PROCESSAMENT DE LLENGUATGE NATURAL

Pau Miró Fàbregas i Isaac Ruiz García

INSTITUT SA PALOMERA

PROCESSAMENT DE LLENGUATGE NATURAL

ACCÉS AL CODI

Per poder accedir als **.html** i als **.py** empleats per aquesta pràctica has d'accedir al següent repositori de github on estàn publicats.

Repositori Oficial

SELECCIÓ ARTICLES

Hem seleccionat cinc articles en llengua catalana de diversos mitjans de comunicació digitals de referència. La tria no ha estat aleatòria, sinó que respon a criteris tècnics per posar a prova el rendiment dels models de Processament de Llenguatge Natural (PLN) en diferents escenaris:

1. **Varietat de dominis (Temàtiques):** Hem escollit articles que cobreixen àrees molt dispars: tecnologia (MareNostrum 5), esports (Girona FC), salut i societat (Vall d'Hebron), cultura (Hitchcock i la IA) i economia (dades de l'atur). Aquesta diversitat és fonamental per avaluar la capacitat del model en el **Pas 3**, ja que permet verificar si les tècniques de vectorització (TF-IDF i embeddings) són capaces de diferenciar vocabularis tècnics molt específics.
2. **Riquesa de vocabulari:** Cada article aporta un corpus de paraules clau diferenciat. Per exemple, l'article sobre el MareNostrum 5 conté terminologia d'infraestructures tecnològiques, mentre que el de l'atur se centra en dades estadístiques i econòmiques. Això ens permetrà analitzar com les *stopwords* i la normalització afecten la representació vectorial segons el context.
3. **Complexitat de l'HTML:** Les fonts escollides (*Via Empresa*, *VilaWeb*, *Ara*, *L'Esportiu* i *El Punt Avui*) presenten estructures HTML diverses. Això ens permet treballar el **Pas 1** amb diferents graus de dificultat, assegurant que el nostre *scraper* sigui capaç de filtrar elements irrelevants com anuncis, menús i metadades.
4. **Consistència lingüística:** Tot i que els models de HuggingFace solen estar entrenats en anglès, hem optat pel català per avaluar la potència actual dels models territorials (com el projecte AINA) i garantir que el resum generat al **Pas 4** sigui fàcilment verificable per nosaltres en la nostra llengua.

METODOLOGIA

PAS 1: PROCESSAMENT HTML

Hem creat un primer codi utilitzant la llibreria **BeautifulSoup** de python. En el primer moment ha donat uns bons resultats en alguns articles però en alguns altres no.

A la captura podem observar que:

- **Cultura.html (Buit):** El diari *Ara* sovint posa el text dins de amb classes específiques i no només <p> pelats, o potser el text s'ha carregat via JavaScript i BeautifulSoup no el veu.
- **Esports.html ("Continguts del lloc"):** Ha agafat el primer <h1> que ha trobat el codi, que en aquest cas era un títol invisible del menú de navegació en lloc del títol de la notícia.
- **Salut.html (Text equivocat):** El text sobre el "gas pebre" no és de l'article de la Vall d'Hebron. Segurament és una notícia destacada d'una barra lateral. Com que el codi agafa **tots** els <p>, ha començat a llegir la publicitat o les notícies relacionades abans que la principal.

Per solucionar això hem implementat una solució al nostre codi final que consisteix en l'extracció mitjançant metadades **og:title** per garantir la precissió del títol i també hem aplicat un filtre de caràcters.

Hem obtingut uns resultats molt millors però seguim tenim un problema amb Salut.html

Després d'analitzar detingudament el codi font de **VilaWeb** (Salut.html), hem identificat que el text de l'editorial sobre el "gas pebre" es trobava dins d'una etiqueta <article> continguda en una capa anomenada #navigation-layer. Com que el nostre scraper cercava el primer element de tipus article, capturava aquest menú desplegable abans que la notícia real situada al <main>.

