Anonimización de Informes Médicos

José Antonio Gaitán Rivas

Máster en Ingeniería y Ciencia de Datos

Málaga, 15 de Septiembre del 2021

Directoras: Raquel Martínez y Lourdes Araujo

Resumen

Con el objetivo de mejorar la salud y seguridad de los pacientes cada vez existe un mayor interés en gestionar eficientemente el contenido de los historiales clínicos electrónicos. Dichos informes médicos están escritos principalmente en lenguaje natural, por lo que contienen información no estructurada generalizadamente, haciéndose imprescindibles tecnologías de Minería de Textos y de PLN (Procesamiento de Lenguaje Natural) para su explotación. Con técnicas apropiadas de dichas tecnologías se ayuda en la toma de decisiones clínicas o se facilita la reutilización de medicamentos, entre muchas otras ventajas.

Sin embargo, los registros clínicos **con información de salud protegida (PHI o *Protected Health Information*)** no pueden ser compartidos directamente debido a restricciones relacionadas con la protección de datos sobre dicha información privada de los pacientes. Es necesaria pues, una anonimización o disociación de dichos registros antes de poder ser usados externamente, debiéndose eliminar total o parcialmente toda información que permita identificar al paciente.

La base del presente trabajo ha sido la tarea de evaluación **MEDDOCAN** (*Medical Document Anonymization*), a la que puede accederse en https://temu.bsc.es/meddocan , que forma parte de la iniciativa **IberLEF 2019**, y con la que se organizó un desafío para la comunidad hispano-hablante, con el objetivo de diseñar sistemas eficientes de anonimización de documentos médicos escritos en español.

La **tarea de MEDDOCAN** se estructura en **dos subtareas**:

- Identificación y clasificación de entidades (nombres de paciente, teléfonos, etc.)
- Detección de texto sensible

La **evaluación** oficial de la tarea, por tanto, engloba los resultados de **ambas** subtareas. El **corpus** está formado por **1.000 estudios de casos clínicos**, y cada uno de ellos cuenta, de forma anexa, con expresiones PHI realizadas por profesionales.

Del total de 1.000 casos, se reservó el 50% (**500 casos**) para **entrenamiento** de la tarea, un 25% (**250 casos**) para labores de desarrollo, y el otro 25% (**250 casos**) para **pruebas**.

En el desafío participaron **18 equipos**, de un total de **8 nacionalidades** distintas, y el mejor resultado, basado en la métrica **F-score**, fue de 0.9360 para la subtarea 1 (*"Identificación y clasificación de entidades")* y de 0.9611 para la subtarea 2 (*"Detección de texto sensible").*

A lo largo del presente trabajo estudiaremos y compararemos los datos proporcionados por los organizadores de la tarea, y propondremos un sistema que implementa una solución simple mediante técnicas de Aprendizaje Automático y Minería de Textos.

Finalmente analizaremos los resultados obtenidos con dicho sistema y serán comparados con los de los participantes en la tarea, exponiendo las ventajas e inconvenientes para la arquitectura escogida, respecto a las presentadas. En dichas conclusiones incorporaremos un listado de posibles mejoras o implementaciones futuras recomendadas para mejorar el rendimiento.

Índice general

Índice de Tablas

Índice de Figuras

1. Introducción

En esta primera sección pasaremos a revisar la motivación que llevó a realizar el presente trabajo, así como las fases y detalles del procedimiento que se ha seguido. Por último analizaremos la meta que perseguimos, que no es otra que diseñar un sistema que reconozca información personal dentro de informes médicos, para hacerla anónima.

1.1 Motivación

Para comprender el alcance de este trabajo, lo primero es conocer el significado del concepto de **Anonimización**. Según la RAE, es el "*proceso mediante el cual se expresa un dato relativo a entidades o personas, eliminando la referencia a su identidad*". En nuestro ámbito vamos a aplicar dicho proceso a informes clínicos, de forma que podamos obtener todo el conocimiento técnico y científico de dichos informes, para su posterior uso diario o para incluirlo en estudios y estadísticas futuras, etc. pero siempre de forma que no se pueda conocer o identificar a la persona objeto de dicho informe.

Como puede leerse en **[1],** la Anonimización se basa en 2 pilares fundamentales: el *enmascaramiento* (**masking**) y la *deidentificación o disociación* (**de-identification**).

