini, llavors el que se pot fer és aprofitar segmentació de temes per identificar aquells que s'han tractat en la reunió i així poder guiar de manera particularitzada l'extracció en cadascun dels dominis.

[3] La semblança es calcula mitjançant alguna funció de mesura com pot ser ROUGE-L que s'explicarà en futurs capítols.

[4] Atenent a la mesura ROUGE-2 que segons veurem més endavant és la mesura que es capaç de valorar la llegibilitat d'un text.

[5] Forma de mesurar la importància d'un fragment.

- **Selecció de contingut:** En aquesta etapa es fa una preselecció de les frases que seran candidates per a formar part del resum definitiu, normalment amb unes limitacions de longitud. Per a aquest fi s'utilitzaven heurístiques, però després es va aplicar també *Integer Linear Programming* (ILP) [31] per tal d'optimitzar una funció objectiu subjecta a una sèrie de limitacions.
- ***Surface realization*:** L'objectiu d'aquesta última subtasca és combinar les frases candidates resultants de la selecció de contingut seguint les regles gramaticals i sintàctiques per generar un resum.

Fonamentalment podem parlar de dos mètodes per portar a terme aquestes 3 subtasques i poder generar un resum abstractiu:

- **Mètodes basats en grafs:** S'aprofita l'expressivitat que tenen els grafs per tal d'implementar les 3 subtasques abans esmentades. Un parell d'exemples d'aquesta aproximació són els *event semantic link networks* (ESLNs) [32] que s'utilitzen per fer conjuntament l'extracció d'informació i selecció de contingut. El graf es construeix a partir d'un text d'entrada de manera que cada node es correspon amb un esdeveniment esmentat en el text i els arcs entre nodes venen donats per una relació semàntica entre dos esdeveniments. I una vegada s'ha construit se pot aplicar ILP de la manera que s'ha explicat prèviament. Altra opció serien els *entailment graphs* [29] que fan la selecció de contingut mitjançant la detecció de frases redundants. Si dos frases tenen el mateix significat (estan vinculades bidireccionalment) una d'elles serà eliminada i si una és més informativa que l'altra (vincle unidireccional) aleshores la menys informativa és eliminada. Mentre que si cadascuna d'elles conté una part que no està superposada a l'altra, es mantindran ambdues.
- **Mètodes basats en plantilles:** Estan fonamentades sobre l'observació que els humans seguim unes estructures semblants a l'hora de resumir textos que pertanyen al mateix àmbit. Per tant aquestes estructures poden aprendre's durant l'entrenament i així codificar una sèrie de plantilles. I quan li arribe un document, s'encarregarà d'emplenar els buits que haja deixat en la plantilla que més s'ajuste al tipus de document d'entre totes les que ha après. En general aquest mètode està composat per 3 fases:

  1. Aprenentatge de les plantilles a base de resums humans, deixant espais en blanc per tal de ser substituits posteriorment.
  2. Extracció de les frases importants del document d'entrada.
  3. Generar un resum emplenant els espais en blanc de la plantilla.

Però les 3 subtasques que s'havien de realitzar per tal d'aconseguir uns resums que foren abstractius: extracció d'informació, selecció de contingut i *surface realization* són tasques gens trivials, mentre que amb l'ús de xarxes neuronals, se pot aconseguir en una única xarxa abstraure a partir d'un document i generar el seu resum corresponent. Des del treball de Rush et al. (2015) [33] 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. Seguidament 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 [34]). La majoria dels models empren vectors de paraules ja preentrenats mitjançant grans corpus, com és el cas de word2vec [35] o GloVe [36], 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 [23].

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 [11], que després van ser substituits pels RNNs [37, 38] 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 [16] o en alguns casos per GRUs que necessiten menys paràmetres i per tant són més ràpides d'entrenar obtenint uns resultats equivalents [18].

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* [33].

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 [39], 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, [40] 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 [41], 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* [42]. Per afrontar aquests problemes s'ha aplicat el *Reinforcement Learning* (RL) [23], 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 [43]. 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'aprenentatge 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 PEGASUS

(*Pre-training with Extracted Gap-sentences for Abstractive Summarization Sequence to sequence models*) [27], perquè es tracta d'un mètode de generació de resums abstractius que ha aconseguit uns resultats d'estat de l'art en diferents tasques i amb *datasets* de temàtiques molt variades. Els resums generats han estat sotmesos a avaluació humana i s'ha conclós que obtenien una qualitat humana en diversos conjunts de dades. A més es va comprovar que tenia una gran capacitat per adaptar-se a conjunts de dades que no havia vist anteriorment i que eren relativament xicotets (només 1000 mostres). Per aconseguir aquests resultats, el model de PEGASUS va ser primer preentrenat afegint dos objectius durant el preentrenament, com es reflexa en la Figura 2.4:

- ***Gap Sentences Generation* (GSG)**: Aquesta tècnica es fa servir per evitar la necessitat de tindre una part extractiva durant la generació del resum. La tècnica consisteix en seleccionar algunes frases del document segons algun criteri (Lead-K, aleatori o Principal, que agafa les millors frases segons ROUGE-1 entre la frase i la resta del document).
- ***Masked Language Model* (MLM)**: Seguint l'estil de BERT, un 15% dels tokens del text d'entrada són seleccionats i d'aquests un 80% són substituits per un token especial $[MASK2]$, un 10% són substituits per un token aleatori i el 10% restant resta intacte.

I posteriorment partint d'aquest model preentrenat es va aplicar unes tasques posteriors per tal d'especialitzar-lo en el resum automàtic.

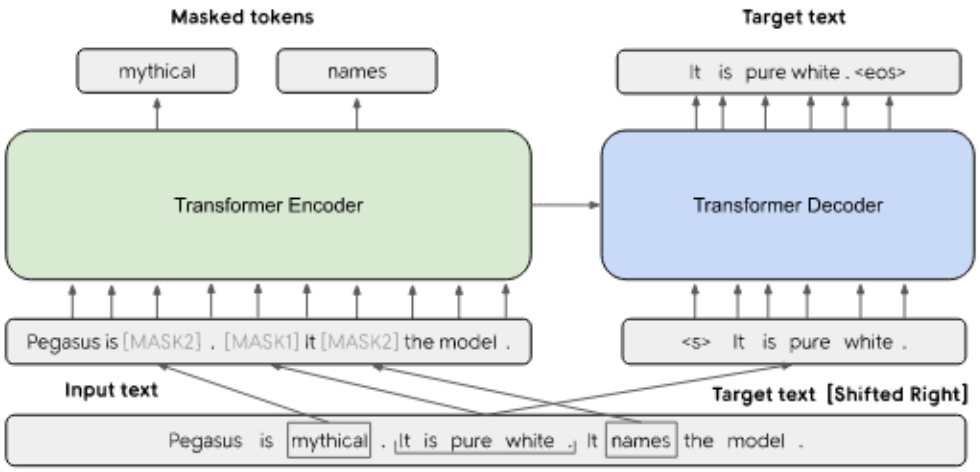

Figura 2.4: Explicació del funcionament de GSG i MLM com a objectius de preentrenament en PEGASUS[27]

2.3 Corpus

Hui en dia, com ja hem comentat en l'estat de l'art de PLN, el que se gasta per treballar en esta matèria són xarxes neuronals, que necessiten una gran quantitat de dades per a poder aprendre els pesos de les seues connexions. Però açò és encara més agreujat en el cas dels *Transformers*, que són les xarxes més emprades ara mateix, perquè se tracta de xarxes molt grans amb una quantitat enorme de paràmetres que aprendre. Per exemple en el cas del model per a resums en català que hem entrenat per a este treball han sigut 415,222,784 paràmetres els que s'han hagut d'aprendre i s'han utilitzat 636,596 notícies només per a l'entrenament [44]. A més també és rellevant posar de manifest que en

este tipus d'entrenament no és necessari exclusivament el corpus que fem servir per al fine-tuning quan especialitzem el model per a una tasca concreta, com en este cas per al resum; sinó que abans d'això s'ha hagut de preentrenar un model amb una quantitat de dades encara major. En el cas del nostre model, es van gastar 2.5 milions de documents per a aquest fi. Per tant, una part de la feina no gens menyspreable és recollir un conjunt de dades que siguen de qualitat i preparar-los perquè puguen ser entesos pel model que volem entrenar. La part positiva és que disposem d'un repositori enorme de textos de liure accés: la World Wide Web on hi ha textos de tot tipus i sobre tots els temes; la part negativa és que aquests textos no poden agafar-se en cru directament de la pàgina web, sinó que cal fer *web scraping* per tal d'extraure els arxius que volem d'una web i després netejar-los per tal de llevar-los el format HTML per exempe. A més com el nostre objectiu és entrenar un model centrat en la tasca de resum de notícies, a esta complicació se li suma el fet que necessitem que les entrades siguen parells d'articles i un resum de referència, que precisen d'un periodista que l'haja redactat i d'altra banda que cada vegada més els diaris estan adaptant-se al format digital i estan protegint-se contra este *web scraping* per evitar que es consumisquen de forma gratuïta les seues notícies.

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, com Sentiment140 per a l'anàlisi de sentiments, WMT14 per a traducció automàtica o per a resum automàtic un corpus molt emprat és CNN/Daily Mail que conté 300,000 parells de notícies i resums procedents d'eixos dos diaris, o NewsRoom que conté 1.3 milions d'articles. Un inconvenient és que la gran majoria dels datasets que tenim disponible són, naturalment, en les llengües més parlades, perquè hi ha més recursos d'on extreure informació, però açò suposa un problema perquè al final entrem en un cercle viciós, perquè com tenim molts documents per exemple en angès, els models i datasets que creem estan en anglès perquè en principi com més recursos tinguem millors resultats s'obtindran, i aleshores quan se generen recursos com hi ha més eines disponibles en anglès, se segueixen creant en anglès. Tot açò, és clar, en detriment de les llengües minoritàries, que per tant passen a més a ser minoritzades en el context tecnològic, la qual cosa fa que no totes les llengües tinguen els mateixos drets i al cap i a la fi no tots els parlants tinguen les mateixes oportunitats.

Per exemple en la Figura 2.5 podem veure la taula dels datasets i models que té disponibles HuggingFace per a entrenar model. Podem veure la diferència que hi ha entre l'anglès i qualsevol altra llengua. És rellevant parlar dels que té disponibles HuggingFace no només perquè siga la tecnologia en la qual ens estem recolzant nosaltres, sinó perquè s'ha convertit en una comunitat molt important en l'àmbit de l'aprenentatge automàtic, en especial dels models basats en *Transformers*. Disposa d'una llibreria que ofereix grans facilitats per entrenar-los que atrauen a molts usuaris, tant sèniors com aquells que estem començant a endinsar-nos en este món.

2.4 Classificació de textos

En aquesta secció farem una explicació de què és la classificació i farem un recorregut per les diferents tècniques que s'han anat emprant al llarg del temps [45].

La classificació en l'àmbit de la Intel·ligència Artificial és una tasca que consisteix en assignar a una entrada una etiqueta d'entre un conjunt predefinit. És una part fonamental també de certes aplicacions del PLN com pot ser l'anàlisi de sentiments o l'etiquetat de temes. Té un paper fonamental, degut a que l'èxit de la capacitat humana deixa molt que desitjar en moltes ocasions perquè som éssers subjectius i la classificació depèn de molts factors com per exemple el bagatge de la persona, el seu cansament en el moment que fa

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 2.5: Taula de datasets i models disponibles en HuggingFace en les diferents llengües

la classificació, etc. Per la qual cosa ens interessa emprar l'aprenentatge automàtic perquè se puga automatitzar aquests processos de manera que s'aconseguisca automatizar el procés i a més a més aconseguir millor rendiment com ha ocorregut per exemple en el cas del ben conegut MNIST, en el que l'error humà oscil·lava al voltant del 0.2% [46] mentre que actualment s'ha aconseguit mitjançant models de *machine learning* baixar-ho al 0.13% [47].

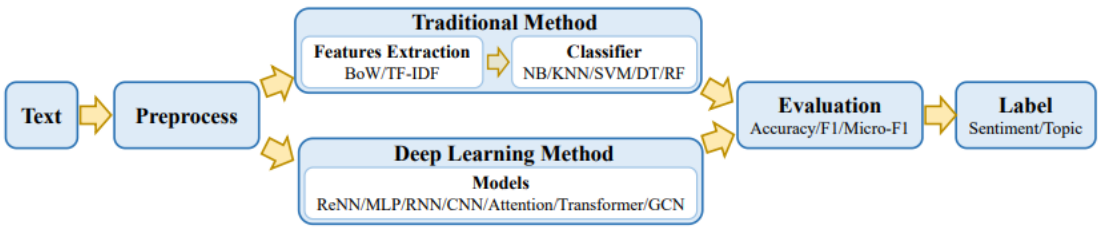

Figura 2.6: Representació del fluxe del procés de classificació de text en mètodes tradicionals i d'aprenentatge profund [45]

Treballar amb text té una gran diferència a treballar amb altra entrada com imatge o característiques numèriques. Perquè la informació cal que siga preprocessada per tal

de deixar-la en un format amb el qual puga treballar el model. Generalment este preprocessament està composat per 3 etapes per a les quals necessitem tècniques de PLN: tokenització, normalització i eliminació de soroll [48]. La principal diferència entre els models clàssics i els que utilitzen aprenentatge profund, com pot apreciar-se clarament en la Figura 2.6 ve precisament en el pas següent al preprocessament de les dades, perquè en el cas dels models clàssics era necessària una primera fase d'extracció de característiques i després a partir d'eixa informació mitjançant algorismes d'aprenentatge automàtic clàssics realitzar una classificació, el problema d'això és que la seua efectivitat estava totalment sotmesa a una bona extracció de característiques. Mentre que en els models d'aprenentatge profund l'extracció de característiques està integrada dins del procés de classificació mitjançant l'aprenenatge de relacions no linials que permeten detectar certes característiques i associar-les directament amb una eixida (etiqueta).

Dels anys 60 fins la dècada de 2010, per a la classificació de textos dominaven els models tradicionals, és a dir, aquells basats en models estadísitics, com per exemple *Naïve Bayes* [49], k-veins més propers (KNN) [50] i Màquines de Vector Suport (SVM) [51]. Aquestes aproximacions milloraven notablement els resultats que s'havien obtingut anteriorment amb métodes basats en regles. No obstant, encara necessitaven una millora, perquè encara necessiten fer l'extracció de característiques que consumeix una gran quantiat de temps i és molt costosa i a més a més perden la informació contextual, complicant més la feina de comprendre la informació semàntica de les paraules. A partir de la dècada de 2010, els models emprats per a la classificació de textos va començar a canviar de tendència, deixant pas als mètodes d'aprenentatge profund, que eviten haver de dissenyar regles per l'extracció de característiques i ofereixen una capacitat per crear automàticament les representacions de textos que a més aconsegueixen mantindre la seua semàntica. Per la qual cosa la major part de la investigació al voltant de la classificació de textos, igual que en la resta de camps del PLN s'ha basat en Xarxes Neuronals Profundes (DNNs) [52]. Entre els models de xarxes neuronals profundes podem destacar CNNs [53] i RNNs [54] dels que ja hem parlat prèviament, entre d'altres. I en temps més recents cal destacar BERT[55], un *Transformer* que originàriament no estava dissenyat per a tasques de classificació, però que ha estat emprat àmpliament com a model preentrenat de partida per fer *fine-tuning* en tasques de classificació.

CAPÍTOL 3
Metodologies i sistemes utilitzats

En aquest capítol 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 del corpus

En aquesta secció explicarem quin és el contingut i estructura del corpus emprat per a l'entrenament, validació i avaluació dels models. Explicarem inicialment com està composat el corpus tant per al castellà com per al català, quines modificacions li hem fet per a poder emprar-lo en l'àmbit d'aquest TFG, en especial l'anonimitzat que s'ha portat a terme; i per últim la combinació dels dos corpus per a l'entrenament bilingüe.

3.1.1. Corpus original

El corpus emprat ha sigut DACSA (*Dataset for Automatic summarization of Catalan and Spanish newspaper Articles*)[1] . Aquest corpus conté parells d'articles periodístics i el seu resum de diferents diaris, tant en català com en castellà. Es va construir seguint un procediment semblant al de *NewsRoom* [56], extraent les notícies de les pàgines web dels diaris triats. Així s'han aconseguit quasi 1 milió de notícies per al català i 5.5 milions per al castellà. La distribució que es va aconseguir per fonts va ser la que veiem en les Taules 3.1 i 3.2. En ambdues Taules es pot veure tant la informació de les mostres incloses en el corpus com les que foren descarregades però finalment es descartaren.

Font periodística	Documents	Exclosos	Inclosos
CA01	288,081	49,848	238,233
CA02	224,705	87,258	137,447
CA03	49,494	13,731	35,763
CA04	234,022	39,325	194,697
CA05	10,170	3,066	7,104
CA06	6,887	1,005	5,882
CA07	56,110	11,729	44,381
CA08	64,180	7,353	56,827
CA09	25,663	20,813	4,850
Total	959,312	234,128	725,184

Taula 3.1: Distribució de la procedència de les notícies en català per a la creació del corpus de DACSA. [57]

Font periodística	Documents	Exclosos	Inclosos
ES01	2,047,404	1,960,950	86,454
ES02	220,806	52,741	168,065
ES03	256,513	60,103	196,410
ES04	17,747	368	17,379
ES05	127,618	11,057	116,561
ES06	178,026	176,652	1,374
ES07	157,092	9,039	148,053
ES08	42,018	25,053	16,965
ES09	112,592	77,480	35,112
ES10	613,954	63,806	550,148
ES11	60,605	3,370	57,235
ES12	82,092	8,068	74,024
ES13	470,987	470,344	643
ES14	73,395	73,240	155
ES15	96,210	61,047	35,163
ES16	598,346	256,301	342,045
ES17	129,581	30,483	99,098
ES18	86,594	4,647	81,947
ES19	50	2,614	164
ES20	119,770	12,608	107,162
ES21	4,100	3,633	467
Total	5,498,064	3,361,154	2,136,910

Taula 3.2: Distribució de la procedència de les notícies en castellà per a la creació del corpus de DACSA. [57]

Per a crear DACSA s'ha aplicat sobre les notícies extretes un seguit de filtres per tal d'assegurar que els documents compliren una sèrie de característiques, per exemple que els resums i l'article tingueren una quantitat mínima de paraules o que la semblança entre l'inici de la notícia i el resum fora inferior a una cota màxima. Aquelles mostres que no compliren amb estes característiques han estat descartades del corpus, de tal manera que DACSA finalment va quedar-se amb 725,000 articles per a la partició en català i 2,100,000 per a la de castellà.

Les mostres que s'han quedat en el corpus s'han distribuït en 4 particions:

- **Entrenament:** És el subconjunt destinat a ensenyar al model a classificar i resumir durant el seu entrenament.
- **Validació:** S'utilitza també durant l'entrenament per tal d'avaluar una sèrie de mètriques que permeten assegurar que s'està realitzant un bon entrenament i a més a més serveix per a triar el millor model.
- **TEST-I:** Conjunt de notícies de fonts que han sigut vistes pel model durant l'entrenament, es fa servir per avaluar les distintes mètriques sobre el model resultant de l'entrenament.
- **TEST-NI:** Aquest conjunt també s'empra per al mateix fi que l'anterior, però està composat per notícies de fonts que el model no ha vist en cap etapa perquè no arribaven a suposar el 5% en el conjunt de test i validació. Per tant aquesta partició ens serveix per avaluar com de bé generalitza el nostre model.

Com s'ha vist en les Taules 3.1 i 3.2 la distribució de les notícies per font dista molt de ser homogènia, i si mantenim aquesta distribució desequilibrada en totes les particions,

tindrà com a conseqüència que el model es focalitze en aprendre aquelles notícies que conformen el grup majoritari. Per evitar eixe biaix el que es va fer és que per al conjunt de test i de validació tots els diaris tingueren el mateix pes. Per a poder garantir això, es van descartar aquelles fonts que representaven menys d'un 5% en estos conjunts, i amb eixes mostres vam construir la partició de *TEST-NI*. De manera que sense comptar les mostres de les fonts descartades, es van dividir en un 90% per a entrenament, un 5% per a validació i altre 5% per a *TEST-I*. A continuació en la Taula 3.3 podem observar com va quedar la distribució de les mostres al llarg de les 4 particions, tant per a català com castellà:

Idioma	Entrenament	Validació	TEST-I	TEST-NI
Català	636,596	35,376	35,376	17,836
Castellà	1,802,919	104,052	104,052	109,626

Taula 3.3: Distribució de les mostres del corpus DACSA entre les 4 particions.

Les Taules 3.4, 3.5, 3.6 i 3.7 mostren les característiques principals de cadascuna de les fonts i les particions.