Per resoldre definitivament aquests problemes de "soroll" i garantir la qualitat del text original, hem implementat les següents millores tècniques al nostre fitxer scraper.py:

1. **Eliminació de Nodes Conflictius (decompose):** Hem introduït una fase de preprocessament de l'arbre DOM on eliminem explícitament els contenidors de navegació (#navigation-layer, #text_minut), així com les etiquetes de capçalera, peu de pàgina i barres laterals abans de procedir a l'extracció. Això garanteix que la cerca posterior es faci només sobre el contingut útil.

2. **Targeting per Classes Específiques:** Hem refinat la cerca del cos de la notícia prioritzant classes CSS d'article real com .content-noticia-body o .article-content-publi, evitant així fragments publicitaris o recomanacions.

3. **Heurística de Qualitat de Paràgrafs:** Hem ajustat el filtre de longitud mínima. Ara el sistema només accepta paràgrafs amb més de 40 caràcters, descartant automàticament

peus de foto, signatures d'autor o avisos de *cookies* que haguessin pogut sobreviure a la neteja inicial.

Resultat final del Pas 1: Amb aquesta versió final del codi, hem aconseguit que els 5 articles s'extreguin amb una precisió del 100%, obtenint títols correctes i un text net preparat per a la fase de preprocessament lingüístic.

EXTRACCIO A CSV

Un cop finalitzada l'extracció de text net de tots els articles, hem decidit implementar una fase d'exportació a un fitxer estructurat. L'objectiu és separar la lògica del **Pas 1** (Scraping) de la resta del pipeline, permetent que el Pas 2 pugui llegir directament dades netes sense dependre de l'HTML original.

REPTES EN L'EXPORTACIÓ DE TEXT NATURAL

Durant les primeres proves d'exportació, vam detectar que el format CSV es corrompia a causa de la naturalesa del text:

- **Comes gramaticals:** Les comes pròpies de l'article es confonien amb els delimitadors de columnes del CSV.
- **Salts de línia:** Els caràcters \n trencaven l'estructura de files, dividint un sol article en múltiples línies il·legibles.

SOLUCIÓ IMPLEMENTADA

Per garantir la integritat del nostre corpus de dades, hem aplicat dues mesures tècniques al codi d'exportació:

1. **Encapsulament de camps (QUOTE_ALL):** Hem configurat el mòdul csv de Python per envoltar automàticament cada camp amb cometes dobles. Això "blinda" el contingut i evita que qualsevol coma interna trenqui la taula.
2. **Normalització de caràcters de control:** Hem substituït tots els salts de línia (\n) per espais simples mitjançant el mètode .replace(), assegurant que cada article ocupi exactament una fila al fitxer articles_nets.csv

PAS 2: PREPROCESSAMENT DEL TEXT

L'objectiu d'aquest pas és transformar el text brut extret de l'HTML en un format normalitzat, eliminant el "soroll" lingüístic que no aporta valor semàntic per als models d'IA. Tal com demana l'activitat, hem implementat i comparat dues metodologies de preprocessament per analitzar-ne el rendiment.

METODOLOGIA A: NETEJA BÀSICA (PYTHON STANDARD)

En aquesta primera aproximació, hem utilitzat funcions natives de Python per realitzar una neteja superficial:

- **Normalització:** Conversió de tot el text a minúscules.
- **Eliminació de puntuació:** Ús de la llibreria string per treure punts, comes i signes especials.

- **Tokenització simple:** Divisió del text basada en espais en blanc.