El **masking**, u **ofuscación** de datos, consiste en *reemplazar datos sensibles, por otros valores, aparentemente válidos, pero diferentes a los originales*. Reduce el riesgo de poder identificar al sujeto, mediante la aplicación de técnicas de transformación de datos sin tener en cuenta ningún aspecto relativo al análisis de dichos datos.

Por otro lado, la **deidentificación** reduce el riesgo igualmente, aunque consiguiendo que los datos resultantes tengan una alta utilidad analítica. Difiere de la ofuscación en que realmente *el conjunto de datos se anonimiza para prevenir que se pueda identificar personalmente a alguien*, por motivos de protección de datos o privacidad. La ofuscación se aplica normalmente a algunos o todos los elementos del conjunto de datos, pero de forma independiente, sin tener en cuenta ninguna relación entre dichos elementos.

Por ejemplo, técnicas de *sustitución de caracteres aleatorios* sobre campos de texto, o asignación de *rango de valores* a campos como fechas o divisas, e incluso eliminación del

total o parte de ciertos campos, son muy comunes para lograr una anonimización efectiva de los datos.

La aplicación práctica de las técnicas comentadas en el párrafo anterior, son de amplio uso en infinidad de sistemas, especialmente por privacidad de datos personales, y entre dichos sistemas encontramos los de índole médico. La información contenida en expedientes clínicos es habitualmente compartida por profesionales, por lo que sobre dichos historiales (***Protected Health Information, PHI***) es necesario aplicar la adecuada anonimización.

1.2 Procedimiento

En nuestro caso, la tipología de los documentos a anonimizar es bien conocida, aunque el propio uso de dichos documentos médicos requiere aprobaciones legales y que las personas implicadas necesitan dar su consentimiento.

La mayoría de los informes se presentan en **texto plano** (información no estructurada), y requieren una transformación para recabar información útil. Además, asumimos que en ciertos casos encontraremos **errores de formato**, es decir, gracias a algún conocimiento previo o predicciones sobre los valores de algunas entidades, esperamos que dichos valores aparezcan bajo algún patrón bien definido. Sin embargo, al analizar su estructura final observamos casos donde no se cumple dicha regla: a veces por caracteres inesperados (tabulaciones, saltos de línea, etc.), o bien por errores humanos al crear los textos. Por ello, nuestro procedimiento se basará en lenguajes y librerías actualizadas, especializadas en la gestión de este tipo de textos.

Por último, para utilizar un **modelo de aprendizaje automático eficiente**, lo ideal será tener el máximo número posible de **PHI anotados**.

En resumen, nuestro proceso estará basado en fases de análisis y desarrollo, de índole técnico, en el que mediante el uso del lenguaje **Python** (por su gran adaptabilidad al entorno de estudio, así como librerías standard adecuadas para tratar informes médicos) desarrollaremos un sistema capaz de anonimizar dichos textos, aprendiendo de ellos y por último evaluando dicho aprendizaje, basado en la tarea **MEDDOCAN**.

1.3 Alcance

La mayoría de técnicas y algoritmos para anonimizar textos se basan en el procesamiento de textos en inglés. Nosotros gestionaremos dichas herramientas, pero adaptadas y enfocadas en el reconocimiento de informes médicos en **español** contenidos en el corpus que ofrece la tarea **MEDDOCAN**.

1.4 Objetivos

El procesamiento de textos correspondientes a historias clínicas es fundamental para avanzar en numerosas aplicaciones en el dominio de la salud, desde sistemas de ayuda al diagnóstico, prevención, tratamientos individualizados, mejora de la gestión de los sistemas de salud, por nombrar algunas. Como hándicap, dicho procesamiento requiere su previa anonimización o de-identificación, ya que están bajo la ley de protección de datos. La anonimización básicamente consiste en la eliminación exhaustiva, o en la sustitución, de toda la información que facilitaría la identificación del sujeto al que se refiere y relacionarlo con su información clínica.

Sobre *registros médicos electrónicos*, donde en su gran mayoría encontramos textos libres manuales escritos en lenguaje natural, tal y como ocurre sobre la base de esta Memoria, la anonimización se vuelve más compleja. Asumiremos que seguramente existirán algunos errores tipográficos, ortográficos o gramaticales. De forma general, existen tres **enfoques** principales para conseguir una buena anonimización de informes basados en texto libre:

- Basado en **modelos**
- Basado en **reglas**
- **Híbrido**, combinación de los dos anteriores

Un enfoque basado en **modelos** usará algún modelo de Machine Learning o estadístico desde un conjunto de datos de entrenamiento, y aplicará dicho modelo a nuevos datos. En el enfoque basado en **reglas** el sistema de anonimización aplica un conjunto predefinido de criterios y funciones directamente en los datos, *sin necesidad de entrenamiento*.