				Article			Resum	
Font	Docs	Tokens	|V|	Sents Per Doc	Words Per Sent	|V|	Sents Per Doc	Words Per Sent
CA01	238,233	114,500,016	614,146	17.68	27.19	115,954	1.14	20.16
CA02	194,697	105,119,526	621,612	19.99	27.01	112,904	1.28	19.14
CA03	137,447	63,683,416	485,286	14.99	30.92	91,975	1.05	22.65
CA04	56,827	24,891,291	276,720	14.84	29.52	58,071	1.21	17.52
CA05	44,381	26,977,332	277,225	18.04	33.69	55,216	1.15	23.86
CA06	35,763	17,181,460	202,931	11.31	42.49	42,289	1.05	22.79
CA07*	7104	3,800,842	83,942	18.04	29.66	19,267	1.02	26.51
CA08*	5882	9,414,192	185,977	66.04	24.24	31,006	2.54	24.84
CA09*	4850	2,667,185	102,024	23.61	23.29	19,584	1.16	28.05
Total	**725,184**	**368,235,260**	**1,326,343**	**17.71**	**28.67**	**223,978**	**1.17**	**20.59**

Taula 3.4: Estadístiques de la partició en català del corpus DACSA. Les fonts marcades amb * no s'han gastat durant l'entrenament, s'han reservat per *TEST-NI* [44]

Com ja s'ha comentat prèviament, en el món de la Intel·ligència Artificial, hi ha una concentració de models i de corpus en anglès, mentre que d'altres llengües, en especial d'aquelles més minoritàries com el català, no hi ha tants recursos disponibles. Pel català el més comú és recórrer al contingut que hi ha en Viquipèdia o al corpus d'Oscar [58], que és el *dataset* que fa servir BART en el seu entrenament, i que és prou reduït en comparació al de la resta de llengües. Per estes raons cal destacar la importància d'haver disposat d'un corpus de la mida del de DACSA amb notícies d'àmbit general.

3.1.2. Modificacions

Per al nostre entrenament no hem utilitzat el corpus DACSA pur sinó que li hem aplicat una sèrie de modificacions per adaptar-lo a les nostres necessitats. Tot seguit comentarem aquestes modificacions:

Anonimitzat

Com l'entrenament d'aquests models tenia com a objectiu addicional al resum automàtic la classificació de la font, es va plantejar efectuar un anonimitzat del corpus per tal d'es-

			Article			Resum		
Font	**Docs**	**Tokens**	**|V|**	**Sents Per Doc**	**Words Per Sent**	**|V|**	**Sents Per Doc**	**Words Per Sent**
ES01	550,148	420,786,144	1,473,628	31.36	24.39	210,079	1.40	19.02
ES02	342,045	174,411,220	907,312	16.66	30.61	148,271	1.06	22.34
ES03	196,410	93,755,039	622,073	15.40	31.00	110,728	1.02	20.59
ES04	168,065	105,628,806	659,054	23.35	26.92	112,908	1.09	22.30
ES05	148,053	105,453,102	626,058	28.35	25.13	109,546	1.47	20.46
ES06	116,561	93,956,373	524,177	26.16	30.81	169,025	1.27	43.20
ES07	107,162	70,944,634	470,244	19.90	33.26	87,901	1.29	25.27
ES08	99,098	65,352,628	495,495	25.03	26.35	81,654	1.25	18.38
ES09	81,947	42,825,867	363,075	15.54	33.63	71,913	1.03	22.41
ES10	74,024	57,782,514	470,826	30.28	25.78	81,793	1.31	20.23
ES11*	70,193	29,692,261	272,248	11.06	38.26	84,898	1.22	44.48
ES12	57,235	28,198,002	294,175	16.06	30.68	58,580	1.21	19.49
ES13	35,163	20,156,337	260,690	19.22	29.83	50,556	1.15	21.20
ES14	35,112	28,408,974	309,194	30.48	26.55	78,751	1.18	28.35
ES15*	17,379	10,099,958	153,598	16.82	34.54	41,512	1.85	26.89
ES16*	16,965	13,791,564	166,446	28.26	28.77	29,955	1.07	25.18
ES17*	2,450	4,545,924	135,761	74.97	24.75	23,588	3.16	26.72
ES18*	1,374	641,752	39,094	17.08	27.34	12,365	1.98	29.43
ES19*	643	398,834	26,797	17.73	34.99	2495	1.04	16.02
ES20*	467	233,873	22,699	18.70	26.78	3857	1.22	24.23
ES21*	155	199,140	19,750	39.06	32.89	2098	1.91	21.79
Total	2,120,649	1,367,262,946	3,189,783	23.44	27.50	516,307	1.24	22.95

Taula 3.5: Taula d'estadístiques tècniques del corpus en castellà. Les fonts marcades amb * no s'han gastat durant l'entrenament, s'han reservat per *TEST-NI* [44]

			Article			Resum		
Partició	**Docs**	**Tokens**	**|V|**	**Sents Per Doc**	**Words Per Sent**	**|V|**	**Sents Per Doc**	**Words Per Sent**
Entrenament	636,596	316,817,625	1,206,292	17.39	28.62	206,616	1.17	20.36
Validació	35,376	17,831,029	258,999	16.17	31.17	51,940	1.15	20.93
TEST-I	35,376	17,704,387	262,148	16.13	31.03	51,958	1.15	20.89
TEST-NI	17,836	15,882,219	247,154	35.38	25.17	45,997	1.56	25.93

Taula 3.6: Taula d'estadístiques tècniques del corpus en català. Organitzades per partició. [44]

			Article			Resum		
Partició	**Docs**	**Tokens**	**|V|**	**Sents Per Doc**	**Words Per Sent**	**|V|**	**Sents Per Doc**	**Words Per Sent**
Entrenament	1,802,919	1,172,626,265	2,920,894	23.94	27.17	454,179	1.24	21.99
Validació	104,052	67,669,381	550,213	23.01	28.27	109,460	1.21	23.36
TEST-I	104,052	67,363,994	550,910	22.93	28.23	109,706	1.21	23.34
TEST-NI	109,626	59,603,306	447,679	16.25	33.46	116,201	1.35	36.84

Taula 3.7: Taula d'estadístiques tècniques del corpus en castellà. Organitzades per partició. [44]

borrar qualsevol pista que puguera quedar en el text i li permetera al model centrar-se en eixos detalls en lloc d'en l'estil que era el nostre objectiu. El procés d'anonimitzat va consistir bàsicament en substituir totes les referència que pugueren aparèixer en cada article al nom del diari, bé fora a través del link a la seua web o bé el propi nom del diari, i van ser substituides per cadenes úniques en cadascun dels documents, per exemple si apareixia la URL: *https://www.nom_diari.cat*, totes les vegades que aparega eixa URL en eixe document, seran substituides per *https://wxy.z* i si torna a aparèixer en altres documents, tindrà un altre identificador únic perquè no puga vincular les notícies a eixa adreça.

Font afegida al resum

Per tal de poder entrenar el nostre model en una tasca de classificació, era necessari que durant l'entrenament i en les distintes avaluacions, el model puguera comparar la seua eixida tant de font predita com de resum generat amb una referència, per a això, com el nostre model genera una única eixida, vam decidir que la millor opció seria que dins d'eixe resum s'incloguera la font en la qual l'havia classificat, de manera que vam haver d'aprofitar la informació de la qual disposàvem per regenerar el corpus i que en el camp *summary* que representa bàsicament el resum de referència amb el qual se compara el model per tal d'entrenar-se i d'avaluar-se, tinguera el format: *FONT <separador> resum*.

3.2 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.

3.2.1. 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.

3.2.2. 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 gastar-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 3.1.

	no	és	un	bon	llibre
és un bon llibre	0	1	1	1	1
no és bon llibre	1	1	0	1	1
és un llibre	0	1	1	0	1

Figura 3.1: Exemple de representació mitjançant Bossa de paraules

3.2.3. 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 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 3.2 pot apreciar-se com paraules que estan relacionades per gènere, per capitalitat o conjugacions verbals són representades pròximes en l'espai vectorial.

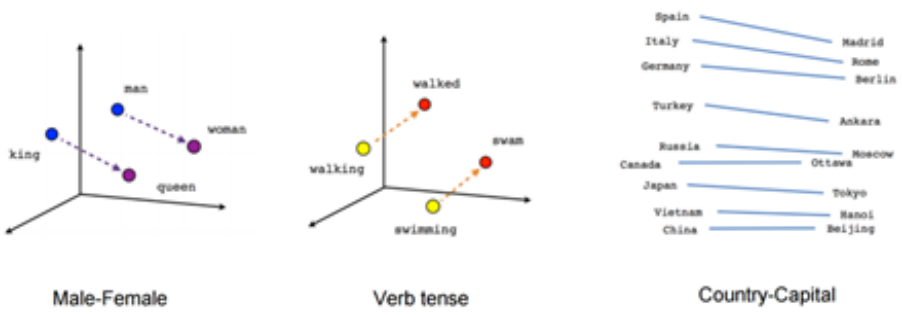

Figura 3.2: Representació de la proximitat en l'espai vectorial entre *word-embeddings* de paraules relacionades (Font: TensorFlow.org)

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:

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

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 [59].

- **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 [35], 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 [60]. 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 este 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 el resultat d'una frase serà una matriu, perquè cada paraula es converteix en tot un vector.

3.3 Mètriques d'avaluació

Per a avaluar el rendiment dels nostres models de resum hem fet servir mètriques ben conegudes. Per a conèixer com de bons han sigut els resums obtinguts, utilitzem les mètriques de ROUGE i BERTscore que són 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. També els hem avaluat amb mesures d'abstractivitat per veure si els resums generats eren més abstractius o extractius. I com l'objectiu del nostre model era doble: resumir i detectar la font de la notícia, també s'han emprat mètriques de classificació.

3.3.1. ROUGE

Es tracta d'una mètrica que 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 escrit, en principi, per un periodista. 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 i que explicarem en l'apartat dedicat a les mètriques d'avaluació de la classificació de la font. 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.

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

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.

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.

ROUGE-Lsum

És una alteració de l'anterior, que en lloc de considerar cadascun dels textos rebuts com una unitat, esquartera el text de referència en frases, i llavors calcula la cadena conicident de major longitud dins del resum generat per a cadascuna de les frases en què s'ha dividit el text de referència.

3.3.2. BERTScore

Es un mètode que no torna valors entre 0 i 1 com el ROUGE, sinó que en llengües com el català o el castellà pot aconseguir arribar a valors de 65 amb molta facilitat i és molt difícil obtindre'n superiors a 85. Llavors això implica que aquesta mesura no aporte 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. Classificació

Dels models entrenats volíem avaluar no només com de bons eren resumint, sinó també com de bons eren classificant la font a la qual pertanyia la notícia. Per avaluar el rendiment del model en la tasca de classificació hem utilitzat unes mètriques també ben conegudes: accuracy, precisió, recall i F1-score, que passarem a explicar a continuació. Per calcular-les vam fer servir la llibreria *scikit-learn*.

Matriu de confusió

La matriu de confussió és una mètrica d'anàlisi, concretament del camp de classificació estadística, que permet visualitzar el rendiment d'un model de classificació. Cada fila de la matriu representa la classe real de les mostres i les columnes la predicció que ha fet el model. Per poder entendre-la cal que expliquem 4 conceptes que també ens serviran d'ajuda per a les pròximes mètriques:

- *True Positive* (TP): La predicció coincideix amb la classe real. Són els valors de la diagonal per a cadascuna de les classes.
- *True Negative* (TN): Per a una determinada classe *C*, serà la suma de tots els valors de la matriu, menys els de la fila i columna que se corresponguen amb la classe *C*. Representa que la mostra no era de la classe *C* i el model efectivament l'ha classificat en una altra classe diferent de C (encara que no siga la correcta).
- *False Positive* (FP): Per a una determinada classe *C* són tots aquells casos en els quals el model ha predit la classe *C* però no era la classe correcta. Es pot calcular com la suma de la columna de la classe on l'ha classificat menys el valor que se correspón amb el TP d'eixa classe, és a dir, l'element de la diagonal per a la classe C.
- *False Negative* (FN): Per a una determinada classe *C* són aquells casos en els quals la mostra era de la classe *C* però ha sigut classificada en qualsevol altra classe. Per tant és la suma de la fila corresponent a la classe *C* a excepció del valor de la diagonal (el TP de la classe C).

Accuracy

Es calcula com el nombre total de prediccions correctes front al total de prediccions.

$$Accuracy = \frac{TruePositives + TrueNegatives}{TruePositives + FalsePositives + TrueNegatives + FalseNegatives}$$

Precisió

Es calcula com el nombre de de prediccions d'una determinada classe que han sigut realment correctes.

$$Precisio = \frac{TruePositives}{TruePositives+FalsePositives}$$

Recall

Mesura quantes de les mostres que realment eren d'una classe, s'han classificat en eixa classe

$$Recall = \frac{TruePositives}{TruePositives+FalseNegatives}$$

F1-Score

És una mesura que permet combinar les dos anteriors, generalment es calcula com la mitjana harmònica d'aquestes dos:

$$F1 - Score = \frac{2*Precisio*Recall}{Precisio+Recall}$$

En el nostre cas com estem fent una classificació multi-classe, tant la precisió com el recall i F1-score poden calcular-se de tres maneres.

- **Macro**: Es calcula el valor de la mètrica per cadascuna de les classes per separat i després es trau la mitjana aritmètica de totes elles sense tindre en compte el nombre de mostres de cadascuna de les classes.
- ***Weighted***: A l'hora de traure la mitjana, els valors de la mètrica de cadascuna de les classes es multiplica pel percentatge de mostres que pertanyen a eixa classe respecte del total de mostres.
- **Micro**: En aquest cas es calculen els TP, FN i FP per a totes les mostres i després es fan els mateixos càlculs, per exemple per al F1-Score seria:

$$F1 - Score = \frac{TP}{TP+\frac{1}{2}(FP+FN)}$$

En el nostre cas utilitzem la versió macro, però no obstant coincideix amb la *weighted* perquè en els conjunts de test i validació totes les classes (fonts) tenen la mateixa quantiat de mostres.

3.3.4. Abstractivitat

El nostre model està entrenat per generar resums abstractius, és a dir, que no siga simplement agafar frases del text original i reordenar-les sinó que siguen de pròpia creació i creiem que és rellevant quantificar com ha sigut el seu rendiment en aquest aspecte, però no hi ha una única mètrica ideal, per això hem triat 4 mètriques *extractive fragment coverage*[56], *abstractivity*$_p$[61], *novel n-grams* [62] i *content reordering* [44]. Ara passarem a descriure aquestes mètriques.

Extractive fragment coverage

Aquesta mesura permet quantificar en quina mesura un resum s'ha derivat del text original, conta quin percentatge de paraules del resum estan presents en l'article.

$$COVERAGE(A,S) = \frac{1}{|S|}\sum_{f \epsilon F(A,S)} |f|$$

On $F(A,S)$ és l'operació que retorna el conjunt de les cadenes extractives comunes més llargues entre l'article A i el resum S i $|S|$ és el nombre de paraules que composen el resum.

Abstractivity$_p$

L'*abstractivity$_p$* permet conèixer quin grau d'abstractivitat té un resum respecte a l'article original, en relació al solapament de les seqüències extractives més llargues. Torna un valor entre 0 i 1, sent 0 totalment extractiu i 1 totalment abstractiu.

$$ABS_p(A,S) = 1 - \frac{\sum_{f \epsilon F(A,S)} |f|^p}{|S|^p}$$

El valor de p serveix per decidir el pes de les seqüències extractives segons la seua longitud, de manera que si és major que 1, penalitzarà aquelles seqüències que siguen més llargues i reduirà el pes que tenen les més curtes.

Novel n-grams

Aquesta mètrica es va introduir en l'entrenament de models com una "recompensa" per tal d'afavorir l'abstractivitat dels resums produïts i va donar molt bons resultats [62]; nosaltres la farem servir per avaluar a posteriori l'abstractivitat del nostre model. Un *novel n-gram* en el resum és aquell n-grama que no està present en el text original. Siga $ng(x,n)$ una funció que calcula el conjunt de n-grames únics presents en el document x, $||s||$ el nombre de paraules en el conjunt s i seguint la mateixa nomenclatura abans expressada, definim la mètrica de *novel n-grams* sense normalitzar com:

$$N(S,A,n) = \frac{||ng(S,n) - ng(A,n)||}{||ng(A,n)||}$$

Però per previndre que s'obtinguen resultats molt alts quan els resums són molt curts, normalitzem aquesta mètrica pel ratio entre la longitud del resum generat i el de referència, els denotarem com S_g al generat i S_r al de referència:

$$R^{nov}(S_g,S_r,A,n) = N(S_g,A,n)\frac{||S_g||}{||S_r||}$$

Content Reordering [44]

Es tracta d'una mètrica que quantifica el percentatge d'informació que s'ha reordenat en el resum respecte de l'ordre original que tenia en l'article. Quan s'incrementa la reordenació, s'està incrementant l'abstractivitat del resum generat. Siga $Reordered(T,S)$ l'operació que conta el nombre de segments extractius reordenats, podem definir la mètrica com:

$$ContentReordering(T,S) = \begin{cases} \frac{\sum_{f \epsilon F(T,S)} |f|}{|S|} \frac{Reordered(T,S)}{|F(T,S)|-1}, & |F(T,S)| > 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

3.4 Arquitectura del model

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 [20] 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 [44]. A continuació parlarem també d'algunes de les raons per les que **NO** hem optat per les altres arquitectures.

- **CNN**: El gran problema, més enllà de que no siga capaç de paral·lelitzar-se, és que per tal de capturar la relació entre n-grames fa falta un kernel de mida n, és a dir, si volem guardar la relació entre parells necessitem un kernel bidimensional, per a tripletes de paraules necessitem un tridimensional i així successivament. La qual cosa implica que per tal de capturar les dependències entre totes les possibles combinacions de paraules en una frase es dispara exponencialment el nombre de kernels necessaris.

- **RNN**: Com les paraules necessiten processar-se una per una perquè necessiten de l'estat ocult anterior, té una naturalesa seqüencial que impedeix la paral·lelització. I a banda, el fet que la seua "memòria" depenga dels estats ocults planteja un parell de problemes, per un costat que només guarda informació del que ha vist amb anterioritat (és a dir que va d'esquerra a dreta) i per al context és necessari tant el que hi ha abans com el que hi ha més endavant (tot i que açò es pot solventar plantejant un model bidireccional). Altre inconvenient és que el nombre de paraules durant el qual pot "recordar" una informació és molt relatiu, però és d'esperar que quan hagen passat unes quantes paraules haja oblidat eixa informació, que podia ser clau per entendre el context. Això és el que intenta mitigar LSTM com s'ha explicat en la secció d'estat de l'art però encara que funciona millor, tampoc és la millor alternativa perquè continua tenint els mateixos problemes d'arrel que RNN.

3.5 Models preentrenats

Com ja hem comentat en l'estat de l'art, en xarxes neuronals existeix una tènica que és el *transfer-learning* molt útil perquè permet reaprofitar el que ja han après altres models prè-

viament 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 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 partim del model NASca que va preentrenar el grup d'investigació ELiRF [44] seguint l'arquitectura de BART que s'especifica a continuació.

BART (*Bidirectional and Auto-Regressive Transformer*) [63] 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 3.3:

- *Token masking*: De forma aleatòria alguns tokens són substituis pel token especial [MASK], aquesta tècnica es feia servir ja en el model de BERT [55].
- *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.

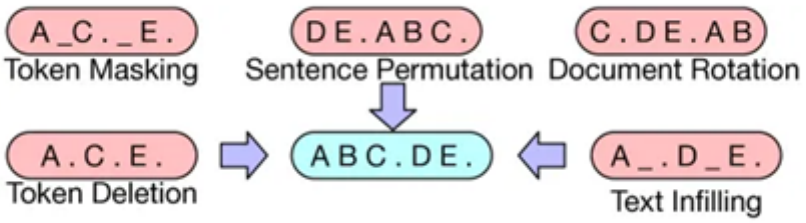

Figura 3.3: Exemples de tècniques per introduir soroll sobre les mostres d'entrada [63]

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 [63]. A més també s'ha pogut comprovar com afegir estes tasques durant el preentrenament milloren l'abstractivitat.

3.6 Entrenament NASca

En el TFG hem partit del model NASca (*News Abstractive Summarization for Catalan*) [44] preentrenat i ajustat pel grup d'investigació ELiRF. La tasca que hem realitzat en el TFG ha sigut continuar entrenant el model NASca (en la fase de *fine-tuning*), durant successives èpoques per tal de continuar amb l'aprenenatge del resum abstractiu de notícies i addicionalment afegir-li la tasca de classificació de les notícies en les diferents fonts. A partir d'ara ens referirem als models proposats en aquest TFG com 'model M1' al primer model que vam entrenar i 'model M2' per referir-nos al segon. En primer lloc passarem a explicar el model NASca, que tenim com a punt de partida.

NASca és un *Transformer* de tipus codificador-descodificador, que segueix la mateixa arquitectura i hiperparàmetres que BART [63]. Durant el preentrenament es va decidir incloure una sèrie de tasques addicionals per aportar-li coneixement lingüístic i amb l'objectiu d'incrementar l'abstractivitat dels resums que després generarà el model definitiu. En concret per a preentrenar el model que s'ha emprat en este projecte, es van aplicar: *text-infilling*, *sentence permutation*, *Gap Sentence Generation*(GSG) [27] i *Next Segment Generation* (NSG) [64]. Amb el *sentence permutation* i *text-infilling* s'espera que millore l'abstractivitat perquè li aporta al model la capacitat de reordenar contingut i substituir frases. Mentre que l'objectiu de GSG és proporcionar al model el coneixement per tal d'entendre un text com a conjunt, ser capaç de generar resums i parafrasejar. En última instància amb NSG es pretén augmentar la cohesió del resum, perquè aquesta tasca consisteix en a partir d'un prefix de partida generar una continuació. Aquest model va ser preentrenat utilitzant els documents en català del corpus DACSA que ja ha estat explicat (incloent aquells que havien estat descartats per a la fase d'entrenament), el subconjunt en català del corpus d'Oscar [58] i els documents que hi havia en Viquipèdia en el moment del preentrenament. En total van ser 2.5 milions de documents, la qual cosa posa de manifest el gran cost que suposa preentrenar un model i per què resulta tan útil la transferència d'aprenentatge que els *Transformers* saben aprofitar molt bé. Aquest model preentrenat és el que es va agafar com a punt de partida en el nostre model M2. De la mateixa manera que NASca es va preentrenar un model de manera equivalent però per a castellà: NASes, per la qual cosa es van haver de gastar 8.5 milions de documents perquè en tant que és una llengua amb més poder també té una major quantitat de recursos disponibles. Després estos dos models preentrenats van aprofitar-se per tal de ser ajustats mitjançant el corpus DACSA que hem descrit. I el resultat d'eixe ajust va ser agafat com a punt de partida en el nostre model M1.

