UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Reconocimiento de entidades nombradas y categorización de textos periodísticos

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Bernabéu Pérez, Pablo

Tutor/a: Segarra Soriano, Encarnación

Cotutor/a: Hurtado Oliver, Lluis Felip

CURSO ACADÉMICO: 2021/2022

Reconocimiento de entidades nombradas y categorización de textos periodísticos

Pablo Bernabeu Pérez

3 de julio de 2022

Resum

L'automatització de la tasca de classificació de textos en un conjunt de categories predeterminat i de la tasca de detecció d'entitats anomenades són temes de gran interés en diversos camps d'aplicació de l'àrea del Processament del Llenguatge Natural. Un d'eixos camps d'aplicació és el treball dels documentalistes sobre textos periodístics en un mitjà de comunicació audiovisual.

La proposta d'aquest treball sorgeix de la col·laboració amb la *Corporació Valenciana de Mitjans de Comunicació* i té un doble objectiu. En primer lloc, el treball consistirà en la construcció i comparació de diversos sistemes basats en xarxes neuronals de classificació de textos periodístics d'un mitjà de comunicació en català. En segon lloc, el treball durà a terme l'entrenament de diversos models de reconeixement d'entitats anomenades que seran comparats entre ells i amb altres sistemes. Després es realitzarà un estudi del comportament d'aquests models sobre textos periodístics del mateix mitjà de comunicació, pel fet que s'han entrenat sobre un domini distint.

Paraules clau: aprenentatge profund, classificació automàtica, detecció i reconeiximent de entitats nomenades, processament del llenguatge natural, textos periodístics en català

Resumen

La automatización de la tarea de clasificación de textos en un conjunto de categorías predeterminado y de la tarea de detección de entidades nombradas son temas de gran interés en diversos campos de aplicación del área del Procesamiento del Lenguaje Natural. Uno de esos campos de aplicación es el trabajo de los documentalistas sobre textos periodísticos en un medio de comunicación audiovisual.

La propuesta de este trabajo surge de la colaboración con la *Corporació Valenciana de Mitjans de Comunicació* y tiene un doble objetivo. En primer lugar, el trabajo consistirá en la construcción y comparación de diversos sistemas basados en redes neuronales de clasificación de textos periodísticos de un medio de comunicación en catalán. En segundo lugar, el trabajo llevará a cabo el entrenamiento de varios modelos de reconocimiento de entidades nombradas que serán comparados entre ellos y con otros sistemas. Después se realizará un estudio del comportamiento de estos modelos sobre textos periodísticos del mismo medio de comunicación, debido a que se han entrenado sobre un dominio distinto.

Palabras clave: aprendizaje profundo, clasificación automática, detección y reconocimiento de entidades nombradas, procesamiento del lenguaje natural, textos periodísticos en catalán

Abstract

The automation of the task of classifying texts into a predetermined set of categories and of the task of named entity recognition are topics of great interest in various fields of application in the area of Natural Language Processing. One of these fields of application is the work of documentalists on journalistic texts in an audiovisual media.

The proposal of this project arises from the collaboration with the *Corporació Valenciana de Mitjans de Comunicació* and has a double objective. Firstly, the work will consist of the construction and comparison of different systems based on neural networks for the classification of journalistic texts of a media in catalan. Secondly, the work will carry out the

training of several named entity recognition models that will be compared among them and with other systems. Then, a study of the behavior of these models on journalistic texts of the same media will be carried out, due to the fact that they have been trained on a different domain.

Key words: automatic classification, deep learning, detection and recognition of named entities, journalistic texts in catalan, natural language processing

Índice general

Índice general v
Índice de figuras vii
Índice de tablas viii

1 Introducción 1
1.1 Motivación . . . 1
1.2 Objetivos . . . 2
1.3 Asignaturas relacionadas . . . 2
1.4 Estructura de la memoria . . . 3
2 Procesamiento del lenguaje natural 4
2.1 Embeddings . . . 4
2.2 Mecanismos de atención . . . 6
2.2.1 Atención en Redes Neuronales Recurrentes . . . 7
2.3 *Transformers* . . . 8
2.3.1 Arquitectura de los *transformers* . . . 8
2.3.2 BERT . . . 10
2.3.3 BERTa . . . 11
2.4 *Transfer Learning* . . . 11
2.5 Tareas . . . 12
2.5.1 Clasificación de textos . . . 12
2.5.2 Reconocimiento de entidades nombradas . . . 12
2.6 Métricas y evaluación . . . 13
2.6.1 Aciertos y errores . . . 13
2.6.2 Métricas . . . 14
2.6.3 Matriz de confusión . . . 15
3 Clasificación de textos 17
3.1 Herramientas utilizadas . . . 17
3.1.1 Entorno de trabajo . . . 17
3.1.2 Librerías utilizadas . . . 17
3.1.3 Librerías descartadas . . . 18
3.2 Corpus . . . 18
3.3 Experimentación y análisis de resultados . . . 21
3.3.1 Modelos de 4 clases . . . 22
3.3.2 Modelos de 29 clases . . . 27
3.3.3 Comparativa de los modelos . . . 33
4 Reconocimiento de entidades nombradas 35
4.1 Herramientas utilizadas . . . 35
4.1.1 Entorno de trabajo . . . 35
4.1.2 Librerías utilizadas . . . 35
4.1.3 Librerías descartadas . . . 36
4.2 Corpus . . . 36
4.2.1 AnCora . . . 36

4.2.2 Wikiann 36
4.3 Experimentación y análisis de resultados 37
4.3.1 Entrenamiento de los modelos 37
4.3.2 Comparativa de los modelos 38
4.3.3 Test con noticias del corpus CVMC 40
5 Conclusiones y trabajo futuro **42**
5.1 Conclusiones 42
5.2 Trabajo futuro 43
Bibliografía **45**
A Objetivos ODS **48**
B Tablas y figuras adicionales **50**

Índice de figuras

2.1 Etiquetado numérico de una frase . . . 5
2.2 *One-Hot Encoding* de una frase . . . 5
2.3 Embeddings en un espacio vectorial de 3 dimensiones . . . 6
2.4 Relaciones entre embeddings en tres dimensiones . . . 6
2.5 Atención entre dos secuencias de texto en una tarea de traducción . . . 7
2.6 Una red neuronal recurrente desenrollada . . . 7
2.7 Dependencias a largo plazo en una RNN . . . 8
2.8 Componentes de un transformer . . . 9
2.9 Transformer decodificando en una tarea de traducción en un transformer 10
2.10 Embedding de una secuencia en el modelo BERT . . . 11
2.11 Texto con entidades nombradas anotadas . . . 13
2.12 Matriz de confusión sin normalizar . . . 16
2.13 Matriz de confusión normalizada . . . 16

3.1 Matriz de confusión del modelo inicial de 4 clases para el conjunto de test de À Punt 2020 . . . 23
3.2 Matriz de confusión del modelo de À Punt 2018 para el conjunto de test de À Punt 2020 . . . 24
3.3 Matriz de confusión del modelo de À Punt 2019 para el conjunto de test de À Punt 2020 . . . 26
3.4 Matriz de confusión del modelo inicial de 29 clases para el conjunto de test de Canal Nou . . . 28
3.5 Matriz de confusión del modelo inicial de 29 clases para la partición de 2020 de À Punt . . . 29
3.6 Matriz de confusión del modelo de 2018 de 29 clases para la partición de 2020 de À Punt . . . 30
3.7 Matriz de confusión del modelo de 2019 de 29 clases para la partición de 2020 de À Punt . . . 31

4.1 Estructura de una pipeline de SpaCy . . . 38
4.2 Noticia anotada manualmente . . . 40
4.3 Entidades detectadas en una noticia por los modelos entrenados con los corpus de AnCora y Wikiann . . . 40

B.1 Matriz de confusión del modelo inicial de 4 clases para el conjunto de test de Canal Nou . . . 50
B.2 Matriz de confusión del modelo inicial de 4 clases para el conjunto de test de À Punt 2018 . . . 50
B.3 Matriz de confusión del modelo inicial de 4 clases para el conjunto de test de À Punt 2019 . . . 51
B.4 Matriz de confusión del modelo de À Punt 2018 de 4 clases para el conjunto de test de Canal Nou . . . 51
B.5 Matriz de confusión del modelo de À Punt 2018 de 4 clases para el conjunto de test de À Punt 2019 . . . 52

B.6 Matriz de confusión del modelo de À Punt 2019 de 4 clases para el conjunto de test de Canal Nou . . . 52
B.7 Matriz de confusión del modelo de inicial de 29 clases para el conjunto de test de À Punt 2018 . . . 53
B.8 Matriz de confusión del modelo de inicial de 29 clases para el conjunto de test de À Punt 2019 . . . 53
B.9 Matriz de confusión del modelo de À Punt 2018 de 29 clases para el conjunto de test de Canal Nou . . . 54
B.10 Matriz de confusión del modelo de À Punt 2018 de 29 clases para el conjunto de test de À Punt 2019 . . . 54
B.11 Matriz de confusión del modelo de À Punt 2019 de 29 clases para el conjunto de test de Canal Nou . . . 55

Índice de tablas

3.1 Noticias de Canal Nou y À Punt y porcentaje sobre el total del corpus . . . 19
3.2 Noticias únicas y duplicadas en el corpus original . . . 19
3.3 Noticias validas y descartadas según su longitud . . . 19
3.4 Noticias con una clase y multiclase . . . 19
3.5 Noticias según su cantidad de etiquetas asignadas . . . 19
3.6 Noticias por clase y porcentaje del total . . . 21
3.7 Precision, recall y F1 macro con el modelo inicial de 4 clases . . . 22
3.8 Macro-F1 para cada clase con el modelo inicial de 4 clases . . . 23
3.9 Precision, recall y F1 con el modelo de 2018 de 4 clases . . . 24
3.10 Macro-F1 para cada clase con el modelo de 2018 de 4 clases . . . 24
3.11 Precision, recall y F1 con el modelo de 2019 de 4 clases . . . 25
3.12 Macro-F1 para cada clase con el modelo de 2019 de 4 clases . . . 25
3.13 Comparativa de los modelos sobre la partición de test de Canal Nou . . . 26
3.14 Comparativa de F1-Macro para cada clase de los modelos sobre la partición de test de Canal Nou . . . 26
3.15 Comparativa de los modelos sobre la partición de 2020 de À Punt . . . 27
3.16 Comparativa de Macro-F1 para cada clase de los modelos sobre la partición de 2020 de À Punt . . . 27
3.17 Precision, recall y F1 con el modelo inicial de 29 clases . . . 28
3.18 Precision, recall y F1 con el modelo de 2018 de 29 clases . . . 29
3.19 Precision, recall y F1 con el modelo de 2019 de 29 clases . . . 30
3.20 Comparativa de los modelos de 29 clases con la partición de test de Canal Nou . . . 31
3.21 Comparativa de los modelos de 29 clases con la partición de 2020 de À Punt 32
3.22 Comparativa clase por clase de los modelos de 29 clases con la partición de 2020 de À Punt . . . 33
3.23 Comparativa de los modelos de 4 y 29 clases con la partición de 2020 de À Punt . . . 33
3.24 Comparativa para cada clase de los modelos de 4 y 29 clases sobre la partición de 2020 de À Punt . . . 34

4.1 Evaluación del modelo de AnCora con la partición de test de AnCora . . . 37

4.2 Evaluación del modelo de Wikiann con la partición de test de Wikiann . . 38
4.3 Comparativa de los modelos con la partición de test de AnCora 39
4.4 Comparativa de la F1 de los modelos para cada clase con la partición de test de AnCora 39
4.5 Comparativa de los modelos con la partición de test de Wikiann 39
4.6 Comparativa de la F1 de los modelos para cada clase con la partición de test de Wikiann 39

CAPÍTULO 1
Introducción

En este primer capítulo de la memoria se introduce la temática del proyecto, que consiste en la construcción de diferentes modelos de clasificación de textos en catalán, con el objetivo de realizar el etiquetado del sistema interno de catalogación de la *Corporació Valenciana de Mitjans de Comunicació* (CVMC) de forma automática o como ayuda para este proceso. Por otro lado se construyen varios modelos para detectar entidades nombradas mediante otro conjunto de datos y se prueba su funcionamiento con noticias pertenecientes al corpus facilitado por la CVMC.

Posteriormente, se expone la motivación para desarrollar el proyecto así como los objetivos y el impacto que se espera lograr. Además, se nombran las asignaturas cursadas durante la carrera que han aportado conocimientos necesarios para la realización del proyecto y que han resultado útiles para el desarrollo de la experimentación. Finalmente, se detalla la estructura del proyecto y el alcance de cada capítulo de la memoria.

1.1 Motivación

La *Corporació Valenciana de Mitjans de Comunicació*, siguiendo la tendencia de la digitalización, extendida a lo largo de la mayoría de sectores empresariales, ha solicitado que se realice un proyecto para estudiar si es posible la automatización del proceso de clasificación de los noticiarios en su sistema interno de catalogación y documentación.

El procesamiento de lenguaje natural [1], en inglés *natural language processing* (NLP), es una rama de la computación y la inteligencia artificial que se ocupa de la investigación y el desarrollo de mecanismos computacionalmente eficaces para la comunicación entre personas y máquinas a través del lenguaje natural. Este campo incluye tareas como la traducción automática [2], el análisis de sentimiento [3] y el procesamiento del habla [4].

El proyecto a realizar se engloba dentro de esta ámbito, ya que su finalidad es el procesamiento de textos periodísticos en catalán para la realización de un sistema que automatice el proceso de clasificación o de otro modo, que sirva como apoyo a los documentalistas, y al mismo tiempo, pueda detectar personas, lugares, organizaciones u otras entidades de interés nombradas dentro de las noticias.

A lo largo del proyecto, se estudiará y procesará un corpus estructurado facilitado por la *Corporació Valenciana de Mitjans de Comunicació*, con el que se entrenarán distintos modelos para la clasificación de noticias. Para ello, se utilizará una estrategia cronológica con el objetivo de prever el funcionamiento de los modelos con noticias de años próximos. Además, se entrenarán otros modelos para el reconocimiento de entidades nombradas y

se probarán sobre noticias del corpus de la CVMC. Finalmente, se extraerán conclusiones del trabajo realizado y se aportarán nuevas ideas y enfoques para la continuación del mismo.

1.2 Objetivos

Este trabajo de fin de grado tiene un doble objetivo. Por un lado, estudiar y depurar un conjunto de noticias en catalán con la intención de elaborar distintos modelos de clasificación, para poder compararlos entre si y hacer una previsión de su desempeño en la clasificación de noticias futuras. El propósito de estos modelos es automatizar el proceso de clasificación o, en su defecto, ser una herramienta de apoyo para el personal destinado a la tarea de documentación y catalogación de estas noticias.

Por otra parte, entrenar distintos sistemas de reconocimiento de entidades nombradas y compararlos entre ellos y con otras herramientas disponibles para afrontar la tarea. Con este sistema, se busca detectar entidades de interés que sean tratadas en los textos periodísticos, en lo que se incluyen personas, organizaciones y localizaciones. La dificultad de este objetivo recae en que el corpus facilitado por la *Corporació Valenciana de Mitjans de Comunicació* no esta etiquetado para la tarea, por lo que será necesario entrenar los modelos en otro dominio y después, evaluar su funcionamiento en noticias del conjunto de datos original.

1.3 Asignaturas relacionadas

A lo largo del proyecto, se han utilizado los conocimientos teóricos adquiridos durante el grado de Ingeniería Informática gracias al temario presentado en las materias y impartido por el profesorado. Si bien la mayoría de asignaturas han sido de utilidad como base teórica o de forma directa, son destacables las asignaturas pertenecientes a la mención en Computación, especialmente aquellas relacionadas con la inteligencia artificial y el aprendizaje automático.

Las asignaturas “Sistemas inteligentes” y “Percepción” impartidas en el tercer año de carrera y “Aprendizaje Automático” perteneciente al plan académico del cuarto curso, han aportado el conocimiento teórico necesario para la comprensión del campo de la inteligencia artificial, así como experiencia práctica útil para el desarrollo de los modelos de aprendizaje automático.

Otra asignatura clave para el desarrollo del proyecto es “Sistemas de almacenamiento y recuperación de la información”. Esta materia, impartida durante el tercer curso, es una introducción al tratamiento de textos y al campo del procesamiento del lenguaje natural. En ella se desarrollan conceptos teóricos que aportan una comprensión intuitiva de las tareas desarrolladas en este proyecto. Además, en las prácticas de la asignatura, se tiene el primer contacto dentro de la carrera con el lenguaje de programación *Python*, que se ha utilizado en la experimentación del proyecto debido a la gran cantidad de librerías y herramientas disponibles para tareas de aprendizaje automático y, en concreto, procesamiento del lenguaje natural.

Finalmente, es importante mencionar las asignaturas “Agentes Inteligentes” y “Técnicas, Aplicaciones y Entornos de Inteligencia Artificial” que si bien no están directamente relacionadas con el proyecto, aportan una visión general del campo de la inteligencia artificial conveniente a la hora de adentrarse en ámbitos más especializados.

1.4 Estructura de la memoria

La memoria del proyecto se compone de un total de cinco capítulos, incluyendo este. En esta sección se realiza una introducción a cada una de estas secciones y los temas tratados en ellas, con el objetivo de tener una visión general del proyecto.

- **Capítulo 1, Introducción**: en el primer capítulo de la memoria, se presenta la temática general del proyecto y las tareas a desarrollar en la experimentación. Además, se manifiesta la motivación detrás de la realización del proyecto así como los objetivos del mismo. Por otro lado, se destacan las asignaturas cursadas a lo largo del grado que han aportado los conocimientos teóricos y prácticos necesarios para llevar a cabo el proyecto. Por último se describe la estructura de la memoria.
- **Capítulo 2, Procesamiento del Lenguaje Natural**: en el segundo capítulo de la memoria, se introduce el campo del procesamiento del lenguaje natural. También se explican los sistemas y las técnicas necesarios para tener una comprensión de la estructura y funcionamiento de los modelos usados en la experimentación del proyecto. Además, se definen las tareas que se van a afrontar en la experimentación, así como las métricas y herramientas para evaluar el desempeño de los modelos en estas tareas.
- **Capítulo 3, Clasificación de textos**: en el tercer capítulo de la memoria, se realiza la experimentación de la tarea de clasificación de textos. En primer lugar, se describe el entorno de trabajo y las herramientas a utilizar durante el desarrollo de la tarea. Después, se hace un estudio y procesado del corpus facilitado por la *Corporació Valenciana de Mitjans de Comunicació* con el que se entrenan distintos modelos de aprendizaje automático para clasificación de textos. Finalmente, se realiza una comparación entre los modelos.
- **Capítulo 4, Reconocimiento de entidades nombradas**: en el cuarto capítulo de la memoria, se realiza la experimentación de la tarea de reconocimiento de entidades nombradas. Primero, se expone el entorno de trabajo, así como las herramientas que se usan durante la experimentación. Tras esto, se describen los conjuntos de datos utilizados para el entrenamiento de los modelos. Luego, se comparan los modelos entrenados con otros sistemas disponibles en herramientas de procesamiento del lenguaje natural. Por último, se realiza una prueba con noticias del corpus de la *Corporació Valenciana de Mitjans de Comunicació*.
- **Capítulo 5, Conclusiones y trabajo futuro**: en el quinto capítulo de la memoria, se sintetiza el trabajo realizado a lo largo del proyecto y se extraen conclusiones de las tareas desarrolladas. Además, se proporcionan ideas para continuar con el trabajo realizado y posibles nuevos enfoques.

CAPÍTULO 2

Procesamiento del lenguaje natural

El procesamiento del lenguaje natural o NLP por sus siglas en inglés, es un campo de la lingüística y la inteligencia artificial que se ocupa de las interacciones entre los ordenadores y el lenguaje humano. El objetivo es conseguir un sistema capaz de entender el lenguaje natural, incluidos los matices contextuales. Dentro de este campo se encuentran tareas como el reconocimiento de texto y habla, el análisis morfológico, sintáctico y semántico o la generación de texto.

A lo largo de los años, el campo ha experimentado varios cambios de enfoque. Entre los años 50 y 90, se estudió el NLP simbólico, que basaba la comprensión del lenguaje natural en un conjunto de reglas. Después, en la década de 1990 y principios de los 2000, se introdujo el aprendizaje automático y el estudio del lenguaje desde un punto de vista estadístico. Desde hace más de una década, los enfoques con mejores resultados están basados en redes neuronales. Recientemente el campo del procesamiento del lenguaje natural ha experimentado un gran avance gracias a los *embeddings*, la auto-atención y los modelos basados en *transformers*.