Respecto a las tareas de entrenamiento en *sistemas de anonimización basados en modelos,* el **contexto** es muy importante: no será lo mismo el training sobre un tipo de documento del área de Patología, que otro del área de Radiología, por ejemplo.

Para conseguir un buen sistema hacen falta buenos conjuntos de datos de entrenamiento locales, y conseguir manualmente **anotaciones** sobre los principales campos. El *propósito* de un sistema de anonimización de texto libre es, primero extraer la información personal contenida en dicho texto, y luego anonimizarla mediante algunas técnicas de ofuscación y deidentificación.

Finalmente la *evaluación* de dicho sistema, mediante métricas estándard, será la clave para determinar su mayor o menor índice de acierto en anonimizar los documentos. El rendimiento se basará en el número de documentos que presenten alguna *pérdida de información*, y no en el número de campos reconocidos. Esto es mucho más estricto que el criterio general seguido en sistemas de anonimización de textos "no médicos", pero es **consistente con las regulaciones en privacidad**. El conjunto de datos necesita tener un *bajo riesgo de re-identificación*.

Entre nuestros objetivos principales se encuentra la propuesta y diseño de **un sistema de anonimización de informes médicos, basado en aprendizaje automático supervisado híbrido, a partir de CRF y redes neuronales**. Dicho sistema se basará en el marco teórico y experimental presentado en la tarea MEDDOCAN. Además evaluaremos el rendimiento de nuestro sistema, comparándolo con los sistemas participantes en dicha tarea.

1.5 Estructura de la Memoria

La memoria del presente trabajo incluye:

- Introducción de **conceptos previos**, básicos para comprender nuestro proyecto
- Análisis del **estado del arte** (propuestas de varios autores)
- Detalle de las **tecnologías** aplicadas
- Presentación del **sistema** de anonimización construido
- **Evaluación** de dicho sistema
- **Conclusiones** y posibles **trabajos futuros** para mejorar el sistema

2. Conceptos preliminares

Una vez presentado en el capítulo anterior el alcance de nuestro trabajo, así como la base en la que se sustenta, vamos a continuación a describir las definiciones y métodos usados durante el proyecto.

La asimilación de estos **conceptos previos** es importante para que la lectura del trabajo, incluyendo las comparativas y conclusiones, sea fructífera.

2.1 Procesamiento del Lenguaje Natural (PLN)

PLN es el área de la Inteligencia Artificial que permite que ordenadores puedan entender el lenguaje humano y realizar tareas complejas sobre distintos objetos lingüísticos (palabras, frases, significados, etc.) Para conseguirlo, se crean sistemas inteligentes capaces de entender, analizar y extraer significado desde un texto escrito o hablado.

Dichos sistemas siguen habitualmente un **pipeline** de varios **niveles**, incluyendo:

- **Morfología** (cómo las palabras se construyen a partir de sus **morfemas**, o unidades más pequeñas con significado propio).
- **Sintaxis** (determina la estructura y roles que conectan las palabras en una frase, basada en una **gramática**, definida por reglas sintácticas del lenguaje).
- **Semántica** (determina el **significado** literal de cada palabra, identificando las interacciones y resolviendo cualquier tipo de ambigüedad).

2.2 Extracción de información mediante PLN

A la hora de afrontar el reto de extraer eficientemente información valiosa desde textos escritos en lenguaje natural, varios **métodos** pueden ser aplicados según la naturaleza de los textos y nuestro conocimiento previo sobre ellos:

- Basados en **Reglas** (codificando manualmente reglas sintácticas y semánticas en base a experiencia humana).
- Basados en **Aprendizaje Automático Supervisado** (un modelo es entrenado con información etiquetada sobre las entidades).
- Basados en **Aprendizaje Automático Semi-supervisado** (cuando no tenemos suficientes datos etiquetados, usaremos un conjunto "semilla" para, partiendo de él, el modelo pueda ser entrenado).

2.3 POS-Tagging

Es el proceso que consiste en **asignar, o etiquetar, a cada palabra de un texto su categoría gramatical** correspondiente. Sin embargo conseguir esto no es algo trivial, ya que una misma palabra puede tener distintas categorías en función del contexto, dando lugar a ambigüedades que deben ser resueltas. Por ejemplo la palabra "*dado*", puede corresponder al verbo "*dar*" o al *nombre singular*.