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 passem a explicar quins són els diferents recursos software que hem decidit utilitzar per a aquest treball així com la justificació de per què els hem triat. Primer parlarem del llenguatge Python en general i després de les diferents llibreries que ens han ajudat a conduir a bon port aquest treball.

4.1.1. Python

Es tracta d'un llenguatge interpretat, d'alt nivell, orientat a objectes que tot i que és fins i tot més antic que Java, ha passat prou desapercebut durant bona part de la seua història, però en els últims temps s'ha convertit en un dels llenguatges més utilitzats i amb major previsió de creixement. Açò es deu a que és un llenguatge bastant senzill per iniciar-se en la programació i molt legible. Hem decidit utilitzar aquest llenguatge i no altre per diverses raons. És triat majoritàriament entre els desenvolupadors de ciència de dades i Intel·ligència Artificial, la qual cosa fa que existisca una àmplia comunitat que recolza el llenguatge i fa que aparega una major quantitat de fòrums de discussions o llibreries. Precisament les llibreries han sigut un altre punt clau, perquè hi ha disponibles una gran quantitat per a tot tipus de tasques, però en especial proporciona un gran ventall de possibilitats per a IA, com per exemple tensorflow, scikit-learn, keras, etc. Entre elles també es troba la llibreria Transformers de Huggingface que hem utilitzat com explicarem més endavant. Si bé podria parèixer una mala tria perquè és cert que Python no és tan ràpid com altres llenguatges, gràcies a l'ús d'algunes llibreries que estan implementades en C o en Rust i estan optimitzades per al tractament de grans quantitats de dades, per exemple permetent la paral·lelització, es converteix en un llenguatge tan potent com els altres.

4.1.2. NLTK

Natural Language Toolkit és un conjunt de llibreries per a ajudar al processament del llenguatge natural en Python. Proporciona llistes de paraules, corpus, i models lingüístics entre altres eines. En el nostre projecte en concret s'ha fet servir per a la subdivisió dels textos generats pel model en oracions, mitjançant la funció *sent-tokenize*.

4.1.3. Scikit-Learn

Es tracta d'una llibreria especialitzada en aprenentatge automàtic que ofereix eines útils tant en l'entrenament com en la posterior avaluació dels resultats. En concret nosaltres l'hem utilitzat per a calcular els resultats de accuracy, precisió, recall i F1 quan avaluàvem la capacitat que tenia el nostre model per classificar correctament les notícies en la seua font. S'ha decidit utilitzar aquesta llibreria per garantir la validesa dels resultats i no córrer el risc de que un error humà ens conduira a conclusions errònies.

4.1.4. NumPy

És una llibreria que dona suport al tractament de vectors i matrius. S'ha gastat per facilitar les operacions amb vectors i per raons de compatibilitat amb altres llibreries.

4.1.5. evaluate

Aquesta llibreria s'ha emprat per fer el càlcul durant la validació i avaluació de les mètriques de BERT-score i ROUGE.

4.1.6. HuggingFace

Una vegada hem decidit que anem a entrenar el nostre model mitjançant una arquitectura Transformers i python com a llenguatge de programació, encara ens quedava decidir quina llibreria utilitzàvem per portar a terme l'entrenament. HuggingFace va desenvolupar una sèrie de llibreries dedicades exclusivament a l'entrenament i avaluació de models Transformers que va fer que ens decantàrem per elles. Parlem en concret de dos llibreries:

- **Datasets**: Permet una gestió eficient de conjunts de dades molt grans, la qual cosa ens interessa especialment en el nostre cas donat que per a entrenar models Transformers (i en general per a xarxes neuronals) es precisen una gran quantitat de dades. Permeten entre altres coses guardar les dades en la memòria cau per no haver d'estar carregant-les i processant-les cada vegada, multiprocessament que permet repartir les tasques de processament per obtindre una acceleració, i una gran problemàtica és que si tenim conjunt de dades tan grans correm el risc que aquestes dades finalment no càpiguen en RAM, llavors per evitar això, aquesta llibreria utilitza un mecanisme de mapeig de memòria a través de l'estructura Apache Arrow que ho soluciona.
- **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 molt el procés de desenvolupament del codi d'entrenament, a més a més ofereixen un curs [65] on ensenyen d'una manera molt detallada però alhora molt didàctica i comprensible com fer servir aquestes llibreries i totes les possibilitats que ofereix. Aquestes raons van fer que fora la perfecta per a un nouvingut al món del PLN. Però no es tracta només de la comoditat que pot oferir al programador, sinó que a més estes llibreries estan optimitzades per tal de processar les dades tant en el preprocés del Dataset com durant l'entrenament, també integra optimizadors com Optuna per tal de fer una recerca prèvia a l'entrenament per decidir quins serien els millors hiperparàmetres i si per alguna raó durant l'entrenament s'ha detingut la màquina o hi ha hagut algun tipus de problema (que de fet s'han patit nombroses vegades durant la realització

d'aquest treball) permet reiniciar l'entrenament des de l'últim punt de control que haja guardat en lloc d'haver de tornar a començar des del principi i perdre tot el progrés fins el moment, funcionalitat gens menyspreable si tenim en compte el temps i recursos que consumeixen aquest tipus de models en entrenar-se. Si bé és cert que ofereix un inconvenient i és que no ens permet modificar l'arquitectura interna dels Transformers, per als objectius d'aquest projecte era més que suficient el que oferia la llibreria.

4.1.7. deepspeed

Es tracta d'una llibreria que permet accelerar entrenaments en PyTorch i permet reduir la potència i ús de memòria per entrenar models grans d'una manera distribuïda aprofitant millor el paral·lelisme del hardware (traient el màxim profit precisament a un dels grans avantatges que oferieix l'arquitectura Transformers que és la possibilitat de paral·lelització). Aquesta llibreria està integrada també per a poder ser utilitzada amb la llibreria Transformers de HuggingFace.

4.1.8. json

A banda de les llibreries vinculades a l'entrenament i l'avaluació, vam decidir generar els resums i guardar-los en fitxers json només acabara cada entrenament, per tal d'estalviar temps, generar-los una sola vegada i després ja poder fer totes les avaluacions que feren falta sobre ells sense haver de tornar a generar-los. I aquesta llibreria ens va proporcionar els mètodes necessaris per a l'escriptura i càrrega d'este tipus de fitxers. També va ser necessària per a la generació dels corpus per als entrenaments.

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 [66] 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.

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

- **Processado**: Intel(R) Core(TM) i9-10940X
- **Memòria RAM**: 128 GB DDR4

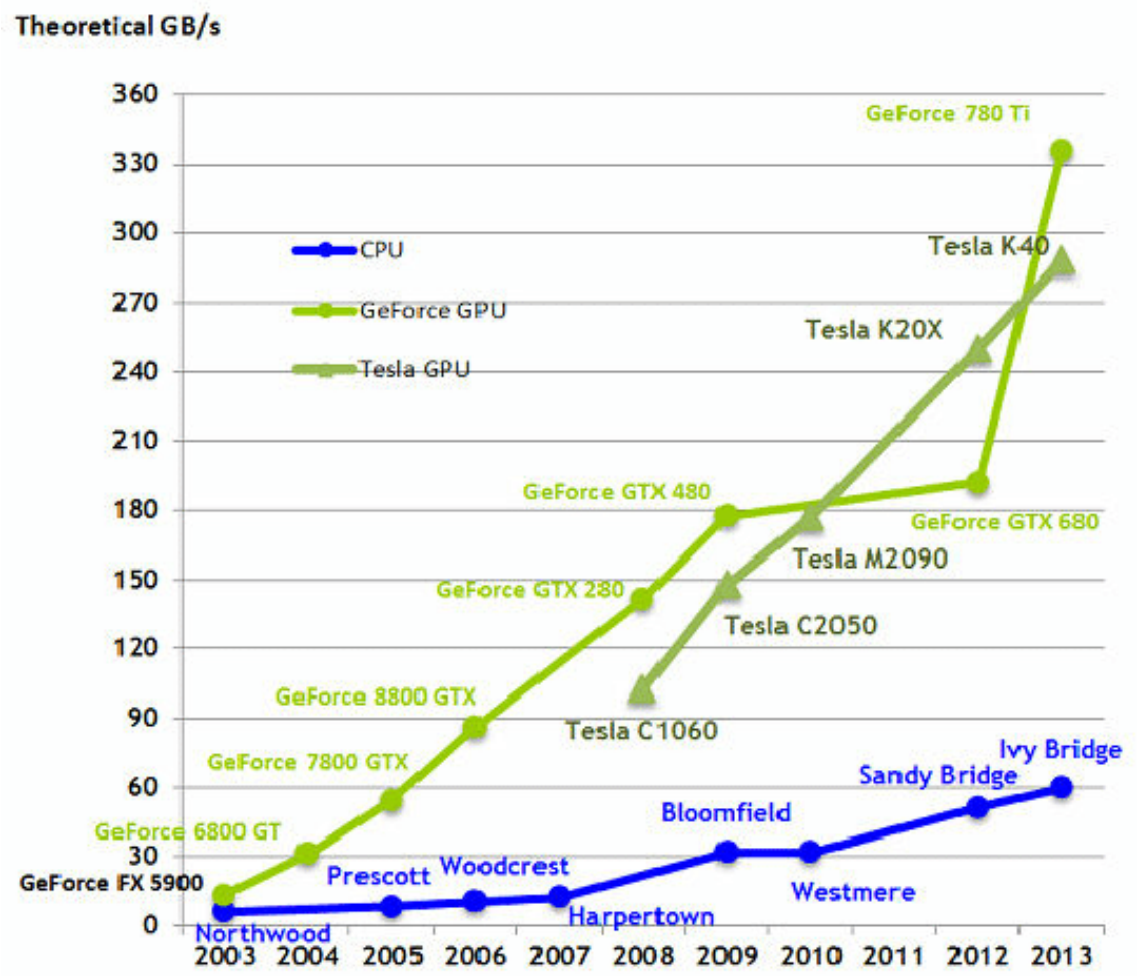

Figura 4.1: Comparativa ús GPU vs CPU per a entrenament de models d'IA [67]

- **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 dos gràfiques.
- **Sistema Operatiu**: Ubuntu 22.04.1 LTS
- **Entorn Python**: conda 23.1.0

CAPÍTOL 5
Experimentació i resultats

En aquesta capítol contarem tot el procés que hem seguit al llarg del desenvolupament del projecte, de quina manera hem preparat els diferents corpus, el codi per als entrenaments i per a l'avaluació posterior. 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. La hipòtesi al voltant de la qual gira tot aquest projecte és la capacitat per a interferir en els resums que genera el model si li donem certa informació prèviament, més enllà del propi article que ha de resumir.

5.1 Entrenament del model M1

5.1.1. Experimentació

En el primer entrenament el que s'hipotetitza en concret, és que entrenant un model que a banda d'especialitzar-se en resumir notícies, siga capaç també de detectar de quin diari (font) procedeix, els resums que generaria serien de major qualitat que els d'un model només dedicat al resum a banda de perquè ha estat entrenant-se durant una major quantitat d'èpoques per tant és d'esperar que ja simplement per eixa raó el rendiment obtingut siga millor. Sobre este model se pretén també estudiar quin és el seu rendiment com a classificador i analitzar si al model li forcem una font, això té algun tipus d'efecte sobre els resums generats, és a dir, si li diem la font correcta de la notícia resumeix amb millor qualitat perquè s'aproxima més al resum de referència? Si li forcem una font distinta a la que realment pertany, preval el seu criteri com a classificador o utilitza la font que li hem forçat i resumeix amb eixe estil? I això ens condueix a una nova pregunta: és capaç de detectar realment els distints estils a l'hora de resumir les notícies de les diferents fonts o les tracta totes d'igual manera?

Per a este primer entrenament vam partir del model NASca [44] que havia estat entrenat pel grup d'investigació ELiRF [2] i partia al seu torn d'un model preentrenat seguint l'arquitectura de BART[63] i també va ser afinat en tasques de resum automàtic, però s'havia entrenat durant només 6 èpoques i a més no se l'havia preparat per a que aprenguera a diferenciar entre les distintes fonts que composaven el corpus. Llavors el vam agafar com a punt de partida per veure quin efecte aconseguíem que tinguera el fet d'afegir la font en els resums.

La primera tasca que vam haver de portar a terme era preparar el corpus de cara a les noves necessitats: que fóra capaç de detectar la font, però el model no ha de generar vàries eixides, sinó que havia de generar una única seqüència que és el resum, llavors el que se va fer va ser que dins del propi resum hi haguera una etiqueta que indicara quina

era la font de la notícia. Per a fer això, com cadascuna de les mostres tenia entre altres, un camp *summary* on guardava el resum de referència, vam alterar-lo per tal que continguera 2 tokens especials al principi del resum, de manera que quedaria amb el següent format: *FONT <separador> resum_original* i això seria el que veuria el model i aprendria que els resums que ell generara, havien de tindre eixe mateix format. A més, també va ser necessari afegir al vocabulari nous tokens especials que representaren cadascuna de les fonts, perquè eren paraules que no havia vist anteriorment aleshores si no els afegim, no sabria tractar-les. Per raons de privacitat vam emprar tokens anonimitzats: CA01, CA02... Perquè no aparegueren els noms dels diaris dels quals s'havien extret les notícies.

Aquest entrenament vam haver de repetir-lo en diverses ocasions, perquè el model del qual partíem, no havia estat entrenat amb *HuggingFace*, i seguien l'esquema que marcava l'arquitectura del paper original de BART i llavors els textos seguien una representació interna lleugerament diferent de la que esperava el model, que el confonia. Per tal de corregir eixe error, el que vam fer va ser deixar d'utilitzar el *DataCollator* que oferia *HuggingFace* i s'encarregava de tokenitzar automàticament els textos d'entrada i aplicar el *padding*[1] i truncat[2], i vam passar a fer aquestes operacions nosaltres per tal de tindre total control i poder mantindre l'estructura original que esperava el model.

Mentre s'anava entrenant la versió ja adaptada d'aquest model, vam anar preparant els *scripts* per avaluar el seu rendiment segons les distintes mètriques que hem explicat sobre els conjunts *TEST-I* i *TEST-NI*. Vam preparar tres scripts, per veure si hi havia canvis substancials en els diferents escenaris. En tots els casos les mètriques analitzades van ser les mateixes que ja han estat explicades, *BERTscore* i *ROUGE* per a la qualitat de resum i precisió, recall i F1-score per a la classificació. Els 3 escenaris que vam plantejar són:

- **Sense alteracions**: Simplement li passàvem al model entrenat el contingut d'un article i ell ja s'encarregava de tokenitzar-lo i generar una eixida que contindria tant la font predita com el resum generat. A més en aquest cas s'ha afegit una matriu de confusió per a la classificació per tindre la informació el més esmicolada possible, agrupada per fonts, per detectar per exemple si hi ha fonts que té més problemes per identificar-les, fonts que tendeix a confondre amb altres, etc.

- **Forçant font correcta**: En aquest cas a banda de donar-li l'article, li passem la font a la qual pertany, que teníem emmagatzemada també en cadascuna de les mostres. Per fer això, el que feiem era forçar el primer token que es generava, i per poder aprofitar l'acceleració que suposava treballar amb batchs de notícies en paral·lel, vam haver de preprocessar el conjunt de dades per tal d'agrupar primer les notícies per font per tal de poder anar forçant els tokens de font correcta d'un en un. Aquest test per tant només podia aplicar-se sobre *TEST-I* perquè els diaris de *TEST-NI* no els ha vist mai i no estan contemplades dins de la classificació.

- **Forçant cadascuna de les fonts**: Per a cadascuna de les mostres que composen el corpus, li forçàvem cadascuna de les possibles fonts que podia predir.

Com ja comentarem més endavant en els resultats, es va detectar que des de les primeres èpoques, el model era capaç de classificar les mostres amb una taxa d'error molt xicoteta, la qual cosa ens va sorprendre i va portar-nos a sospitar que pot ser hi havia alguna petjada en el text dels articles que feia que el model de seguida fora capaç de detectar quina era la font i encertar-la. Per això vam tornar a entrenar el model exactament de la mateixa manera i amb el corpus DACSA, però aquesta vegada anonimitzant-lo.

[1] El padding consisteix en afegir al text tokens especials que no tenen significat, <*pad*>, per tal que tots els textos s'ajusten a la mateix mida, 512 tokens en el nostre cas.

[2] Si els textos superaven eixa mida de 512 tokens, se truncava i se deixava en 511 tokens més un </*s*> per indicar la fi del resum

Aquest entrenament va durar 3 dies sencers en executar-se, amb una duració de 12h per època.

5.1.2. Analisi de resultats

En aquest apartat comentarem els resultats que hem obtingut dels distints tests que s'han efectuat sobre aquest model, així com dels que es van anar obtenint durant la validació.

Resultats de la validació

En les Figures 5.1 i 5.2, que representen els valors de la *loss* i de les mètriques *ROUGE* respectivament obtingudes durant la validació. Pot observar-se que el nostre model va anar millorant durant tot el seu entrenament perquè les mètriques de *ROUGE* van anar en increment, però com se comentava en l'apartat d'experimentació, per alguna raó la *loss* va anar pujant llavors els models que resultaven de cada època a ulls del programa estaven sent pitjors, per tant emmagatzemava els primers models que eren els millors, perquè per defecte intenta minimitzar la *loss* si no li especifiquem un altre criteri. Les mètriques que se van tindre en compte en la validació van ser les de *ROUGE*-1, *ROUGE*-2, *ROUGE-L*, *ROUGE-Lsum*, i en tots els casos segueixen la mateixa tendència, perquè podem veure que la forma de les quatre rectes és la mateixa, tenen una tendència ascendent durant les quatre primeres èpoques i després hi ha un davallada en l'última. Tot i que en tots els casos el canvi que hi ha és molt súbtil, donat que el marge és de 1.5 centèssimes. A continuació passem a exposar els resultats que s'han obtingut durant la fase de test sobre els dos subconjunts preparats per a tal efecte: *TEST-I* i *TEST-NI*.

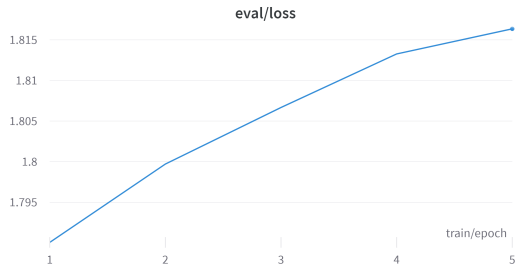

Figura 5.1: Evolució de la *loss* en valdiació durant la fase de *fine-tuning* del model M1

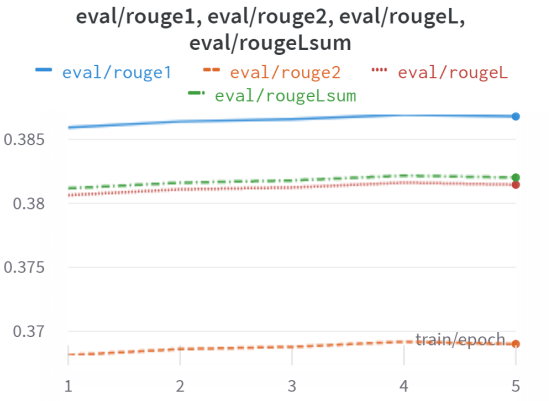

Figura 5.2: Evolució de les diferents mètriques de ROUGE en validació durant la fase de *fine-tuning* del model M1

Resultats del model cru

En la Taula 5.1 s'observa una comparativa de les mètriques ROUGE i BERTscore entre el nostre model M1 respecte del model NASca [44] original per a ambudes particions de test. Podem observar que malgrat que el model original s'havia entrenat només durant 6 èpoques i que ara l'hem continuat entrenant per a resum automàtic durant 5 èpoques més i que en la validació sí que semblava que millorava els resultats, realment no podem afirmar que resumisca millor, perquè en el subconjunt de *TEST-I ROUGE*-1 emoitjora mentre que la resta milloren; però en qualsevol cas els valors varien en un interval molt

xicotet, és a dir que té bàsicament la mateixa capacitat de resum. I pel que fa a *TEST-NI*, s'observa tant un empitjorament respecte del nostre model en *TEST-I*, la qual cosa era d'esperar perquè és el mateix que va passar en NASca, però la proporció en la qual baixa és major, i a més s'empitjoren els resultats respecte de NASca per a este mateix subconjunt. Açò pot explicar-se perquè és un subconjunt conformat per notícies escrites per diaris dels quals el model mai ha vist cap notícia, llavors en totes les mètriques de *ROUGE* s'han obtingut pitjors resultats, la qual cosa té sentit en tant que ara el nostre model està especialitzat en dos tasques: classificació i resum, però el primer que fa és intentar classificar la notícia, i per tant ho intenta fer evidentment en alguna de les fonts que ja ha vist, però mai és correcta, i intentarà resumir-la pensant-se que és una font incorrecta, per la qual cosa és d'esperar que els resums obtinguts siguen més diferents dels de referència. No obstant la variació dels resultats des de *TEST-I* a *TEST-NI* no és tan gran, la qual cosa ens indica que el model continua tenint una bona capacitat de generalització, encara que siga una miqueta pitjor que NASca.