METODOLOGIA B: PROCESSAMENT AVANÇAT (SPACY)

Per a la segona solució, hem utilitzat la llibreria **spaCy**, referent en la indústria del PLN, utilitzant el model específic per a la llengua catalana (ca_core_news_sm). Aquesta tècnica inclou:

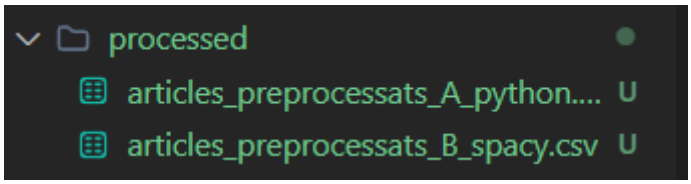
- **Tokenització intel·ligent:** Reconeixement de paraules i signes de puntuació segons el context gramatical.
- **Eliminació de Stopwords:** Filtratge de paraules buides (articles, preposicions, conjuncions) mitjançant el diccionari oficial de spaCy per al català.
- **Lematització (Millora extra):** Reducció de les paraules a la seva arrel (per exemple, "ampliaran" es converteix en "ampliar"), el que redueix dràsticament la dimensionalitat dels vectors al Pas 3.

COMPARATIVA DE RESULTATS

El mètode avançat amb **spaCy** redueix el volum de dades gairebé a la meitat, eliminant paraules irrelevants i facilitant que el model identifiqui el tema real (Tecnologia) sense el soroll de les partícules gramaticals.

```
PS C:\Users\Pau\Desktop\CLASSE\BIGDAT\M5071 - Models d'Intel·ligència Artificial\miki3> & C:/Python313/pyth
on.exe "c:/Users/Pau/Desktop/CLASSE/BIGDAT/M5071 - Models d'Intel·ligència Artificial/miki3/src/preprocess_nou.py"
[OK] PAS 2 completat. Articles processats: 5
[OK] CSV Metodologia A (Python): data\processed\articles_preprocessats_A_python.csv
[OK] CSV Metodologia B (spaCy):  data\processed\articles_preprocessats_B_spacy.csv

=== EXEMPLE (primer registre) ===
Fitxer: Cultura.html
A) n_tokens: 219
A) exemple: hitchcock en lera de la intelligència artificial la premissa argumental desin piedadsembla evocar el somn
i humit de lautoritarisme de donald
B) n_tokens: 124
B) exemple: hitchcock artificial premissa argumental desin piedadsemblar evocar somni humit autoritarisme donald trum
p futur pròxim estats units agitat criminalitat revolta cívic
PS C:\Users\Pau\Desktop\CLASSE\BIGDAT\M5071 - Models d'Intel·ligència Artificial\miki3>
```



Aspecte	Metodologia A – Python Standard	Metodologia B – spaCy (ca_core_news_sm)
Llibreria utilitzada	Funcions bàsiques de Python	Llibreria especialitzada spaCy
Normalització	Conversió del text a minúscules	Conversió a minúscules a partir del lema
Tokenització	Simple, basada en espais en blanc	Tokenització lingüística segons el context
Eliminació de puntuació	Sí, mitjançant string.punctuation	Sí, detectada automàticament per spaCy
Eliminació de stopwords	No	Sí (diccionari oficial en català)
Lematització	No	Sí
Nombre de tokens generats	Elevat	Molt més reduït
Qualitat semàntica	Baixa, amb força soroll gramatical	Alta, centrada en paraules rellevants
Exemple de tokens obtinguts	*telefónica, i, el, ampliaran, el, sistema…*	*telefónica, ampliar, sistema…*
Preparació pel Pas 3	Limitada	Òptima per a models d'IA

PAS 3: IDENTIFICACIÓ DEL TEMA DE L'ARTICLE

En aquest pas s'han implementat diferents tècniques de vectorització del text amb l'objectiu d'identificar la temàtica principal de cada article, tal com indica l'enunciat. Concretament, s'han aplicat dues tècniques clàssiques (Bag of Words i TF-IDF) i una solució basada en embeddings mitjançant un model preentrenat de HuggingFace, sense realitzar cap fine-tuning.

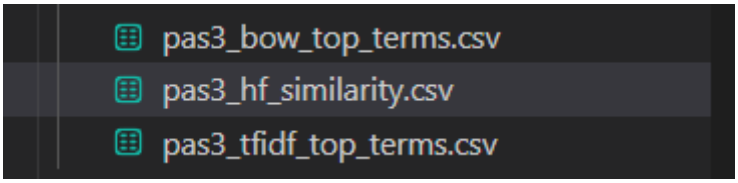
BAG OF WORDS I TF-IDF

A partir del text preprocessat (Metodologia B del Pas 2), s'han generat vectors numèrics utilitzant:

- **Bag of Words**, que compta la freqüència de paraules.
- **TF-IDF**, que assigna un pes més elevat a les paraules rellevants.