2.4 Reconocimiento de Entidades Nombradas (NER)

En todo texto escrito en lenguaje natural podemos localizar "**Entidades Nombradas**", que consisten en **categorías predefinidas** como nombres de personas, organizaciones, localizaciones, cantidades, tiempo, valores monetarios, etc.

Sin embargo bajo algunas circunstancias, un mismo "valor" puede ser considerado en una categoría u otro, dependiendo del **contexto**. Se hace necesaria una clasificación de los **roles** de dichas entidades, dentro de una oración, párrafo o incluso considerando todo el texto.

Esa clasificación puede seguir enfoques diversos:

- Basados en **Reglas** (el tipo de entidad nombrada es asignada siguiendo un conocimiento humano previo sobre la secuencia de ciertas palabras o etiquetas).
- Basados en **Métodos Probabilísticos** (definen la probabilidad de que una cierta etiqueta aparezca siguiendo una determinada secuencia, asignándose el tipo de

entidad que mejor se ajuste). Por ejemplo, **CRF** (Conditional Random Fields) es una técnica que sigue esta metodología, y será una de las que apliquemos.

- Basados en **Métodos de Aprendizaje Automático Supervisado** (predicen secuencias de etiquetas que forman una entidad). Un ejemplo es **RNN** (Redes Neuronales Recurrentes), que incluiremos en nuestro trabajo.

2.5 Evaluación de tareas NER

En todo sistema que use técnicas de extracción de información basadas en PLN, nos interesará normalmente conocer su **rendimiento** (cómo de bien predice los resultados). Para ello evaluaremos la calidad de la salida producida por dicha tarea.

El procedimiento habitual será primero **separar el corpus** (conjunto de textos) en **dos** partes: entrenamiento ("**training**") y pruebas ("**test**"). Una proporción con el 75% del total de los textos como training, y el resto 25% para tests, suele ser común, aunque dicho porcentaje dependerá de la naturaleza del corpus y de los resultados obtenidos. Lo importante es **entrenar al modelo** con el conjunto de **training**, conteniendo valores ya etiquetados previamente, y a continuación ejecutar las **predicciones** sobre el conjunto de **test**, y evaluar el **porcentaje** de acierto obtenido.

Nuestra base será la **matriz de confusión**, donde los elementos de la **diagonal** denotarán las **predicciones correctas**, y está definida como:

		Clase Real	
		SÍ	**NO**
Clase Predicha	**SÍ**	TP	FP
	NO	FN	TN

Figura 1: Matriz de confusión

TP (True Positives): nº muestras clasificadas **correctamente** como **positivas**

FP (False Positives): nº muestras clasificadas **incorrectamente** como **positivas**

TN (True Negatives): nº muestras clasificadas **correctamente** como **negativas**

FN (False Negatives): nº muestras clasificadas **incorrectamente** como **negativas**

A partir de esta matriz podemos calcular diversas **métricas de evaluación** a la tarea:

- **Accuracy**: *"¿Qué tan bien clasificó correctamente el modelo?"*

  Accuracy = (TP + TN) / (TP + TN + FP + FN)

Esta métrica es útil en problemas de categorización donde las clases están bien balanceadas.

- **Precision**: *"Proporción de muestras clasificadas positivas fueron correctas"*

  Precision = TP / (TP + FP)

Métrica a usar cuando necesitemos estar muy seguro de las predicciones.

- **Recall**: *"Proporción de muestras positivas reales clasificadas correctamente"*

  Recall = TP / (TP + FN)

Útil para capturar tantos positivos como sea posible.

Y tomando como base las anteriores métricas podemos calcular *"F_1-Score"*:

F_1-Score = 2 (Precision x Recall) / (Precision + Recall)

F_1-Score es la media armónica de "Precision" y "Recall", y tiene un valor máximo de 1.0 y mínimo de 0.0. A mayor valor, indicará un mejor rendimiento en la tarea.

2.6 Algoritmos de clasificación

Podemos entender en nuestro contexto, "clasificación" como la predicción de patrones, junto al posterior reconocimiento y asignación a una determinada **clase**. Para dicho proceso de reconocimiento existen como vimos diversas técnicas basadas en reglas, aprendizaje automático supervisado, etc. Y sobre estas técnicas encontramos distintos tipos de algoritmos. El resultado, no obstante, siempre será una **etiqueta discreta**, es decir, nuestro problema siempre recaerá en un conjunto **finito** de resultados posibles. Genéricamente una clasificación puede ser **binaria** (solo 2 clases) o **multicategoría** (más de 2 clases).