Partició	Model	*ROUGE*-1	*ROUGE*-2	*ROUGE-L*	*ROUGE-Lsum*	*BERTscore*
TEST-I	NASca	28.84	11.68	22.78	23.30	71.85
	model M1	28.51	11.82	22.87	23.33	72.02
TEST-NI	NASca	28.19	11.20	21.45	22.44	70.14
	model M1	27.40	11.03	21.10	21.98	70.13

Taula 5.1: Taula comparativa mètriques resum automàtic NASca vs model M1

Com també volíem analitzar la qualitat d'aquest model com a classificador, hem plantejat una sèrie de mètriques per analitzar-ho com podem veure en la Taula 5.2 on es mostren les mètriques explicades prèviament per avaluar el nostre model com a classificador de fonts periodístiques: *accuracy*, precisió, *recall* i *F1-socre* així com el percentatge de resums que tenien el format esperat. En este cas només s'ha pogut efectuar sobre la patició *TEST-I* perquè són les notícies de les quals sí ha vist prèviament la seua font i per tant pot detectar-la. Aquests percentatges reflexen uns resultats molt positius, és a dir, el nostre model és capaç de predir quasi sense enganyar-se a quina font pertany cada notícia, a més a més, este comportament s'ha pogut observar ja des de les primeres èpoques, és a dir que ho ha après amb molta rapidesa. També s'ha incorporat la comprovació del format correcte perquè com ja s'ha comentat vam enfrontar-nos al principi a una sèrie de problemes precisament de format, i una vegada aplicats els canvis que s'han explicat per deixar de fer servir el *DataCollator* i fer-ho amb el nostre propi codi, des de la primera època aconseguia el format correcte en el 100% de les ocasions. Però també volíem tindre una informació més esmicolada, repartida per fonts, llavors vam calcular la matriu de confusió, on les files representen la font correcta (etiqueta) de la notícia i les columnes la font on el model les ha classificat.

Accuracy	Precisió	Recall	F1-Score
92.36	93.03	92.36	92.19

Taula 5.2: Mètriques classificació per al model M1 sobre TEST-I

La Taula 5.3 mostra una matriu de confusió en la qual en cada fila es representa la font original de les notícies avaluades i les columnes la font en la qual ha classificat el model M1 la notícia, i el valor en cada cel·la $M[i][j]$ representa el percentatge de notícies que són originalment de la font i i que han sigut classificades en j. D'aquesta manera podem extraure informació més enllà de simplement concloure si els resultats de classificació han sigut bons o roïns. Com era d'esperar a la vista dels excel·lents resultats de classificació, la suma de la diagonal supera el 90%. El que queda fora de la matriu és en quasi tots

els casos residual, però sí que podem destacar que el classificador es confón prou més vegades a l'hora de classificar *CA*03 i especialment *CA*04 del que ho fa amb la resta de fonts. Açò pot deure's a que siguen fonts que realment no tenen un estil tan marcat a l'hora de realitzar els articles o que se confonen amb l'estil d'altres diaris dels quals ha vist més mostres, perquè si parem atenció, la major part de les vegades que s'ha enganyat ha sigut a favor de la font *CA*01 que és precisament de la qual hi ha més mostres en el conjunt d'entrenament. I també resulta curiós que la font *CA*04 la confundisca bastants vegades amb la *CA*03 però en canvi a l'inrevès ocorre en poques ocasions. Un exemple d'aquesta confusió el podem veure en l'apèndix B, on els resums sense forçar la font i forçant la font CA03 (que és en la qual classifica la notícia) són idèntics. Però el resum que obté en forçar la font CA04 és idèntic a excepció que no menciona Ortega i Rigau, sinó que utilitza una anàfora com si s'haguera referit ja a elles anteriorment. La qual cosa reforça la idea que aquesta confusió ve donada perquè les notícies de CA04 tenen un estil molt semblant a les de CA03.

	CA01	CA02	CA03	CA04	CA05	CA06
CA01	**16.11**	0.05	0.36	0.12	0.02	0.01
CA02	0.09	**16.55**	0.03	0.00	0.00	0.00
CA03	1.00	0.05	**15.21**	0.40	0.01	0.01
CA04	2.13	0.03	2.79	**11.52**	0.01	0.01
CA05	0.03	0.01	0.04	0.01	**16.58**	0.01
CA06	1.90	0.02	0.03	0.00	0.03	**16.40**

Taula 5.3: Matriu de confusió per al model M1 sobre TEST-I

En la Taula 5.4 trobem una altra matriu de confusió que representa el mateix que l'anterior sols que aquesta vegada s'ha avaluat contra l'altra partició de test: *TEST-NI*. Evidentment es queda tot concentrat en les files que se corresponen amb les fonts que pertanyen a aquest test: *CA*07, *CA*08 i *CA*09 i en les columnes de les fonts de *TEST-I* que són les que ha vist: *CA*01-*CA*06, en este cas clarament les classifica totes incorrectament, però podem destacar un fet que va resultar prou sorprenent, i és que en un 71.19% de les ocasions les ha classificat en *CA*01, és a dir, o bé les 3 fonts dona la casualitat que tenen un estil molt semblant al de *CA*01 o el que és més probable, és que quan li arriba una notícia d'una font que no ha vist mai, és a dir, està escrita amb un estil que no li sona de res, doncs sempre tendeix a classificar-ho en aquella que més ha vist, que és *CA*01. En canvi per a *CA*08 també hi ha un alt percentatge de notícies que són classificades en la *CA*06, la qual cosa no pot explicar-se per esta mateixa raó, perquè aquesta és la font que menys pes té dins del conjunt d'entrenament. En aquest cas sí que podria explicar-se perquè *CA*06 i *CA*08 tinguen un estil de redacció semblant. En un intent de demostrar aquesta hipòtesi o trobar alguna particularitat que puguera explicar aquest fenòmen, revisant les fonts que composen el corpus, es va descobrir que precisament aquestes dos fonts són els únics diaris valencians que hi ha en tot el corpus, tota la resta són catalans. La qual cosa és prou impressionant, perquè en este cas fins i tot ha sigut capaç de diferenciar les variants dialectals, perquè quan li arriben notícies d'un diari que no ha vist mai, però que empra el dialecte valencià, tendeix a classificar-lo o bé en aquella font que més ha vist, o bé en l'únic diari valencià que coneix. I a més açò reforça la idea què la nostra Intel·ligència Artificial és capaç de classificar les notícies a base de l'estil de redacció, el vocabulari, etc. Que en definitiva és el que ha fet que detectara una notícia com més semblant a aquelles dels diaris valencians que dels catalans.

	CA01	CA02	CA03	CA04	CA05	CA06
CA01	0.00	0.00	0.00	0.00	0.00	0.00
CA02	0.00	0.00	0.00	0.00	0.00	0.00
CA03	0.00	0.00	0.00	0.00	0.00	0.00
CA04	0.00	0.00	0.00	0.00	0.00	0.00
CA05	0.00	0.00	0.00	0.00	0.00	0.00
CA06	0.00	0.00	0.00	0.00	0.00	0.00
CA07	**38.50**	**0.36**	**0.31**	**0.06**	**0.57**	**0.04**
CA08	**14.96**	**2.32**	**1.59**	**0.16**	**2.64**	**11.32**
CA09	**17.73**	**4.81**	**0.46**	**0.02**	**1.08**	**3.09**

Taula 5.4: Matriu de confusió per al model M1 sobre TEST-NI

Resultats del model forçant fonts

Però l'objectiu d'aquest treball era analitzar si podia condicionar-se d'alguna manera la generació de resum. En aquest model en concret, es pretenia aconseguir-ho indicant-li al model una font (correcta o no) a la qual pertany la notícia que està resumint. La Taula 5.5 veurem els resultats que es van obtindre forçant la font correcta i forçant cadascuna de les altres fonts que ha vist, de manera indiscriminada. Els resultats forçant la font correcta, només ha pogut efectuar-se sobre el *TEST-I* perquè les fonts correctes de *TEST-NI* no les ha vist mai.

Partició	Font forçada	*ROUGE*-1	*ROUGE*-2	*ROUGE-L*	*ROUGE-Lsum*	*BERTscore*
TEST-I	Correcta	28.49	11.81	22.85	23.30	72.02
	CA01	28.42	11.76	22.81	23.26	72.00
	CA02	28.40	11.74	22.79	23.22	71.99
	CA03	28.40	11.73	22.76	23.20	71.98
	CA04	28.43	11.76	22.80	23.25	71.99
	CA05	28.43	11.77	22.79	23.23	71.99
	CA06	28.44	11.78	22.82	23.27	72.00
TEST-NI	CA01	26.25	10.29	20.02	20.89	69.05
	CA02	26.25	10.30	20.06	20.91	69.04
	CA03	26.25	10.36	20.07	20.93	69.06
	CA04	26.21	10.29	20.00	20.86	69.03
	CA05	26.29	10.37	20.01	20.95	69.05
	CA06	26.31	10.39	20.10	20.97	69.06

Taula 5.5: Taula comparativa de mètriques resum automàtic model M1 forçant fonts

Els resultats van ser un poc descoratjadors, donat que esperàvem que realment el fet de forçar o no la font tinguera un efecte per mínim que fóra, i realment no ha sigut així. En el cas de forçar la font correcta, els resultats són pràcticament idèntics, la qual cosa sí era d'esperar, perquè com ja hem vist en la Taula 5.2, aquest model no s'enganya pràcticament a l'hora de classificar les fonts de les notícies, per la qual cosa li donarà igual que li la diguem inicialment o no, perquè si no li la diem quasi sempre serà capaç d'encertar-la, per tant s'esperava que els resultats no es veieren afectats per aquesta mesura. El que sí ha resultat més sorprenent ha sigut que en forçar cadascuna de les fonts, al *TEST-I* no li ha importat, els resultats han continuat sent els mateixos, la qual cosa ens porta a pensar que realment els resums que genera li diguem la font que li diguem continuen sent els mateixos, és a dir que a l'hora de resumir realment no segueix un estil diferent segons siga la font. Una altra opció seria que encara que li hagem dit una font incorrecta, el model manté el seu propi criteri i resumeix de forma correcta com si no li haguérem forçat cap font. Per eixir de dubtes i comprovar què està passant amb estos resums i si realment hi

ha diferència entre el resum que genera només donant-li la notícia i el que genera quan li diem una font, hem preparat un *script* que genera dos matrius, on les files són la font original i les columnes la font que s'ha forçat, en una es trau el percentatge de resums en els quals hi ha algun tipus de diferència entre el resum generat sense forçar la font i forçant la font que indica la columna. I una altra que les compara mitjançant la mètrica *ROUGE-Lsum* per poder quantificar com de diferents són:

Les Taules 5.6 i 5.7 representen respectivament, quin percentatge de notícies s'han resumit de manera diferent quan hem forçat una font i quan no ho hem fet i per avaluar també com de diferents han sigut eixos resums hem afegit una altra matriu on se compararen mitjançant *ROUGE-Lsum*. S'ha decidit mantindre una representació matricial per tal de poder obtindre el màxim d'informació possible, de manera que les files novament marquen quina és la font original i les columnes ara representen la font forçada.

	CA01	CA02	CA03	CA04	CA05	CA06
CA01	2.19	10.39	25.92	15.33	12.48	13.55
CA02	10.96	0.22	20.05	13.33	10.39	10.04
CA03	21.61	18.73	6.36	22.34	20.49	18.54
CA04	19.79	18.27	21.81	13.35	18.96	18.67
CA05	9.79	7.62	18.44	9.89	0.29	8.21
CA06	11.67	9.23	24.03	13.18	9.26	0.71

Taula 5.6: Matriu de percentatge de resums diferents forçant i no forçant la font pel model M1 sobre *TEST-I*

	CA01	CA02	CA03	CA04	CA05	CA06
CA01	99.01	95.96	91.02	93.64	94.89	94.55
CA02	95.81	99.89	92.08	94.37	95.72	96.02
CA03	91.71	92.39	97.13	91.45	91.90	92.57
CA04	91.94	92.45	91.48	94.66	92.12	92.29
CA05	96.11	97.04	93.33	95.99	99.88	96.68
CA06	94.97	95.78	91.47	94.42	95.92	99.64

Taula 5.7: Matriu de *ROUGE-Lsum* entre resums generats forçant i no forçant la font pel model M1 sobre *TEST-I*

El que podem atendre en les taules 5.6 i 5.7 és que, com era d'esperar, en les diagonals els percentatges de la primera matriu són molt baixets i els de la segona molt alts, perquè la primera com més baixet siga vol dir que hi ha hagut una menor quantitat de mostres per a les quals els resums no són idèntics i en la segona, com més gran siga el valor vol dir que major és la seua semblança. En el cas de la diagonal es deu a que els resums que se generen forçant la font i no forçant-la són gairebé els mateixos, perquè com ja hem explicat anteriorment, si quasi sempre encerta la font, és d'esperar que el resum generat siga el mateix, realment és com si no li estiguerem aportant nova informació. Tot i que sí que cal destacar que en els casos de $CA03$ i $CA04$ la diferència entre els resums generats és notablement major tot i que la font forçada siga la correcta, açò està vinculat amb el fet que hem comentat en la Figura 5.3 què aquestes dos fonts precisament són en les que troba una major dificultat a l'hora de classificar-les, aleshores quan ha generat el resum sense dir-li quina font era, en una major proporció de casos s'ha enganyat, i per tant ha generat resums que són més diferents que quan li hem dit quina era la font correcta. Pel que fa als resultats que estan fora de la diagonal el que es pot observar és que hi ha una major quantitat de diferències, és a dir, que el model sí té en compte la font que li hem proporcionat a l'hora de resumir una notícia, i segons la font que siga, resumeix d'una

manera o altra, no obstant, encara que estos percentatges ronden el 10 i 20% de resums que són diferents, si atenem al valor del *ROUGE* realment són resums molt semblants els que se generen.

5.2 Entrenament del model M2

5.2.1. Experimentació

Com vam veure que els resultats de classificació van ser tan bons i en canvi no es veia una gran diferència en els resums que es generaven quan es forçaven les diferents fonts; encara quan vam fer servir el corpus anonimitzat, ens vam adonar que durant eixes 6 èpoques que el model de NASca s'havia estat entrenant i que nosaltres agafem com a punt de partida per al nostre entrenament, les mostres que havia vist no havien estat anonimitzades, llavors cabia la possibilitat que tot i que ara estiguera veient els articles anonimitzats, encara recordara part d'eixa informació que no estava anonimitzada, per tant l'objectiu que es va buscar amb este segon entrenament era descartar ja per complet que els resultats tan exitosos en la classificació vingueren donats per la presència de pistes en el text i que fora perquè efectivament es tractava d'un bon classificador que era capaç de detectar quasi sempre la font correcta.

Per a l'entrenament d'aquest model, al qual ens referirem d'ací endavant com 'model M2', hem utilitzat pràcticament el mateix codi que en l'entrenament anterior, l'única cosa que canvia ha sigut el model que s'agafava com a punt de partida per fer el *fine-tuning*, que va ser, el model que es gastava també com a preentrenat per entrenar NASca. Es va afegir una opció perquè en cada epoch es guardaren els resums que s'havien generat, per tal de poder anar fent proves a la vegada que s'estava entrenant, es va determinar que el criteri per seleccionar (i guardar) el millor model no fora minimitzar la *loss* (criteri per defecte), sinó maximitzar el *ROUGE-Lsum* perquè ja s'havia vist que per alguna raó la *loss* anava pujant en entrenaments anteriors, i com realment la classificació ja se feia amb percentatges de rendiment molt alts des de les primeres èpoques, llavors ens interessava que se centrara en millorar la seua capacitat de resum; no obstant es van afegir les mètriques de classificació a la validació, per tal de poder anar veient la seua evolució al llarg de l'entrenament, tot i que no es tingueren en compte per triar millor model.

Pel que fa als *scripts* d'avaluació dels resultats d'aquest entrenament ens serveixen els mateixos que ja teníem, perquè el que hem canviat ha sigut el model del qual partíem per fer el *fine-tuning* però el resultat generat i el que volem avaluar segueix sent el mateix. No obstant, per diferents raons vam decidir incorporar noves avaluacions, que no havíem aplicat sobre el model anterior perquè encara no era el definitiu i no valia la pena exprimir-lo tant. Els escenaris que s'han avaluat en aquest apartat són:

- **Quantificar qualitat entre forçar la font i no forçar-la**: Front als resultats que demostraven que havia hagut una diferència més significativa en este nou model quan li donàvem la font com a entrada i quan no ho feiem, vam decidir que no era prou amb comparar els resums entre sí, sinó que seria interessant comparar ambdós amb el resum de referència i vam calcular una matriu on cada valor $M[i][j]$ representa el *ROUGE-Lsum* entre el resum generat forçant la font j sobre una notícia de la font i, respecte del seu resum de referència; i un vector amb les notícies també agrupades per font però sense forçar res i amb això puguérem traure una matriu normalitzada que mostra si hi ha una millora o un empitjorament de la qualitat dels resums segons aquesta mètrica.

- **Resums més llargs**: Els resultats que es van obtindre entre forçar una font qualsevol i no forçar-la en els resums generats per aquest model van ser pràcticament idèntics i revisant alguns dels resums generats es va observar que tenia igual forçar una font o una altra, el resum era idèntic en moltes ocasions. Llavors es va plantejar que potser podia donar-se perquè els resums que s'estan generant realment són molt curts degut a que s'agafa com a referència el titular i aleshores no és suficient per introduir l'estil propi de cada diari, llavors per a això era interessant comprovar què passava si se li demanaven resums que foren més llargs. En este cas vam traure la longitud mitjana de tokens de tots els resums de test i ho vam multiplicar per 2, quedant així doncs uns resums amb longitud mínima de 70 tokens i 420 de màxima. El problema dels resultats que obtinguem en esta avaluació, és que com els resums que nosaltres tenim de referència són realment més curts, la comparativa amb ells no és profitosa perquè no està en les mateixes condicions, llavors el més pròxim que tenim a un resum correcte i que per tant fem servir com a referència, són els resums que ha generat el propi model quan li hem forçat la font correcta, és a dir que realment els resultats de les mètriques de resum automàtics no són rigurosos, però el que volem avaluar en este cas tampoc no era això, sinó que el que ens interessa són dos coses: veure quin efecte té sobre l'abstractivitat el generar resums que siguen més llargs i sobretot veure si hi ha una major diferència entre els resums generats segons la font que se li diga. Perquè la nostra hipòtesi és que si els resums que genera el model són més llargs, és capaç d'imprimir una major quantitat de trets estilístics que diferencien a unes fonts d'altres.

- **Generar n-resums**: Per veure si tenia algun efecte el fet d'estar agafant només la primera seqüència que generara el model, vam decidir també que per cada notícia es generara en lloc de només una seqüència d'eixida que se generaren vàries per veure si en les subsegüents el resultat variava o seguia en la mateixa línia, en el nostre cas en concret vam decidir generar 4 resums per cadascuna de les mostres. I es van avaluar per separat els resums que se generaven en cada posició, és a dir se van avaluar per separat tots els primers resums, els segons, etc.

- **Avaluar abstractivitat**: Com ja s'ha explicat, existeixen dos tipus de resums: els abstractius i els extractius, en el nostre cas el model que hem entrenat està preparat per ser abstractiu, que és el que tendim a fer els éssers humans, no només extraure frases del text sinó reformular les idees més importants que extraem d'ell. Però no basta amb afirmar que el nostre model genera resums abstractius, i com existeixen una sèrie de mètriques per avaluar-ho les apliquem. De fet les aplicarem sobre totes les combinacions que ja hem parlat: forçar les distintes fonts (correctes i incorrectes), resums més llargs, etc. Per veure si aplicant estes variacions s'aconsegueix variar els resultats de l'abstractivitat.

En este cas com partíem del model preentrenat i hem hagut d'aplicar més èpoques, la duració de l'entrenament ha sigut de pràcitcament 9 dies sencers amb una duració de 13h per època. I esta vegada a l'hora d'avaluar el model, en lloc de llançar cada vegada els *scripts* que teníem preparats de l'anterior vegada que generaven cadascun els resums, el que vam pensar va ser generar una sola vegada tots els possibles resums que puguerem necessitar: forçant la font, sense forçar-la, amb resums més llargs, etc. I així en qualsevol moment podríem realitzar els tests d'una manera molt més ràpida i eficient sense haver de regenerar el mateix resum que ja s'ha generat en altres ocasions i que tardava al voltant d'1 hora per a TEST-I i mitja per a TEST-NI.

5.2.2. Analisi de resultats

Resultats de la validació

La Figura 5.3 representa l'evolució de la *loss* durant les successives validacions al llarg de l'entrenament. En aquest entrenament sí que podem observar com té un comportament més predictible durant la validació, donat que va reduint-se, no obstant, baixa només fins la huitena època, després comença a incrementar-se, tenint llavors un comportament més semblant del que pareix a primera ullada a l'entrenament anterior, perquè en aquest cas partim del model *pre-NASca* i *NASca* realment ja havia fet 6 èpoques, per la qual cosa ambdós models després de més o menys el mateix temps comencen a empitjorar la seua *loss*. I en la Figura 5.4, que descriu l'evolució del ROUGE durant la validació, podem observar com amb el *ROUGE* també passa el mateix que en l'anterior, que és que les quatre mètriques tenen la mateixa tendència i totes milloren, encara que el marge en el qual millora és també xicotet.