En este capítulo de la memoria, se introducen los conceptos teóricos previos y necesarios para entender los modelos utilizados, se definen las tareas a afrontar durante la experimentación y se definen las métricas y sistemas de evaluación usados para valorar el desempeño de los modelos.

2.1 Embeddings

Las redes neuronales son modelos computacionales muy potentes y con muchas aplicaciones, pero debido a su funcionamiento interno solo trabajan con números. En el campo del procesamiento del lenguaje natural se trabaja con textos, así que para poder utilizar las redes neuronales para procesar texto debemos usar un sistema que nos permita representar el texto en forma de vectores. La representación de palabras como vectores es conocida como *word embeddings* [5].

Un primer enfoque podría ser asignar un número a cada palabra distinta, de tal manera que cada palabra este representada por un identificador único. Para asignar las etiquetas a las palabras, podemos utilizar cualquier estrategia, como seguir el orden alfabético o el orden de aparición en un texto. Sin embargo, al utilizar esta técnica para representar mediante números, se establecen relaciones entre las palabras que no son ciertas. Según

las etiquetas de la figura 2.1, las palabras “perro” y “gato” están a la misma distancia que “amigos” y “el” o también que “y” es el doble de "gato".

Figura 2.1: Etiquetado numérico de una frase

Para evitar estos problemas, podemos utilizar la representación *One-Hot Encoding* [6]. Esta codificación usa para cada palabra un vector del tamaño del vocabulario, en el que cada posición representa una palabra del vocabulario, de tal manera que solo el valor en la posición que simboliza la palabra es 1 y todos los demás valores del vector son 0.

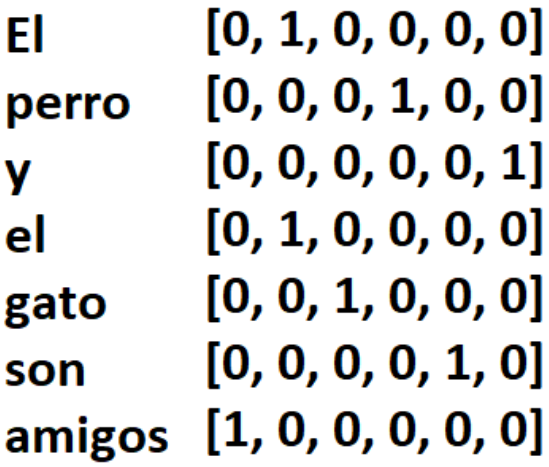

Figura 2.2: *One-Hot Encoding* de una frase

Mediante esta representación, todas las palabras están a la misma distancia entre sí, lo que evita que se creen relaciones numéricas entre palabras que no están relacionadas a nivel semántico, pero, al mismo tiempo, tampoco se representan las relaciones entre palabras que sí están relacionadas. Por otro lado, esta codificación tiene la desventaja de que se usa una gran cantidad de espacio para representar el texto. Para representar un texto que contiene N palabras con un vocabulario de tamaño M hace falta una matriz de NxM donde sólo N valores son 1. Pese a que la representación *One-Hot Encoding* tiene ciertos problemas, codifica las palabras en vectores, por lo que pueden ser procesados por redes neuronales.

Word2vec [7] es una técnica que utiliza una red neuronal para producir *embeddings*. Esta red es entrenada con grandes cantidades de texto con el objetivo de producir *embeddings* de dimensiones reducidas a partir de vectores *One-Hot Encoding*. Además el vector que se obtiene como salida encapsula la relación de esta palabra con las demás palabras del vocabulario, ya que dos palabras cercanas en el espacio vectorial tienen alguna relación semántica. La cercanía de dos palabras se puede cuantificar mediante la similitud coseno de los dos *embeddings* que las codifican.

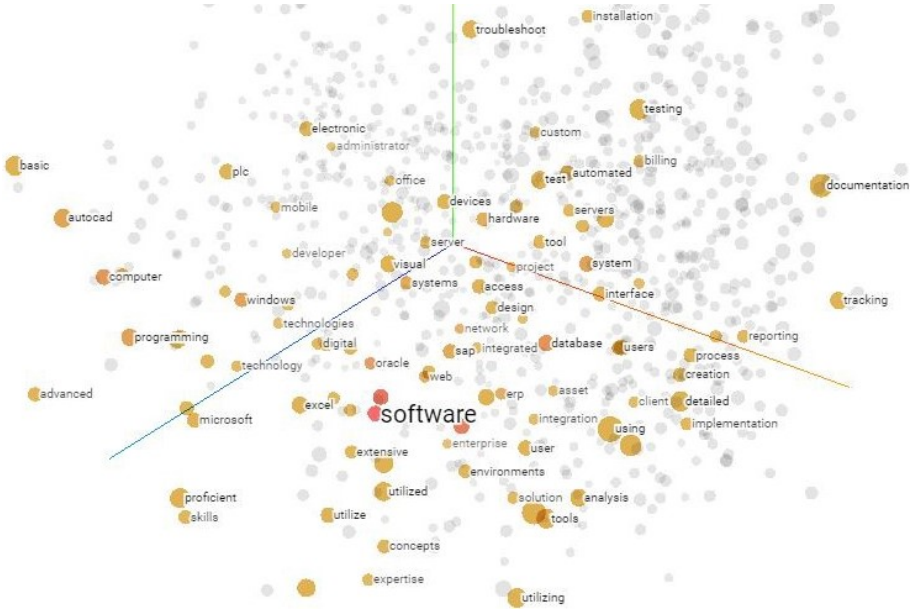

Figura 2.3: Embeddings en un espacio vectorial de 3 dimensiones

En el espacio vectorial, dos parejas de palabras con la misma relación entre si estarán a la misma distancia, por lo que, teóricamente, se pueden realizar operaciones como tomar el *embeddings* de la palabra "profesor" sumarle el de "mujer" y obtener como resultado el vector que codifica "profesora". En la figura 2.4 se puede ver la relación entre varias parejas de palabras como género, tiempo verbal y país-capital.

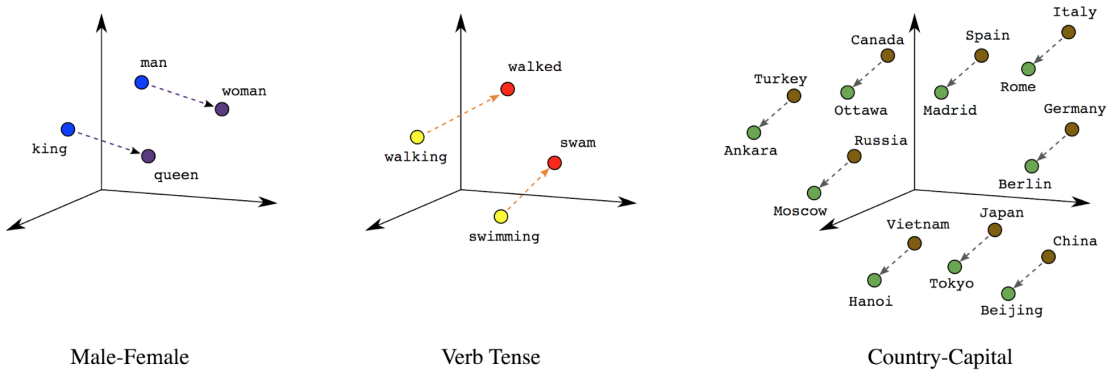

Figura 2.4: Relaciones entre embeddings en tres dimensiones

Desde la publicación de *word2vec* se han creado extensiones como *paragraph2vec* y *doc2vec* que transforman párrafos y documentos respectivamente en vectores. También se han desarrollado otros sistemas de *embeddings* con mejores resultados que *word2vec*, como GloVe [8] y fastText [9].

2.2 Mecanismos de atención

Los mecanismos de atención permiten a las redes neuronales tener una visión general de los datos y fijarse en las partes que consideran más relevantes [10].

La atención, en términos matemáticos, es un vector cuyos componentes serán usados como pesos de una suma ponderada en combinación con la salida de una red neuronal. Para obtener este vector, o mejor dicho, conjunto de vectores de atención (ya que cada uno representa la atención de cada dato por separado) necesitamos los vectores *key* y *query*. Dado un dato dentro de una secuencia, el vector clave define las propiedades de este dato y el vector de búsqueda lo que "atrae" su atención en otro dato. El vector de atención del dato se obtiene realizando el producto escalar entre su vector de búsqueda y la matriz de vectores clave. Repitiendo la operación con todos los vectores de búsqueda obtenemos una matriz de atención.

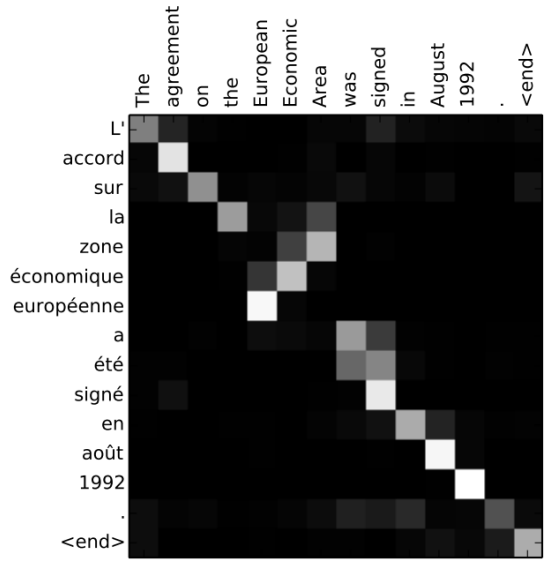

Figura 2.5: Atención entre dos secuencias de texto en una tarea de traducción

2.2.1. Atención en Redes Neuronales Recurrentes

Los mecanismos de atención se introdujeron en la década de los noventa y se han utilizado desde entonces en combinación con otros sistemas, como las Redes Neuronales Recurrentes (RNN por sus siglas en inglés). Las RNN son redes que utilizan la salida del paso previo como parte de la entrada del siguiente. Este mecanismo dota a la red de "memoria", dándole la capacidad de relacionar los datos previos con el que esta siendo procesado actualmente.

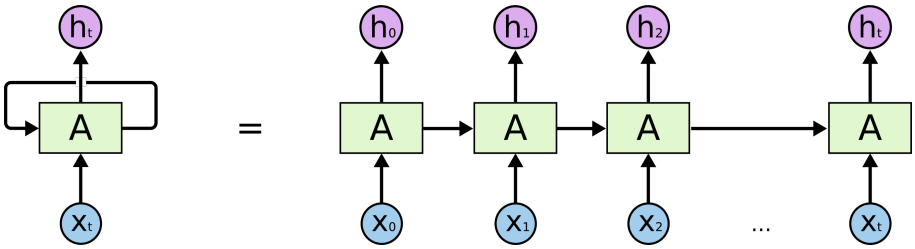

Figura 2.6: Una red neuronal recurrente desenrollada

Sin embargo, al comprimir la información como un único vector de salida, se pierde información, especialmente aquella procesada en pasos más antiguos. Esto dificulta encontrar relaciones entre datos separados, causando un efecto de "perdida de memoria" a largo plazo.

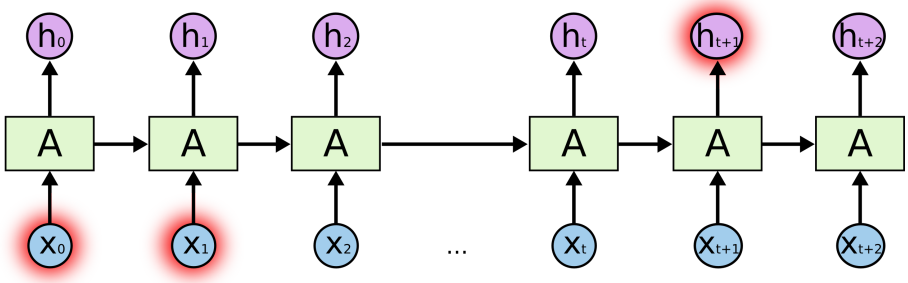

Figura 2.7: Dependencias a largo plazo en una RNN

La atención en combinación con las redes neuronales recurrentes, permite que la red se centre en las partes de la entrada que considera más importantes, independientemente de su posición, facilitando el aprendizaje y generando mejores resultados. Esta combinación empodera a las redes neuronales recurrentes, que ya habían sido utilizadas en solitario con éxito en muchas tareas, mejorando su rendimiento. [11]

2.3 *Transformers*

Los *transformers* son modelos de aprendizaje profundo diseñados para el procesamiento de datos secuenciales basados en mecanismos de auto-atención [12]. Los *transformers* fueron presentados en 2017 por un equipo de Google Brain y han sustituido en gran medida otros modelos de aprendizaje profundo, como las redes neuronales recurrentes, por su buen desempeño en las tareas de procesamiento del lenguaje natural y visión por ordenador, así como su arquitectura paralelizable, que conlleva tiempos de entrenamiento reducidos y la posibilidad de escalar los modelos. Esto ha permitido crear grandes modelos del lenguaje que tienen una comprensión avanzada del mismo y que pueden especializarse en tareas concretas a través de la técnica conocida como *fine-tuning*.

2.3.1. Arquitectura de los *transformers*

Los *transformers* están formados por dos módulos, un componente de codificación y otro de decodificación, conectados entre sí. El componente de codificación, está formado por un número de codificadores o *encoders* apilados, y de la misma manera, el componente de decodificación esta constituido por el mismo número de decodificadores o *decoders*.

Antes de entrar al componente de codificación y decodificación, el texto es procesado por un sistema de *embeddings*, que transforma las palabras en vectores. Después, se asocia un vector a cada *embedding* que representa la posición de la palabra en el texto en un módulo de codificación posicional o *positional encoding* en inglés. Los componentes del vector de posición son determinados por funciones trigonométricas como el seno y el coseno. Este proceso es necesario ya que, a diferencia de las redes neuronales recurrentes, donde las palabras se procesan de forma secuencial, el *transformer*, al procesar las palabras en paralelo, no tiene una noción del orden de las mismas.

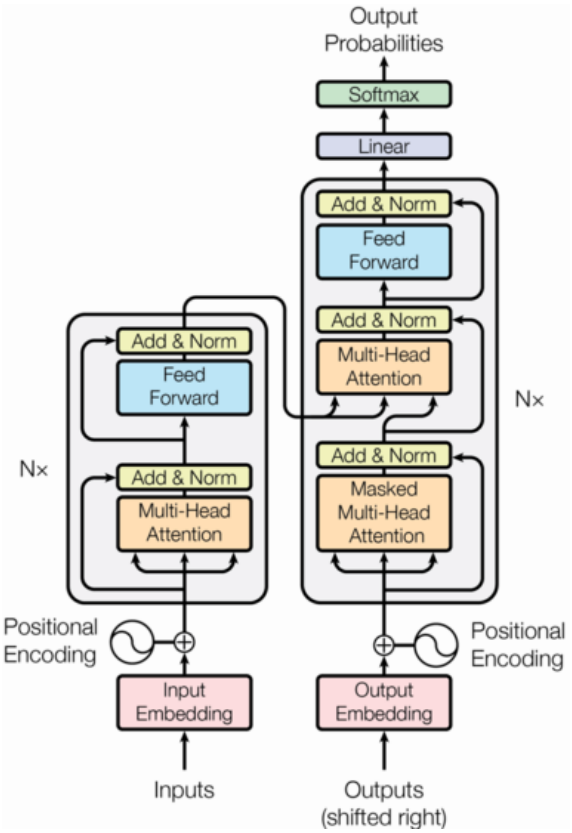

Figura 2.8: Componentes de un transformer

Los *encoders* están compuestos de dos elementos. En primer lugar, una capa de auto-atención que permite al *encoder* prestar atención a otras palabras de la secuencia mientras codifica una palabra. La salida de este vector, se añade a la entrada original y después se normaliza. El resultado de este proceso es tomado como entrada por el segundo elemento, una red neuronal prealimentada o *feed-forward*. Al igual que con el primer elemento, se suma la salida de la red prealimentada con su entrada (que es la salida de la capa de atención) y se normaliza. Cuando la entrada es procesada por el primer *encoder*, el resultado se usa como entrada del siguiente hasta llegar al último, que emite la salida final del módulo de codificación: una pareja de vectores *key* y *query* para cada palabra de la secuencia de entrada.

Al igual que en el *encoder*, cada *decoder* recibe como entrada la salida del *decoder* anterior y proporciona su salida como entrada al siguiente *decoder*. Sin embargo, a diferencia del módulo de codificación que recibe como entrada el texto original, la entrada del de decodificación es doble: la salida del módulo de codificación y su propia salida. Por este motivo, los *decoders* tienen una composición similar a los *encoders*, pero además del módulo de auto-atención y la red neuronal prealimentada, tienen también una capa de atención adicional, que es la que procesa la salida del último *encoder* y está situada entre las otras dos capas.

La salida generada por el módulo de codificación es un vector, y para convertirlo en una palabra existen dos ultimas capas: una capa lineal y una capa *softmax*. La capa lineal esta formada por una red neuronal totalmente conectada que genera un vector del tamaño del vocabulario y asigna una puntuación a cada palabra. Después, la capa *softmax* interpreta la salida asignando probabilidades a cada palabra, es decir, un valor entre 0 y 1 de tal manera que la suma de todas las probabilidades sea igual a 1. La palabra con la mayor probabilidad es escogida como resultado.

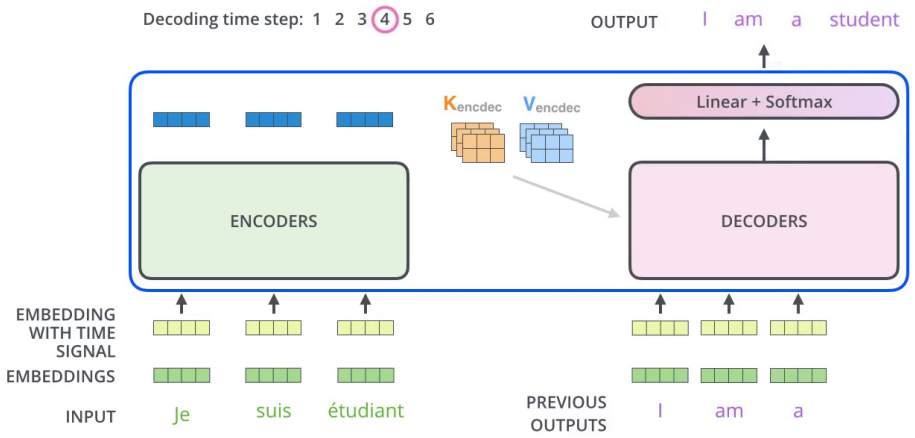

Figura 2.9: Transformer decodificando en una tarea de traducción en un transformer

Tanto el módulo de codificación como el de decodificación se pueden usar juntos, como se ha explicado, o por separado, dependiendo de la tarea:

- **Modelos *encoder***: Son los modelos que solo utilizan el módulo de codificación y destacan en tareas como la clasificación de textos y el reconocimiento de entidades nombradas.
- **Modelos *decoder***: Estos modelos solo constan de el módulo de decodificación y son usados para tareas generativas, como la generación de textos.
- **Modelos *encoder-decoder***: También conocidos como *sequence-to-sequence models* (modelos secuencia a secuencia), están diseñados para tareas de generación en las que se requiere una entrada, como la traducción o el resumen.

2.3.2. BERT

En 2018 se publicó un modelo de procesamiento del lenguaje natural llamado BERT (Bidirectional Encoder Representations from Transformers) [13], capaz de resolver una gran variedad de tareas de clasificación y generación de textos. BERT ha obtenido nuevos resultados del estado del arte en 11 tareas distintas, incluyendo *benchmarks* como GLUE [14], RACE [15] y SQuAD [16].

El modelo BERT trabaja a nivel de tokens dividiendo las palabras mediante el sistema *WordPiece* [17], lo que permite mejorar el tratamiento de palabras poco comunes y ofrece un buen equilibrio entre la flexibilidad de los modelos que trabajan a nivel de caracteres y la eficacia de los modelos que usan palabras. BERT ha sido entrenado con una gran cantidad de datos (3300 millones de palabras) mediante dos tareas: predecir la palabra dada una frase con una palabra enmascarada (*Masked Language Model o MLM*) y determinar si dos frases son consecutivas (*Next Sentence Prediction* o NSP). Estas tareas no limitan el contexto a las palabras o frases previas, sino que le permiten fijarse también en las siguientes, lo que lleva a un aprendizaje bidireccional.

El aprendizaje bidireccional es posible gracias a los *embeddings* contextuales [18], que tienen en cuenta el contexto de una palabra para su representación. Los *embeddings* contextuales están compuestos por 3 *embeddings*: el *embedding* del token, que encapsula el significado del token, el *embedding* de la frase, que representa la frase completa y aporta contexto, y el *embedding* posicional, que indica en que posición de la secuencia se encuentra el token. BERT utiliza tres tokens especiales: "[CLS]" que indica el inicio de una secuencia, "[SEP]" que marca el final de una secuencia y "[MASK]" que indica que se ha

enmascarado una palabra. El token de separación y mascara se usan durante el preentrenamiento en las tareas de predicción de la siguiente frase y predicción de la palabra enmascarada respectivamente.