Existen algoritmos de clasificación para aprendizaje **supervisado** (conocemos a priori el número de clases y todas ellas están bien etiquetadas), **no supervisado** (datos no etiquetados que se intentan agrupar tras la aplicación del algoritmo) y **semisupervisado** (usamos datos de entrenamiento etiquetados y no etiquetados).

Cada uno de esos algoritmos usa uno o varios de los **modelos** que en el siguiente apartado vamos a enumerar brevemente, sirviendo así dicho apartado como "referencia" para secciones posteriores de esta Memoria. Además de los modelos comentados también agregaremos una breve presentación de algunos de los algoritmos de aprendizaje aplicados para la tarea MEDDOCAN por parte de algunos participantes.

2.7 Descripciones (modelos, procesos y algoritmos)

2.7.1 Conditional Random Field (CRF)

Es un modelo **discriminativo,** idóneo para tareas de reconocimiento de entidades nombradas, y en general para tareas de clasificación o regresión. El modelo asigna una etiqueta $\mathbf{S_i}$ a cada elemento $\mathbf{O_i}$ dada una secuencia de datos $\mathbf{O_1}$ …$\mathbf{O_N}$ modelando la probabilidad de la secuencia correcta de etiquetas condicionada por las observaciones, $\mathbf{P(S|O)}$ de ahí que sean discriminativos, y no generativos.

Se puede representar con un grafo no dirigido **G = (V, E)** en el que cada vértice represente una variable aleatoria cuya distribución de probabilidad debe ser deducida, y cada arista indique una dependencia entre las variables de los vértices que conecta.

El modelo necesita ser entrenado con **N** muestras, conteniendo cada una el conjunto de observaciones así como las etiquetas asociadas a esas observaciones. Se extraerá un conjunto de características, representando las dependencias entre diferentes estados y las observaciones. Durante el entrenamiento el modelo asignará un peso a cada característica, indicando su mayor o menor importancia.

2.7.2 Modelo oculto de Márkov (HMM)

Es un modelo **estadístico,** por tanto basado en probabilidades, y especificado mediante un conjunto de ecuaciones que relacionan diversas variables aleatorias, y en las que pueden aparecer otras variables no aleatorias.

El objetivo del modelo será **determinar los parámetros desconocidos** (u ocultos, de ahí el nombre) **a partir de los parámetros observables**. Cada estado tendrá una distribución de probabilidad sobre los posibles símbolos de salida.

Este modelo es ampliamente usado en el reconocimiento de la escritura manual y del habla, en procesos de etiquetado gramatical y en bioinformática.

2.7.3 Máquinas de vectores de soporte (SVM)

Son un conjunto de algoritmos de **aprendizaje supervisado**, diseñados para resolver problemas de clasificación y regresión. Identifican la clase a la que pertenece un valor mediante vectores de características. La selección de las correctas características (de entre el conjunto de atributos) para el contexto correspondiente, representará un mayor o menor éxito en la clasificación.

Dado un conjunto de entrenamiento, se etiquetan las clases y se entrena una SVM para construir un modelo predictivo para nuevas muestras. Los datos siempre son mapeados buscando la máxima separación entre las clases. El vector entre los dos puntos más cercanos de cada par de clases se denomina **vector soporte**.

2.7.4 Árboles de decisión

Son **modelos de predicción** que, dado un conjunto de datos, construyen y categorizan una serie de condiciones que ocurren de forma sucesiva, similares a los sistemas basados en reglas.

Están formados por **nodos, vectores de números, flechas y etiquetas.**

- *Nodo*: momento en el que se ha de tomar una decisión
- *Vector de número*: solución final en función de las diversas posibilidades
- *Flecha*: unión entre nodos, representando cada acción distinta posible
- *Etiqueta*: se usa en nodos y flechas para su identificación

Como norma, solo existirá un **único nodo inicial**, no apuntado por ninguna flecha, mientras que el resto de nodos son apuntados por una única flecha. Esto implica que las **decisiones son todas excluyentes entre sí**, y solo habrá un único camino desde el nodo inicial a cualquier solución final.

2.7.5 Boosting

Es un algoritmo de **aprendizaje supervisado**, cuyo principal objetivo es **convertir uno o varios clasificadores débiles en uno robusto**, consiguiendo un mejor rendimiento en la asignación de los valores a las clases correctas.