Figura 5.3: Evolució de la *loss* en valdiació durant la fase de *fine-tuning* del model M2

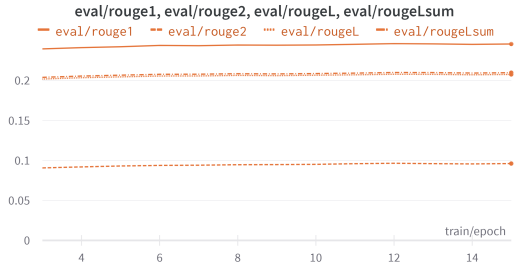

Figura 5.4: Evolució de les diferents mètriques de ROUGE en valdiació durant la fase de *fine-tuning* del model M2

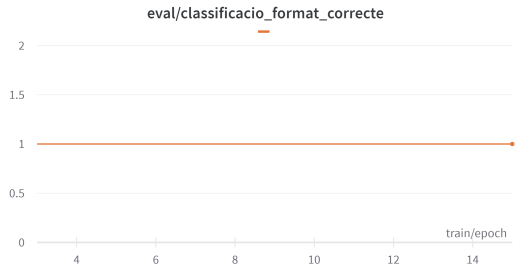

Figura 5.5: Evolució de la correcció del format de resum en valdiació durant la fase de *fine-tuning* del model M2

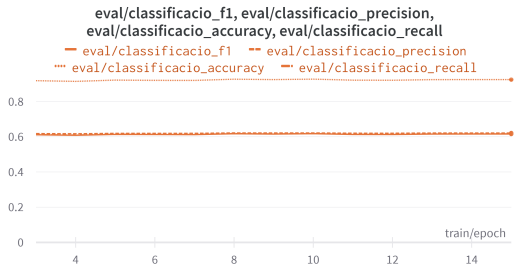

Figura 5.6: Evolució de les diferents mètriques de classificació en valdiació durant la fase de *fine-tuning* del model M2

Per a este entrenament també vam incloure en la validació l'avaluació de les mètriques relatives a la classificació. Les Figures 5.5 i 5.6 il·lustren l'evolució de les mètriques de format correcte i classificació de fonts respectivament. En elles pot apreciar-se com des del primer moment el model sap quina estructura ha de tindre el resum que genere, perquè té sempre un 100% de casos on ho encerta. I després a l'hora de la classificació, l'*accuracy* sí que té uns bons valors com en els del primer entrenament, però les altres mètriques tenen un valor sospitosament xicotet en comparació amb l'anterior, la qual cosa va fer que durant l'entrenament pensàrem que fora correcta la nostra hipòtesi i realment el primer model haguera aconseguit els bons resultats perquè durant l'entrenament de NASca havia vist les mostres sense anonimitzar i ara que les veu des del primer moment anonimitzades no és capaç de classificar-les tan correctament. Però no obstant per poder confirmar la hipòtesi no basta amb fer els resultats de la validació, sinó que fa falta

esperar-se a veure els resultats després del test. Altre detall important és que per a la tasca de classificació no hi ha a penes variació al llarg de l'entrenament, això vol dir que és pràcticament igual de bo en la classificació després d'una sola època que després de 15, la qual cosa ens porta a pensar que troba prou "fàcil" especialitzar-se en descobrir quina és la font de les notícies.

Resultats del model cru

La Taula 5.8 compara les mètriques que ha obtingut el model M2 amb les que va obtindre el NASca original, per veure quin dels dos generava uns resums de major qualitat (més semblants als de referència). El que podem extraure d'aquests resultats és que empitjoren els resultats respecte de NASca, encara que siga poquet, també empitjoren els resultats per a *TEST-I* respecte del primer model que teníem en la Taula 5.1 però en canvi per a *TEST-NI* els resultats milloren, la qual cosa ens porta a pensar que aquest model té una major capacitat de generalització per haver aplicat l'anonimitzat del corpus. Pel que fa a les mètriques relacionades amb la classificació.

Partició	Model	*ROUGE-1*	*ROUGE-2*	*ROUGE-L*	*ROUGE-Lsum*	*BERTscore*
TEST-I	NASca	28.84	11.68	22.78	23.30	71.85
	model M1	28.51	11.82	22.87	23.33	72.02
	model M2	28.22	11.58	22.55	23.00	71.35
TEST-NI	NASca	28.19	11.20	21.45	22.44	70.14
	model M1	27.40	11.03	21.10	21.98	70.13
	model M2	27.44	11.11	21.09	22.00	71.15

Taula 5.8: Taula comparativa mètriques resum automàtic NASca vs model M2

En la Taula 5.9 podem veure reflectits els resultats que ha obtingut el model M2 com a classificador de fonts periodístiques. I podem comprovar que realment la hipòtesi no era certa, ha sigut un cas aïllat del corpus de validació, perquè els resultas que ha obtingut el test són els mateixos que va obtindre el primer model, llavors la raó per la qual el model obté tan bon rendiment en la classificació no és perquè quedara cap "pista" dins de l'article, sinó que és perquè el model troba suficient informació en el text com per poder triar la font correctament. No obstant per acabar de descartar qualsevol sospita, se van entrenar dos models especialitzats només en la classificació d'articles i de resums respectivament, per veure amb més detall que és el que està passant i que discutirem en la següent secció.

Accuracy	Precisió	Recall	F1-Score
92.35	92.35	93.09	92.19

Taula 5.9: Taula mètriques classificació per al model M2 sobre TEST-I

En les Figures 5.10 i 5.11 tenim novament unes matrius de confusió que representen els percentatges de notícies que eren d'una determinada font (fila) i el model M2 ha classificat en la font que indica la columna.

Pel que fa a la matriu de confusió de *TEST-I* de la Taula 5.10 podem veure que els resultats obtinguts són molt semblants als del model anterior, com era d'esperar, perquè ja hem vist que els resultats de la classificació continuaven sent els mateixos. La major part dels percentatges s'acumula en la diagonal perquè és on indica que ha encertat en la seua predicció i després hi ha valors residuals en la resta de cel·les de la matriu, i tornen a ser les notícies de $CA03$ i especialment $CA04$ on més s'enganya a l'hora de fer la classificació. Per a *TEST-NI*, com veiem en Taula 5.11 són també iguals que els del primer

model, amb la diferència que sembla que el que destacàvem que era capaç de detectar les variacions dialectals i en el moment que rebia una notícia en valencià tendia a classificar-la en l'únic diari valencià que coneixia, ara amb l'anonimitzat pareix que ha perdut un poc eixa capacitat, perquè ha baixat el percentage de notícies de $CA08$ que classifica en $CA06$ en favor de classificar-la en la font que més ha vist: $CA01$.

	CA01	CA02	CA03	CA04	CA05	CA06
CA01	**16.29**	0.05	0.24	0.08	0.01	0.00
CA02	0.08	**16.55**	0.03	0.01	0.00	0.00
CA03	1.22	0.06	**14.99**	0.38	0.01	0.01
CA04	2.41	0.03	2.72	**11.49**	0.01	0.01
CA05	0.03	0.01	0.03	0.00	**16.59**	0.00
CA06	0.20	0.00	0.01	0.00	0.01	**16.44**

Taula 5.10: Matriu de confusió per al model M2 sobre TEST-I

	CA01	CA02	CA03	CA04	CA05	CA06
CA01	0.00	0.00	0.00	0.00	0.00	0.00
CA02	0.00	0.00	0.00	0.00	0.00	0.00
CA03	0.00	0.00	0.00	0.00	0.00	0.00
CA04	0.00	0.00	0.00	0.00	0.00	0.00
CA05	0.00	0.00	0.00	0.00	0.00	0.00
CA06	0.00	0.00	0.00	0.00	0.00	0.00
CA07	**37.87**	**0.31**	**0.37**	**0.03**	**1.22**	**0.02**
CA08	**16.94**	**1.84**	**2.20**	**0.17**	**2.69**	**9.13**
CA09	**17.91**	**3.51**	**0.59**	**0.28**	**1.71**	**3.18**

Taula 5.11: Matriu de confusió per al model M2 sobre TEST-NI

A continuació veurem si per a este nou model, té algun efecte sobre els resums generats el fet de forçar la font correcta o qualsevol altra font en general amb els resultats de les mètriques de resum *ROUGE* i *BERTscore* quan forcem les distintes fonts i a més afegim una matriu normalitzada que compara els resultats de *ROUGE-Lsum* quan forcem cadascuna de les fonts contra el resum de referència front a la mateixa mètrica però amb els resums generats sense dir-li la font front al mateix resum de referència per veure amb major claredat l'efecte d'empitjorament o millora de la qualitat dels resums organitzat per font.

Resultats del model forçant fonts

La Taula 5.12 mostra els resultats relatius a la qualitat dels resums (ROUGE i BERTscore) que ha obtingut el model M2 forçant tant la font correcta com cadascuna de les fonts possibles. Podem observar el mateix comportament que en el model anterior, que quan forcem la font correcta el resultat és idèntic a no dir-li res perquè en una alt percentatge dels casos ja encerta el model per ell mateix la font, i quan forcem qualsevol de les altres fonts, en especial quan forcem les fonts que pitjor classifica, les mètriques s'empitjoren indicant que els resums generats tenen pitjor qualitat. En general la davallada de les mètriques és major en aquest model que en l'anterior en *TEST-I* i si mirem *TEST-NI* abans els resultats es quedaven sense alterar, mentre que ara en alguns casos fins i tot millora.

Les Taules 5.13 i 5.14 il·lustren les matrius normalitzades, on novament la fila representa la font original de les notícies i la columna la font forçada; un valor major que 1 vol dir que forçant la font s'ha obtingut millors resultats de *ROUGE-Lsum* que sense forçar-

Partició	Font forçada	*ROUGE-1*	*ROUGE-2*	*ROUGE-L*	*ROUGE-Lsum*	*BERTscore*
TEST-I	Correcta	28.16	11.52	22.50	22.95	71.82
	CA01	27.91	11.36	22.81	22.71	71.62
	CA02	27.94	11.35	22.27	22.80	71.65
	CA03	27.58	11.09	22.08	22.46	71.54
	CA04	27.38	10.92	21.98	22.34	71.56
	CA05	27.99	11.38	22.34	22.72	71.69
	CA06	27.87	11.27	22.29	22.67	71.68
TEST-NI	CA01	27.33	11.04	21.00	21.89	70.09
	CA02	27.82	11.18	21.27	22.31	70.54
	CA03	27.27	10.81	20.89	21.80	70.28
	CA04	27.21	10.95	20.98	21.86	70.39
	CA05	27.73	11.18	21.29	22.15	70.05
	CA06	27.85	11.38	21.42	22.28	70.24

Taula 5.12: Taula comparativa mètriques resum automàtic model M2 forçant fonts

la i menor que 1 que s'ha empitjorat. Per tant, en realitat la diferència entre forçar i no forçar és molt súbtil en ambdues particions, per a *TEST-I* la tònica general és que els resums obtinguts quan forcem les fonts són pitjors que si no les forcem, especialment quan forcem les fonts sobre les quals sembla tenir menor control: $CA03$ i $CA04$, i només hi ha un cas en el qual millora els resultats, que és quan les notícies són de $CA04$ i li fem creure al model que és del $CA02$. Mentre que en la partició de *TEST-NI* que són aquelles notícies de diaris que no ha llegit mai, millora en més casos (i en major proporció) que *TEST-I* i en els casos que empitjora, la davallada de rendiment és menor. És a dir que sobre *TEST-NI* forçar la font té un efecte més positiu i sobre *TEST-I* és més negatiu.

	CA01	CA02	CA03	CA04	CA05	CA06
CA01	0.9986	0.9771	0.9625	0.9657	0.9911	0.9907
CA02	0.9660	0.9998	0.9682	0.9594	0.9631	0.9601
CA03	0.9910	0.9890	0.9908	0.9704	0.9892	0.9849
CA04	0.9959	**1.0001**	0.9869	0.9939	0.9956	0.9957
CA05	0.9783	0.9868	0.9702	0.9684	0.9995	0.9927
CA06	0.9972	0.9973	0.9807	0.9827	0.9964	0.9993

Taula 5.13: Matriu normalitzada *ROUGE-Lsum* forçant fonts i sense forçar per al model M2 sobre TEST-I

	CA01	CA02	CA03	CA04	CA05	CA06
CA07	0.9982	**1.0332**	**1.0119**	**1.0061**	**1.0191**	**1.0194**
CA08	0.9973	**1.0062**	0.9885	0.9882	0.9845	**1.0067**
CA09	0.9961	**1.0167**	0.9812	0.9905	**1.0120**	**1.0295**

Taula 5.14: Matriu normalitzada *ROUGE-Lsum* forçant fonts i sense forçar per al model M2 sobre TEST-NI

Veient que els resultats obtinguts són molt semblants forcem o no forcem distintes fonts, vam decidir avaluar novament mitjançant una matriu quants casos de resums eren diferents i mitjançant *ROUGE-Lsum* com de diferents eren, comparant els resums generats sense dir-li cap font i donant-li cadascuna de les fonts (columnes), els resultats poden trobar-se en les Taules 5.15 i 5.16. I estan agrupats per la font original de les notícies (files).

Amb aquestes dos matrius (5.15 i 5.16) el que confirmem és que gràcies a haver fet el *fine-tuning* del model amb un corpus anonimitzat en tot moment, el model resultant sembla més sensible a la font proporcionada, perquè si comparem estes matrius amb

les del model anterior en les Taules 5.6 i 5.7 es detecta que en este cas hi ha una major quantitat de resums que són diferents en forçar la font i no forçant-la i a més d'eixos resums que són diferents, també són més diferents com ens indica la matriu del *ROUGE-Lsum*. Fins i tot en els casos de *CA*03 i *CA*04 que ja hem comentat en diverses ocasions que són les fonts que el model té menys controlades, quan els forcem la font correcta, generen uns resums diferents en quasi un 10 i 23% de les vegades respectivament, cosa que contrasta amb la resta de fonts on el màxim és un 2.12% de resums diferents. Aquesta diferència de percentatges s'explica perquè precisament estes dos fonts són en les que més s'enganya a l'hora de classificar-les, llavors hi ha més vegades en les que quan genera el resum creu que la font de la notícia no és la real, per la qual cosa quan forcem la font vertadera, hi ha major diferència que en la resta de casos en els quals quasi sempre ja havia encertat el model per ell mateix la font. No obstant encara que els resums siguen més diferents en este nou model, no s'ha vist afectada la qualitat dels resums generats com ja hem vist en la Taula 5.12. En l'apèndix C podem trobar diferents mostres que exemplifiquen casos particulars en els quals quan hem forçat la font el resultat obtingut ha estat diferent al resum generat sense dir-li res respecte a la font al model. Aquestes diferències s'ha pogut constatar que van des de resums que res tenen a veure entre ells, fins a casos en els quals l'única alteració han estat els símbols de puntuació. I resulta destacable com, en certs casos, hi ha diferències entre el resums obtinguts forçant i sense forçar la font, però en canvi els resums obtinguts forçant les distintes fonts són molt més semblants entre ells.

	CA01	CA02	CA03	CA04	CA05	CA06
CA01	2.12	73.30	78.14	75.07	61.86	62.53
CA02	72.83	0.68	76.9	74.20	70.68	72.49
CA03	66.28	72.41	9.89	61.35	67.3	65.98
CA04	60.24	69.66	52.59	23.02	64.84	65.60
CA05	68.81	72.914	74.39	71.57	0.44	58.53
CA06	57.92	67.94	77.51	74.95	52.75	1.20

Taula 5.15: Matriu de percentatges de resums diferents forçant i no forçant la font pel model M2 sobre *TEST-I*

	CA01	CA02	CA03	CA04	CA05	CA06
CA01	99.08	68.32	67.23	68.69	73.41	71.97
CA02	65.95	99.68	63.68	64.63	67.87	63.66
CA03	71.76	67.46	95.72	74.72	71.48	70.87
CA04	75.52	69.25	78.53	90.46	72.82	70.51
CA05	68.38	67.90	66.34	67.11	99.78	70.91
CA06	74.24	70.14	66.74	65.68	76.47	99.45

Taula 5.16: Matriu de *ROUGE-Lsum* entre resums generats forçant i no forçant la font pel model M2 sobre *TEST-I*

També ens interessa veure si en generar resums més llargs, aconseguim que eixos resums estiguen més impregnats de l'estil de cadascuna de les fonts, en tant que sent més llargs poden donar més joc per introduir més construccions i vocabulari propis, i per tant siguen més diferents en comparar els resums generats forçant la font i sense forçar-la.

Les Taules 5.17 i 5.18 representen dos matrius que valoren respectivament el percentatge de resums que ha generat el model M2 sense marcar la font i són diferents dels que ha obtingut forçant-li una font (la de la columna) i la segona avalua com de diferents eren mitjançant *ROUGE-Lsum*. El que podem observar és un gran creixement dels percentatges de resums que són diferents quan forcem la font i quan no ho fem i un descens de

la semblança entre eixos resums, mesurada mitjançant el *ROUGE-Lsum*, ara que li hem indicat al model que els resums que ha de generar han de ser més llargs. Quan els resums són més llargs, efectivament és verifica la nostra hipòtesi de que en un resum més llarg té un major impacte la font, perquè aconsegueix deixar una major petjada que diferencia els resums segons quina font crega que ha produit el resum.

	CA01	CA02	CA03	CA04	CA05	CA06
CA01	2.21	91.37	89.69	93.37	83.59	83.94
CA02	95.19	0.75	94.79	97.01	93.77	94.92
CA03	85.23	93.50	80.57	32.34	90.72	90.96
CA04	79.65	93.91	80.57	32.34	90.72	90.96
CA05	91.72	95.18	92.99	95.99	0.53	86.00
CA06	79.17	89.65	91.51	93.31	74.54	1.55

Taula 5.17: Matriu de resums generats diferents forçant i no forçant la font pel model M2 sobre *TEST-I* amb resums llargs

	CA01	CA02	CA03	CA04	CA05	CA06
CA01	99.21	66.55	67.95	65.18	71.01	70.62
CA02	61.93	99.69	61.33	58.88	63.18	60.07
CA03	69.82	65.44	95.53	65.34	67.17	66.82
CA04	71.82	65.90	72.37	88.67	68.07	67.86
CA05	64.42	63.85	61.14	60.21	99.79	66.97
CA06	72.03	66.66	64.98	62.43	73.74	99.43

Taula 5.18: Matriu de *ROUGE-Lsum* entre resums generats forçant i no forçant la font pel model M2 sobre *TEST-I* amb resums llargs

Resultats d'abstractivitat

Ara canviem l'objecte a analitzar i passem a l'abstractivitat, anem a avaluar com d'abstractiu ha sigut el nostre model, si és més o menys abstractiu que el model NASca original i comprovarem si el fet d'allargar la longitud dels resums fa que siguen més abstractius que quan són més curts. La Taula 5.19 mostra els resultats de les distintes mètriques d'abstractivitat proposades: *Extractive Fragment Coverage*, *Content Reordering*, Abtractivitat$_p$ i *Novel N-grames*, les quals s'han aplicat sobre les dos particions de test. Es mostren resultats del model NASca original juntament amb els del model M2 generant resums de longitud estàndard i resums de llargària major.

Partició	Model	*Extractive Fragment Coverage*	*Content Reordering*	Abstractivitat$_p$ $(p = 2)$	Novel $1-Grames$	Novel $4-Grames$
TEST-I	NASca	96.99	46.17	47.19	3.21	28.65
	model M2	97.42	62.12	43.34	2.74	26.21
	model M2 $^{resums\_llargs}$	96.76	80.27	65.01	3.62	26.01
TEST-NI	NASca	96.66	42.37	41.89	3.52	26.32
	model M2	97.15	52.36	38.98	3.00	24.53
	model M2 $^{resums\_llargs}$	97.20	65.64	55.02	3.14	20.86

Taula 5.19: Taula comparativa de mètriques d'abstractivitat NASca vs model M2

Les conclusions que es poden extraure dels resultats de la Taula 5.19 són pel que fa a la cobertura els resultats són molt semblants en totes les particions i models, perquè en

general tendeix a fer servir el mateix vocabulari. La resta de mètriques les analitzarem per separat per a *TEST-I* i *TEST-NI*:

- TEST-I: El que podem concloure de la reordenació de contingut, és que els nostres models agafen una tendència major de la que tenia NASca original a resumir el contingut de l'article mitjançant la reordenació del text d'entrada, especialment en els casos dels resums llargs. Pel que fa a l'abstractivitat$_p$ podem veure que hi ha una davallada en el model entrenat amb resums de mida normal, però en el moment en el que generem resums més llargs, s'incrementa considerablement. Per últim, els Novel $4-grames$ es veu reduit en ambdós casos, tot i que en un lleuger percentatge, indicant per tant que s'inventa menys construccions i agafa més $4-grames$ de l'article original, i en els $1-grames$ disminueix amb els resums curts, però amb els llargs fins i tot millora el valor de referència de NASca.

- TEST-NI: La reordenació de contingut en aquest cas també augmenta per al model que hem entrenat respecte a NASca original en ambdós casos, però amb una pujada menys pronunciada que en la partició de *TEST-I*. L'abstractivitat$_p$ té també el mateix comportament que en *TEST-I*, baixa una miqueta en el model entrenat amb resums normals i quan l'obliguem a generar resums més llargs l'augmenta. Pel que fa als Novel $n-grames$ en ambdós casos baixa, però quan els resums són més llargs baixa encara més el Novel $4-grames$ i en canvi el $1-grames$ puja lleugerament.

Per tant podem traure en clar que el nostre model segueix la mateixa tendència que el de NASca en el sentit que per a *TEST-I* continua generant resums que són més abstractius que en *TEST-NI*. També està clar que el nostre model té una tendència bastant gran a resumir reordenant el contingut original del text. Amb aquestes dades no podem concloure del tot si el nostre model és més abstractiu que NASca o no, perquè depén d'amb quina mètrica decidim comparar-lo, en canvi el que sí es pot afirmar és que quan el nostre model genera resums llargs estos són més abstractius que els curts i en eixe cas sí que aconsegueix treure uns resultats més abstractius que el propi NASca, tot i que agafa més seqüències ($4-grames$) de l'article original, per tant és menys innovador amb el vocabulari en eixe sentit.