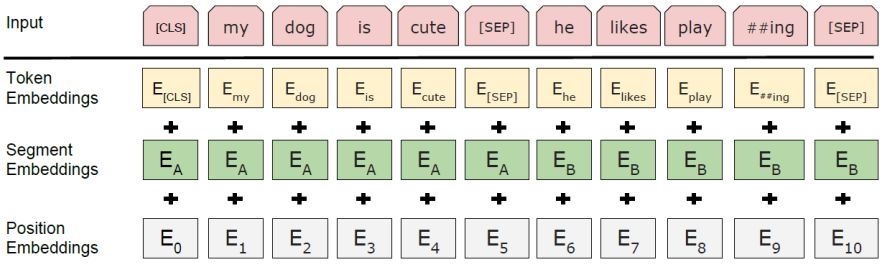

Figura 2.10: Embedding de una secuencia en el modelo BERT

Desde la publicación de BERT, se han desarrollado nuevos modelos como DistilBERT [19], que reduce el tamaño de BERT en un 40 % lo que lo hace un 60 % más rápido y retiene el 97 % del rendimiento, XLNet [20] que predice los tokens en orden aleatorio y mejora el rendimiento de BERT en 20 tareas, o XLM [21] que extiende el uso de BERT a diferentes lenguajes.

2.3.3. BERTa

En 2019 se publicó RoBERTa (Robustly Optimized BERT Pretraining Approach) [22], un modelo robusto basado en BERT que mejora los resultados obtenidos por BERT y los modelos posteriores. Los investigadores que desarrollaron RoBERTa demuestran en su estudio que BERT estaba infraentrenado así como la importancia de la selección de los hiperparámetros del modelo. Para el entrenamiento se descarta el uso de *NSP* y solo se entrena el modelo usando palabras enmascaradas, pero con enmascaramiento dinámico, que se basa en realizar varias copias de los datos enmascarando cada vez tokens distintos.

BERTa [23] es un modelo de procesamiento del lenguaje natural en catalán basado en RoBERTa, desarrollado por la unidad de minería de textos biomédicos [24] del Barcelona Supercomputing Center [25] bajo el Plan de Impulso de las Tecnologías del Lenguaje [26]. Este modelo fue desarrollado para demostrar la importancia de los modelos monolingües para lenguas con pocos recursos, incluso para un lenguaje proveniente del latín como el catalán, que podría dar ventaja a los modelos plurilingües entrenados con otras lenguas románicas. BERTa fue entrenado con un compendio de corpus que incluyen páginas web, noticias, documentos oficiales y subtítulos de películas. Durante el estudio se desarrolla un nuevo *benchmark* en catalán, llamado CLUB, en el que BERTa supera a los modelos plurilingües.

2.4 *Transfer Learning*

El *transfer learning* es una técnica mediante la que se transfiere el conocimiento obtenido por un modelo durante la resolución de un problema para aplicarlo en otro problema relacionado [27]. Para aplicar esta técnica se divide el proceso de entrenamiento en dos partes, *pretraining* y *fine-tuning*.

El *pretraining* es un entrenamiento inicial desde cero, en el que se toma un modelo con pesos inicializados aleatoriamente y sin ningún conocimiento previo. Para no con-

dicionar el conocimiento del modelo a una tarea concreta, se le da un objetivo genérico, como predecir la siguiente palabra de una secuencia o en una frase con una palabra oculta, sugerir palabras para rellenar el hueco. Para este entrenamiento se requiere una gran cantidad de datos y el proceso puede llevar semanas, incluso con una gran capacidad de computo.

El *fine-tuning* se realiza después de que el modelo haya sido preentrenado y tomando este modelo se realiza un entrenamiento adicional para la tarea final del modelo. Para este entrenamiento se requiere un corpus específico de la tarea para la que se esta entrenando, pero mucho más pequeño, ya que el modelo ya tiene cierta comprensión del ámbito sobre el que se trabaja.

Mediante *transfer learning* se crea un único modelo versátil y adaptable a distintas tareas permitiendo obtener buenos resultados con muchos menos datos, ya que el modelo es capaz de transferir el conocimiento del preentrenamiento a la nueva tarea, reduciendo así el tiempo y los recursos requeridos para entrenar el modelo.

El *transfer learning* se ha aplicado con éxito a muchos campos del aprendizaje automático y es una práctica común en el procesamiento del lenguaje natural y la visión por ordenador, debido al gran cantidad de recursos necesarios a nivel de datos y capacidad computacional necesarios para el entrenamiento de un modelo desde cero.

2.5 Tareas

2.5.1. Clasificación de textos

La clasificación de textos o secuencias es una tarea dentro del procesamiento del lenguaje natural que consiste en asignar una etiqueta a un texto [28].

Según el número de clases entre las que se puede clasificar el texto podemos encontrar problemas de clasificación binaria y clasificación multiclase. Algunos ejemplos de clasificación binaria son el análisis de sentimiento o la detección de *spam*. Dentro de la clasificación multiclase encontramos la detección de idiomas o la identificación de intenciones.

Uno de los problemas que existen en la clasificación multiclase es el desequilibrio de clases, que ocurre cuando existe una desproporción en el numero de muestras de cada clase. En un problema de clasificación desbalanceado se debe prestar especial atención al realizar la partición de los datos en los conjuntos de entrenamiento, validación y test para asegurar que existen muestras de todas las categorías en todos los conjuntos y, idealmente, que las clases tengan la misma proporción en todos los conjuntos.

Por otra parte, un conjunto de datos desequilibrado puede conducir a otro problema conocido como precisión desequilibrada. Esta situación sucede cuando un modelo tiene una tasa de acierto elevada (*accuracy*) y sin embargo no es capaz de clasificar correctamente. En estos casos, y para resolver este problema, es especialmente importante seleccionar métricas que permiten hacer una evaluación significativa del desempeño de los modelos.

2.5.2. Reconocimiento de entidades nombradas

El reconocimiento de entidades nombradas o NER (por sus siglas en inglés) es una tarea de procesamiento del lenguaje natural que consiste en localizar y clasificar las entidades nombradas encontradas en un texto en categorizas predefinidas [29].

La propia tarea puede ser dividida en dos: detección de entidades nombradas y clasificación de las mismas. La detección de entidades nombradas consiste en identificar que palabras o secuencias de palabras conforman una entidad. Las entidades no pueden solaparse ni estar anidadas, es decir, "Banco de España" es una única entidad, por lo que la palabra "España" contenida en la entidad no puede serlo, pese a que si estuviera por separado lo sería. La clasificación de entidades nombradas es la categorización de las entidades, detectadas en la fase anterior, en una de las categorías definidas a priori. Algunas de estas categorías pueden ser: persona, localización, organización, expresiones de tiempo, cantidades... Si bien ambas partes de la tarea pueden ser llevadas a cabo por separado, en muchas ocasiones, incluyendo los modelos desarrollados en la experimentación de este proyecto, se realizan de forma simultánea por un único modelo.

Figura 2.11: Texto con entidades nombradas anotadas

Existen distintos sistemas de etiquetado como IOB, IOE, IOBES o BILOU. Los corpus que se utilizarán en la experimentación utilizan el etiquetado IOB y IOB2. El sistema IOB (Inside-Outside-Beggining) utiliza una etiqueta I cuando el token actual se encuentra dentro de una entidad, O cuando no lo está y B cuando el token actual es el principio de una entidad consecutiva a otra del mismo tipo. El sistema IOB2 sigue las mismas normas excepto que la etiqueta B se utiliza siempre que un token es el primero de una entidad, independientemente de si sigue a otra del mismo tipo o no. De esta manera, la etiqueta I solo se utiliza para tokens dentro de una entidad que no son el primero.

2.6 Métricas y evaluación

En esta sección se definen lo que representan un acierto y un error en cada tarea, las métricas que se van a utilizar para evaluar los modelos y las matrices de confusión utilizadas para visualizar la confusión entre clases.

2.6.1. Aciertos y errores

Existen cuatro términos comunes a la mayoría de métricas diseñadas para evaluar modelos de clasificación. Si bien representan lo mismo, se definen de distinta forma para cada tarea:

Aciertos y errores en clasificación de textos

Aunque los términos están definidos originalmente para clasificación binaria son extensibles a clasificación multiclase:

- **Verdadero positivo** o *true positive* (TP): una predicción que indica correctamente que la muestra pertenece a la clase.
- **Verdadero negativo** o *true negative* (TN): una predicción que indica correctamente que la muestra no pertenece a la clase.
- **Falso positivo** o *false positive* (FP): una predicción que indica erróneamente que una muestra pertenece a una clase.
- **Falso negativo** o *false negative* (FN): una predicción que indica erróneamente que una muestra no pertenece una clase.

Aciertos y errores en reconocimiento de entidades nombradas

Al igual que en la clasificación de textos, la tarea de reconocimiento de entidades es multiclase. Además hemos de definir exactamente lo que se considera un acierto, ya que en ocasiones una entidad puede estar formada por más de una palabra:

- **Verdadero positivo** o *true positive* (TP): la palabra o conjunto de palabras que se identifica como entidad coincide exactamente con la entidad real y el tipo de ambas coincide.
- **Verdadero negativo** o *true negative* (TN): la palabra que se determina fuera de una entidad no pertenece a ninguna entidad.
- **Falso positivo** o *false positive* (FP): la palabra o grupo de palabras que se identifica como entidad no coincide exactamente con la entidad real y/o la entidad predicha no pertenece a la clase real.
- **Falso negativo** o *false negative* (FN): la entidad no se identifica correctamente porque no se detecta que pertenece a la clase real y/o porque la palabra o grupo de palabras que se identifica como entidad no coincide exactamente con la entidad real.

2.6.2. Métricas

Accuracy

La exactitud es el ratio entre el número total de aciertos (verdaderos positivos y verdaderos negativos) y el total de muestras. Esta métrica puede ser engañosa cuando la cantidad de muestras de las clases están desbalanceadas.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

La precisión es el porcentaje de muestras que se han identificado como cierta clase correctamente. Esta métrica favorece a los modelos que no confunden muestras de otras clases con la clase que se esta evaluando y penaliza aquellos que asignan etiquetas de la clase actual a muestras de otra clase, es decir, los que dan muchos falsos positivos.

$$Precision = \frac{TP}{TP + FP}$$

Recall

La sensibilidad, exhaustividad o cobertura es el ratio de muestras etiquetadas correctamente respecto del total de muestras pertenecientes a la clase. Esta métrica favorece a los modelos que no confunden muestras de la clase que se esta evaluando con otras clases y penaliza a aquellos que no son capaces de detectar estas muestras, es decir, que tienen muchos falsos negativos.

$$Recall = \frac{TP}{TP + FN}$$

F1 score

El valor-F es una media armónica entre la precisión y la sensibilidad. Esta métrica es muy utilizada ya que permite resumir las métricas de precisión y sensibilidad en un único valor que refleja las debilidades del modelo, ya que la media armónica penaliza los valores pequeños. De esta manera un modelo con una alta precisión pero sensibilidad deficiente, o viceversa, obtendrá un valor-F mediocre.

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

En un problema de clasificación multiclase, las métricas de *precision*, *recall* y *F1* se pueden computar con distintas estrategias:

- **Media Macro**: La media macro calcula la métrica para cada clase por separado y realiza una media no ponderada, equivalente a una media común si todas las clases tienen el mismo número de muestras, por lo que no tiene en cuenta el desbalanceo de clases.
- **Media Micro**: La media micro no tiene en cuenta las clases y utiliza los valores totales de verdaderos positivos, falsos positivos y falsos negativos para realizar el calculo de las métricas.
- **Media ponderada**: La media ponderada se realiza computando las métricas para cada clase y ponderando cada clase según su numero de muestras al realizar la media. Usando esta media, en ocasiones, el valor de la métrica F1 puede no estar entre la *precision* y el *recall*.
- **Por clase**: Esta estrategia no resume los datos en un único valor final, sino que calcula el valor de la misma para cada clase. Resulta útil a la hora de evaluar el desempeño del modelo para cada clase individualmente.

A lo largo de la experimentación se usará generalmente la media macro ya que da la misma importancia a todas las clases, independientemente del número de muestras. También se utilizarán las métricas por clase, para poder identificar las clases que mejor o peor clasifica el modelo.

2.6.3. Matriz de confusión

Una matriz de confusión es una herramienta que permite visualizar el desempeño de un modelo de clasificación y identificar que clases confunde entre si. En la matriz, la fila representa la clase real de la muestra y la columna la clase predicha por el modelo. De esta manera, dada una matriz de confusión A el elemento a_{ij} con valor n, representa que n muestras de la i-ésima clase real han sido clasificadas en la j-ésima clase por el modelo.

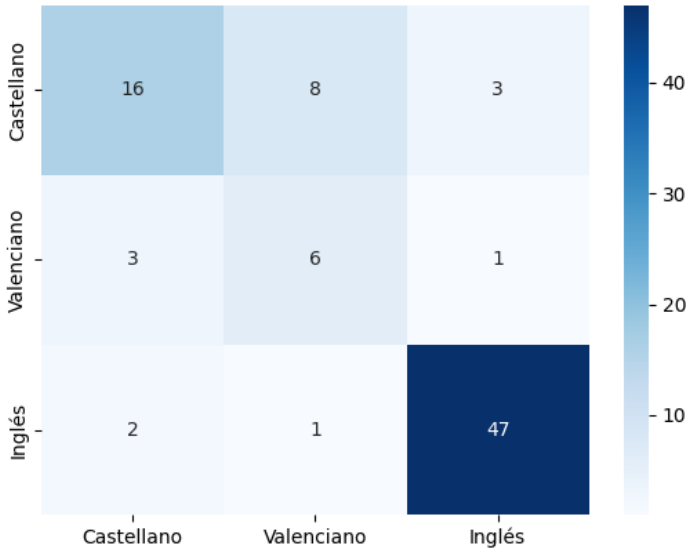

Figura 2.12: Matriz de confusión sin normalizar

En el caso de que las clases estén desbalanceadas, se pueden normalizar las filas y usar porcentajes para obtener una mejor representación visual.

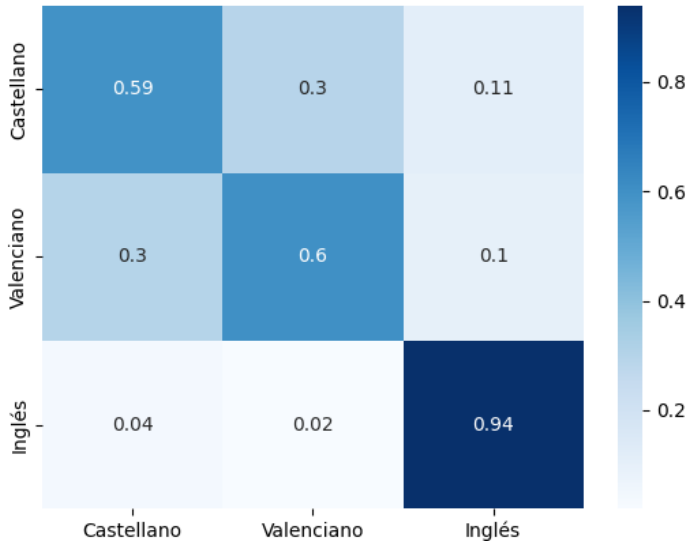

Figura 2.13: Matriz de confusión normalizada

CAPÍTULO 3
Clasificación de textos

En este capítulo se presenta el entorno y las tecnologías utilizadas durante la tarea de clasificación de textos para realizar la experimentación de la misma. Para la realización de la tarea se empleará un modelo de aprendizaje automático basado en *transformers*, explicado a nivel teórico en el capítulo anterior. En este capítulo también se estudiará y depurará el corpus facilitado por la *Corporació Valenciana de Mitjans de Comunicació* para su uso en el entrenamiento de los modelos. Finalmente, se realizará la evaluación de los modelos entrenados con las métricas previamente explicadas.

3.1 Herramientas utilizadas

3.1.1. Entorno de trabajo

El desarrollo de la experimentación se ha llevado a cabo en un dispositivo con sistema operativo Windows 10 con una tarjeta gráfica NVidia GTX 1060 con 6GB de VRAM. Gracias al uso de la tarjeta gráfica con la herramienta de computación paralela CUDA [30] en la versión 11.6 se ha podido paralelizar y acelerar el proceso de entrenamiento de los modelos. Como entorno de desarrollo se ha escogido Visual Studio Code [31] debido a su rapidez y versatilidad gracias a la facilidad para descargar e instalar extensiones y funcionalidades.

El lenguaje de programación utilizado ha sido *Python* [32] en su versión 3.9.5, mediante ficheros .py pero en ocasiones también con *Jupyter Notebooks* [33] o con el interprete del lenguaje por su flexibilidad al hacer pruebas. Esta decisión se basa en la importancia del lenguaje en el desarrollo del aprendizaje automático y tratamiento de datos, así como el gran número de librerías disponibles para la realización de estas tareas.

3.1.2. Librerías utilizadas

Pandas [34] es una librería de análisis y tratamiento de datos. La librería define una nueva estructura de datos llamada *DataFrame* que permite una gestión más estandarizada, intuitiva y eficaz de los datos. Los *DataFrames* se pueden crear mediante objetos como listas o diccionarios y también a partir de ficheros csv y xlsx. Esta librería se ha utilizado en su versión 1.4 para leer el archivo de *Excel* original que contenía el corpus y realizar el tratamiento de los datos.

Scikit-learn [35], también conocida como *Sklearn*, es una librería de aprendizaje automático que incluye varios algoritmos de clasificación, regresión y análisis de grupos así

como múltiples funcionalidades para tratamiento de datos, selección y evaluación de modelos entre otras. La librería se ha usado con su versión 1.0.1 para realizar las particiones de los datos y la evaluación de los modelos a través del computo de métricas y matrices de confusión.

Hugging Face [36] es una librería de aprendizaje automático que permite construir, entrenar y desplegar modelos compatibles con los principales *frameworks* de aprendizaje automático: *PyTorch* [37], *TensorFlow* [38] y *Jax* [39]. Además es una comunidad en la que se comparten modelos y colecciones de datos. La web cuenta actualmente con más de 50000 modelos y 6000 conjuntos de datos. Para esta tarea, se ha utilizado un modelo preentrenado disponible en el *Hub* de *Hugging Face* y se ha usado la librería en su versión 4.15 con *PyTorch* en la versión 1.10.1 para entrenar el modelo para clasificación de textos.

Seaborn [40] es una librería de visualización de datos que permite representar gráficas informativas. Se ha utilizado en su versión 0.11.2 para representar las matrices de confusión.

3.1.3. Librerías descartadas

SpaCy [41] es una librería de procesamiento del lenguaje natural orientada al desarrollo de software y despliegue de modelos en producción. Actualmente, soporta más de 66 idiomas, entre ellos el catalán, y además, cuenta con diversos modelos y *embeddings* preentrenados. Su uso se ha descartado en esta tarea debido a que no cuenta con una pipeline en catalán para la clasificación de textos y el modelo de propósito general está disponible en *Hugging Face* y el entrenamiento es más lento con esta librería.

Se han considerado otras librerias de visualización de datos como *MatPlotLib* [42] o *mlxtend* [43]. Ambas se han descartado ya que hacían la misma función que *Seaborn* pero la representación de la matriz de confusión esta librería era mejor al trabajar con muchas clases, como ocurre en uno de los experimentos de esta tarea.

3.2 Corpus

El corpus CVMC se origina en el sistema de documentación y catalogación de la *Corporació Valenciana de Mitjans de Comunicació*. El sistema lleva en funcionamiento desde su implementación en 1999 y cuenta con registros desde este año hasta el cierre de Canal 9 en 2013, y desde la reapertura de la cadena como À Punt en 2018 hasta la actualidad.

En el sistema se almacena información de cada noticia retransmitida en los telediarios de la cadena, así como algunas que no se han llegado a emitir. Además de lo dicho por los presentadores, los reporteros y la voz en *off* (almacenado conjuntamente como "texto"), también se registra el título de la noticia, la clase a la que pertenece, la sección y capítulo dentro del telediario, la fecha de emisión y palabras clave de la noticia. Para el estudio del corpus se ha utilizado el texto de la noticia, la clase y la fecha. Para el entrenamiento de los modelos se ha tomado el texto como entrada y la clase como salida esperada, ya que se trata de entrenamiento supervisado.

El corpus contiene 416610 noticias, de las cuales, 311604 son de la partición de Canal Nou, lo que representa el 74.8 % del total y el 25.2 % restante, 105006 noticias, de À Punt. Las noticias pueden pertenecer a una o varias clases entre las 38 que existen.