Durante cada iteración el algoritmo escoge un clasificador de una sola característica, respetando la medida del error conjunto, que debe ser previamente definida. Así los detectores de cada categoría se pueden entrenar conjuntamente. Los clasificadores son escogidos en función de la exactitud de sus predicciones, de modo que presenten diferente peso.

2.7.6 Red Neuronal Recurrente (RNN)

En una red neuronal **clásica** los vectores de entrada producen vectores de salida, pero **entre ejecuciones no se mezcla esa información**. En RNN las **salidas anteriores sirven como entrada** en posteriores ejecuciones. Cada entrada se multiplica por pesos aleatorios inicialmente, y sesgo, transformados mediante funciones de activación. A mayor secuencia temporal a analizar, mayor será el número de capas necesarias para la RNN.

2.7.7 Long short-term memory (LSTM)

Arquitectura RNN con conexiones hacia delante, pero también hacia atrás. Puede procesar secuencias completas de datos. Una **unidad** LSTM común se compone de una **celda,** una **puerta de entrada,** una **puerta de salida y** una **puerta de olvido**.

La *celda* actúa como memoria, recuerda los valores sobre intervalos de tiempo arbitrarios, mientras que las *puertas* regulan el flujo de información desde y hacia la celda.

La *puerta de entrada* ayuda a actualizar el estado de las celdas. La información de la entrada actual y del estado previo pasa por las funciones *sigmoide* y *tanh*, regulándose la red.

La *puerta de salida* será donde se determine cuánta información del estado interno actual se enviará al estado externo.

La *puerta de olvido* es quien decide mantener u olvidar la información. Únicamente la entrada actual e información procedente de capas previamente ocultas se mantiene con la función sigmoide. Valores cercanos a 1 permanecen, y cercanos a 0 desaparecerán.

Las redes LSTM son adecuadas para tareas de clasificación sobre datos en series temporales, y además resuelven el problema de desvanecimiento de gradiente (común en RNN tradicionales)

2.7.8 Gated Recurrent Unit (GRU)

Versión mejorada de la arquitectura RNN, introducida en 2014 por Kyunghyun Cho. Similar a LSTM, sin embargo necesita menos parámetros para su configuración.

GRU sólo tiene un *estado oculto* transferido en el tiempo, mientras que LSTM mantiene dicho estado oculto pero también el *estado de la celda*.

Además, la estructura de puertas en GRU es más simple, solo existen dos tipos: **puerta de actualización** (fusión de las puertas de entrada y olvido) y **puerta de reinicio**.

La *puerta de actualización* es la encargada de controlar los estados históricos para la salida actual.

La *puerta de reinicio* determina si el estado candidato en el momento actual dependerá del estado de la red en el momento anterior y en qué medida.

3. Estado del arte

En este capítulo presentaremos las técnicas generales base de anonimización, y a continuación realizaremos un recorrido histórico de los sistemas propuestos en el ámbito de nuestro trabajo, junto con las tareas actuales en las que se basa nuestro estudio.

3.1 Anonimización: datos y técnicas

Para anonimizar datos personales se necesita detectar los atributos que, en nuestro **contexto de datos**, representen *identificadores* de una persona.

En general podemos clasificar los campos en:

- **Identificadores** (identifican unívocamente a una persona: DNI, etc.)
- **Cuasi-identificadores** (no son identificadores, pero combinados con otros campos sí pueden desvelar la identidad de una persona: fecha nacimiento, edad, etc.)
- **No identificadores** (ni por sí solos ni combinados identifican a una persona)

Además de la clasificación anterior, hay un tipo de dato más a considerar, y son los **datos sensibles**, que no se pueden tratar sin el consentimiento del individuo, ya que tienen relación con sus derechos y libertades. Los datos de **salud** corresponden a esta categoría.

Respecto a las **técnicas** de anonimización más usadas encontramos:

- **Seudonimización** (la información personal es tratada, excluyendo los datos identificativos, pero sin suprimir totalmente la *vinculación* entre dichos datos)
- **Generalización** (reemplaza datos por otros menos específicos)
- **Aleatorización** (se aplica una modificación aleatoria, por ejemplo usando permutaciones o introduciendo ruido en los datos)
- **Eliminación** (eliminamos atributos y/o registros no necesarios, siempre y cuando no presente su borrado un riesgo de re-identificación)

3.2 Propuestas de Sistemas

A la hora de estudiar distintas propuestas realizadas con anterioridad por autores relacionadas con el alcance de nuestro proyecto, como ya expusimos, encontramos que la mayoría detectan y procesan textos con información sensible, en inglés.