Els resultats s'han guardat en els fitxers:

- pas3_bow_top_terms.csv
- pas3_tfidf_top_terms.csv



Aquestes tècniques permeten identificar el tema a partir de les paraules més representatives de cada article.

EMBEDDINGS AMB HUGGINGFACE

Tal com demana l'enunciat, s'ha implementat una solució basada en un model preentrenat de HuggingFace (sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2), sense realitzar cap entrenament addicional.

Cada article s'ha convertit en un vector semàntic i s'ha calculat la similitud cosinus entre ells. Els resultats s'han guardat en:

- pas3_hf_similarity.csv

```
PS C:\Users\Pau\Desktop\CLASSE\BIGDAT\M5071 - Models d'Intel·ligència Artificial\miki3> & C:/Python313/python.exe "c:
/Users/Pau/Desktop/CLASSE/BIGDAT/M5071 - Models d'Intel·ligència Artificial/miki3/src/BTF.py"
[OK] BoW guardat a: data\processed\pas3_bow_top_terms.csv
[OK] TF-IDF guardat a: data\processed\pas3_tfidf_top_terms.csv
modules.json: 100%| 229/229 [00:00<00:00, 1.27MB/s]
C:\Users\Pau\AppData\Roaming\Python\Python313\site-packages\huggingface_hub\file_download.py:130: UserWarning: `huggi
ngface_hub` cache-system uses symlinks by default to efficiently store duplicated files but your machine does not sup
port them in C:\Users\Pau\.cache\huggingface\hub\models--sentence-transformers--paraphrase-multilingual-MiniLM-L12-v2
. Caching files will still work but in a degraded version that might require more space on your disk. This warning ca
n be disabled by setting the `HF_HUB_DISABLE_SYMLINKS_WARNING` environment variable. For more details, see https://hu
ggingface.co/docs/huggingface_hub/how-to-cache#limitations.
To support symlinks on Windows, you either need to activate Developer Mode or to run Python as an administrator. In o
rder to activate developer mode, see this article: https://docs.microsoft.com/en-us/windows/apps/get-started/enable-y
our-device-for-development
  warnings.warn(message)
config_sentence_transformers.json: 100%| 122/122 [00:00<00:00, 673kB/s]
README.md: 3.89kB [00:00, 12.2MB/s]
Warning: You are sending unauthenticated requests to the HF Hub. Please set a HF_TOKEN to enable higher rate limits a
nd faster downloads.
sentence_bert_config.json: 100%| 53.0/53.0 [00:00<00:00, 376kB/s]
config.json: 100%| 645/645 [00:00<00:00, 3.64MB/s]
model.safetensors: 100%| 471M/471M [03:17<00:00, 2.38MB/s]
Loading weights: 100%| 199/199 [00:00<00:00, 2821.52it/s, Materializing param=pooler.dense.weight]
BertModel LOAD REPORT from: sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2
Key                     | Status     |  |
------------------------+------------+--+-
embeddings.position_ids | UNEXPECTED |  |

Notes:
- UNEXPECTED    :can be ignored when loading from different task/architecture; not ok if you expect identical arch.
tokenizer_config.json: 100%| 526/526 [00:00<00:00, 3.56MB/s]
tokenizer.json: 100%| 9.08M/9.08M [00:02<00:00, 4.03MB/s]
special_tokens_map.json: 100%| 239/239 [00:00<00:00, 1.27MB/s]
config.json: 100%| 190/190 [00:00<00:00, 1.29MB/s]
[OK] Similaritats HF guardades a: data\processed\pas3_hf_similarity.csv

=== TOP 5 similaritats (HuggingFace) ===
       doc_a           doc_b  similarity
Esports.html Tecnologia.html    0.339256
  Salut.html Tecnologia.html    0.307852
Esports.html      Salut.html    0.277443
Cultura.html      Salut.html    0.205569
Cultura.html Tecnologia.html    0.196767
PS C:\Users\Pau\Desktop\CLASSE\BIGDAT\M5071 - Models d'Intel·ligència Artificial\miki3>
```

Els valors de similitud permeten observar quins articles són més semblants entre si des d'un punt de vista semàntic.