	Noticias	Porcentaje
Canal Nou	311604	74.8 %
À Punt	105006	25.2 %
Total	416610	100 %

Tabla 3.1: Noticias de Canal Nou y À Punt y porcentaje sobre el total del corpus

Sobre este conjunto de datos original se ha realizado un preprocesado. En primer lugar, se han eliminado las muestras duplicadas. En la partición de Canal Nou se han encontrado 1705 muestras duplicadas. Por otra parte, 52885 de las noticias de À Punt son duplicadas, lo que supone más de la mitad de las noticias.

	Noticias Únicas	Porcentaje Noticias Únicas	Noticias Duplicadas	Porcentaje Noticias Duplicadas
Canal Nou	309899	99.45 %	1705	0.55 %
À Punt	52121	49.64 %	52885	50.36 %
Total	362020	86.90 %	54590	13.10 %

Tabla 3.2: Noticias únicas y duplicadas en el corpus original

Después se han descartado las noticias que tuvieran menos de 10 palabras, ya que entre ellas se encontraban muchas muestras con contenido poco representativo o irrelevante. Se han encontrado 454 de estas noticias en el conjunto de Canal Nou y otras 2576 en el de À Punt.

	Noticias Validas	Porcentaje Noticias Validas	Noticias Descartadas	Porcentaje Noticias Descartadas
Canal Nou	309445	99.86 %	454	0.14 %
À Punt	49545	95.06 %	2576	4.94 %
Total	358990	99.16 %	3030	0.84 %

Tabla 3.3: Noticias validas y descartadas según su longitud

Una vez realizado el preprocesado, se ha estudiado el número y proporción de muestras con múltiples etiquetas. De las 358990 noticias, menos del 4 % pertenecen a múltiples clases (entre 2 y 5 clases en total).

	Noticias Uniclase	Porcentaje Noticias Uniclase	Noticias Multiclase	Porcentaje Noticias Multiclase
Canal Nou	299996	96.95 %	9449	3.05 %
À Punt	45698	92.24 %	3847	7.76 %
Total	345694	96.30 %	13296	3.70 %

Tabla 3.4: Noticias con una clase y multiclase

	1 Etiqueta	2 Etiquetas	3 Etiquetas	4 Etiquetas	5 Etiquetas
Canal Nou	299996	8916	500	32	1
À Punt	45698	3760	86	1	0
Total	345694	12676	586	33	1

Tabla 3.5: Noticias según su cantidad de etiquetas asignadas

Pese que se ha considerado la posibilidad de realizar un modelo de clasificación multi-etiqueta, se ha descartado debido al número reducido de muestras. Por este motivo, solo se han utilizado las noticias con una única etiqueta asignada.

De esta manera, finalmente han quedado 345694 noticias, 299996 pertenecientes al conjunto de Canal 9 y 45698 a À Punt. Al aplicar todos los criterios de filtrado, la proporción sobre el total de noticias de ambos conjuntos ha cambiado del 74.8 % de Canal Nou y 25.2 % de À Punt, a 86.78 % y 13.22 % respectivamente.

	Canal Nou	Porcentaje Canal Nou	À Punt	Porcentaje À Punt	Total	Porcentaje Total
esports	80425	26.81 %	14572	31.89 %	94997	27.48 %
justícia i ordre públic	34639	11.55 %	4356	9.53 %	38995	11.28 %
política	23705	7.90 %	6203	13.57 %	29908	8.65 %
societat	21121	7.04 %	2376	5.20 %	23497	6.80 %
accidents i catàstrofes	19959	6.65 %	1176	2.57 %	21135	6.11 %
festes i tradicions	12335	4.11 %	1078	2.36 %	13413	3.88 %
medicina i sanitat	8541	2.85 %	3535	7.74 %	12076	3.49 %
economia, comerç i finances	10597	3.53 %	1411	3.09 %	12008	3.47 %
agricultura, ramaderia i pesca	7564	2.52 %	630	1.38 %	8194	2.37 %
medi ambient	7226	2.41 %	814	1.78 %	8040	2.33 %
fenòmens naturals	7549	2.52 %	481	1.05 %	8030	2.32 %
transports i comunicacions	6673	2.22 %	1041	2.28 %	7714	2.23 %
terrorisme	5942	1.98 %	145	0.32 %	6087	1.76 %
conflictes armats	5679	1.89 %	273	0.60 %	5952	1.72 %
cultura	4034	1.34 %	624	1.37 %	4658	1.35 %
música	3717	1.24 %	598	1.31 %	4315	1.25 %
urbanisme i habitatge	3686	1.23 %	402	0.88 %	4088	1.18 %
turisme	3357	1.12 %	624	1.37 %	3981	1.15 %
cinema	3396	1.13 %	436	0.95 %	3832	1.11 %
obres públiques i infraestructures	3535	1.18 %	266	0.58 %	3801	1.10 %
treball	2980	0.99 %	793	1.74 %	3773	1.09 %
arts	3015	1.01 %	438	0.96 %	3453	1.00 %
ensenyament	2476	0.83 %	701	1.53 %	3177	0.92 %
ciència i investigació	2649	0.88 %	328	0.72 %	2977	0.86 %
religió	2779	0.93 %	147	0.32 %	2926	0.85 %
relacions internacionals	2359	0.79 %	499	1.09 %	2858	0.83 %

espectacles	2119	0.71 %	275	0.60 %	2394	0.69 %
mitjans de comunicació	1495	0.50 %	501	1.10 %	1996	0.58 %
indústria	1620	0.54 %	317	0.69 %	1937	0.56 %
energia	1256	0.42 %	88	0.19 %	1344	0.39 %
defensa	1154	0.38 %	70	0.15 %	1224	0.35 %
gastronomia	913	0.30 %	98	0.21 %	1011	0.29 %
literatura	461	0.15 %	152	0.33 %	613	0.18 %
història	415	0.14 %	183	0.40 %	598	0.17 %
miscel·lània	210	0.07 %	25	0.05 %	235	0.07 %
actualitat rosa	194	0.06 %	0	0.00 %	194	0.06 %
recursos territorials	101	0.03 %	38	0.08 %	139	0.04 %
tauromàquia	120	0.04 %	4	0.01 %	124	0.04 %

Tabla 3.6: Noticias por clase y porcentaje del total

Una vez realizado el preprocesado, es importante estudiar la representación de cada clase en el corpus. Como se puede ver en la tabla 3.6, la proporción de las 38 clases esta extremadamente desequilibrada. La clase más común es "esports", que contiene 94997 muestras, representando el 27.48 % del total. Por otra parte, "tauromàquia" representa únicamente el 0.04 %, con 124 noticias.

En la tabla 3.6 también podemos ver el cambio en las proporciones de las noticias entre los conjuntos de À Punt y Canal Nou. Por ejemplo, "medicina i sanitat" pasa de ser la octava clase más común en el conjunto de Canal Nou, representando el 2.85 % del total de noticias, a la cuarta, con un 7.74 %. En el otro extremo, la clase "actualitat rosa", que únicamente tenia 194 noticias en Canal Nou, desaparece completamente en À Punt.

3.3 Experimentación y análisis de resultados

La experimentación en esta tarea se divide en dos partes: por una parte, una serie de modelos de clasificación para las cuatro clases mayoritarias del corpus, y, por otro lado, otros modelos para clasificar noticias en las 29 clases más comunes en el corpus.

Se han tomado las cuatro clases con más muestras, que son "esports", "justícia i ordre públic", "política" y "societat", ya que estas cuatro clases representan el 54.21 % del corpus con un total de 187397 muestras. De esta manera, se dispone de suficientes datos para entrenar un modelo de clasificación, y al mismo tiempo se reduce la complejidad del problema al limitar el número de clases. De esta manera es posible evaluar el rendimiento de los modelos generados y determinar si la tarea es viable y se puede escalar añadiendo más clases.

Por otra parte, el segundo experimento se plantea una vez analizados los resultados del anterior. Para realizarlo, se han tomado aquellas clases que representan al menos un 0.5 % del total de noticias del corpus. De esta manera, quedan 29 clases, que representan el 98.42 % del corpus con un total de 340222 noticias. Este experimento plantea un escenario más cercano a la dimensión del problema real, pero se descartan aquellas clases con pocas noticias, ya que el modelo tendría dificultades para aprender a clasificarlas.

En ambos experimentos se han entrenado 3 modelos distintos utilizando una estrategia cronológica:

- **Modelo *Canal Nou***: Partiendo del modelo BERTa se realiza un *fine-tuning* con una parte de las noticias de Canal Nou y se evalúa con otra parte de este conjunto, así como las noticias de À Punt divididas por años.
- **Modelo *À Punt 2018***: A partir del modelo anterior, el de Canal Nou, se realiza un nuevo entrenamiento usando las muestras de À Punt pertenecientes al año 2018. Después se evalúa con las mismas noticias del conjunto de Canal Nou que se evaluó el modelo anterior y las noticias de À Punt de los años 2019 y 2020.
- **Modelo *À Punt 2019***: Tomando el modelo previo (À Punt 2018) como punto de partida, se entrena este modelo utilizando las noticias de À Punt del año 2019. Al finalizar el entrenamiento, se realiza una evaluación con las noticias de Canal Nou, al igual que los otros modelos, y también con las muestras de À Punt de 2020.

Con este procedimiento se consigue simular en tres ocasiones una situación en la que un modelo se utiliza para clasificar noticias nuevas, es decir, no solo que no haya visto durante el entrenamiento, sino también noticias más recientes según su fecha de emisión. Gracias a esto, se puede hacer una previsión de qué ocurriría al usar el modelo entrenado con todos los datos para clasificar muestras del 2021.

3.3.1. Modelos de 4 clases

A partir del conjunto de noticias de Canal Nou, se han tomado las noticias pertenecientes a las siguientes clases: "esports", "justícia i ordre públic", "política" y "societat". Las 159890 muestras se han dividido en tres particiones: entrenamiento, validación y test con una proporción 80/10/10. Por otra parte, el conjunto de noticias de À Punt pertenecientes a las mismas cuatro clases, que contiene 27507 muestras, se ha dividido por años, obteniendo así tres particiones: 2018, 2019 y 2020, que contienen 5179, 12128 y 10200 muestras respectivamente.

Entrenamiento con la partición de entrenamiento de Canal Nou

Usando el modelo BERTa, referenciado en Hugging Face como "roberta-base-ca" [44], se ha realizado el *fine-tuning* para la tarea de clasificación de textos con el conjunto de entrenamiento constituido de 5 épocas. El conjunto de validación se ha utilizado para ver la evolución del modelo durante el entrenamiento y seleccionar el mejor modelo.

Una vez finalizado el entrenamiento, se ha evaluado el modelo usando la partición de test de Canal Nou y las tres particiones de À Punt por separado.

	Test Canal Nou	Test À Punt 2018	Test À Punt 2019	Test À Punt 2020
Precision	0.9089	0.8770	0.8875	0.8728
Recall	0.9036	0.8598	0.8679	0.8562
F1	0.9060	0.8665	0.8767	0.8632

Tabla 3.7: Precision, recall y F1 macro con el modelo inicial de 4 clases

El conjunto que mejor clasifica el modelo, como se puede ver en la tabla 3.7, es la partición de test de Canal Nou. Esto es esperable, ya que las muestras de este conjunto son más parecidas a los datos con los que se ha entrenado al modelo. Dentro de las particiones de À Punt, el conjunto que mejor clasifica el modelo es el que contiene noticias de 2019.

	Test Canal Nou	Test À Punt 2018	Test À Punt 2019	Test À Punt 2020
esports	0.9851	0.9877	0.9878	0.9864
justícia i ordre públic	0.9111	0.8456	0.8665	0.8449
política	0.9111	0.8999	0.9257	0.8841
societat	0.8167	0.7328	0.7268	0.7374

Tabla 3.8: Macro-F1 para cada clase con el modelo inicial de 4 clases

En la tabla 3.8 podemos observar el desempeño del modelo en la clasificación de las clases por separado. En este caso, la categoría de "esports" se ha clasificado mejor en las particiones de À Punt que en la de canal Nou, lo que no ocurre con el resto de clases. Si bien con la clase "política" no hay diferencias significativas, tanto con la clase de "societat" como con "justícia i ordre públic" se puede ver una disminución del rendimiento del modelo en las particiones de À Punt respecto del conjunto de test de Canal Nou.

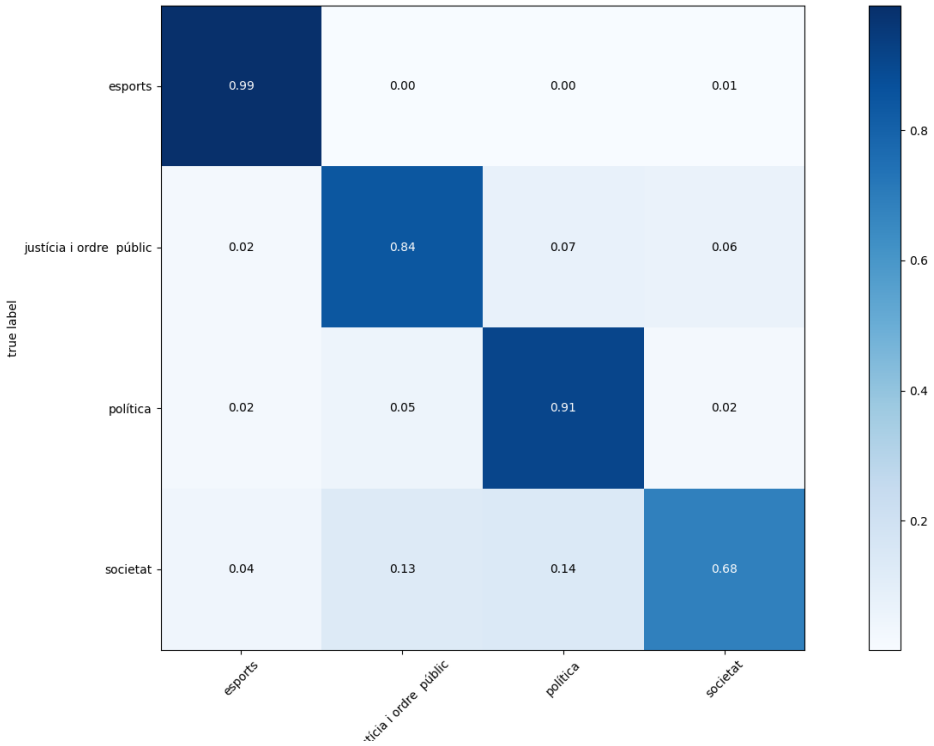

Figura 3.1: Matriz de confusión del modelo inicial de 4 clases para el conjunto de test de À Punt 2020

En la figura 3.1 se puede ver la matriz de confusión del modelo para la partición de test de À Punt 2020. En ella podemos observar que las noticias de la clase "justícia i ordre públic" son confundidas habitualmente con la clase "política" y "societat", así como que el modelo confunde las muestras de la clase "societat" mucho menos con la clase "esports" que con las otras dos.

Entrenamiento con la partición de 2018 de À Punt

Usando el modelo entrenado en el experimento anterior, se ha realizado un nuevo entrenamiento, esta vez con la partición de 2018 de À Punt. El modelo resultante se ha

evaluado usando tanto la partición de test de Canal Nou como las particiones de 2019 y 2020 de À Punt.

	Test Canal Nou	Test À Punt 2019	Test À Punt 2020
Precision	0.9023	0.8820	0.8656
Recall	0.8987	0.8808	0.8636
F1	0.9004	0.8814	0.8640

Tabla 3.9: Precision, recall y F1 con el modelo de 2018 de 4 clases

Comparando las tablas 3.7 y 3.9 podemos observar que tras realizar el entrenamiento con las muestras de À Punt de 2018, el modelo ha mejorado clasificando las particiones de 2019 y 2020 de À Punt, pero su desempeño ha empeorado en el conjunto de test de Canal Nou.

	Test Canal Nou	Test À Punt 2019	Test À Punt 2020
esports	0.9840	0.9900	0.9861
justícia i ordre públic	0.9046	0.8657	0.8394
política	0.9061	0.9254	0.8845
societat	0.8071	0.7443	0.7461

Tabla 3.10: Macro-F1 para cada clase con el modelo de 2018 de 4 clases

La diferencia más notable en la tabla 3.10 respecto de la tabla 3.8 es la mejora en la clasificación de la clase "societat" en ambas particiones del conjunto de À Punt. Otro cambio destacable es que el modelo ha empeorado clasificando las muestras de la clase "justícia i ordre públic", que es la segunda clase que peor clasifican los modelos, respecto del modelo anterior.

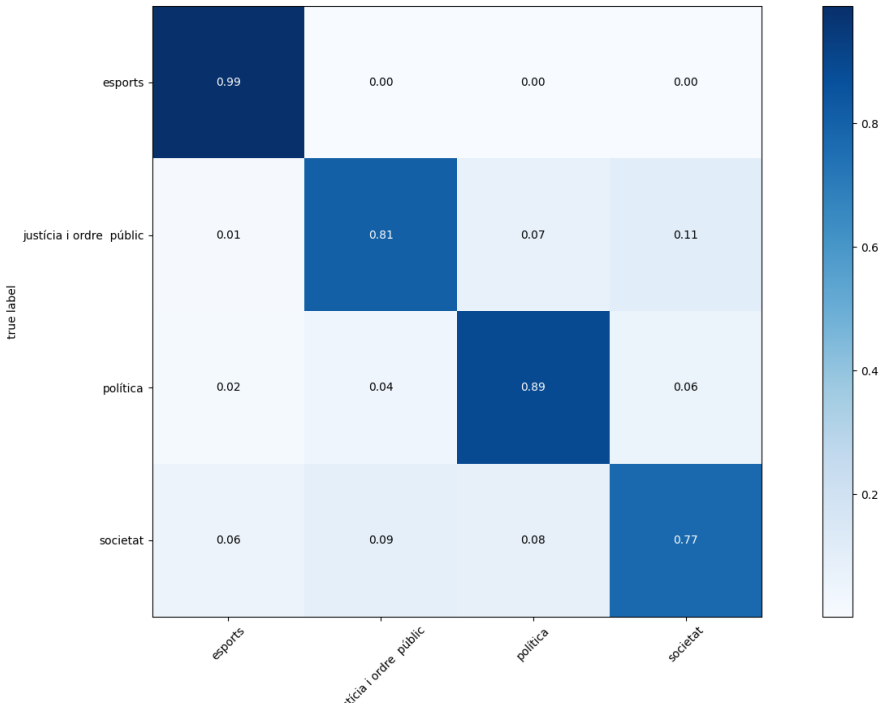

Figura 3.2: Matriz de confusión del modelo de À Punt 2018 para el conjunto de test de À Punt 2020

En la matriz de confusión del modelo de À Punt 2018 para figura la partición de test de À Punt 2020, que se puede ver en la figura 3.2, se puede observar la evolución respecto del modelo anterior 3.1. La mejora más importante es que el nuevo modelo clasifica correctamente el 77 % de las noticias frente a un 68 % del modelo anterior, un cambio del 9 %. Esto se debe a que ahora confunde menos las muestras de esta clase con las de "justícia i ordre públic" i "política", sin embargo el modelo ha empeorado clasificando las noticias de estas dos clases.

Entrenamiento con la partición de 2019 de À Punt

Utilizando el modelo del apartado anterior como punto de partida, se ha realizado un otro entrenamiento, usando el conjunto de noticias de À Punt de 2019. Tras el entrenamiento, se ha evaluado el modelo usando el conjunto de test de Canal Nou y la partición de 2020 de À Punt.

	Test Canal Nou	Test À Punt 2020
Precision	0.9042	0.8753
Recall	0.8936	0.8673
F1	0.8985	0.8710

Tabla 3.11: Precision, recall y F1 con el modelo de 2019 de 4 clases

En la tabla 3.11 podemos observar que la tendencia descrita en la comparación entre los dos modelos anteriores continua. Es decir, el modelo ha mejorado en la clasificación de las noticias de À Punt y ha empeorado ligeramente clasificando muestras de Canal Nou.

	Test Canal Nou	Test À Punt 2020
esports	0.9834	0.9894
justícia i ordre públic	0.9056	0.8491
política	0.9029	0.8925
societat	0.8022	0.7531

Tabla 3.12: Macro-F1 para cada clase con el modelo de 2019 de 4 clases

Por otra parte, en la tabla 3.12 se refleja la mejora del modelo en cada clase por separado. Esta vez, el modelo ha mejorado en la clasificación de todas las clases respecto del anterior.