Prácticamente la mayoría se basan en **NER** (*Named Entity Recognition*), es decir "**Reconocimiento de Entidades Nombradas**", ya que en el proceso a aplicar sobre nuestros informes médicos obviamente lo primero será tratar de identificar personas, organizaciones y localizaciones.

Estos sistemas suelen estar basados en **modelos** que usan:

- Diccionarios y reglas (expresiones regulares)
- Machine Learning (ML), o aprendizaje automático
- Híbridos (reglas y aprendizaje automático)

3.3 Sistemas basados en diccionarios y reglas

Históricamente los primeros sistemas diseñados, como ***Scrub*** [2], manejaban información sensible basándose en expresiones regulares y diccionarios. *Scrub* usaba múltiples algoritmos para detectar las clases, y luego diccionarios para la clasificación de las entidades. Este método era bastante fiable, detectando la gran mayoría de entidades en informes médicos, aunque muy costoso de mantener.

Otro sistema basado en diccionarios fue ***Concept-Match*** [3]. Primeramente eliminaba todas las palabras que no aportaban información práctica (*stopwords*) para finalmente buscar las entidades en vocabularios biomédicos y en Metatesauros, como el UMLS (libro nacional de medicina de EEUU). Desafortunadamente este sistema demostró ser poco fiable, con muchos casos de falsos positivos.

3.4 Sistemas basados en Machine Learning

El **aprendizaje automático (Machine Learning)** es una disciplina científica del ámbito de la Inteligencia Artificial. Se crean sistemas que aprenden autónomamente, identificando patrones complejos, mediante un algoritmo que revisa los datos y es capaz de predecir comportamientos futuros. Estos sistemas mejoran sus resultados con el tiempo, sin intervención humana. Veamos a continuación algunos de estos sistemas, que aplican principalmente métodos NER y clasificadores.

El sistema ***MITRE*** [4] usaba dos clasificadores: uno basado en **CRF** y otro en **HMM**. Al final del algoritmo se usan expresiones regulares.

El sistema presentado en [5] usaba **Boosting** y **árboles de decisión**. No se usó **POS-tagging**. Mediante tres clasificadores diferentes se analizaban los datos y si al menos dos de dichos clasificadores votaban positivamente, entonces se aceptaba el token.

Otro sistema basado en CRF fue [6], y era capaz de detectar entidades mediante el uso de diferentes categorías y clasificaciones (morfológicas, sintácticas e incluso diccionarios externos). Entidades tipo fecha, identificación o nombres, eran muy bien identificadas mediante características de la sentencia.

3.5 Sistemas Híbridos

Los mejores resultados, en general, se obtienen **combinando** sistemas basados en reglas con modelos de aprendizaje automático, pues se abarcan más categorías de PHI distintas. Por ejemplo en [7] el sistema fue basado en clasificadores **SVM** (Máquinas de Vector Soporte) y expresiones regulares.

El sistema ***HIDE*** descrito en [8] usaba un clasificador CRF como primer componente, a continuación las entidades se enlazaban y finalmente se anonimizaban los datos.

Otro sistema basado en un clasificador CRF fue [9], el cual, además de usar diccionarios, expresiones regulares y un procesamiento final de ajuste, iba obteniendo "PHI de confianza", en cada iteración.

3.6 Trabajos de la tarea MEDDOCAN

La importancia práctica de anonimizar textos clínicos motivó en el pasado la propuesta de diversas tareas por parte de la **Comunidad i2b2** (*i2b2.org*), aunque dichas tareas inicialmente estaban enfocadas todas al idioma inglés. Sin embargo, años después, el creciente interés en la anonimización de estos textos en otros idiomas (francés, alemán, holandés, etc.) provocó la aparición de sistemas orientados también en español, y con ello la primera tarea planteada como reto a la comunidad íntegramente en castellano: **MEDDOCAN**.

El objetivo de la tarea era **evaluar** el rendimiento de los sistemas propuestos para **identificar y clasificar información sensible** en casos clínicos escritos en español. Por tanto la tarea fue dividida en dos subtareas complementarias:

1. *Identificación y clasificación de entidades* (nombres de paciente, teléfonos, etc.)
2. *Detección de texto sensible*
    a. *Estándar*: identifica y enmascara datos confidenciales, individualmente.
    b. *Mejorado con uso de fusiones:* versión avanzada del Estándar, combinando elementos y uniéndolos por categorías, mejorando la detección,

El corpus, compuesto por 1.000 informes clínicos escritos en lenguaje natural, con anotaciones incorporadas por expertos, contenía **29 tipos de entidades** diferentes. Participaron **18 equipos**, de 8 países distintos, que aportaron 63 sistemas diferentes para la subtarea 1, y 61 sistemas diferentes para la subtarea 2.