També volíem analitzar si sobre l'abstractivitat tenia algun efecte tant per a resums de mida normal com per als llags, el fet de forçar la font i no forçar-la.

La Taula 5.20 mostra els resultats de les mètriques d'abstractivitat quan al model li forcem cadascuna de les fonts. S'observa el mateix comportament, quan forcem la font correcta els valors són molt pareguts als resultats que s'obtenen sense forçar la font, i quan se van forçant la resta de fonts se pot veure que més o menys totes oscil·len en valors al voltant dels de referència que tenim en la Taula 5.19. No obstant en ambdues particions les fonts més dificils de classificar ($CA03$ i $CA04$) i la que té més mostres ($CA01$), són les úniques en les quals empitjoren les mètriques d'abstractivitat, en els altres 3 casos s'incremeneten. També podem destacar que la mètrica que més canvia en termes quantiatius és la reordenació de contingut, que entre el valor més xicotet i el més gran hi ha una diferència de fins a 9 punts, mentre que en la cobertura per exemple se'n va 0.35 punts.

En la Taula 5.21 podem distingir les mètriques d'abstractivitat aplicades sobre el model M2 quan li forcem les distintes fonts i el fem generar resums llargs. S'observa novament que quan forcem la font correcta els resultats són els mateixos, però en aquest cas sembla que la font ha perdut influència, perquè quan forcem les distintes fonts els resultats de les mètriques es mantenen més estables, a excepció de $CA01$ que sí que demostra una davallada en rendiment més forta que la resta, en $TEST-I$. En $TEST-NI$ sí que

Partició	Font forçada	*Extractive Fragment Coverage*	*Content Reordering*	Abstractivitat$_p$ $(p = 2)$	Novel $1 - Grames$	Novel $4 - Grames$
TEST-I	Correcta	97.45	61.69	43.03	2.71	26.08
	CA01	97.74	58.61	42.35	2.40	25.59
	CA02	97.44	64.53	46.30	2.71	28.71
	CA03	96.59	58.51	43.20	3.53	27.22
	CA04	97.94	55.18	39.33	2.17	24.03
	CA05	97.49	64.02	43.95	2.69	26.46
	CA06	97.37	63.28	44.25	2.80	27.17
TEST-NI	CA01	97.25	51.05	38.11	2.87	24.17
	CA02	97.32	55.19	42.93	2.79	25.76
	CA03	97.07	48.78	36.38	3.03	21.75
	CA04	97.36	48.54	37.35	2.73	22.91
	CA05	96.86	58.23	40.42	3.36	24.69
	CA06	96.63	57.58	43.06	3.54	27.47

Taula 5.20: Taula comparativa de les mètriques d'abstractivitat del model M2 forçant les distintes fonts

manté la mateixa tendència que amb els resums de longitud normal, una baixada dels resultats per a les fonts $CA01$ i $CA03$, però $CA04$ ha fet millorar els resultats.

Partició	Font forçada	*Extractive Fragment Coverage*	*Content Reordering*	Abstractivitat$_p$ $(p = 2)$	Novel $1 - Grames$	Novel $4 - Grames$
TEST-I	Correcta	96.75	80.31	65.17	3.62	26.04
	CA01	97.49	74.46	61.31	2.83	23.24
	CA02	96.89	81.04	67.76	3.45	27.89
	CA03	95.37	77.28	65.97	4.93	27.07
	CA04	97.02	77.49	65.53	3.26	24.89
	CA05	96.85	81.94	65.48	3.56	26.43
	CA06	96.87	79.68	64.58	3.56	26.24
TEST-NI	CA01	97.43	64.15	53.79	2.91	19.95
	CA02	97.23	68.61	60.27	3.03	22.85
	CA03	96.71	64.43	53.84	3.56	19.18
	CA04	97.12	64.68	58.66	3.14	21.08
	CA05	96.82	71.72	57.95	3.51	22.77
	CA06	96.69	70.53	57.86	3.74	23.51

Taula 5.21: Taula comparativa mètriques abstractivitat sobre resums llargs del model M2 forçant les distintes fonts

Resultats de la generació de n-resums

Per últim vam voler comprovar si d'alguna manera es veien modificats els resums generats quan demanàvem que generara més d'una seqüència d'eixida. En concret vam forçar el model a generar 4 seqüències (resums). La Taula 5.22 mostra quin van ser els resultats per a cadascun dels 4 resums generats considerant les mètriques de ROUGE.

Dels resultats de la Taula 5.22 no podem extraure conclusions, perquè no hi ha un comportament unificat. Per exemple en *TEST-I* totes les eixides obtenen més o menys els mateixos resultats però els resums generats en segon lloc tenen una davallada del rendiment, i en *TEST-NI* succeix més o menys el mateix però el que decau és el de l'últim. Si comparem el *TEST-I* amb el model que generava només una eixida, sense considerar la

Partició	nombre de resum	*ROUGE*-1	*ROUGE*-2	*ROUGE-L*	*ROUGE-Lsum*
TEST-I	1	29.34	12.60	21.59	22.83
	2	26.42	11.98	19.73	21.19
	3	29.74	12.89	21.43	22.84
	4	29.69	12.47	21.55	22.81
TEST-NI	1	24.49	10.71	18.14	18.73
	2	23.97	11.37	17.66	18.31
	3	22.66	9.89	17.03	18.05
	4	23.43	8.18	15.67	15.96

Taula 5.22: Taula comparativa mètriques resum automàtic NASca vs model M2 quan genera 4 resums seguits

2a eixida que ha tingut un pitjor rendiment generalitzat, la resta han aconseguit millorar els *ROUGE*-1 i *ROUGE*-2 mentre que els que tenen en compte les seqüències comunes més llargues (LCS) per als càlculs: *ROUGE-L* i *ROUGE-Lsum* les han empitjorat. Per a *TEST-NI* sí que hi ha una davallada generalitzada del rendiment a l'hora de generar més d'un resum.

En l'apèndix A podem trobar una mostra d'una notícia i un exemple de totes les versions de resums de les quals hem parlat en aquesta secció.

5.3 Classificador de notícies i resums

5.3.1. Experimentació

Analitzant els resultats que hem observat en el model anterior, realment l'èxit de la classificació no venia donat perquè les mostres no estigueren anonimitzades des d'un primer moment, perquè els resultats obtinguts han sigut els mateixos. Llavors ací apareix una incògnita, com és possible que siga capaç de distingir tan bé de quina font és cada notícia i després genere uns resums que independentment de la font forçada obtenen els mateixos resultats? És a dir detecta l'estil o vocabulari de les diferents fonts en els articles, però després quan va a resumir-les independentment de la font que se li force, resumeix de la mateixa manera. Llavors per poder comprovar que estava passant es van fer dos experiments, entrenant per una banda un model dedicat a classificar els articles que va obtindre uns resultats pràcticament igual de bons, fins i tot quan es van realitzar alteracions sobre els articles originals, com eliminar tots els símbols de puntuació, reordenar alfabèticament totes les paraules, etc. També es va entenar un model per classificar resums, perquè de moment només estem classificant articles. Aquest model pot servir-nos per veure si el problema està en que en l'article sí que hi ha suficient informació, suficients petjades estilístiques, per detectar la font, però en els resums de referència o bé no hi ha cap pista que li permeta aprendre a resumir de formar diferent segons la font o bé és massa curt per ser capaç d'imprimir eixe estil. Per poder respondre a aquestes preguntes, anem a aplicar cadascun dels models sobre els articles originals així com sobre els resums de referència, i en especial sobre els resums que genera el nostre model, tant els curts com els llargs i també incloem els que s'han generat forçant les diferents fonts, perquè com hem vist en este nou model quan forçàvem les fonts existia una major diferència entre el resum obtingut i el que s'obtenia sense forçar la font, en especial en el cas dels resums llargs. Per realitzar aquest entrenament s'ha seguit la mateixa estratègia que en la resta de models, s'ha utilitzat la llibreria de *HuggingFace* per aplicar una fase de *fine-tuning* sobre un model preentrenat. Per poder fer-ho es va emprar el mateix corpus

que en els altres entrenaments, amb la diferència que en lloc d'aportar-li un resum de referència com a "etiqueta", el que s'aportava era la font correcta perquè volem entrenar-lo perquè siga capaç de realitzar una tasca de classificació simplement.

5.3.2. Analisi de resultats

La Taula 5.23 es mostren els resultats que han obtingut en la tasca de classificació els dos models que s'han entrenat a tal efecte. El primer d'ells s'ha entrenat en classificació veient els articles, i l'altre veient els seus resums de referència. I després per traure aquests resultats s'han utilitzat tant els articles i resums que conformen la nostre partició de *TEST-I* (de *TEST-NI* no perquè no es capaç de classificar eixes fonts que mai ha vist), com aquells que ha generat el nostre model M2. Dins dels que ha generat el nostre model podem agrupar entre les proves que s'han fet emprant resums de longitud estàndard o uns de més llargs. També s'han fet diferents experimentacions jugant amb la font: sense forçar-li'n cap, forçant la correcta i forçant cadascuna de les fonts vistes.

El primer que podem extraure dels resultats d'aquesta taula és que els articles tenen molta informació com per permetre a una Intel·ligència Artificial detectar amb facilitat la seua font, perquè els resultats que obté el model que ha sigut entrenat amb articles és el mateix que va obtindre el nostre model quan ha de classificar un article, perquè en canvi quan fem servir este model per classificar resums, que tenen una longitud menor, perd molt de rendiment, bé siga perquè acostumat a tindre molta informació en un article, ara en té menys o perquè els resums tenen una menor petjada estilística que permeta detectar la font. Tot apunta a la primera hipòtesi perquè després veiem que en el model que s'ha entrenat a base de resums, sí que és capaç de detectar la font dels resums de referència, és a dir que aquests tenen elements identitaris que les delaten. Podem destacar que quan intenta classificar els resums llargs, millora substancialment els resultats que obtenia contra resums de longitud normal, per la qual cosa podem confirmar la nostra hipòtesi que si feiem més llargs els resums, aconseguíem que el model imprimira una major marca estilística. En general, el fet de forçar una font no té un gran impacte en el rendiment, però sí és cert, que forçant una font qualsevol, és a dir, enganyant al model i fent que resumisca la notícia com si fora d'un diari que realment no és, fa que després a l'hora de realitzar la classificació se confonga més vegades que si li donem la font correcta o no li diem res, és a dir que forçar un font, per poc efecte que tinga sí que fa que l'estil del resum siga diferent, això passa especialment quan generem resums llargs. Com era d'esperar, forçar la font correcta o no dir-li res obté els mateixos resultats perquè els resums que es generaven eren gairebé idèntics. Un altre fet destacable és que el nostre model genera uns resums que són pràcticament igual d'identificables que els resums de referència, és a dir que el nostre model ha generat uns resums que podrien fer-se passar per resums que ha escrit un periodista d'eixe diari, i si anem al cas dels resums llargs encara més.

D'altra banda tenim el model que ha sigut entrenat per a classificar a base dels resums de referència, en este cas evidentment millora el rendiment a l'hora de classificar els resums, i empitjora a l'hora de classificar els articles, però no obstant la diferència no és tan gran com en el classifiador anterior, perquè al cap i a la fi encara que estiga acostumat a veure resums i ara li arribe un article, aquest és de major longitud aleshores té més espai per a deixar les petjades estilístiques del diari en qüestió. Aquest classificador, com és lògic encerta la font dels resums generats pel nostre model en un major percentatge d'ocasions, perquè està acostumat a classificar resums, però a diferència del que ocorria en el classificador anterior, en este cas, els resums que genera el nostre model no són tan fàcilment reconeixibles, té un rendiment pitjor amb ells que amb els resums de referència, no obstant sí que millora quan se tracta de resums llargs. Per a este classificador en comparació amb l'entrenat amb articles, són més visibles les diferències en els resums

obtinguts en forçar les diferents fonts, perquè en eixos casos s'enganya més a l'hora de classificar-los que quan no li hem dit la font o li hem donat la correcta.

Model [3]	Text avaluat	Font forçada	Accuracy	Precisió	Recall	F1-Score
Article	**Article TEST-I**	**-**	**92.92**	**93.4**	**92.92**	**92.79**
	Resum TEST-I	-	38.48	55.93	38.48	32.31
	Resums generats	-	37.34	57.28	37.33	31.26
		Correcta	37.82	57.32	37.81	31.62
		CA01	36.5	59.4	36.5	30.85
		CA02	34.4	58.65	34.39	29.01
		CA03	32.74	57.17	32.73	27.72
		CA04	34.75	57.68	34.74	29.21
		CA05	36.36	55.52	36.36	30.66
		CA06	36.3	59.93	36.3	30.59
	Resums llargs generats	-	52.13	63.94	52.13	46.81
		Correcta	53.33	64.51	53.32	47.72
		CA01	50.28	67.02	50.27	45.7
		CA02	47.76	65.93	47.75	43.43
		CA03	44.28	62.51	44.27	40.19
		CA04	48.35	62.29	48.35	43.99
		CA05	49.70	62.88	49.69	45.28
		CA06	49.8	64.27	49.79	45.32
Resum	Article TEST-I	-	56.21	60.34	56.21	54.06
	Resum TEST-I	**-**	**66.34**	**72.52**	**66.34**	**66.94**
	Resums generats	-	59.21	67.69	59.21	59.92
		Correcta	59.75	68.11	59.75	60.47
		CA01	50.95	60.96	50.96	51.29
		CA02	50.64	59.22	50.64	50.84
		CA03	48.8	58.36	48.8	49.46
		CA04	51.62	69.29	51.62	52.20
		CA05	53.86	62.29	53.86	54.56
		CA06	52.81	62.11	52.81	53.38
	Resums llargs generats	-	69.74	76.06	69.74	70.00
		Correcta	71.22	77.15	71.22	71.44
		CA01	56.25	66.03	56.25	55.00
		CA02	57.81	66.48	57.81	56.12
		CA03	54.8	67.38	54.8	54.71
		CA04	56.45	63.53	56.45	55.56
		CA05	62.29	69.33	62.29	62.35
		CA06	58.63	68.09	58.63	58.05

Taula 5.23: Taula comparativa dels resultats del model de classificació d'articles i resums

5.4 Model bilingüe

Per ampliar les experimentacions d'aquest treball, es va decidir no quedar-se només en un model de resum de notícies i classificació de la seua font de procedència, sinó ampliar també cap a un model que fora bilingüe (català i castellà) i llavors permetre que pugerem

dir-li en quina llengua està escrita la notícia i en quina llengua volem obtindre el resum [4]. D'aquesta manera podem veure també si és capaç de detectar la llengua del text d'origen i si hi ha diferències de rendiment d'una llengua a altra, si li donem la informació addicional de quina és la llengua de la notícia és capaç de resumir millor, perquè en tractar-se de dos llengües amb prou semblances és possible que les confonga, o si intentem enganyar-li i fer-li creure que la notícia està escrita en una llengua distinta, encara així serà capaç d'entendre el seu contingut i resumir-lo amb el mateix rendiment?

A més en este cas partim d'una problemàtica addicional, que és el fet de que hem d'ajustar un model preentrenat, i com en el grup d'investigació no es disposa de cap model bilingüe, cal partir del model preentrenat en català, llavors serà durant l'etapa de *fine-tuning* durant la qual haurà d'aprendre no només a resumir en dos llengües, sinó a més a comprendre text en castellà que no ha vist mai.

5.4.1. Descripció del corpus

Si bé amb DACSA ja disposem d'un corpus complet tant per al català com per al castellà, no podíem agafar sense més aquest corpus per a l'entrenament del model bilingüe, per diverses raons. La primera és la descompensació de la quantitat de documents, que farien que es centrara en els documents en castellà, perquè com hem comentat en aquesta secció, després del filtrat de notícies, encara hi ha aproximadament el triple de mostres en castellà que en català. Per tant el que vam plantejar va ser fer una equivalència de cada font en català amb una font en castellà i vam agafar la mateixa quantitat de documents que hi havia en cadascuna de les particions originals, com pot veure's en la Taula 5.24.

Font català	Font castellà	Entrenament	Validació	TEST-I	TEST-NI
CA01	ES01	226441	5896	5896	0
CA02	ES02	182905	5896	5896	0
CA03	ES03	125655	5896	5896	0
CA04	ES04	45035	5896	5896	0
CA05	ES05	32589	5896	5896	0
CA06	ES06	23971	5896	5896	0
CA07	ES11	0	0	0	7104
CA08	ES15	0	0	0	5882
CA09	ES16	0	0	0	4850

Taula 5.24: Distribució de les notícies en el corpus bilingüe

Una vegada teníem el mateix nombre d'articles en català que en castellà, encara faltava fer una cosa, donat que volíem que el model fora capaç de generar resums tant en castellà com en català per a articles catalans, vam afegir per cadascuna de les mostres una nova mostra amb el mateix article, però amb el resum traduit al castellà mitjançant un model de traducció automàtica proporcionat pel grup d'investigació[5]. Per al cas dels articles en castellà, vam deixar exclusivament els resums en castellà perquè no era segur si el model seria capaç de comprendre el castellà, aleshores per reduir la complexitat de la tasca que havia de portar a terme, a més suposava reduir el cost computacional per tal de poder tindre el model entrenat i avaluat en unes dates apropiades per a la duració del TFG. La Figura 5.7 mostra gràficament com va quedar el corpus per a l'entrenament del model bilingüe.

[4] En el cas dels articles en castellà el resum resultant només pot ser en castellà

[5] Estava desplegada en el servidor *boso* del DSIC i mitjançant una petició HTTP se li enviava el text a resumir, la font d'origen i de destí i retornava el text traduït.

Figura 5.7: Representació del corpus bilingüe

5.4.2. Experimentació

Primerament vam configurar un corpus com s'ha descrit en l'apartat anterior. Això va generar un problema a l'hora de començar l'entrenament perquè els fitxers eren massa grans a l'hora de carregar-los, per tant es va haver de canviar el format en que guardàvem els *datasets* en lloc de gastar un json, gastar json lines, que escriu cada mostra en una llista separada, amb la qual cosa no és necessari anar mantenint tanta informació en el buffer d'entrada que és el que ens estava donant problemes.

Per les limitacions que hem comentat prèviament, què el model amb que treballem en este cas, no és bilingüe de partida perquè mai ha vist castellà, es fa necessari deixant-lo entrenant-se durant una major quantiat d'èpoques, concretament 15, però a més a més com s'ha triplicat el nombre de mostres que conformen el corpus, cadascuna de les èpoques dura 48h, per la qual cosa este entrenament ha tardat 30 dies en completar-se.

5.4.3. Analisi de resultats

Resultats ROUGE

De la mateixa manera que en la resta del treball, s'han avaluat els resultats ROUGE obtinguts pels resums generats pel model bilingüe en comparació als de referència.

La Taula 5.25 mostra els resultats de les 4 mètriques de ROUGE per al model bilingüe en ambdues particions i amb totes les combinacions lingüístiques que es capaç de gestionar: notícies en català amb resums en català(*CA-CA*); notícies en català amb resums en castellà (*CA-ES*) i notícies en castellà amb resum en castellà (*ES-ES*). A la taula podem observar per una banda que els resums que genera en CA-CA que seria l'escenari equivalent al que hem tractat fins ara en els models monolingües, els resultats són pràcticament idèntics als de model M2, millora molt subtilment en totes les mètriques menys en *ROUGE-Lsum* que empitjora 3 centèssimes en la partició *TEST-I*, també estan molt pròxims als de NASca [44]. Mentre que en la partició de *TEST-NI* els resultats per a CA-CA empitjoren respecte al que havia obtingut model M2, tot i que també ho fan de manera atenuada, no obstant el fet que haja millorat per a *TEST-I* i empitjorat per

TEST-NI ens pot fer sospitar que aquest model té menor capacitat de generalitzar, si bé la diferència és tan xicoteta que no ens permet arribar a cap conclusió definitiva. Com és lògic, i ha ocorregut en tota la resta d'experimentacions, els resultats per a *TEST-NI* en qualsevol mètrica han estat pitjors que el seu equivalent en *TEST-I*.

Partició	Llengua article	Llengua resum	*ROUGE*-1	*ROUGE*-2	*ROUGE-L*	*ROUGE-Lsum*
TEST-I	CA	CA	28.31	11.83	22.88	22.97
	CA	ES	29.37	11.47	23.74	23.78
	ES	ES	29.18	11.85	23.24	23.36
TEST-NI	CA	CA	26.49	10.55	20.48	20.57
	CA	ES	27.06	10.21	21.11	21.14
	ES	ES	28.54	12.02	22.85	22.87

Taula 5.25: Taula de mètriques dels resums automàtic pel model bilingüe

D'altra banda tenim els resultats per al castellà, que han sigut molt sorprenents. Perquè partíem d'un model preentrenat en català, i precisament comentàvem anteriorment, que corríem el risc que aquest model bilingüe realment no fora capaç de comprendre o de resumir en castellà, perquè havia d'aprendre a fer-ho tot durant la fase de *fine-tuning* i això era una complexitat afegida. No obstant, els resultats el que demostren és que malgrat eixes circumstàncies i que la quantitat de notícies en castellà ha sigut la mateixa que les de català com ja s'ha explicat en l'apartat de **Descripció del corpus**, els resums generats tenen una major semblançat al resum de referència tant quan la notícia a resumir estava en castellà com quan la notícia era en català però el resum a generar havia de ser en castellà, això s'ha donat tant en la partició *TEST-I* com *TEST-NI*. En ambdós casos els resultats milloren el obtinguts per NASca, que utilitzem com a referència. No obstant això, no arriba a obtindre tan bons resultats com va obtindre NASes, donat que aquest havia estat ja preentrenat amb un model lingüístic en castellà, i la quantitat de notícies amb que va estar entrenat va ser molt superior.