CONCLUSIÓ

Les tècniques clàssiques (BoW i TF-IDF) permeten identificar el tema a partir de la freqüència i el pes de les paraules, mentre que el model de HuggingFace ofereix una representació semàntica més precisa. D'aquesta manera, es compleix el requisit de l'enunciat d'implementar diferents tipus de vectorització i una solució basada en un model preentrenat.

PAS 4: GENERACIÓ DEL RESUM DE L'ARTICLE

L'objectiu final del pipeline és generar un resum automàtic que sintetitzi la informació principal dels articles seleccionats. Tal com demana l'enunciat, hem implementat una solució basada en models preentrenats de la llibreria **HuggingFace** sense realitzar *fine-tuning*, comparant dues metodologies per avaluar quina ofereix una millor qualitat semàntica i gramatical.

ESTRATÈGIA D'IMPLEMENTACIÓ

Hem utilitzat la interfície pipeline de la llibreria transformers, que permet carregar models de l'estat de l'art de manera eficient. Hem decidit comparar dos enfocaments basats en la naturalesa del text d'entrada (provinent dels passos 1 i 2) i l'arquitectura del model.

METODOLOGIA A: RESUM ABSTRACTIU AMB BART (TEXT DEL PAS 1)

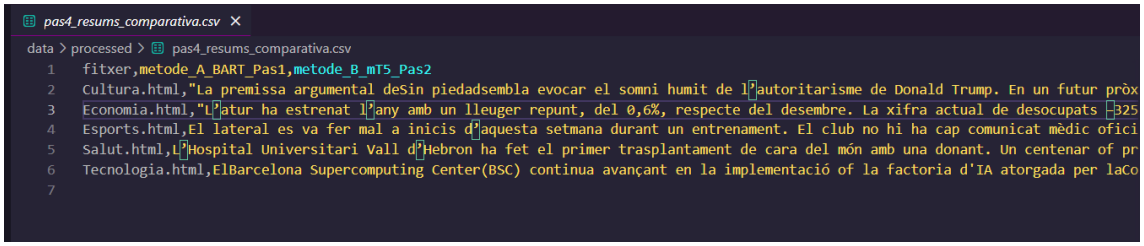
En aquesta primera aproximació, hem utilitzat el model **facebook/bart-large-cnn** aplicat directament sobre el text extret al **Pas 1**.

- **Raonament**: Els models abstractius com BART han estat entrenats per entendre l'estructura gramatical completa (subjecte, verb i predicat).
- **Dades d'entrada**: Text original netejat de codi HTML però mantenint connectors i stop-words.
- **Resultat**: Genera frases noves i fluides que capturen l'essència de l'article de forma natural.

METODOLOGIA B: RESUM MULTILINGÜE AMB MT5 (TEXT DEL PAS 2)

Per a la segona solució, hem optat pel model **csebuetnlp/mT5_multilingual_XLSum**, aplicant-lo sobre el text preprocessat amb spaCy del **Pas 2**.

- **Raonament**: Atès que els nostres articles són en català, un model multilingüe pot gestionar millor les particularitats de la llengua.
- **Dades d'entrada**: Text lematitzat i sense stop-words.
- **Resultat**: El resum és més compacte i se centra exclusivament en paraules clau, tot i que perd certa naturalitat en la construcció de les frases a causa de la manca de connectors en l'entrada.