Finalmente, en la figura 3.3 esta representada la matriz de confusión del modelo para la partición de test de À Punt 2020. El modelo ha empeorado clasificando muestras de la clase de "societat", pero ha mejorado su rendimiento con las clases de "política" i "justícia i ordre públic", en esta última se ha reducido en un 3 % el porcentaje total de muestras de la clase confundidas con la clase "societat".

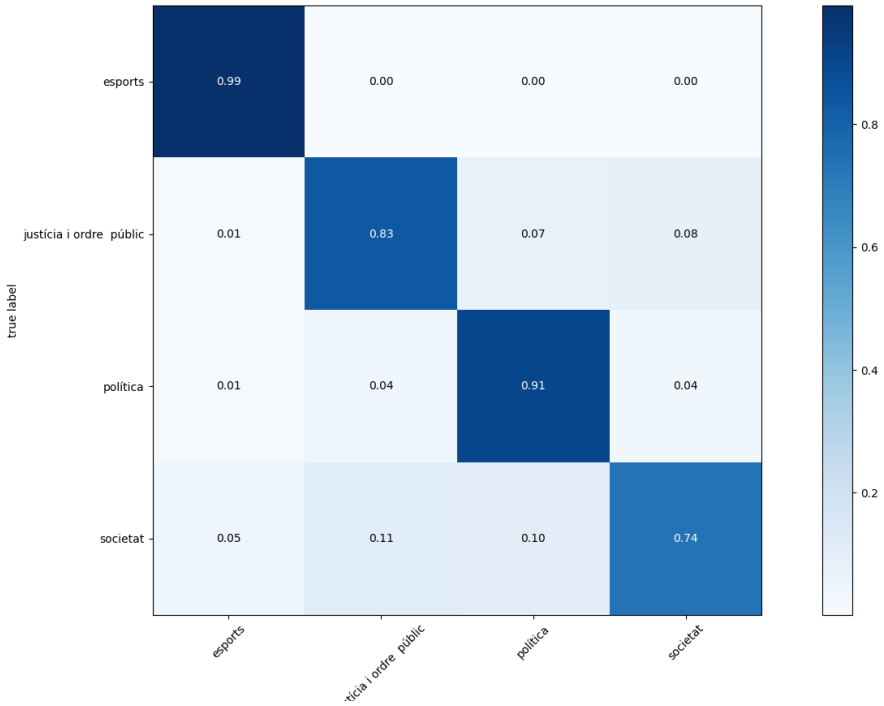

Figura 3.3: Matriz de confusión del modelo de À Punt 2019 para el conjunto de test de À Punt 2020

Comparativa de los modelos

Una vez realizados los experimentos, podemos comparar los tres modelos usando la partición de test de Canal Nou y la de 2020 de À Punt.

	Canal Nou	À Punt 2018	À Punt 2019
Precision	0.9089	0.9023	0.9042
Recall	0.9036	0.8987	0.8936
F1	0.9060	0.9004	0.8985

Tabla 3.13: Comparativa de los modelos sobre la partición de test de Canal Nou

	Canal Nou	À Punt 2018	À Punt 2019
esports	0.9851	0.9840	0.9834
justícia i ordre públic	0.9111	0.9046	0.9056
política	0.9111	0.9061	0.9029
societat	0.8167	0.8071	0.8022

Tabla 3.14: Comparativa de F1-Macro para cada clase de los modelos sobre la partición de test de Canal Nou

En la tabla 3.13 podemos observar cómo para la partición de test de Canal Nou, el modelo va empeorando ligeramente al ser entrenado con muestras de À Punt. Podemos extraer las mismas conclusiones de la tabla 3.14, donde se ve la evolución de cada clase por separado. En este caso, todas las clases empeoran en los modelos entrenados con noticias de À Punt respecto al modelo inicial.

	Canal Nou	À Punt 2018	À Punt 2019
Precision	0.8728	0.8656	0.8753
Recall	0.8562	0.8636	0.8673
F1	0.8632	0.8640	0.8710

Tabla 3.15: Comparativa de los modelos sobre la partición de 2020 de À Punt

	Canal Nou	À Punt 2018	À Punt 2019
esports	0.9864	0.9861	0.9894
justícia i ordre públic	0.8449	0.8394	0.8491
política	0.8841	0.8845	0.8925
societat	0.7374	0.7461	0.7531

Tabla 3.16: Comparativa de Macro-F1 para cada clase de los modelos sobre la partición de 2020 de À Punt

Finalmente, en la tabla 3.15, podemos ver la evolución de los modelos en la clasificación de la partición de 2020 de À Punt. Si bien existe una ligera mejora en el desempeño del modelo de 2018 respecto del original, el modelo mejora significativamente al ser entrenado con los datos de 2019. En la tabla 3.16 podemos ver la evolución de la métrica F1-Macro para cada clase. Así, podemos ver que el modelo de À Punt 2019 es mejor clasificando todas las clases, pero la mayor diferencia es en la clase de "societat".

En conclusión, podemos determinar que existe una mejora en la clasificación para las noticias de À Punt tras entrenar a los modelos con muestras más recientes. Si bien esto conlleva una pérdida de rendimiento en la clasificación de las noticias de Canal Nou, el uso de los modelos entrenados con noticias recientes es más interesante, ya que nos permite clasificar noticias nuevas con mayor efectividad.

3.3.2. Modelos de 29 clases

Para el entrenamiento de estos modelos se ha seguido el mismo procedimiento que con los modelos de cuatro clases, pero esta vez con las veintinueve más comunes. Por un lado, se han tomado las 295172 noticias pertenecientes a estas clases del conjunto de noticias de Canal Nou. Este conjunto se ha dividido en las particiones de entrenamiento, validación y test que contienen el 80 %, 10 % y 10 % de las muestras respectivamente. Por otro lado, se ha dividido el conjunto de noticias de À Punt en tres particiones según el año al que pertenecen las muestras. De esta manera se ha constituido la partición de 2018, que contiene 7917 muestras, la de 2019 con 17987 muestras, y la de 2020, a la que pertenecen 19138 muestras.

Entrenamiento con la partición de entrenamiento de Canal Nou

Al igual que con el modelo de cuatro clases, se ha usado como punto de partida el modelo BERTa, sobre el que se ha realizado el *fine-tuning* para la tarea de clasificación de textos usando la partición de entrenamiento del conjunto de datos de Canal Nou. Este proceso ha constado de 5 épocas y posteriormente se ha utilizado el conjunto de validación para seleccionar el mejor modelo.

Tras el entrenamiento se ha utilizado la partición de test de Canal Nou y el conjunto de À Punt para evaluar el modelo.

	Test Canal Nou	Test À Punt 2018	Test À Punt 2019	Test À Punt 2020
Precision	0.7112	0.7256	0.7124	0.6594
Recall	0.7049	0.7032	0.6972	0.6543
F1	0.7074	0.7107	0.7025	0.6528

Tabla 3.17: Precision, recall y F1 con el modelo inicial de 29 clases

A diferencia de los modelos de cuatro clases (tabla 3.7), quitando la partición de 2020, no existe una gran diferencia en las métricas macro entre los conjuntos de Canal Nou y À Punt, incluso se obtiene mejor F1 para la partición de 2018 de À Punt que Canal Nou.

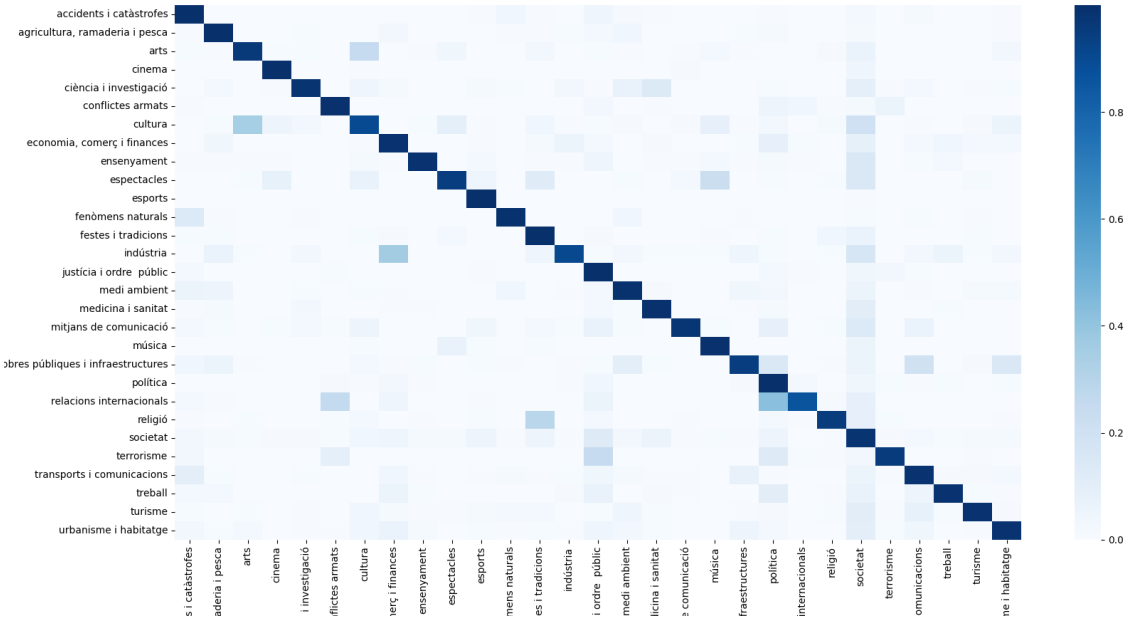

Figura 3.4: Matriz de confusión del modelo inicial de 29 clases para el conjunto de test de Canal Nou

En la figura 3.4 está representada la matriz de confusión del modelo con la partición de test de Canal Nou. Con su ayuda se puede ver aquellas clases que el modelo tiene más dificultad para clasificar, en este caso son relacions internacionals", “cultura” e “indústria”.

En la matriz también se puede observar las clases que el modelo confunde más a menudo, como “arts” y “cultura”. Es importante recalcar que la matriz no es simétrica. Se puede ver con las clases “economia, comerç i finances” y “indústria”, ya que el modelo confunde frecuentemente la segunda clase con la primera, lo que no ocurre al revés.

Otra información que nos aporta la matriz de confusión es las clases con las que se suelen confundir las muestras a menudo, es decir, que tienen muchos falsos positivos. La clase que con la que se confunde más a menudo el modelo al clasificar muestras de otras clases es “societat”, como se puede ver en su columna, ya que contiene más casillas con colores oscuros fuera de la diagonal que otras columnas.

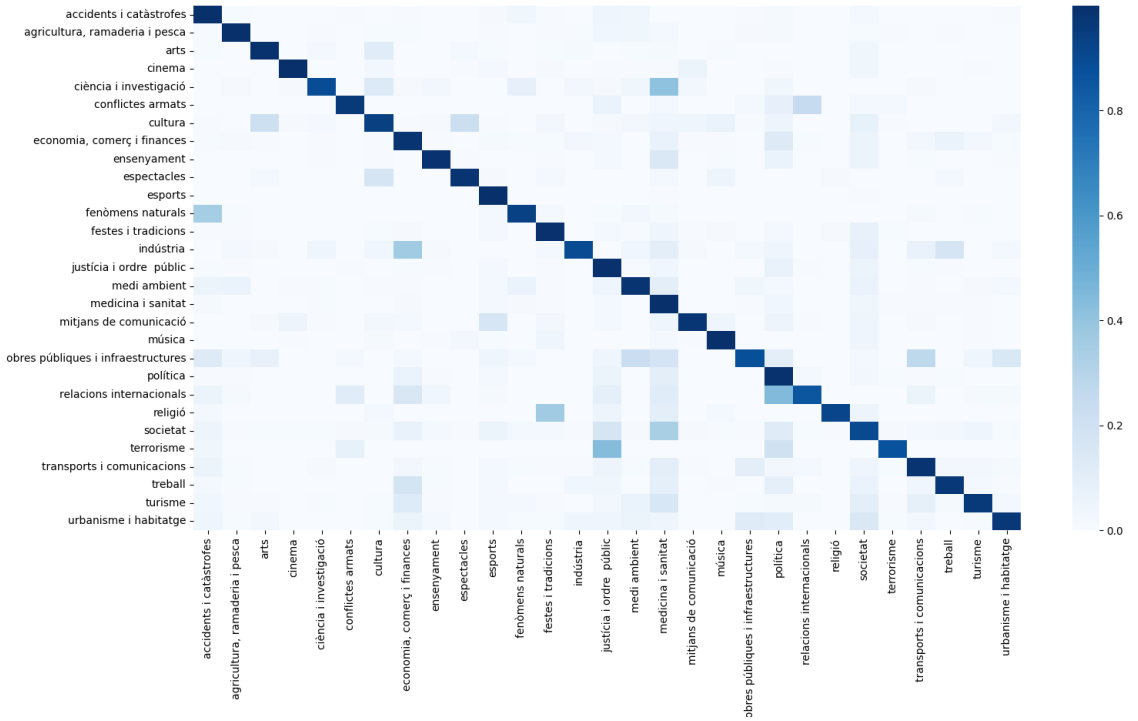

Figura 3.5: Matriz de confusión del modelo inicial de 29 clases para la partición de 2020 de À Punt

La matriz de confusión de la figura 3.5 ha sido obtenida con la partición de 2020 de À Punt. En ella se puede ver como se repiten algunos de los casos comentados en la figura 3.4 como la confusión de la clase "indústria" con "economia, comerç i finances".

Sin embargo, también hay diferencias. En primer lugar, se observa a simple vista que hay más casillas de color oscuro fuera de la diagonal que en la figura anterior. Esto es debido a que el modelo es menos preciso al clasificar estas muestras que las del conjunto de Canal Nou, como se puede ver en la tabla 3.17. Por otro lado, al igual que ocurría antes con la clase "societat", ahora existen más clases con muchos falsos positivos, como "medicina i sanitat" o "política".

Entrenamiento con la partición de 2018 de À Punt

A partir del modelo entrenado con las noticias de Canal Nou, se ha realizado un nuevo entrenamiento, esta vez con las muestras de À Punt de 2018. Su rendimiento se ha comprobado usando la partición de test de Canal Nou y el resto de noticias de À Punt (2019 y 2020).

	Test Canal Nou	Test À Punt 2019	Test À Punt 2020
Precision	0.6953	0.7026	0.6642
Recall	0.7097	0.7124	0.6793
F1	0.7004	0.7041	0.6672

Tabla 3.18: Precision, recall y F1 con el modelo de 2018 de 29 clases

Como se puede ver en la tabla 3.18, al igual que ocurría en el experimento anterior, el nuevo entrenamiento ha causado una mejora del modelo para las particiones de À Punt a cambio de empeorar su rendimiento clasificando muestras de Canal Nou. En este

caso la mejora más significativa se encuentra en el conjunto de 2020, que ha mejorado su F1-Macro en 0.0144.

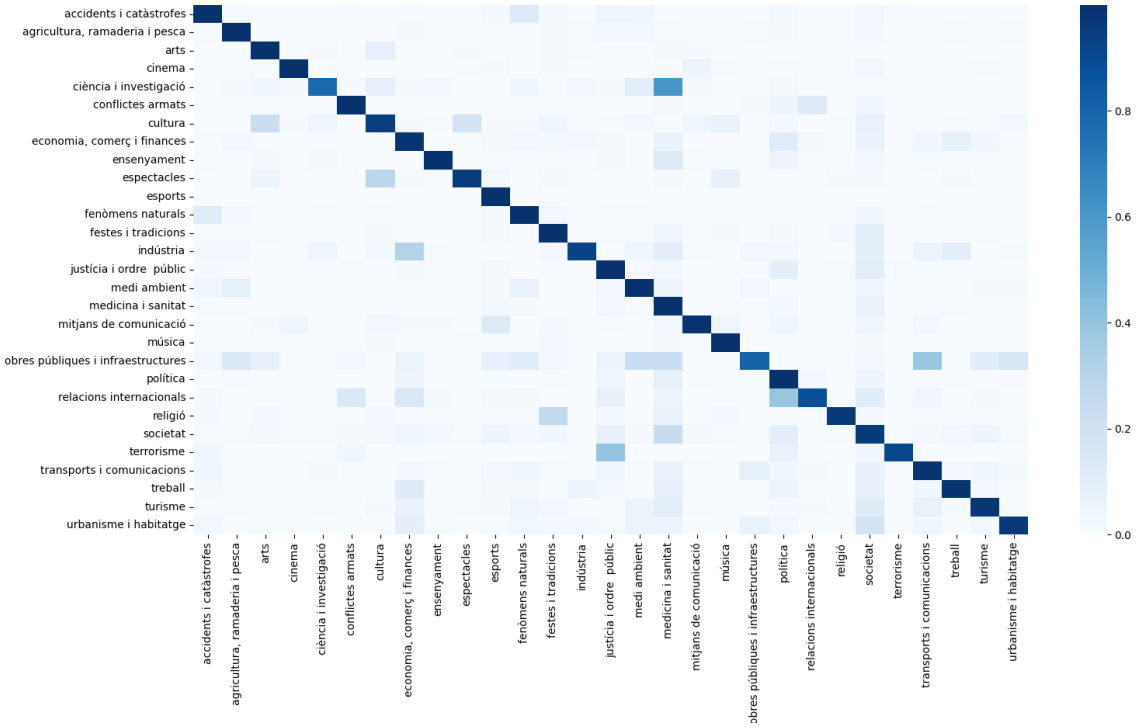

Figura 3.6: Matriz de confusión del modelo de 2018 de 29 clases para la partición de 2020 de À Punt

Comparando la matriz de confusión de la figura 3.6 con la del modelo anterior (figura 3.5) para la partición de À Punt 2020, se puede ver que, generalmente, hay menos casillas de colores oscuros fuera de la diagonal o han reducido su intensidad. Esto se debe a que el modelo, en general, clasifica mejor que antes. Sin embargo, hay ciertas clases que se clasifican peor que antes, como "ciència i investigació", cuyas noticias son confundidas más a menudo por el modelo con la clase "medicina i sanitat".

Entrenamiento con la partición de 2019 de À Punt

Finalmente, se ha entrenado un último modelo a partir del modelo del experimento anterior y usando las noticias de À Punt de 2019. Al finalizar el entrenamiento, se ha evaluado el modelo resultante con la partición de test de Canal Nou y las muestras de À Punt de 2020.

	Test Canal Nou	Test À Punt 2020
Precision	0.6897	0.6697
Recall	0.6985	0.6695
F1	0.6928	0.6658

Tabla 3.19: Precision, recall y F1 con el modelo de 2019 de 29 clases

Como se puede ver en la tabla 3.19, a diferencia del resto de modelos, en este caso el entrenamiento con muestras recientes no ha mejorado el rendimiento del modelo en la clasificación de las noticias de À Punt de 2020 según la métrica *F1*. Sin embargo, la métrica

accuracy si ha mejorado, esto quiere decir que el numero total de aciertos ha aumentado, aunque la media de las métricas de *precision* y *recall* ha disminuido ya que ha aumentado el numero de aciertos en las clases con más muestras y se ha reducido en las de menos muestras. Además, en el conjunto de test de Canal Nou el modelo ha empeorado más que su predecesor.

Esto se ve reflejado en la matriz de confusión de en la figura 3.7, donde en las filas de clases minoritarias como "espectacles", "indústria" y "terrorisme" se encuentran casillas más oscuras que en la matriz del modelo previo.

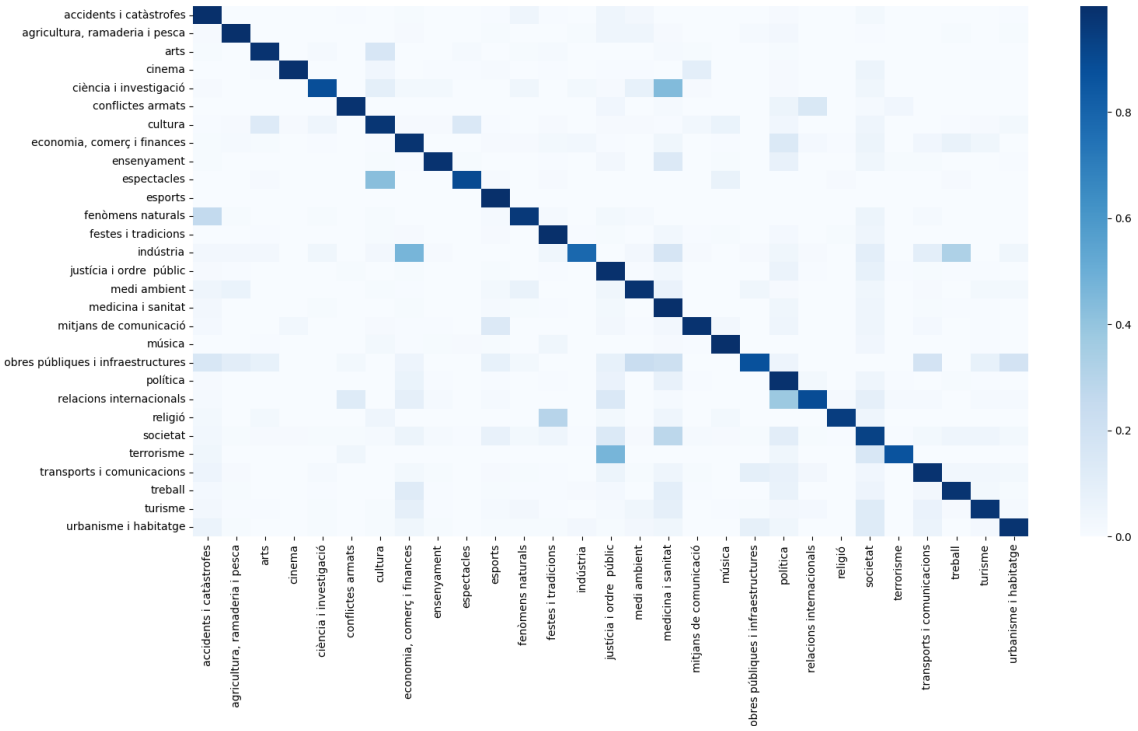

Figura 3.7: Matriz de confusión del modelo de 2019 de 29 clases para la partición de 2020 de À Punt

Comparativa de los modelos

Tras realizar todos los experimentos, vamos a comparar los modelos utilizando aquellas muestras que no se han usado para ningún entrenamiento de los modelos, es decir, la partición de test de Canal Nou y la de À Punt de 2020.