Resultats abstractivitat

Per a avaluar l'abstractivitat hem decidit avaluar exclusivament els parells resum-article que estigueren en la mateixa llengua, perquè les altres avaluacions no hagueren tingut sentit en tant que es tracta de llengües diferents i per tant gasten paraules diferents, la qual cosa haguera confós els resultats incremenetant-los notablement. A més, hem tornat a optar per avaluar l'abstractivitat de resums d'una longitud "normal" però també dels resums amb una major longitud, per analitzar si té el mateix efecte que en les altres experimentacions, on s'ha observat que quan forçàvem al model a generar uns resums que foren més llargs, aquests eren més abstractius.

La Taula 5.26 mostra les diferents mètriques que hem triat per avaluar l'abstractivitat sobre ambdues particions de test, i agrupades per la llengua de l'article. De l'anàlisi de la taula podem extraure dos grans conclusions: per una banda que els resultats que ha obtingut el català milloren en totes les mètriques als que havia obtingut el model M2 i per tant els resums són més abstractius. Però en castellà el que s'observa són uns pitjors resultats tant respecte del català, com dels models anteriors, la qual cosa ens porta a pensar que quan ha vist un text en una llengua desconeguda per a ell fins el moment, ha tendit més a copiar el text original en lloc d'innovar i generar el seu propi contingut. Com ocorria amb la partició *TEST-NI* que com tampoc havia vist mai les notícies d'eixa font, no tenia tan clar com resumir-les i per tant tenia una major tendència a mantindre el contingut original.

La Taula 5.27 mostra els resultats de les mètriques d'abstractivitat, també agrupades segons la llengua de l'article a resumir i avaluat sobre les dos particions de test, però amb

Partició	Llengua article-Llengua resum	*Extractive Fragment Coverage*	*Content Reordering*	Abstractivitat$_p$ $(p = 2)$	Novel $1-Grames$	Novel $4-Grames$
TEST-I	CA-CA	96.78	67.88	47.94	3.39	32.04
	ES-ES	97.64	56.22	39.87	2.51	25.50
TEST-NI	CA-CA	96.30	58.29	42.83	3.84	30.01
	ES-ES	97.52	53.88	38.36	2.62	24.69

Taula 5.26: Taula de mètriques d'abstractivitat model bilingüe

la diferència que esta vegada, quan s'han generat els resums s'ha forçat a que aquests tingueren una major longitud. Els resultats no han sigut gens sorprenents perquè posen de manifest que s'obtenen resums més abstractius en la partició de $TEST\text{-}I$ que en la de $TEST\text{-}NI$, en català que en castellà i quan els resums són llargs front a quan tenen una longitud estàndard. El que sí resulta interessant és que també en aquesta avaluació pot constatar-se que quan el model bilingüe resumeix notícies en català amb resums llargs, ho fa d'una manera més abstractiva del que ho feia el model M2 sota les mateixes condicions.

Partició	Llengua article-Llengua resum	*Extractive Fragment Coverage*	*Content Reordering*	Abstractivitat$_p$ $(p = 2)$	Novel $1-Grames$	Novel $4-Grames$
TEST-I	CA-CA	95.47	84.84	70.59	4.95	33.47
	ES-ES	97.19	73.67	52.69	3.16	22.30
TEST-NI	CA-CA	96.06	72.68	61.20	4.35	27.32
	ES-ES	97.69	69.45	51.06	2.64	19.20

Taula 5.27: Taula de mètriques d'abstractivitat model bilingüe amb resums llargs

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 tres models d'Intel·ligència Artificial, basat en l'arquitectura *Transformer* que parteixen en tots els casos del mateix model preentrenat amb un corpus en català seguint l'esquema d'entrenament de BART. Els dos primers models han estat destinats al resum abstractiu de notícies i la classificació de les seues fonts, amb la lleugera diferència que el primer va ser entrenat partint d'un model que ja havia passat per la fase d'ajust durant unes quantes èpoques, i el segon directament del model preentrenat. D'aquesta manera es compleixen els dos primers objectius d'aquest projecte que eren utilitzar les xarxes neuronals per classificar i resumir (abstractivament) notícies. També s'han complit els objectius relatius a comprovar l'efecte estilístic que podia tindre el fet de forçar una font periodística a l'hora què el model genera un resum. De l'anàlisi dels resultats obtinguts s'ha conclós que sí té efecte, especialment quan els resums generats són més llargs, malgrat que no canvia la qualitat [1] del resum generat. També s'ha pogut observar que en afegir la tasca de classificació, no s'ha produit una millora en les mètriques triades per avaluar la qualitat dels resums; si bé és cert que com a classificador ha quedat demostrat que funciona molt bé i que en els nostres models, si generem resums que siguen més llargs, s'incrementa notablement la seua abstractivitat respecte al que obtenia NASca.

El tercer objectiu també ha quedat complert donat que un tercer model bilingüe ha estat entrenat. Cal destacar que el model, no només ha aconseguit ser capaç de resumir notícies indistintament de la llengua de l'article d'entrada i la del resum d'eixida, sinó que ha estat capaç d'entendre el castellà i resumir en eixa llengua, sense haver vist cap mostra d'ella fins la fase de *fine-tuning* a la qual li hem dedicat 15 èpoques. A més cal posar de relleu que les 3 avaluacions que hem fet sobre aquest model han permès constatar que resumir notícies en dues llengües ha ajudat al model a obtindre uns millors resultats en les notícies en català. El model bilingüe obté millors resultats que el model M2 per a la mateixa llengua, i a més genera uns resums que són més abstractius, que també era l'objectiu d'aquest treball. A més a més, els resultats han posat de manifest que per a les notícies en castellà, els resums eren d'una abstractivitat inferior al català però obtenien una major semblança als resums que tenien de referència.

[1] Entenent la qualitat en aquest cas com la semblança mitjançant les mètriques de ROUGE entre el resum generat i els resums de referència redactats per periodistes.

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 gran problema al que ens hem d'enfrontar sempre que entrenem models d'Intel·ligència Artificial, en especial quan s'entrenen models *Transformers* que són xarxes neuronals enormes amb una gran quantitat de paràmetres que ha d'aprendre, per la qual cosa necessita una gran quantitat de mostres d'entrenament. Tot això es tradueix en un consum de recursos computacionals molt alt, que impliquen un alt consum energètic i temporal. No obstant, com s'ha explicat en el capítol d'**Eines utilitzades**, gràcies als avenços en el hardware i a l'ús de llibreries com *deepspeed* que permeten explotar al màxim les GPUs, estos temps s'han pogut reduir a uns més considerables. Si bé els entrenament són amb diferència les execucions que més temps han consumit, tampoc és menyspreable el que s'havia de dedicar a l'execució dels *scripts* d'avaluació, perquè hi havia moltes mètriques que comprovar, molts tipus de resums (normal, forçant la font correcta, forçant cadascuna de les fonts, resums llargs...), però això va aconseguir millorar-se una miqueta gràcies a l'ús de les GPUs per una banda i per altra gràcies a la idea de generar els resums una sola vegada i guardar-los en un fitxer de tipus *json* i així l'avaluació s'estalviava els temps dedicats a generar una vegada rere altra els mateixos resums. Açò va implicar haver de refer els scripts d'avaluació. Totes les preparacions de corpus, els entrenaments i proves de la qualitat dels models. han sigut possibles gràcies a la total disponibilitat de la màquina tardis per a aquest entrenament.

- **Entrenament multi-GPU:** Donat que la màquina està dotada de dos targetes gràfiques, inicialment es va plantejar l'entrenament distribuït en aquestes dos targetes, que permetria accelerar notablement les execucions, especialment els entrenaments. El que creíem que seria tan senzill com llançar l'*script* mitjançant *deepspeed* va resultar que era un problema generalitzat entre els usuaris de *HuggingFace* i també en general per a entrenar models d'Intel·ligència Artificial. La solució no semblava trivial, per la qual cosa es va acabar optant per entrenar en una única GPU. La qual cosa no vol dir que l'altra quedara ociosa, perquè quan hem estat treballant amb diversos models simultàniament, s'ha pogut entrenar un model en una GPU i aprofitar l'altra per accelerar els *scripts* de test, estratègia que tampoc ha estat exempta de problemes.

- **Caigudes de tardis:** Altre problema al que ens vam haver d'enfrontar però que estava fora del nostre control va ser que en diverses ocasions la màquina va desconnectar-se per manteniment, la qual cosa va provocar no només que no puguerem treballar durant els dies que estava inoperativa, sinó en especial que es va perdre el progrés dels entrenaments i validacions que estaven en marxa. Si bé és cert que gràcies a la llibreria de *HuggingFace* pot recuperar-se l'entrenament a partir d'un checkpoint guardat i no perdre tot el progrés i haver de començar de zero. De totes formes les desconnexions de tardis sí van afectar l'entrenament, perquè els guardats els efectuàvem cada època, llavors el progrés a meitat epoch sí que es perdia (i un epoch en el model més gran que hem entrenat dura dos dies, per tant significava perdre

el progrés de dos dies en el pitjor dels casos). Açò també afectà les avaluacions que en alguns casos eren *scripts* que tardaven molt de temps en executar-se.

- **Format dels resums generats:** Com ja s'ha comentat en l'entrenament del model M1 vam haver de tornar a començar l'entrenament diverses vegades perquè el model estava tenint un comportament molt estrany l'hora de generar els primers tokens i es va acabar descobrint que era perquè estava agafant el format que seguia *HuggingFace* a l'hora de preparar els tokens per passar-li'ls al model, però el model NASca original havia estat entrenant sense utilitzar el pretokenitzat de *HuggingFace* i estava esperant un altre format. La qual cosa va suposar una sensació d'estar fent constantment les coses malament i de no estar avançant el projecte. Quan per fi es va descobrir quina era la causa del problema, es va haver de corregir el codi per deixar de tokenitzar automàticament les mostres i passar a fer-ho de manera més "artesana".

- **Guardar model amb menor loss:** En l'entrenament d'un dels models no vam especificar quin havia de ser el criteri a seguir durant la validació per guardar el millor model. Per defecte el criteri és minimitzar la *loss* i al llarg de l'entrenament la *loss* pujava, mentre que la resta de mètriques estaven millorant-se, per la qual cosa s'estava guardant com a millor model el primer que havia entrenat perquè era el que tenia una millor *loss* i això va portar a que les avaluacions es feren amb el primer model fins que ens hi vam adonar i vam haver d'agafar l'últim model i tornar a començar les avaluacions, i encara així sort que no es guardava només el millor sinó els tres millors i no vam haver de repetir tot l'entrenament.

- **Entrenar un *Transformer*:** 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 model d'intel·ligència artifical des de zero sense tindre coneixement de les eines que anàvem a utilitzar i sense haver entrenat mai cap model, perquè el que s'havia vist fins el moment al llarg de la carrera havia estat un coneixement més bé teòric.

- **Entrenar un model bilingüe que mai ha vist el castellà en preentrenament:** Una gran dificultat a la qual ens vam enfrontar va ser que havíem d'entrenar un model que fora capaç d'entendre textos tant en català com en castellà i després els resumira. Però havíem de partir d'un model preentrenat que només havia vist textos en català, això vol dir que el model ha d'aprendre a fer una nova tasca: resumir, i a més ha d'aprendre una nova llengua que, encara que siguen llengües veïnes, no és menyspreable la càrrega de treball durant l'entrenament.

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.

- **Model de resum automàtic en castellà**: Un punt amb el qual seria interessant continuar la línia d'investigació, seria realitzar la mateixa experimentació que s'ha portat a terme per al català però amb la part del corpus DACSA en castellà, perquè aquest disposa d'un volum de notícies molt major que podria permetre analitzar si el fet de tindre més notícies segueix permetent millorar els resultats respecte del català com ocorria entre NASca i NASes, i si amb un corpus amb una quantitat de fonts significativament major, segueix sent capaç de classificar les notícies amb tant d'èxit. L'entrenament d'un model en castellà de fet estava plantejat inicialment per a

aquest treball, però precisament per la quantitat de notícies el procés d'entrenament és molt major, per la qual cosa es va decidir continuar entrenant models en català perquè eren més ràpids d'entrenar i per tant es podia dedicar el temps disponible a fer una major quantitat d'experimentacions sobre eixos models en lloc d'entrenar més models i tindre una experimentació més pobre.

- **Major varietat dialectal en el corpus DACSA**: Un descobriment molt interessant d'aquest treball ha estat el fet que quan el model veia notícies de fonts que no havia vist mai, però que resultaven ser d'un diari valencià, tendia a classificar-les en l'únic diari valencià que havia vist. Aquest podria ser un punt molt interessant en el qual aprofondir-se augmentant el corpus DACSA amb major presència de diaris valencians i balears per poder fer una analisi més precisa d'aquest fenómen. Possiblement també podria tindre el seu interès realitzar la mateixa operació amb la part de castellà, però per a això ja caldria anar-se'n a barrejar castellà i espanyol sud-americà que tenen uns trets lingüístics més diferenciadors que no les varietats dialectals del castellà presents en l'Estat Espanyol.

- **Model bilingüe classificador**: Una altra línia en la qual podria ampliar-se aquest treball seria al model bilingüe afegir-li la tasca de classificació de notícies per tal de poder analitzar tant la seua capacitat classificadora com l'efecte que podria tindre sobre aquest model el fet d'anar forçant-li les distintes fonts. Aquest cas tindria gran interès perquè es podria forçar a que una notícia catalana és d'algun diari en castellà i que intente traure el resum en castellà o qualsevol combinació que puguem imaginar i tinguera interès analitzar. També seria interessant en el model bilingüe actual poder analitzar quin efecte tindria sobre els seus resums el fet de forçar l'idioma d'entrada, de manera que per exemple li donàrem una notícia en català però li diguérem que es tracta d'una notícia en castellà, això el confondria o mantindria indistintament el seu criteri?

Bibliografia

[1] E. Segarra Soriano, V. Ahuir, L.-F. Hurtado, and J. González, "DACSA: A large-scale dataset for automatic summarization of Catalan and Spanish newspaper articles," pp. 5931–5943, July 2022.

[2] "Pàgina web del grup d'investigació ELiRF." http://elirf.dsic.upv.es/.

[3] 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.

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

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

[6] 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.

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

[8] 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.

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

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

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

[12] W. Wang and J. Gang, "Application of convolutional neural network in natural language processing," in *2018 international conference on information Systems and computer aided education (ICISCAE)*, pp. 64–70, IEEE, 2018.

[13] Y. Kim, "Convolutional neural networks for sentence classification," *CoRR*, vol. abs/1408.5882, 2014.

[14] 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.

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

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

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

[18] 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.

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

[20] 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.

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

[22] 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.

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

[24] W.-T. Hsu, C.-K. Lin, M.-Y. Lee, K. Min, J. Tang, and M. Sun, "A unified model for extractive and abstractive summarization using inconsistency loss," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Melbourne, Australia), pp. 132–141, Association for Computational Linguistics, July 2018.

[25] L. Scanlon, S. Zhang, X. Zhang, and M. Sanderson, "Evaluation of cross domain text summarization," p. 1853–1856, 2020.

[26] 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.

[27] 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.

[28] H. Lin and V. Ng, "Abstractive summarization: A survey of the state of the art," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 9815–9822, Jul. 2019.

[29] Y. Mehdad, G. Carenini, and R. T. Ng, "Abstractive summarization of spoken and written conversations based on phrasal queries," pp. 1220–1230, June 2014.

[30] P.-E. Genest and G. Lapalme, "Fully abstractive approach to guided summarization," pp. 354–358, July 2012.

[31] K. Woodsend and M. Lapata, "Learning to simplify sentences with quasi-synchronous grammar and integer programming," pp. 409–420, July 2011.

[32] W. Li, L. He, and H. Zhuge, "Abstractive news summarization based on event semantic link network," pp. 236–246, Dec. 2016.

[33] 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.

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

[35] 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.

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

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

[38] 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.

[39] 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.

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

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

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

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

[44] V. Ahuir, L.-F. Hurtado, J. González, 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.

[45] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, and L. He, "A survey on text classification: From shallow to deep learning," *CoRR*, vol. abs/2008.00364, 2020.

[46] Y. LeCun, L. D. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. A. Muller, E. Sackinger, P. Simard, *et al.*, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the statistical mechanics perspective*, vol. 261, no. 276, p. 2, 1995.

[47] A. Byerly, T. Kalganova, and I. Dear, "No routing needed between capsules," *Neurocomputing*, vol. 463, pp. 545–553, 2021.

[48] A. Bhavani and B. Santhosh Kumar, "A review of state art of text classification algorithms," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1484–1490, 2021.

[49] M. E. Maron, "Automatic indexing: an experimental inquiry," *Journal of the ACM (JACM)*, vol. 8, no. 3, pp. 404–417, 1961.

[50] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[51] T. Joachims, "Text categorization with support vector machines," *Proc. European Conf. Machine Learning (ECML'98)*, 01 1998.

[52] R. Aly, S. Remus, and C. Biemann, "Hierarchical multi-label classification of text with capsule networks," pp. 323–330, July 2019.

[53] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," pp. 1–6, 2017.

[54] P. Liu, X. Qiu, and X. Huang, "Recurrent neural network for text classification with multi-task learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, p. 2873–2879, AAAI Press, 2016.

[55] 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.

[56] M. Grusky, M. Naaman, and Y. Artzi, "Newsroom: A dataset of 1.3 million summaries with diverse extractive strategies," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, (New Orleans, Louisiana), pp. 708–719, Association for Computational Linguistics, June 2018.

[57] P. Marco García, "Resum abstractiu de noticies basat en xarxes neuronals," 2021.

[58] P. Ortiz Suarez, B. Sagot, and L. Romary, "Asynchronous pipelines for processing huge corpora on medium to low resource infrastructures," 07 2019.

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

[60] 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.

[61] R. Bommasani and C. Cardie, "Intrinsic evaluation of summarization datasets," pp. 8075–8096, Nov. 2020.

[62] W. Kryściński, R. Paulus, C. Xiong, and R. Socher, "Improving abstraction in text summarization," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Brussels, Belgium), pp. 1808–1817, Association for Computational Linguistics, Oct.-Nov. 2018.

[63] 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.

[64] Y. Zou, X. Zhang, W. Lu, F. Wei, and M. Zhou, "Pre-training for abstractive document summarization by reinstating source text," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Online), pp. 3646–3660, Association for Computational Linguistics, Nov. 2020.

[65] "Curs HuggingFace sobre la seua llibreria *transformer*." https://huggingface.co/course.

[66] 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.

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

APÈNDIX A
Resums d'exemple

En aquest annex veurem un exemple tangible de com ha resumit el nostre model una notícia real que forma part del conjunt de test. Com el model M2 és al que hem dedicat més temps mostrarem només els resums generats per aquest, però en tots els escenaris dels quals hem anat parlant al llarg de tot aquest treball (resums de longitud estàndard i llarg i forçant cadascuna de les fonts).

Notícia

SFJ / Madrid.. El punt sis de l'ordre del ple que ha celebrat aquest dimarts el Congrés espanyol era ben clar: demanar explicacions al govern del PP que lidera Mariano Rajoy "sobre els reiterats ajornaments i incompliments en la construcció del Corredor Mediterrani". La moció l'ha presentada Joan Baldoví (Compromís), del Grup parlamentari Mixt, i ha rebut el suport en l'hemicicle d'ERC, del PSOE, de Podemos, del PNV així com, de manera vetllada, de Ciudadanos. No obstant això, tot i que les reclamacions tenien la intenció d'interpel·lar Rajoy, aquest no s'hi trobava al ple, així com tampoc la presidenta del Congrés espanyol, l'exministra de Foment i responsable de les infraestructures de l'Estat, Ana Pastor, qui s'ha deixat caure al final de la moció i ha presidit la Mesa durant la votació.. Baldoví ha insistit en fer veure el greuge compartiu que hi ha entre altres territoris de l'Estat i els territoris de l'arc mediterrani i ha reiterat que s'alegra "sincerament de cada inversió" feta. Després de l'alegria venia la reprimenda al PP, perquè el "govern paralitzà en 2012 el Corredor Mediterrani. D'açò no m'alegre", ha afegit. "Hi ha bones notícies a altres territori, però en el nostre no", per això "és hora de dir prou a les promeses incomplides". El diputat de Compromís ha recordat que la moció recull esmenes de C's, PSOE i PP, per això "volem que s'aprove, perquè hi haja compromisos", ha dit. "Volem pressupostos de 2017, 2018 i 2019 que tinguen partides obertes, que es tinga en compte i es treballe amb les comunitats autònomes i que el Corredor tinga via d'ample europeu. Coses raonables", ha afegit el diputat valencià, qui en última instància ha afegit que "voldria el mateix tuit amb la foto de Revilla (el president de Cantàbria) amb el ministre de Foment, però amb el meu president Ximo Puig, amb Puigdemont, etc.", ha acabat dient Baldoví, tot fent el paral·lelisme amb l'anunci de les infraestructures que rebrà la comunitat càntabra. . Joan Baldovi durant la seua intervenció al congrés espanyol. Tot seguit, han fet ús del seu temps diferents diputats valencians com Cantó (C's), qui, segons ha recordat Lizondo, fou "el primer en dur el tema" del Corredor al Congrés espanyol.