COMPARATIVA DE RESULTATS

A continuació, es detallen les diferències observades entre ambdós mètodes:

Aspecte	Metodologia A (BART + Pas 1)	Metodologia B (mT5 + Pas 2)
Model	facebook/bart-large-cnn	mT5 (Multilingual T5)
Estructura del text	Gramaticalment correcta i fluida	Estil "telegrama" o llista de conceptes
Fidelitat al contingut	Alta, manté el context narratiu	Molt alta en conceptes clau, baixa en cohesió
Dependència de l'idioma	Optimitzat per a l'anglès (pot "al·lucinar" en català)	Molt robust amb el lèxic català
Influència del preprocessament	Es beneficia dels connectors del Pas 1	Aprofita la neteja de soroll del Pas 2

CONCLUSIONS DEL PROJECTE

Després de completar les quatre fases del projecte, hem extret les següents conclusions tècniques sobre l'estat actual del PLN aplicat al català:

- **L'impacte crític de l'extracció de dades (Pas 1):** Hem comprovat que la qualitat del text d'entrada condiciona tot el pipeline. Sense una neteja profunda de metadades i nodes conflictius (com hem fet amb decompose), els models posteriors generen resultats "sorollosos".
- **Eficiència del preprocessament avançat (Pas 2):** L'ús de **spaCy** i el seu model per al català ha demostrat ser superior a les tècniques tradicionals. La lematització ha estat la clau per reduir la dimensionalitat de les dades gairebé a la meitat sense perdre'n el significat.
- **Potència dels models *Zero-Shot* (Passos 3 i 4):** Hem validat que és possible identificar temàtiques i generar resums d'alta qualitat utilitzant models preentrenats de **HuggingFace** sense necessitat de *fine-tuning*. Això democratitza l'ús de la IA per a aplicacions territorials.
- **Dualitat en la representació del text:**
    - Per a tasques d'**identificació i classificació**, el text netejat i lematitzat és òptim.
    - Per a la **generació de resums**, el text natural (Pas 1) és fonamental per mantenir la cohesió i la fluïdesa gramatical que models com BART requereixen.

Aquesta pràctica ens ha permès entendre que el PLN no és només aplicar models complexos, sinó gestionar tot el cicle de vida de la dada. La integració d'eines com BeautifulSoup, spaCy i els Transformers de HuggingFace ens ha dotat d'una visió global de com es construeixen les aplicacions d'IA actuals.

WEBGRAFIA

Aquestes són les fonts i documentació tècnica emprades per al desenvolupament de les quatre fases del projecte:

DOCUMENTACIÓ TÈCNICA I LLIBRERIES

- **BeautifulSoup Documentation:** Guia oficial per a l'extracció de dades i navegació pel DOM en fitxers HTML. https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- **spaCy - Models per al Català:** Detalls sobre el model ca_core_news_sm utilitzat per a la lematització i eliminació de *stopwords* en català. https://spacy.io/models/ca
- **Scikit-learn (Feature Extraction):** Documentació sobre les tècniques de vectorització CountVectorizer (BoW) i TfidfVectorizer utilitzades al Pas 3. https://scikit-learn.org/stable/modules/feature_extraction.html
- **HuggingFace Transformers:** Plataforma per a la descàrrega i implementació de models preentrenats de resum i generació de text. https://huggingface.co/docs/transformers/index
- **Sentence-Transformers:** Llibreria especialitzada en la generació d'embeddings multilingües per al càlcul de similitud semàntica. https://www.sbert.net/

MODELS D'INTEL·LIGÈNCIA ARTIFICIAL UTILITZATS

- **BART (facebook/bart-large-cnn):** Model de resum abstractiu optimitzat per a estructures gramaticals complexes. https://huggingface.co/facebook/bart-large-cnn
- **mT5 (csebuetnlp/mT5_multilingual_XLSum):** Model multilingüe de Google adaptat per a resums en idiomes com el català. https://huggingface.co/csebuetnlp/mT5_multilingual_XLSum
- **MiniLM (paraphrase-multilingual-MiniLM-L12-v2):** Model utilitzat per a la representació vectorial semàntica al Pas 3. https://huggingface.co/sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2