	Canal Nou	À Punt 2018	À Punt 2019
Precision	0.7112	0.6953	0.6897
Recall	0.7049	0.7097	0.6985
F1	0.7074	0.7004	0.6928

Tabla 3.20: Comparativa de los modelos de 29 clases con la partición de test de Canal Nou

Al igual que en el experimento anterior, reflejado por la tabla 3.13, en la tabla 3.20 se ve el empeoramiento de los modelos sobre el conjunto de test de Canal Nou tras ser entrenados con muestras de À Punt. En este caso, es especialmente destacable la reducción en la métrica *precision*, que ha disminuido en 0.0215 en total.

	Canal Nou	À Punt 2018	À Punt 2019
Precision	0.6594	0.6642	0.6697
Recall	0.6543	0.6793	0.6695
F1	0.6528	0.6672	0.6658

Tabla 3.21: Comparativa de los modelos de 29 clases con la partición de 2020 de À Punt

En la tabla 3.21 se refleja la evolución de los modelos sobre la partición de À Punt de 2020. A diferencia del modelo de cuatro clases, como veíamos en la tabla 3.15, la evolución del modelo no es continua según las métricas con ponderación macro. Específicamente, entre el modelo de 2018 y 2019 la puntuación de *recall* y *F1* empeoran. Sin embargo, como la métrica de *precision* ha aumentado de 0.7940 a 0.7963. Esto quiere decir que el modelo de 2019 clasifica más muestras correctamente que el de 2018, pero esta mejora ocurre en las clases con más muestras y al mismo tiempo ha empeorado clasificando clases minoritarias. En este caso, el modelo que mejor clasifica dependerá de la métrica elegida para evaluarlo, que dependerá del uso que se le vaya dar al modelo.

Finalmente, en la tabla 3.22 se puede ver la evolución de los modelos clase por clase con el conjunto de noticias de À Punt de 2020. Las dos últimas filas de la tabla se incluye la F1 del modelo con ponderación macro y micro.

	Canal Nou	À Punt 2018	À Punt 2019
accidents i catàstrofes	0.6946	0.7300	0.6829
agricultura, ramaderia i pesca	0.8067	0.7929	0.8023
arts	0.7423	0.7360	0.7546
cinema	0.8557	0.8693	0.8377
ciència i investigació	0.5242	0.4955	0.5301
conflictes armats	0.6434	0.6832	0.6797
cultura	0.5489	0.5624	0.5933
economia, comerç i finances	0.6268	0.6328	0.6321
ensenyament	0.7759	0.7978	0.7784
espectacles	0.6577	0.6275	0.6162
esports	0.9713	0.9732	0.9791
fenòmens naturals	0.5949	0.5870	0.6292
festes i tradicions	0.7602	0.7078	0.7931
indústria	0.5252	0.5675	0.4382
justícia i ordre públic	0.7799	0.7724	0.7805
medi ambient	0.6497	0.6822	0.6907
medicina i sanitat	0.8058	0.8169	0.8178

mitjans de comunicació	0.6858	0.6828	0.7079
música	0.8493	0.8858	0.8582
obres públiques i infraestructures	0.3415	0.3353	0.3568
política	0.7362	0.7484	0.7430
relacions internacionals	0.3860	0.4196	0.4170
religió	0.6422	0.6250	0.6786
societat	0.4683	0.4979	0.4722
terrorisme	0.4000	0.5435	0.4783
transports i comunicacions	0.6345	0.6536	0.6409
treball	0.6299	0.6554	0.6602
turisme	0.6205	0.6406	0.6423
urbanisme i habitatge	0.5733	0.6263	0.6173
macro	0.6528	0.6672	0.6658
micro	0.7884	0.7940	0.7963

Tabla 3.22: Comparativa clase por clase de los modelos de 29 clases con la partición de 2020 de À Punt

3.3.3. Comparativa de los modelos

Para finalizar el análisis de la experimentación de la tarea de clasificación de textos, realizaremos una comparativa final entre los últimos modelos de cuatro y veintinueve clases, es decir, aquellos entrenados con los datos de À Punt de 2019.

	Modelo 4 Clases	Modelo 29 Clases
Precision	0.9042	0.6697
Recall	0.8936	0.6695
F1	0.8985	0.6658

Tabla 3.23: Comparativa de los modelos de 4 y 29 clases con la partición de 2020 de À Punt

Si bien podemos fijarnos en las métricas macro, reflejadas en la tabla 3.23, para tener una visión general del desempeño de los modelos, no aportan demasiada información a la hora de comparar los modelos, ya que el numero de clases influye en los resultados de estas métricas. Por tanto, resulta más representativo en fijarnos en el desempeño de los modelos en las clases que comparten.

Como se puede ver en la tabla 3.24, todas el modelo de 29 clases tiene peor rendimiento clasificando todas las clases respecto al de 4 clases. En particular, la clase que más perjudicada se ve es la de "societat", cuyo valor de F1 disminuye en más de 0.28. Centrándonos en los valores de *precision* y *recall* podemos ver que el aumento de clases afecta en ambas métricas, ya que el modelo confunde las noticias de las 4 clases que estamos estudiando con otras clases, pero también las noticias de las otras clases con estas. Así, podemos concluir que el aumento de clases afecta al rendimiento del modelo en general y en cada clase en particular en la clasificación multiclase.

	Modelo 4 Clases			Modelo 29 Clases		
	Precision	Recall	F1	Precision	Recall	F1
esports	0.9856	0.9933	0.9894	0.9719	0.9863	0.9791
justícia i ordre públic	0.8642	0.8345	0.8491	0.7810	0.7800	0.7805
política	0.8798	0.9056	0.8925	0.7494	0.7366	0.7430
societat	0.7713	0.7531	0.7531	0.4792	0.4653	0.4722

Tabla 3.24: Comparativa para cada clase de los modelos de 4 y 29 clases sobre la partición de 2020 de À Punt

CAPÍTULO 4

Reconocimiento de entidades nombradas

En este capítulo se realiza la experimentación de la tarea de reconocimiento de entidades nombradas. En primer lugar, se presenta el entorno y las tecnologías utilizadas durante la realización de la tarea. Después se introducen dos corpus nuevos que serán utilizados para el entrenamiento de dos modelos distintos. Tras el entrenamiento, se evaluarán ambos modelos junto a otros dos modelos preentrenados utilizando las métricas explicadas en el capítulo 2. Finalmente se realiza una prueba con noticias pertenecientes al corpus de la *Corporació Valenciana de Mitjans de Comunicació*.

4.1 Herramientas utilizadas

4.1.1. Entorno de trabajo

El desarrollo de la experimentación se ha llevado a cabo en el mismo sistema que la tarea de la tarea de clasificación de textos, es decir, un dispositivo con sistema operativo Windows 10 con una tarjeta gráfica NVidia GTX 1060 con 6GB de VRAM. Al igual que para la tarea anterior, se ha utilizado la tarjeta gráfica con la herramienta de computación paralela CUDA para paralelizar y acelerar el proceso de entrenamiento de los modelos.

Al tener que usar herramientas similares para la realización de esta tarea se ha vuelto a utilizar *Python* 3.9.5 como lenguaje de programación en combinación con el entorno de desarrollo Visual Studio Code.

Durante la experimentación se ha realizado una prueba con un subconjunto del corpus CVMC, que no esta etiquetado para la tarea de reconocimiento de entidades nombradas. Para realizar este etiquetado se ha hecho uso de una herramienta *online* de código abierto para anotación de entidades nombradas [45] que genera archivos de JSON compatibles con *SpaCy*.

4.1.2. Librerías utilizadas

Al igual que en la tarea anterior, se ha utilizado la librería de aprendizaje automático *Hugging Face*. Esta vez, además del modelo preentrenado, también se han utilizado dos conjuntos de datos disponibles en el *Hub*. Para el entrenamiento del modelo en la tarea

de reconocimiento de entidades nombradas, se ha utilizado la librería *Hugging Face* en su versión 4.15 con *PyTorch* en la versión 1.10.1.

Además de *Hugging Face*, esta vez si se ha utilizado *SpaCy*. En la web de *SpaCy* encontramos 4 pipelines para procesamiento del lenguaje natural en catalán que usan distintas tecnologías y de diferentes tamaños. Todas ellas tienen un módulo de reconocimiento de entidades nombradas.

4.1.3. Librerías descartadas

Como en este caso, como los conjuntos de datos utilizados en la tarea estaban disponibles en *Hugging Face*, no ha sido necesario el uso de *Pandas* y *Scikit-learn* para el tratamiento de datos.

En este caso no ha sido necesario realizar ninguna gráfica, por lo que se ha descartado el uso de librerías de visualización de datos como *Seaborn*, *MatPlotLib* o *mlxtend*.

4.2 Corpus

Como se ha observado durante la experimentación en la tarea de clasificación de textos, los modelos obtienen mejores resultados cuanto más parecidas son las muestras de entrenamiento y test. Siguiendo este razonamiento, para crear un modelo de reconocimiento de entidades que obtenga buenos resultados en el reconocimiento de entidades nombradas en noticias de Canal Nou y À Punt sería útil entrenar al modelo con noticias del corpus de la *Corporació Valenciana de Mitjans de Comunicació*. Sin embargo, la tarea de reconocimiento de entidades nombradas requiere un entrenamiento supervisado y el conjunto de datos de CVMC no está etiquetado para esta tarea, por lo que necesitamos utilizar un conjunto de datos en catalán que sí este etiquetado para esta tarea.

4.2.1. AnCora

AnCora [46] es un corpus en catalán y español anotado para diferentes tareas de procesamiento del lenguaje natural como lema y categoría morfológica o constituyentes y funciones sintácticas. El corpus de cada lengua contiene 500.000 palabras y están constituidos fundamentalmente por textos periodísticos.

En el *Hub* de *Hugging Face* se encuentra el corpus de AnCora en catalán para la tarea de reconocimiento de entidades nombradas. Este conjunto de datos ha sido depurado por la unidad de minería de textos biomédicos del *Barcelona Supercomputing Center* para separar las entidades multipalabra y está anotado con el formato IOB. Las palabras pueden estar etiquetadas como una (o ninguna si no son entidades) de las siguientes cuatro categorías: persona, localización, organización y miscelánea.

El conjunto de datos contiene un total de 13581 muestras divididas en los conjuntos de entrenamiento, validación y test. Las particiones contienen 10628, 1427 y 1526 muestras respectivamente.

4.2.2. Wikiann

El corpus de Wikiann [47] es el resultado de un proyecto para desarrollar un sistema de etiquetado y enlazado plurilingüe para los 282 idiomas que existen en la Wikipedia.

Este corpus esta pensado para entrenar modelos monolingues y plurilingües para reconocimiento de entidades nombradas y también para evaluar las capacidades de modelos plurilingües en idiomas no vistos durante el entrenamiento.

En *Hugging Face* está disponible una versión del corpus etiquetada con el formato IOB2 con 176 de los 282 idiomas originales, entre los que se encuentra el catalán. El conjunto de datos de catalán contiene 40000 muestras que han sido divididas en las particiones de entrenamiento, validación y test en proporciones de 50/25/25. A diferencia del corpus de AnCora no existe la categoría de "miscelánea", por lo que las palabras pueden estar etiquetadas como "persona", "localización", "organización" o ninguna en el caso de que no sean entidades.

4.3 Experimentación y análisis de resultados

En la experimentación de esta tarea se han entrenado dos modelos distintos a partir del modelo BERTa, disponible en *Hugging Face* bajo el nombre "roberta-base-ca". Uno de los modelos ha sido entrenado con el corpus de AnCora y el otro con el de Wikiann. Tras el entrenamiento, ambos han sido evaluados con la partición de test de ambos modelos para compararlos. Finalmente, los modelos se han probado usando un conjunto reducido de noticias del corpus CVMC etiquetado manualmente.

4.3.1. Entrenamiento de los modelos

Antes del entrenamiento de ambos modelos existe un problema que se ha de solucionar, y es que en el corpus las etiquetas están asignadas de entidad palabra a palabra sin embargo, el modelo no procesa las palabras completas, sino tokens. Por tanto, debemos tokenizar el corpus con el tokenizador y después alinear los tokens generados con las etiquetas para poder entrenar y evaluar el modelo.

Entrenamiento del modelo con el corpus AnCora

En primer lugar, tomando el modelo "roberta-base-ca" como punto de partida, se ha realizado el *fine-tuning* para la tarea de reconocimiento de entidades nombradas usando el corpus de AnCora. Utilizando las 10628 muestras del conjunto de entrenamiento se han realizado 10 épocas y se ha utilizado el conjunto de validación para elegir el mejor modelo. Las métricas resultantes de la evaluación del modelo con la partición de test del conjunto de datos se pueden consultar en la tabla 4.1.

	LOC	ORG	PER	MISC
Precision	0.8607	0.7805	0.9462	0.5976
Recall	0.8723	0.8265	0.9614	0.5962
F1	0.8679	0.8028	0.9537	0.5969

Tabla 4.1: Evaluación del modelo de AnCora con la partición de test de AnCora

Entrenamiento del modelo con el corpus Wikiann

Por otro lado, también a partir del modelo "roberta-base-ca", se ha entrenando un modelo usando el corpus de Wikiann para la tarea de reconocimiento de entidades nombradas. Utilizando la partición de entrenamiento del conjunto de datos se han realizado

10 épocas y se ha utilizado la partición de validación para seleccionar el mejor modelo. Tras el entrenamiento se ha probado el modelo usando el conjunto de entrenamiento del mismo corpus obteniendo las métricas que se pueden ver en la tabla 4.2.

	LOC	ORG	PER
Precision	0.9035	0.8995	0.9230
Recall	0.9291	0.8951	0.9382
F1	0.9161	0.8973	0.9305

Tabla 4.2: Evaluación del modelo de Wikiann con la partición de test de Wikiann

4.3.2. Comparativa de los modelos

Tras el entrenamiento de ambos modelos y su evaluación vamos a proceder a comparar los modelos con ambos corpus. Es importante destacar los modelos no son comparables directamente, ya que el modelo entrenado con los datos de Wikiann clasifica en 3 grupos de etiquetas, "persona", "localización" y "organización", y el de AnCora en 4, los mismos tipos que el de Wikiann y "miscelánea".

Para la comparativa de los modelos se han usado también dos de las cuatro pipelines en catalán disponibles en la librería de *SpaCy*, en concreto "ca_core_news_trf" ("SpaCy Trf" en adelante) y "ca_core_news_lg" (referenciado como "Spacy CNN" a partir de ahora). El modelo "SpaCy Trf" parte de BERTa, el mismo modelo base que los modelos entrenados durante la experimentación, pero además del módulo de reconocimiento de entidades nombradas contiene otros módulos destinados a otras tareas que se han desactivado durante la evaluación ya que no afectan al desempeño del modelo. Por otro lado, "Spacy CNN" es una red neuronal convolucional (CNN) entrenada con *embeddings tok2vec*. Al igual que el modelo "SpaCy Trf", también consta con otros módulos, a parte del de reconocimiento de entidades nombradas, que se han desactivado para evaluar el modelo. Para poder evaluar estos modelos con los corpus, se ha realizado un procesado del corpus para adaptarlo a un formato compatible con las pipelines de *SpaCy*.

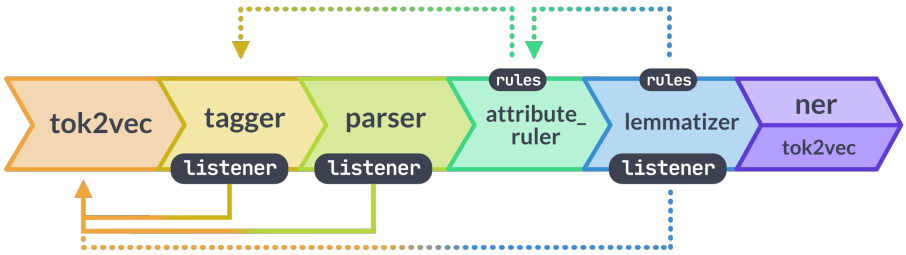

Figura 4.1: Estructura de una pipeline de SpaCy

Comparativa con el corpus de AnCora

En la tabla 4.3 se puede ver que ambas pipelines de *SpaCy* obtienen mejores resultados en el conjunto de datos de AnCora en comparación con los modelos entrenados durante la experimentación. Como es esperable el modelo entrenado con el corpus de Wikiann obtiene resultados muy inferiores a los de los otros modelos. Esto se acentúa especialmente ya que todas aquellas palabras o grupos de palabras que deberían ser etiquetados como miscelánea o no se les asigna etiqueta o se les asigna una etiqueta de otra clase, empeorando la métrica de precisión, y en consecuencia la F1.

	AnCora	Wikiann	SpaCy CNN	SpaCy Trf
Precision	0.7963	0.5300	0.8589	0.8963
Recall	0.8148	0.6827	0.8623	0.9048
F1	0.8054	0.5870	0.8605	0.9005

Tabla 4.3: Comparativa de los modelos con la partición de test de AnCora

Por otro lado, en la tabla 4.4 se muestra el valor de la métrica F1 para cada una de las etiquetas por separado. Del modelo entrenado con el corpus de AnCora es destacable su mal rendimiento respecto a las pipelines de *SpaCy* con las entidades de organización y especialmente las de miscelánea. El modelo de Wikiann clasifica mucho mejor las entidades de persona que el resto, y pese a su mal rendimiento general obtiene un valor elevado en la métrica F1 para este tipo de etiquetas.

	AnCora	Wikiann	SpaCy CNN	SpaCy Trf
LOC	0.8679	0.5310	0.8705	0.9059
ORG	0.8028	0.4034	0.8606	0.9211
PER	0.9537	0.8264	0.9621	0.9770
MISC	0.5969	-	0.7489	0.7982

Tabla 4.4: Comparativa de la F1 de los modelos para cada clase con la partición de test de AnCora

Comparativa con el corpus de Wikiann

Para el conjunto de datos de Wikiann son el modelo de AnCora y las pipelines de *SpaCy* los que se encuentran en desventaja, ya que pese a que en las muestras del conjunto de Wikiann no hay palabras o grupos de palabras etiquetadas como miscelánea, estos modelos si clasificaran algunas como tal, lo que puede afectar al recall y, por tanto, al F1.

	AnCora	Wikiann	SpaCy CNN	SpaCy Trf
Precision	0.5496	0.9086	0.4813	0.6378
Recall	0.5394	0.9212	0.4425	0.5529
F1	0.5307	0.9149	0.4602	0.5904

Tabla 4.5: Comparativa de los modelos con la partición de test de Wikiann

En la tabla 4.5 podemos ver las métricas de los modelos para el corpus de Wikiann. En esta evaluación el modelo que mejor rendimiento tiene es el entrenado con las muestras de Wikiann. A diferencia de la comparativa anterior, el modelo entrenado con el corpus de AnCora supera a la pipeline de *SpaCy* basada en redes neuronales convolucionales ("SpaCy CNN"), aunque sigue siendo peor que la basada en transformers ("SpaCy Trf").