“Han passat quasi 30 anys i seguim igual”, ha dit Cantó, qui també ha aprofitat per posar el dit a l’ull de diferents diputats sobre temes que poc o res tenien a veure amb la moció plantejada.. José Luís Àbalos, del PSOE, ha recordat les inversions de Zapatero, perquè “els socialistes sempre hem apostat per aquest corredor i anem a donar suport a la moció”. Un to diferent ha fet servir Miguel Barrachina fent veure que el Corredor potser no cal tant, perquè segons el diputat del PP els estrangers “valoren les infraestructures d’Espanya quasi amb un excel·lent”. A més, ha incidit amb el fet que en el corredor s’han invertit 13.500 milions dels 17 pressupostats. . Votació de la moció de Compromís sobre els "reiterats ajornaments i incompliments en la construcció del Corredor". Iñigo Barandiaran, del PNV, ha estat més decidit des del primer moment en dir reiteradament que donaran suport a la moció perquè l’eix mediterrani és vertebrador. Per la seua banda, Teresa Jordà, d’ERC, s’ha situat en el to exigent que primerament havia manifestat Baldoví. Jordà, però, ha fet servir un conte: “Madrid, capital del Corredor Mediterrani”. Aquest ha acabat amb l’afirmació que aquesta infraestructura no és un capritx, “és una prioritat” i finalment ha retret al Govern espanyol allò que el secretari autonòmic d’Habitatge, Obres públiques i Vertebració del Territori, Josep Vicent Boira, va afirmar fa unes setmanes en fer-se ressò que el Govern espanyol es gastà diners europeus destinats al Corredor Mediterrani per a unir les estacions madrilenyes.. Finalment, Fèlix Alonos, de Podemos, ha manifestat que donaran suport a la moció, alhora que ha retret que Rajoy no s’hi trobara en el ple i ha mostrat el seu disgust pel fet que el PP “vote una cosa a les Corts i una altra al Congrés”.. En el torn de votacions tots els grups han votat favorablement la moció, de manera que la transacció presentada posteriorment per Baldoví ha rebut el suport de tots els grups, inclòs un PP sense Rajoy, que acceptava la reprimenda per no “impulsar i ajornar reiteradament” el Corredor Mediterrani.

Resum original

Tots els grups voten a favor de la proposta de Compromís i Joan Baldoví ha dit que "volem pressupostos de 2017, 2018 i 2019 que tinguen partides obertes i que el Corredor tinga via d’ample europeu".

Resum generat pel model M2

El diputat de Compromís ha recordat que la moció recull esmenes de C’s, PSOE i PP, per això “volem que s’aprove, perquè hi haja compromisos”.

Resum generat pel model M2 forçant font correcta

El diputat de Compromís ha recordat que la moció recull esmenes de C’s, PSOE i PP, per això “volem que s’aprove, perquè hi haja compromisos”.

Resum generat pel model M2 forçant font CA01

> Joan Baldoví (Compromís), del Grup parlamentari Mixt, rep el suport en l'hemicicle d'ERC, del PSOE, de Podemos, del PNV així com, de manera vetllada, de Ciudadanos.

Resum generat pel model M2 forçant font CA02

> El diputat de Compromís ha recordat que la moció recull esmenes de C's, PSOE i PP, per això "volem que s'aprove, perquè hi haja compromisos".

Resum generat pel model M2 forçant font CA03

> El diputat de Compromís ha recordat que la moció recull esmenes de C's, PSOE i PP, per això 'volem que s'aprove, perquè hi haja compromisos'.

Resum generat pel model M2 forçant font CA04

> El diputat de Compromís ha recordat que la moció recull esmenes de C's, PSOE i PP, per això "volem que s'aprove, perquè hi haja compromisos".

Resum generat pel model M2 forçant font CA05

> Joan Baldoví (Compromís), del Grup parlamentari Mixt, ha rebut el suport en l'hemicicle d'ERC, del PSOE, de Podemos, del PNV així com, de manera vetllada, de Ciudadanos.

Resum generat pel model M2 forçant font CA06 (correcta)

> El diputat de Compromís ha recordat que la moció recull esmenes de C's, PSOE i PP, per això "volem que s'aprove, perquè hi haja compromisos".

Resum llarg generat pel model M2

> El diputat de Compromís, Joan Baldoví, ha recordat que la moció recull esmenes de C's, PSOE i PP, per això "volem que s'aprove, perquè hi haja compromisos", ha dit el diputat valencià en la seua intervenció durant la votació al congrés espanyol d'aquest dimarts. El diputat valencià ha afegit que "voldria el mateix tuit amb la foto de Revilla (el president de Cantàbria) amb el ministre de Foment, però amb Puig, amb Puigdemont, etc.".

Resum llarg generat pel model M2 forçant font correcta

El diputat de Compromís, Joan Baldoví, ha recordat que la moció recull esmenes de C's, PSOE i PP, per això "volem que s'aprove, perquè hi haja compromisos", ha dit el diputat valencià en la seua intervenció durant la votació al congrés espanyol d'aquest dimarts. El diputat valencià ha afegit que "voldria el mateix tuit amb la foto de Revilla (el president de Cantàbria) amb el ministre de Foment, però amb Puig, amb Puigdemont, etc.".

Resum llarg generat pel model M2 forçant font CA01

Joan Baldoví (Compromís), del Grup parlamentari Mixt, ha rebut el suport en l'hemicicle d'ERC, del PSOE, de Podemos, del PNV així com, de manera vetllada, de Ciudadanos. No obstant això, tot i que les reclamacions tenien la intenció d'interpel·lar Rajoy, aquest no s'hi trobava al ple.

Resum llarg generat pel model M2 forçant font CA02

El diputat de Compromís, Joan Baldoví, ha recordat que la moció recull esmenes de C's, PSOE i PP, per això "volem que s'aprove, perquè hi haja compromisos". El diputat valencià ha afegit que "voldria el mateix tuit amb la foto de Revilla (el president de Cantàbria) amb el ministre de Foment, però amb el meu president Ximo Puig, amb Puigdemont, etc.

Resum llarg generat pel model M2 forçant font CA03

El diputat de Compromís ha recordat que la moció recull esmenes de C's, PSOE i PP, per això "volem que s'aprove, perquè hi haja compromisos", ha dit. 'Volem pressupostos de 2017, 2018 i 2019 que tinguen partides obertes, que es tinga en compte i es treballe amb les comunitats autònomes'.

Resum llarg generat pel model M2 forçant font CA04

Joan Baldoví (Compromís), del Grup parlamentari Mixt, ha rebut el suport en l'hemicicle d'ERC, del PSOE, de Podemos, del PNV així com, de manera vetllada, de forma vetllada. No obstant això, tot i que les reclamacions tenien la intenció d'interpel·lar Rajoy, aquest no s'hi trobava al ple.

Resum llarg generat pel model M2 forçant font CA05

Joan Baldoví (Compromís), del Grup parlamentari Mixt, ha rebut el suport en l'hemicicle d'ERC, del PSOE, de Podemos, del PNV així com, de manera vetllada, de Ciudadanos. No obstant això, tot i que les reclamacions tenien la intenció d'interpel·lar Rajoy, aquest no s'hi trobava al ple.

Resum llarg generat pel model M2 forçant font CA06 (correcta)

El diputat de Compromís, Joan Baldoví, ha recordat que la moció recull esmenes de C's, PSOE i PP, per això "volem que s'aprove, perquè hi haja compromisos", ha dit el diputat valencià en la seua intervenció durant la votació al congrés espanyol d'aquest dimarts. El diputat valencià ha afegit que "voldria el mateix tuit amb la foto de Revilla (el president de Cantàbria) amb el ministre de Foment, però amb Puig, amb Puigdemont, etc.".

APÈNDIX B
Exemple de confusió de fonts

Notícia

La Fiscalia Superior de Catalunya ha demanat deu anys d'inhabilitació per a càrrec públic o de govern per a l'expresident de la Generalitat Artur Mas i nou per a l'exvicepresidenta Joana Ortega i l'exconsellera d'Educació, Irene Rigau, per l'organització del procés participatiu del 9-N. El ministeri públic considera Mas autor dels delictes de desobediència greu i prevaricació administrativa, i a Ortega i Rigau les considera cooperadores necessàries dels mateixos delictes. Segons els fiscals, tots tres eren çonscients"que amb les accions preparatòries del procés participatiu "trencaven l'obligat compliment de les decisions del Tribunal Constitucional", que el 4 de novembre, cinc dies abans de les votacions, va ordenar suspendre el procés. En el seu escrit, de 38 pàgines, la fiscalia recorda que el TC va suspendre el 29 de setembre del 2014 el decret de convocatòria de la consulta popular no referendària que s'havia aprovat el 26 de setembre. Després d'uns dies d'indecisió, el 14 d'octubre Mas va comparèixer en públic per anunciar un "procés de participació ciutadana"per manifestar la seva "opinió sobre el futur polític de Catalunya". Allò va activar diversos procediments administratius "encaminats a organitzar la votació", com que els col·legis electorals serien instituts públics o l'encàrrec a empreses privades per a feines com el sistema informàtic o la distribució logística. La majoria d'aquests encàrrecs es van fer en el marc de contractes previs. Entre els principals actes administratius que la fiscalia considera que demostren els delictes hi ha la creació de la pàgina web www.participa2014.cat i del sistema informàtic, la compra de 7.000 ordinadors portàtils per a les meses de votació, la preparació dels instituts com a seus electorals, la fabricació d'urnes i butlletes per part del Centre d'Iniciatives per a la Reinserció (CIRE), una empresa pública que dóna feina a presos, una campanya de publicitat institucional i d'informació per correu postal i una assegurança per als voluntaris. En el cas dels instituts, el fiscal diu que els directors territorials van convocar reunions amb els directors dels centres i els van "sol·licitar de forma vehement la seva col·laboració"per tal de poder preparar les urnes, paperetes i ordinadors els dies anteriors a la votació. També fa notar que els 7.000 ordinadors coincideixen pràcticament amb les 6.700 meses electorals, i que després d'aquella jornada electoral es van tornar a repartir per diversos centres diferents dels que havien estat. Tots aquests actes es van fer abans del 4 de novembre. El 31 d'octubre el govern espanyol va impugnar aquests actes i va presentar un conflicte de competència davant del TC, que va emetre una providència el 4 de novembre on suspenia els actes realitzats des del 31 d'octubre i els futurs.

El 7 de novembre el Govern va presentar davant del TC un recurs de súplica i una petició d'aclariment, on demanava una resolució ràpida per tal de no deixar passar la data del 9 de novembre. Això, segons el fiscal, demostraria que la Generalitat sabia que el procés participatiu estava suspès. A més, des del 3 de novembre fins el dia abans de la jornada decisiva una empresa postal privada va repartir la publicitat institucional a tots els domicilis, el 4 de novembre es va fer efectiva l'ampliació de l'assegurança per las voluntaris, la web de participació va seguir activa, la campanya institucional va seguir endavant, i els dies 7 i 8 de novembre les empreses informàtiques van deixar a punt els ordinadors, les aplicacions, el sistema informàtic i la preparació dels voluntaris. També els dies 7 i 8 van arribar a les seus electorals les urnes i les butlletes, i es va habilitar un pavelló de la Fira de Barcelona perquè els periodistes seguissin el procés participatiu. El dia 9 de novembre del 2014 es van obrir els instituts públics i les seus de la Generalitat a l'estranger per poder votar, que van estar disponibles fins el 25 de novembre, i les empreses informàtiques subcontractades van col·laborar en tot moment perquè el sistema informàtic funcionés. Així, els fiscals Francisco Bañeres i Emilio Sánchez Ulled consideren que Mas, Ortega i Rigau van "articular una estratègia de desafiament complet i efectiu a la suspensió"del TC. "Mas, emparat simplement en la seva voluntat, que va convertir gens raonablement en aparent font de normativitat, actuant amb plena consciència i voluntat, va deixar de suspendre oficialment la convocatòria", afegeix, ja que no va desconvocar el procés participatiu ni va aturar totes les actuacions públiques que ja estaven en marxa. De fet, recorda que els actes administratius suspesos al setembre sí que es van aturar, mentre que els de novembre "van continuar fins a completar-los, això sí, amb la convenient discreciói un cert çlima d'opacitat", "generant l'aparença que les actuacions administratives públiques es paralitzaven i el procés quedava exclusivament en mans de ciutadans voluntaris, tot i que en realitat no era així". "Els acusats eren plenament conscients que amb això trencaven l'obligat acatament de les decisions del TC", afirma la fiscalia, que considera que des de l'inici eren conscients de la possible impugnació i per això van actuar "disposats en tot moment a eludir el control jurisdiccional". Per últim, els fiscals recorden que el juny del 2015 el TC va declarar inconstitucionals tots els actes preparatius del procés participatiu. Per tot això, la fiscalia demana deu anys d'inhabilitació especial per a càrrec públic electe sigui d'àmbit local, autonòmic o estatal, així com per a funcions de govern en àmbit autonòmic o estatal, per a Mas, o alternativament nou anys i mig d'inhabilitació i 36.000 euros de multa. Per a Ortega i Rigau, demana, com a cooperadores necessàries, nou anys d'inhabilitació o, alternativament, vuit anys i set mesos d'inhabilitació i 30.000 euros de multa.

Resum generat

El ministeri públic considera Mas autor dels delictes de desobediència greu i prevaricació administrativa, i a Ortega i Rigau les considera cooperadores necessàries dels mateixos delictes.

Resum generat forçant la font CA03

> El ministeri públic considera Mas autor dels delictes de desobediència greu i prevaricació administrativa, i a Ortega i Rigau les considera cooperadores necessàries dels mateixos delictes.

Resum generat forçant la font CA04 (correcta)

> El ministeri públic considera Mas autor dels delictes de desobediència greu i prevaricació administrativa, i les considera cooperadores necessàries dels mateixos delictes.

APÈNDIX C
Exemples de resums condicionant l'estil

En aquest annex veurem tot un seguit d'exemples per al model M2 que evidencien com forçar una font a l'hora de generar el resum, és capaç de condicionar el resultat d'una o altra manera. Trobem exemples en els quals els canvis són de caràcter estilístic, com ocorre amb els distints tipus de cometes, en altres directament canvia la semàntica del resum i també exemplifiquem els casos on forçar la font a penes altera el resultat, que són els més sovintejats. En tots els exemples hi ha dos resums d'una mateixa notícia, per una banda el resum obtingut sense donar al model informació sobre quina és la font original, i un segon resum en el qual s'ha forçat una font que pot ser o no la correcta.

Primer veiem quatre exemples en els quals forçant la font s'han obtingut resums que no tenen res a veure a nivell de contingut l'ún amb l'altre; el cinquè exemplica precisament la situació contrària: dos resums que malgrat forçar-se la font s'ha mantingut el mateix resum; i per acabar hem posat un parell de casos on s'exemplifiquen les diferències focalitzant-se en l'estil i no tant en el contingut.

Exemple 1

Comencem amb un exemple prou interessant perquè podem observar que la font que s'ha forçat és precismaent la font correcta, i amb els resultats que hem estudiat prèviament amb més d'un 90% d'encerts en la classificació resulta sorprenenet que hagen sigut dos resums tan diferents. El més probable en aquest cas és que el model s'haja enganyat a l'hora de fer classificació i per tant haja generat el resum a l'estil d'una font que no era la correcta i llavors quan li hem donat la correcta ja ho ha fet a l'estil que s'esperava.

Font original: CA01 Font forçada: CA01 Resum a cegues: Des del 13 de març fins al 14 d'abril els Bombers han trobat 42 persones mortes al entrar als domicilis. Resum font forçada: La Creu Roja alerta que el confinament afecta especialment aquest col·lectiu i demana que l'entorn dels avis no perdi el contacte amb ells.

Exemple 2

En aquest cas ens trobem amb un exemple en què sent diferents la font original i la font que s'ha forçat, també s'han obtingut resums que encara que es pot veure fàcilment que tracten sobre la mateixa temàtica, la part del discurs en què es centra cadascun és diferent.

Font original: CA02
Font forçada: CA01
Resum a cegues: Els guardons honorífics s'entreguen des de fa sis anys en una cerimònia a part de la gala dels Oscars.
Resum forçat: El guionista Hayao Miyazaki i l'actriu Maureen O'Hara també rebran el guardó.

Exemple 3

En cadascun dels resums ha decidit centrar-se en una de les parts de la negociació. I gasta en ambdós casos les cometes «», que com veurem més endavant és un fet distintiu de cadascuna de les fonts i que té un cert interés.

Font original: CA03
Font forçada: CA02
Resum a cegues: El primer secretari del PSC espera que «es produeixi un acord» al voltant dels «escenaris de diàleg» que s'estan plantejant.
Resum font forçada: ERC no retirarà l'esmena a la totalitat contra els Pressupostos si no cessa «la repressió» contra l'independentisme.

Exemple 4

En aquest cas no hi ha un canvi semàntic, perquè es continua dient el mateix però es presenta la informació d'una manera diferent: en un cas parla del secretari de treball i de xifres exactes, i en l'altre de l'EPA i de percentatges.

Font original: CA03
Font forçada: CA02
Resum a cegues: El secretari de Treball, Josep Ginesta, assegura que en els primers tres mesos del 2020 s'han perdut 118.500 feines.
Resum font forçada: La taxa d'atur ha pujat fins al 10,66% entre gener i març, segons l'EPA.

Exemple 5

També podem veure en altres exemples, que de fet són els majoritaris, on per molt que forcem una font distinta de l'original, el model manté el seu propi criteri de resum i genera un resum gairebé idèntic al que generava sense dir-li cap font; encara que no n'hem posat tants casos perquè resulten de menor interès, perquè les dades ja reflecteixen aquesta situació. En este cas només hi ha una diferència: *del decret* front a *del NOU decret*.

Font original: CA06
Font forçada: CA01
Resum a cegues: El conseller d'Educació, Vicent Marzà, ha defensat aquest dimarts que l'esborrany del decret d'admissió d'alumnes al País Valencià inclou novetats com tornar a prevaldre la proximitat al centre escolar o eliminar els punts que donaven els col·legis per circumstàncies específiques o ser antic alumne.
Resum forçat: El conseller d'Educació, Vicent Marzà, ha defensat aquest dimarts que l'esborrany del nou decret d'admissió d'alumnes al País Valencià inclou novetats com tornar a prevaldre la proximitat al centre escolar o eliminar els punts que donaven els col·legis per circumstàncies específiques o ser antic alumne.

Exemple 6

En aquest exemple a banda de tornar a veure com els resums no tenen res a veure, volem destacar que les diferències que aconseguim, no són només a nivell semàntic, sinó també en l'estil, en concret en el tipus de cometes emprat, que s'ha pogut observar com cada font gasta un tipus de cometes diferent i eixe fet diferencial ha estat après per la nostra Intel·ligència Artificial. En cada cas veiem com per resumir a mode de citació ha decidit emprar un tipus de cometes diferents.

Font original: CA05
Font forçada: CA03
Resum a cegues: «La derrota del plebiscit pot ser una llosa pel lideratge de Mas, malgrat l'èxit de la seva coalició que podrà governar sense altres entrebancs que els originats per les seves peculiaritats internes».
Resum font forçada: 'Si el seny polític existeix, i IC i PSC es mantenen fidels al seu catalanisme polític (comptant que PSOE i Podemos ho puguin entendre, cosa gens fàcil), això faria inviable el front dit constitucionalista'.

Exemple 7

Aquest exemple mostra el cas d'una notícia especialment conflicitva, perquè s'ha detectat que forçant la font que forçàrem s'obtenia un resultat diferent a l'obtingut sense forçar. Però amb la peculiaritat, que forçant i no forçant sí que eren totalment diferents en contingut, però en canvi una vegada se forçava, es forçara la que es forçara, el resultat era pràcticament idèntic, els únics canvis eren precisament en les cometes i en la forma de referir-se al brexit. Tot això és per a les fonts CA03-CA06, perquè en les fonts CA01 i CA02 (l'original) els resums obtinguts han estat idèntics a l'obtingut sense forçar cap font. En cadascun dels exemples es veu com per a esciure *aixecar-se* extret de la notícia original, i per això es posa entre cometes, decidint en cadascuna de les fonts gastar un o altre tipus. A més a més també resulta curiós com per a referir-se al brèxit en uns casos ho posa entre cometes simples, indicant que es tracta d'una paraula estrangera, i en altres el que fa és donar la seua definició. De fet s'ha constatat com en els diferents resums que genera, quan parla de brexit només apareix de dos maneres: 'brexit' o Brexit.

Font original: CA02

Resum a cegues: "Si és amb un segon referèndum o amb un altre mètode, això és secundari", afirma l'exlíder laborista.

Forçant font CA03: El primer ministre britànic crida els britànics a la insurrecció i a «aixecar-se» contra el 'brexit'.

Forçant font CA04: El primer ministre britànic crida els britànics a la insurrecció i a "aixecar-se" contra el 'brexit'.

Forçant font CA05: El primer ministre britànic crida els britànics a "aixecar-se" contra la sortida del Regne Unit de la UE.

Forçant font CA06: El primer ministre britànic crida els britànics a "aixecar-se" contra la sortida del Regne Unit de la UE.

APÈNDIX D

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 D.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 D.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 notícies en **català** i per a això s'ha fet servir un corpus de grans dimensions (DACSA) que suposa un dels majors corpus de notícies en ambdós llengües. Disposar de corpus de grans dimensions en una determinada llengua és de vital importància per poder entrenar models que puguen treballar en eixa llengua. Tractant-se el català d'una llengua minoritària, però sobretot minoritzada, que ha aplegat a ser censurada i es troba en un constant estat de diglòssia que porta a una desigualtat negativa fins i tot en territoris catalanoparlants. Amb la utilització d'aquest corpus i l'entrenament de models com els que es presenten en aquest treball

s'incentiva la recerca en català i es facilita el desenvolupament d'eines de PLN en aquesta llengua per a projectes futurs.

- **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 notícies que és el que ens pertoca en aquest projecte es permet 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