	AnCora	Wikiann	SpaCy CNN	SpaCy Trf
LOC	0.5493	0.9161	0.4449	0.5082
ORG	0.2679	0.8973	0.3697	0.4963
PER	0.7747	0.9305	0.5659	0.7668

Tabla 4.6: Comparativa de la F1 de los modelos para cada clase con la partición de test de Wikiann

Finalmente, en la tabla 4.6 se observa el rendimiento de los modelos clase por clase. Aquí se puede ver que el modelo entrenado con los datos de AnCora supera la pipeline

de “SpaCy Trf” con el etiquetado de entidades de localización y persona, pero su F1 para las entidades de organización es tan bajo (el peor de los cuatro modelos) que hace que el macro-F1 sea inferior.

4.3.3. Test con noticias del corpus CVMC

Tras la comparativa de los modelos, ambos se han probado con un conjunto de 40 noticias tomadas del corpus del CVMC. Las noticias han sido anotadas manualmente usando una herramienta *online* de anotación de entidades nombradas. Este conjunto de noticias contiene un total de 122 entidades de las cuales 49 son de localización, 36 de persona, 35 de organización y 2 de miscelánea.

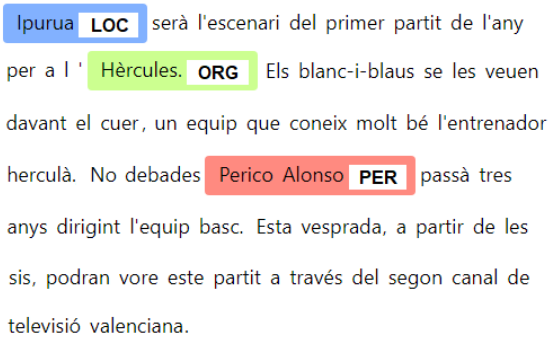

Figura 4.2: Noticia anotada manualmente

Generalmente, el modelo de AnCora tiende a detectar menos entidades de las anotadas manualmente, esto puede ser debido a que las muestras con las que esta entrenado suelen ser noticias largas y con pocas entidades. Por el contrario, el modelo entrenado con Wikiann suele identificar más entidades de las que se anotan manualmente, también podría estar relacionado con su corpus de entrenamiento, ya que contenía frases cortas y las entidades representaban un porcentaje alto de cada muestra.

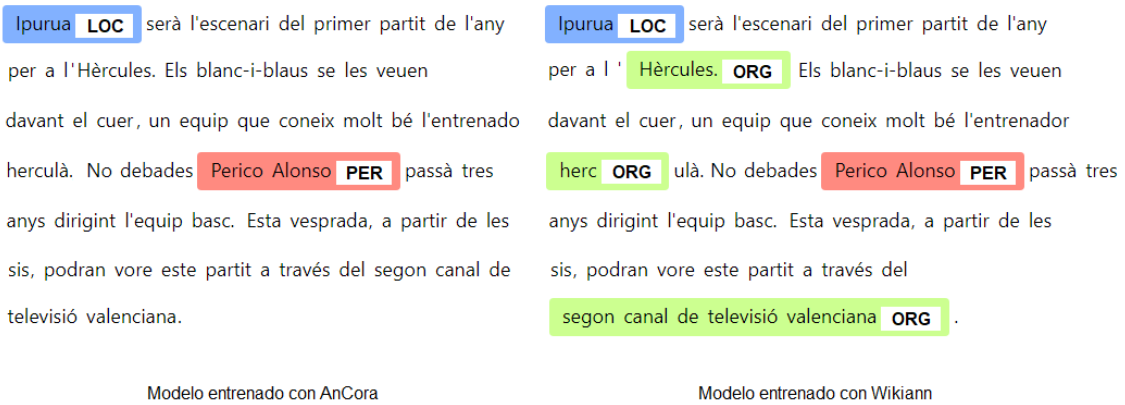

Figura 4.3: Entidades detectadas en una noticia por los modelos entrenados con los corpus de AnCora y Wikiann

En la figura 4.3 podemos ver la noticia de la figura 4.2 procesada por los modelos de reconocimiento de entidades nombradas entrenados con los corpus de AnCora y Wikiann. En el primer caso, el modelo entrenado con el corpus de AnCora no ha identificado “Hércules” como entidad. Por otra parte, el modelo entrenado con el conjunto de datos de

Wikiann ha detectado todas entidades correctamente, pero también ha detectado “herc” y “según canal de televisió valenciana” como entidades de organización.

El hecho de que el corpus de CVMC no esté etiquetado para NER supone un problema a la hora de evaluar los modelos. Al no disponer del tiempo necesario para etiquetar una cantidad de muestras que representen un porcentaje significativo del total del corpus, no se pueden generar unas métricas representativas del desempeño de los modelos. Al mismo tiempo también limita el rendimiento de los propios modelos, porque, como se ha observado en la experimentación de la tarea de clasificación de textos, los modelos rinden mejor con muestras más parecidas a aquellas vistas durante el entrenamiento.

CAPÍTULO 5
Conclusiones y trabajo futuro

En este capítulo se presentan las conclusiones finales extraídas tras la realización completa del proyecto. Además se aportan nuevas ideas y enfoques para continuar profundizando en el ámbito del proyecto.

5.1 Conclusiones

El proyecto tenía un doble objetivo inicialmente. Por una parte, desarrollar un sistema de clasificación de textos periodísticos que automatizara el proceso de catalogación de los documentalistas de la *Corporació Valenciana de Mitjans de Comunicació* o, en su defecto, que ayudara en la clasificación. Por otro lado, crear un modelo de reconocimiento de entidades nombradas que permita identificar personas, organizaciones y localizaciones u otras entidades de interés dentro de las noticias.

Respecto al primer objetivo, se han desarrollado diferentes modelos capaces de clasificar noticias entre un subconjunto de las categorías del corpus de la CVMC. En primer lugar, se han desarrollado varios modelos de clasificación con las 4 clases mayoritarias. Entre estos modelos, el entrenado con las noticias de À Punt de 2019, clasifica correctamente cerca del 93 % las noticias del año 2020. En segundo lugar, motivado por los resultados de los modelos de cuatro clases expuestos a lo largo de la experimentación, se ha decidido afrontar un problema más cercano a la tarea real, clasificar noticias en veintinueve clases. El modelo entrenado con muestras de 2019 del conjunto de À Punt, es capaz de clasificar en la categoría correcta casi el 80 % de las noticias del año siguiente. Es importante destacar que, al haber usado una enfoque cronológico durante el entrenamiento y la experimentación, se puede argumentar que los resultados son buenos indicadores del rendimiento de los modelos con noticias de años futuros. Estos resultados son bastante positivos, pero aún habiendo reducido el número de categorías a aquellas que representan al menos el 0.5 % del total del corpus, el modelo confunde más del 20 % de las noticias, así que podemos concluir que no es posible realizar un sistema de clasificación automática con los modelos entrenados durante la experimentación.

Acerca del segundo objetivo, se han entrenado dos modelos con corpus distintos y se han evaluado y comparado entre ellos y con otras herramientas de reconocimiento de entidades nombradas. Los corpus tenían un numero distinto de categorías de entidades, ya que uno incluía "miscelánea" y el otro no. Por este motivo, la comparación realizada no es un indicativo directo de cual de los dos modelos funciona mejor. Pese a que los modelos han tenido un buen rendimiento, el modelo disponible en SpaCy basado en transformers supera al modelo entrenado en la experimentación del proyecto con el mismo corpus y

que tiene el mismo número de categorías, lo que puede indicar que los modelos están infraentrenados y podrían beneficiarse de más épocas de entrenamiento. Se ha hecho una prueba con un conjunto reducido de noticias pertenecientes al corpus de la CVMC y se ha observado ciertas tendencias en cada uno de los modelos. El no disponer del propio corpus de la CVMC etiquetado ha limitado el rendimiento y la evaluación de los modelos para esta tarea.

En general, se puede concluir que, pese al buen rendimiento del modelo de clasificación, no se puede automatizar el sistema de categorización de noticias utilizando este modelo, pero si puede servir como una herramienta de apoyo a los documentalistas. Por otra parte, el modelo de reconocimiento de entidades nombradas puede resultar útil a la hora de encontrar palabras clave en los textos, que también constituye una parte del proceso de catalogación.

5.2 Trabajo futuro

Durante el planteamiento inicial del proyecto y la realización del mismo han surgido ideas que no se han podido llevar a cabo debido a limitaciones temporales o de recursos. Además, una vez terminado el proyecto existen distintas vías por las que seguir ahondando en las tareas abordadas.

Respecto al corpus de la *Corporació Valenciana de Mitjans de Comunicació*, de los distintos campos disponibles en el mismo solo se ha utilizado el texto de la noticia y la categoría la que pertenece durante el entrenamiento de los modelos. Existe la opción de incluir otras categorías que puedan añadir contexto u información para ayudar a los modelos en la clasificación. En este sentido, cabe la posibilidad combinar ambos sistemas desarrollados en la experimentación, ya que la presencia de ciertas entidades en un texto podría apuntar a que un texto pertenece a una categoría concreta. Por otra parte, se ha descartado el uso de las noticias con más de una etiqueta y resulta interesante el desarrollo de un modelo con la capacidad de clasificar las muestras en más de una categoría, especialmente considerando la ambigüedad en la diferenciación de ciertas categorías en algunos casos.

En la tarea de clasificación de textos, se ha probado un único modelo base. Esto se debe a que es el único disponible basado en *transformers* para procesamiento de lenguaje natural que se ha encontrado. Si hubieran más modelos disponibles se podría realizar *fine-tuning* para la tarea y comparar su rendimiento. Con el modelo BERTa que se dispone actualmente se podría realizar *hyperparameter tuning* para tratar de encontrar unos parámetros más optimizados y posiblemente obtener mejores resultados. Si se considera que los resultados con veintinueve clases son suficientemente buenos, se puede tratar de abordar la tarea de clasificación completa, con las treinta y ocho clases.

En relación a la tarea de reconocimiento de entidades nombradas, el mayor punto de interés sería disponer del corpus de la CVMC etiquetado para la tarea. Esto permitiría entrenar y evaluar a los modelos con este corpus y ver el desempeño de estos respecto de los disponibles en *SpaCy*. Al igual que para la clasificación de textos sería interesante evaluar el efecto de el *hyperparameter tuning* en el rendimiento de los modelos. Por otro lado, se propone la posibilidad de usar los modelos de reconocimiento de entidades nombradas en combinación con los de clasificación, como una herramienta para aportar información adicional para la clasificación, así como para sugerir nuevas categorías cuando se observe que existe una etiqueta muy común y resulta interesante diferenciarla de las categorías ya existentes.

Finalmente, se plantea la implementación de la herramienta de apoyo a la catalogación. Pese que esta implementación queda fuera del alcance del proyecto, se puede realizar un sistema de *software* que permita el uso de los modelos por parte de usuarios no especializados. Además, existe la posibilidad de mostrar las probabilidades de pertenencia a cada clase para sugerir más de una clase entre las que elegir a la hora de clasificar.

Bibliografía

[1] Aditya Koli, Diksha Khurana, Kiran Khatter, Sukhdev Singh. *Natural Language Processing: State of The Art, Current Trends and Challenges.* 2017

[2] Dan Jurasfsky, James H. Martin. Chapter 10: Machine Translation, *Speech and Language Processing.* Tercera Edición, 2022.

[3] Dan Jurasfsky, James H. Martin. Chapter 4: Naive Bayes and Sentiment Classification, *Speech and Language Processing.* Tercera Edición, 2022.

[4] Dan Jurasfsky, James H. Martin. Chapter 26: Automatic Speech Recognition and Text-to-Speech, *Speech and Language Processing.* Tercera Edición, 2022.

[5] Dan Jurasfsky, James H. Martin. Chapter 6: Vector Semantics and Embeddings, *Speech and Language Processing.* Tercera Edición, 2022.

[6] One-Hot Encoding. https://www.sciencedirect.com/topics/computer-science/one-hot-encoding.

[7] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space.* 2013.

[8] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. *GloVe: Global Vectors for Word Representation.* 2014.

[9] *fastText: Library for efficient text classification and representation learning.* https://fasttext.cc/.

[10] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate.* 2014.

[11] Chris Olah, Shan Carter. *Attention and Augmented Recurrent Neural Networks.* https://distill.pub/2016/augmented-rnns/

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. *Attention Is All You Need.* 2017.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* 2018.

[14] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman. *GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding.* 2018.

[15] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, Eduard Hovy. *RACE: Large-scale ReAding Comprehension Dataset From Examinations.* 2018.

[16] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, Percy Liang. *SQuAD: 100,000+ Questions for Machine Comprehension of Text.* 2016.

[17] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation.* 2016.

[18] Alan Akbik, Duncan Blythe, Roland Vollgraf. *Contextual String Embeddings for Sequence Labeling.* 2018.

[19] Victor Sanh, Lysandre Debut, Julien Chaumond, Thomas Wolf. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.* 2019.

[20] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le. *XLNet: Generalized Autoregressive Pretraining for Language Understanding.* 2019.

[21] Guillaume Lample, Alexis Conneau. *Cross-lingual Language Model Pretraining.* 2018.

[22] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* 2019.

[23] Jordi Armengol-Estapé, Casimiro Pio Carrino, Carlos Rodriguez-Penagos, Ona de Gibert Bonet, Carme Armentano-Oller, Aitor Gonzalez-Agirre, Maite Melero, Marta Villegas. *Are Multilingual Models the Best Choice for Moderately Under-resourced Languages? A Comprehensive Assessment for Catalan.* 2021.

[24] *Biomedical Text Mining Unit.* https://temu.bsc.es/.

[25] *Barcelona Supercomputing Center. Centro Nacional de Supercomputación.* https://www.bsc.es/es.

[26] *Plan de Impulso de las Tecnologías del Lenguaje.* https://plantl.mineco.gob.es/Paginas/index.aspx.

[27] Dan Jurasfsky, James H. Martin. Chapter 11: Transfer Learning with Contextual Embeddings and Pre-trained language models, *Speech and Language Processing.* Tercera Edición, 2022.

[28] Chris Manning, Hinrich Schütze. Chapter 16: Text Categorization, *Foundations of Statistical Natural Language Processing.* 1999.

[29] Dan Jurasfsky, James H. Martin. Chapter 8: Sequence Labeling for Parts of Speech and Named Entities, *Speech and Language Processing.* Tercera Edición, 2022.

[30] *CUDA.* https://developer.nvidia.com/cuda-zone.

[31] *Visual Studio Code.* https://code.visualstudio.com/.

[32] *Python.* https://www.python.org/.

[33] *Project Jupyter.* https://jupyter.org/.

[34] *Pandas.* https://pandas.pydata.org/.

[35] *Scikit-learn.* https://scikit-learn.org/stable/index.html.

[36] *Hugging Face.* https://huggingface.co/.

[37] *PyTorch.* https://pytorch.org/.

[38] *TensorFlow.* https://www.tensorflow.org/.

[39] *JAX.* https://jax.readthedocs.io/en/latest/.

[40] *Seaborn: statistical data visualization*. https://seaborn.pydata.org/.

[41] *SpaCy*. https://spacy.io/.

[42] *Matplotlib: Visualization with Python*. https://matplotlib.org/.

[43] *Mlxtend*. https://rasbt.github.io/mlxtend/.

[44] *BERTa: RoBERTa-based Catalan language model*. https://huggingface.co/PlanTL-GOB-ES/roberta-base-ca.

[45] *Tecoholic's NER Annotator*. https://github.com/tecoholic/ner-annotator.

[46] Maria Antònia Martí, Mariona Taulé, Manu Bertran and Lluís Màrquez *AnCora: Multilingual and Multilevel Annotated Corpora*. 2008.

[47] Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, Heng Ji *Cross-lingual Name Tagging and Linking for 282 Languages*. 2017.

APÉNDICE A
Objetivos ODS

Los Objetivos de Desarrollo Sostenible (ODS) son una iniciativa de la Organización de las Naciones Unidas (ONU) para realizar avances en asuntos de interés mundial como el fin de la pobreza, la protección el planeta y la mejora de las vidas y las perspectivas de las personas en todo el mundo. En 2015, todos los Estados Miembros de la ONU aprobaron 17 Objetivos como parte de la Agenda 2030 para el Desarrollo Sostenible, en la cual se establece un plan para alcanzar los Objetivos en 15 años.

Objetivos de Desarrollo Sostenible	**Alto**	**Medio**	**Bajo**	**No procede**
ODS 1. **Fin de la pobreza.**				X
ODS 2. **Hambre cero.**				X
ODS 3. **Salud y bienestar.**			X	
ODS 4. **Educación de calidad.**			X	
ODS 5. **Igualdad de género.**			X	
ODS 6. **Agua limpia y saneamiento.**				X
ODS 7. **Energía asequible y no contaminante.**				X
ODS 8. **Trabajo decente y crecimiento económico.**				X
ODS 9. **Industria, innovación e infraestructuras.**		X		
ODS 10. **Reducción de las desigualdades.**	X			
ODS 11. **Ciudades y comunidades sostenibles.**				X
ODS 12. **Producción y consumo responsables.**				X
ODS 13. **Acción por el clima.**				X
ODS 14. **Vida submarina.**				X
ODS 15. **Vida de ecosistemas terrestres.**				X
ODS 16. **Paz, justicia e instituciones sólidas.**				X
ODS 17. **Alianzas para lograr objetivos.**				X

Los Objetivos de Desarrollo Sostenible (ODS) son una iniciativa de la Organización de las Naciones Unidas (ONU) para realizar avances en asuntos de interés mundial como el fin de la pobreza, la protección el planeta y la mejora de las vidas y las perspectivas de las personas en todo el mundo. En 2015, todos los Estados Miembros de la ONU aprobaron 17 Objetivos como parte de la Agenda 2030 para el Desarrollo Sostenible, en la cual se establece un plan para alcanzar los Objetivos en 15 años.

El proyecto realizado está relacionado con tres de los objetivos ODS. El objetivo con el que esta más estrechamente relacionado es el número diez: "Reducción de las desigualdades". Actualmente, la lengua catalana se encuentra en una situación de diglosia respecto de la lengua castellana, siendo el catalán el idioma perjudicado en la situación. Si bien oficialmente ambas lenguas son cooficiales en la Comunitat Valenciana y desde las insti-

tuciones públicas se intenta fomentar el uso del valenciano su presencia es muy reducida en ciertos ámbitos de la sociedad. Por esto, la colaboración con la *Corporació Valenciana de Mitjans de Comunicació* puede ayudar a la labor de la organización en la normalización del uso de la lengua.

Por otro lado, el proyecto contribuye al noveno objetivo ODS: "Industria, innovación e infraestructuras". El proyecto desarrollado usa un modelo de aprendizaje automático desarrollado por el *Barcelona Supercomputing Center* bajo el Plan de Impulso de las Tecnologías del Lenguaje, que tiene como objetivo fomentar el desarrollo del procesamiento del lenguaje natural, la traducción automática y los sistemas conversacionales en lengua española y lenguas cooficiales. El proyecto tiene como objetivo ayudar en la labor de clasificación y etiquetado de las noticias realizada por los documentalistas de la *Corporació Valenciana de Mitjans de Comunicació*, por lo que se puede considerar como una manera de introducir y promover nuevas tecnologías para agilizar y mejorar el proceso y hacer una mejor gestión de los recursos humanos y económicos.

Finalmente, el proyecto también está vinculado con los objetivos número tres, cuatro y cinco: "Salud y bienestar", "Educación de calidad" y "Igualdad de género". La *Corporació Valenciana de Mitjans de Comunicació*, a través de À Punt, realiza una función didáctica y de preservación de la cultura y la lengua en la sociedad valenciana. En la web de À Punt podemos encontrar la "Carta de valors per als continguts infantils i juvenils" donde se marcan los principios de los contenidos dirigidos a los niños y adolescentes y se incentivan aquellos que transmitan valores relacionados con la formación, la vida saludable, la igualdad de género y la cultura entre muchos otros. Estos valores están directamente vinculados a los objetivos ODS nombrados, por lo que el proyecto puede ayudar de manera indirecta a fomentar estos valores.

APÉNDICE B
Tablas y figuras adicionales

Las figuras B.1, B.2, B.3 son matrices de confusión del modelo inicial de 4 clases.

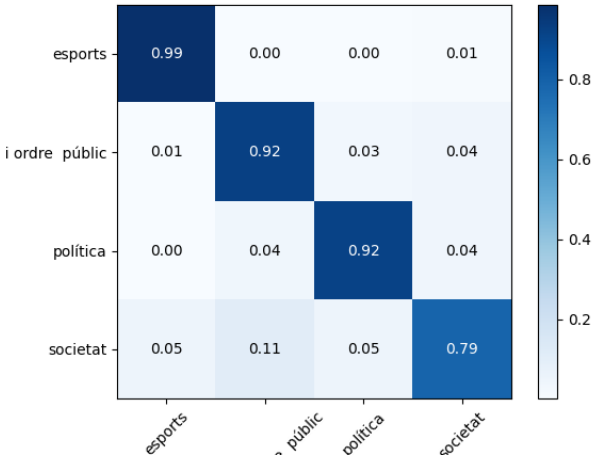

Figura B.1: Matriz de confusión del modelo inicial de 4 clases para el conjunto de test de Canal Nou

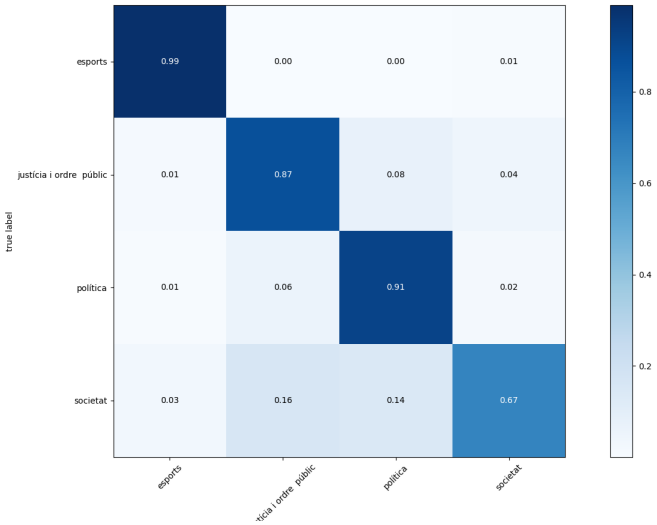

Figura B.2: Matriz de confusión del modelo inicial de 4 clases para el conjunto de test de À Punt 2018

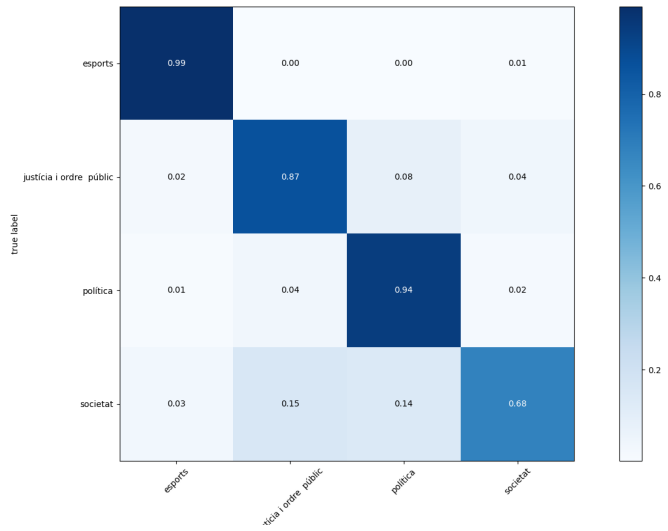

Figura B.3: Matriz de confusión del modelo inicial de 4 clases para el conjunto de test de À Punt 2019

Las figuras B.4, B.5 son matrices de confusión del modelo de À Punt 2018 de 4 clases.

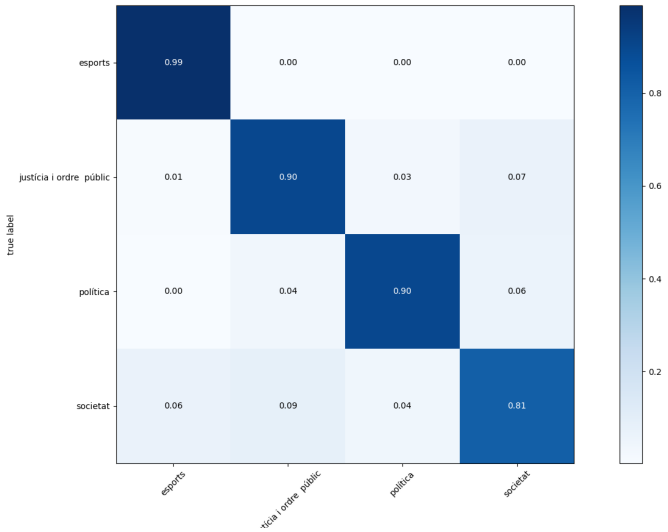

Figura B.4: Matriz de confusión del modelo de À Punt 2018 de 4 clases para el conjunto de test de Canal Nou

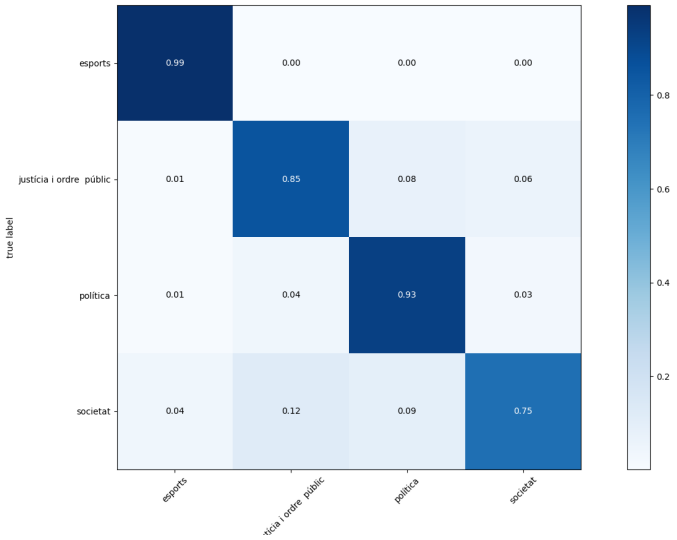

Figura B.5: Matriz de confusión del modelo de À Punt 2018 de 4 clases para el conjunto de test de À Punt 2019

Las figuras B.6 son matrices de confusión del modelo de À Punt 2019 de 4 clases.

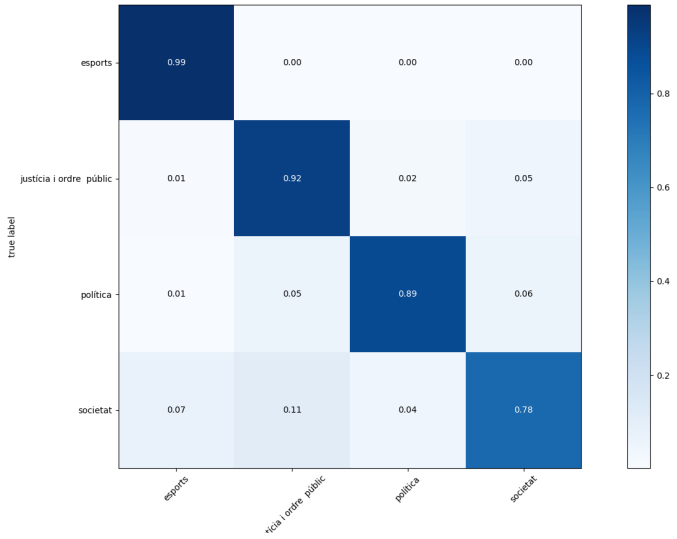

Figura B.6: Matriz de confusión del modelo de À Punt 2019 de 4 clases para el conjunto de test de Canal Nou

Las figuras B.7, B.8 son matrices de confusión del modelo inicial de 29 clases.

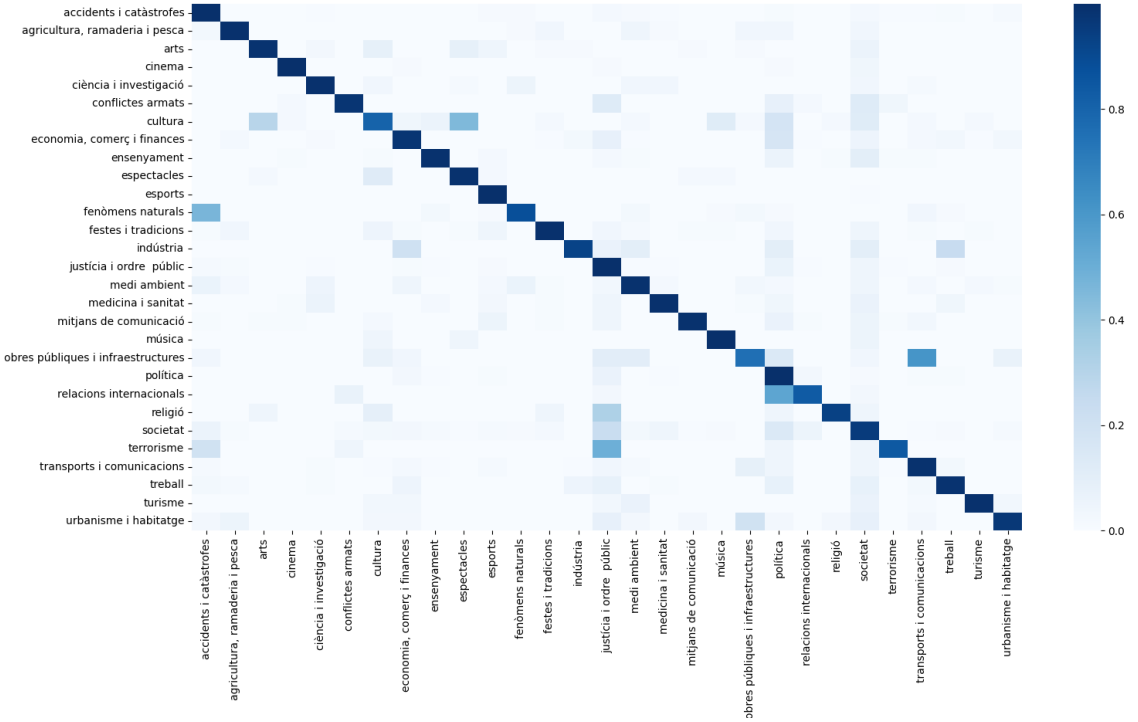

Figura B.7: Matriz de confusión del modelo de inicial de 29 clases para el conjunto de test de À Punt 2018

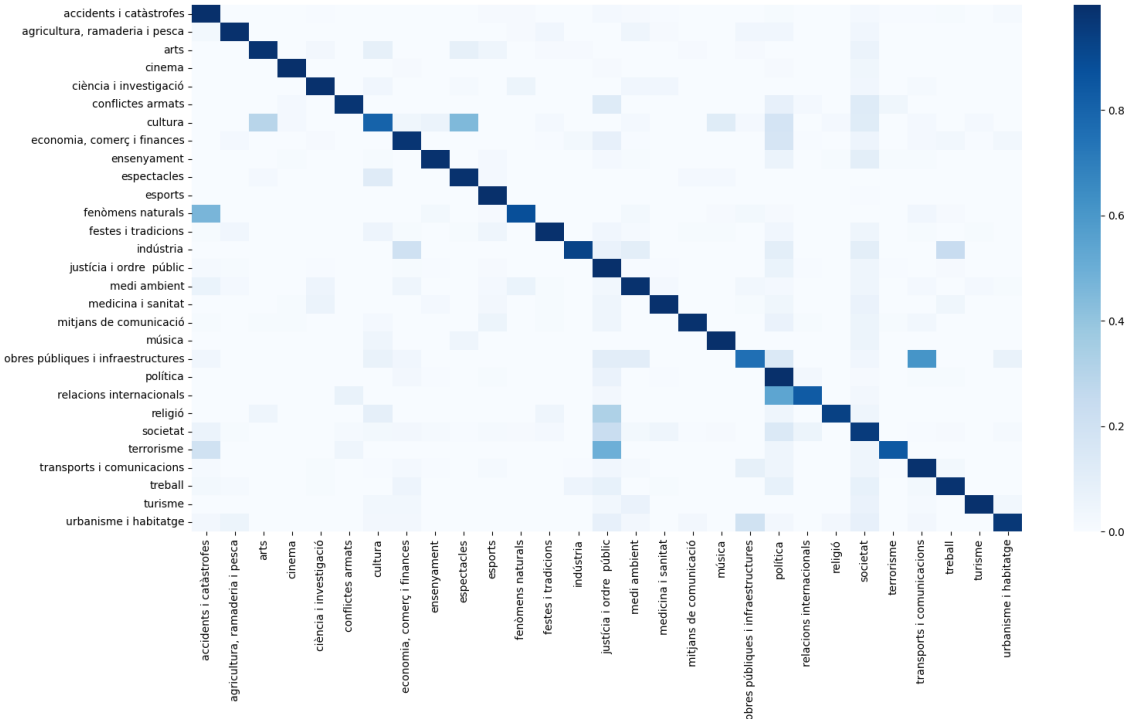

Figura B.8: Matriz de confusión del modelo de inicial de 29 clases para el conjunto de test de À Punt 2019

Las figuras B.9, B.10 son matrices de confusión del modelo de À Punt 2018 de 29 clases.

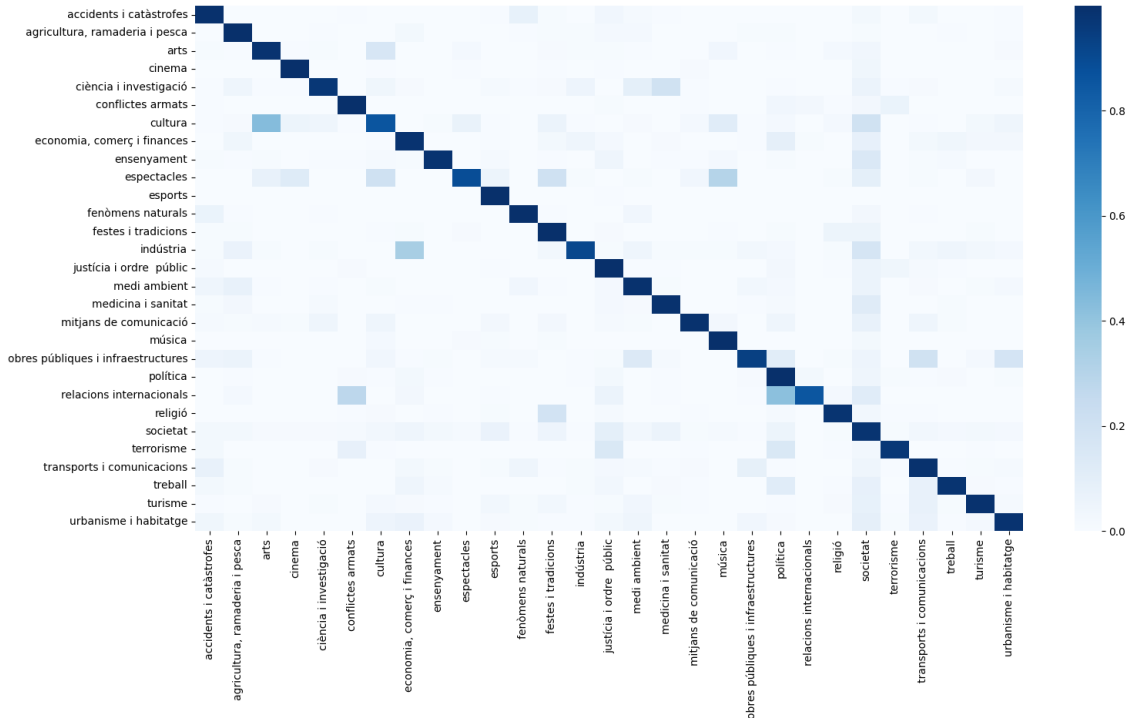

Figura B.9: Matriz de confusión del modelo de À Punt 2018 de 29 clases para el conjunto de test de Canal Nou

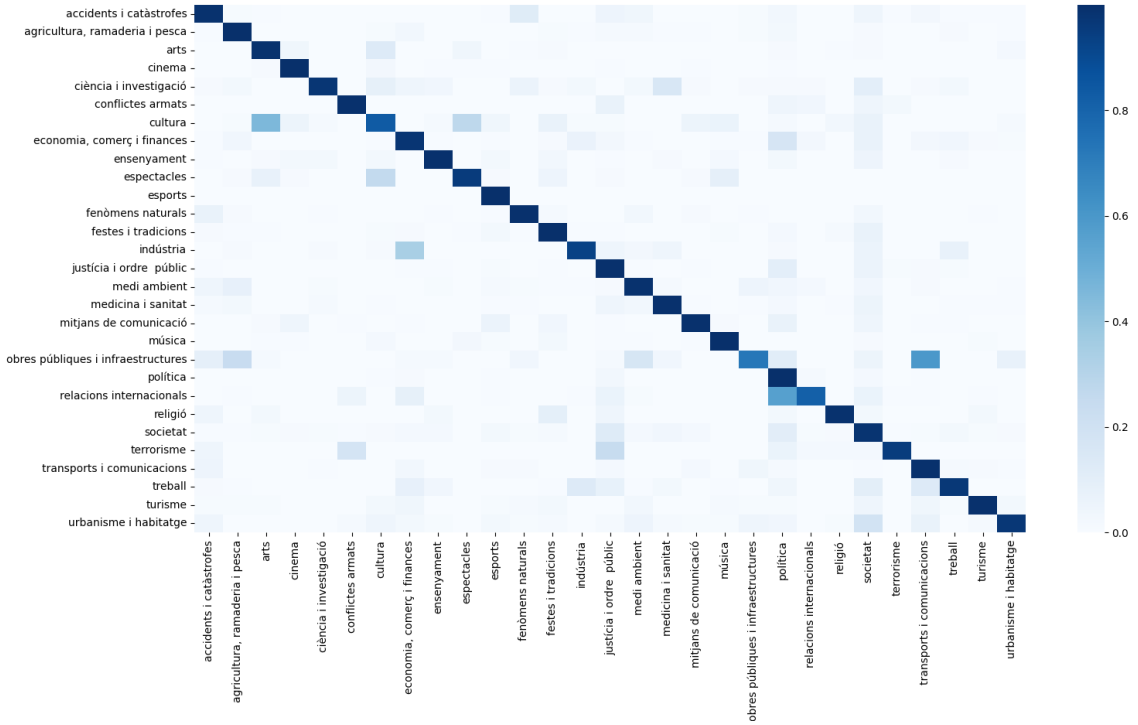

Figura B.10: Matriz de confusión del modelo de À Punt 2018 de 29 clases para el conjunto de test de À Punt 2019

La figura B.11 es una matriz de confusión del modelo de À Punt 2019 de 29 clases.

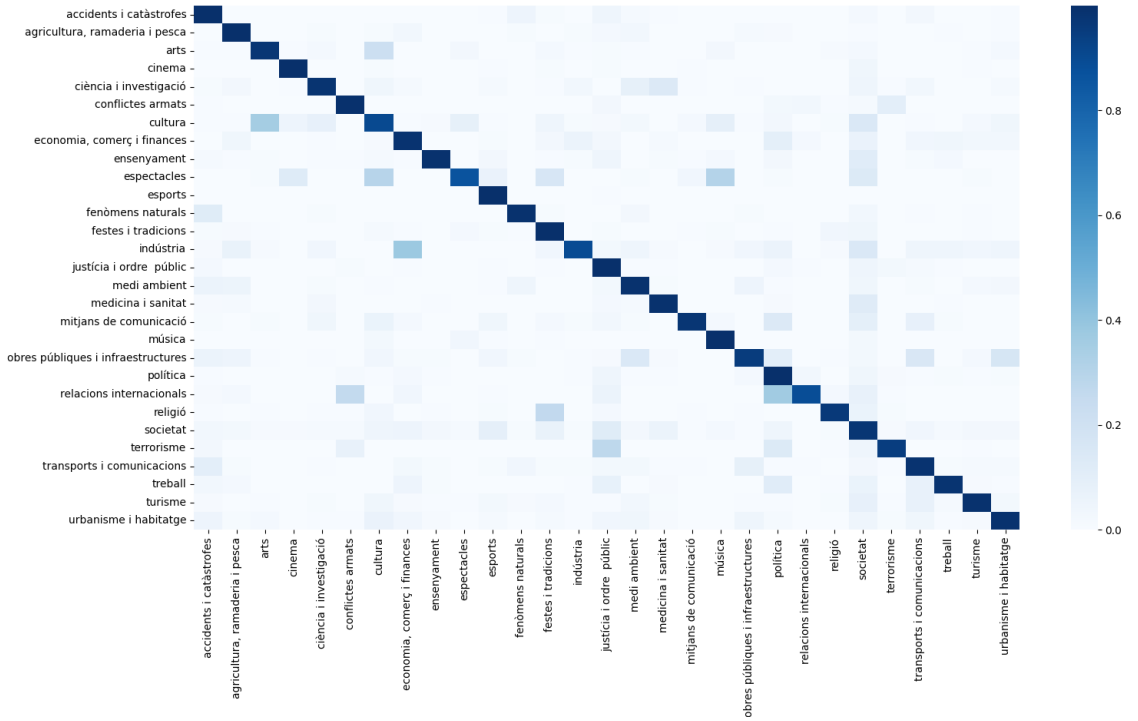

Figura B.11: Matriz de confusión del modelo de À Punt 2019 de 29 clases para el conjunto de test de Canal Nou