De los sistemas que obtuvieron mejor rendimiento en la tarea, podemos destacar:

- **NLNDE (Neither-Language-Nor-Domain-Experts)** [15]

Usa una combinación de diferentes enfoques del estado del arte, entrenando RNN con capas de salida usando CRF. La salida del modelo es ajustada finalmente con una capa adicional, manejando y sobrescribiendo algunas expresiones regulares bien conocidas, como URLs, etc.

- **ReCRF** [16]

Sistema híbrido que combina CRF y reglas creadas automáticamente a partir de los datos. Inicialmente se tokeniza el texto usando expresiones regulares manuales y un modelo pre-entrenado en SpaCy. A continuación el generador de reglas analiza las ocurrencias de todas las categorías PHI y crea las reglas correspondientes, que serán más tarde usadas por CRF.

- **CRF + XGBoost + (CNN+LSTM)** [17]

Sistema híbrido basado en la combinación de salidas no secuenciales para múltiples clasificadores XGBoost ("*eXtreme Gradient Boost*"). Este algoritmo se ha demostrado como una de las mejores opciones en problemas supervisados, y además destaca porque es paralelizable, permitiendo así realizar entrenamiento de conjuntos de datos gigantescos en mucho menor tiempo. La primera parte del sistema global usa varios modelos XGBoost entrenados con diferentes esquemas de etiquetado, y finalmente un clasificador CRF obtiene los resultados definitivos.

Además, en un sistema complementario, se usaron CNN ("*Convolutional Neural Networks*") o redes convolucionales, y LSTM ("*Long Short-Term Memories*"), de forma que las capas de la red neuronal sirvieron para representar la secuencia de caracteres, y la capa bidireccional LSTM para la secuencia de palabras. Finalmente una capa adicional de salida mediante CRF obtenía los resultados. Globalmente se usó un sistema de votación ("*voting*") para escoger el mejor resultado de entre los sub-sistemas.

- **Deep Learning (BERT+CRF, Flair)** [18]

Este sistema consta de dos sub-sistemas independientes, cada uno basado en un método Deep Learning: *BERT+CRF* y *Flair*. El primero agrega una capa basada en CRF a BERT ("*Bidirectional Encoder Representations from Transformers*"), el cual, por definición, realiza los entrenamientos de forma bidireccional sobre el texto, usando *Transformer*, un mecanismo que lee toda una sentencia a la vez, no secuencialmente de izquierda a derecha o viceversa. Así mejora sustancialmente, por ejemplo, el aprendizaje del contexto de cada palabra. Finalmente, *Flair* representa un método de etiquetado secuencial basado en la gestión de cadenas por contextos. El mejor resultado fue obtenido con BERT+CRF.

Con respecto a la evaluación final de los trabajos, en la siguiente [Tabla 1] encontramos los resultados para la **subtarea 1 (por etiquetas):**

Category	Sub-category	Best Team(s)	Leak	Precision	Recall	F1
AGE	EDAD_SUJETO_ASISTENCIA	jiangdehuan	0.0004	0.9828	0.9942	0.9885
CONTACT	CORREO_ELECTRONICO	lukas.lange nperez	0.0001	0.9920	0.9960	0.9940
	NUMERO_FAX	jimblair jiangdehuan lsi_uned	0.0000	1.0000	1.0000	1.0000
	NUMERO_TELEFONO	jiangdehuan	0.0000	1.0000	1.0000	1.0000
DATE	FECHAS	jiangdehuan lukas.lange	0.0004	0.9935	0.9951	0.9943
ID	ID_ASEGURAMIENTO	FSL jiangdehuan jimblair lsi_uned lukas.lange m.domrachev mhjabreel nperez sohrab	0.0001	1.0000	0.9950	0.9975
	ID_CONTACTO_ASISTENCIAL	lsi2_uned lukas.lange mhjabreel nperez sohrab vcotik	0.0000	1.0000	1.0000	1.0000
	ID_SUJETO_ASISTENCIA	jiangdehuan	0.0001	0.9758	0.9965	0.9860
	ID_TITULACION_PERSONAL_SANITARIO	jiangdehuan jimblair lsi_uned lsi2_uned lukas.lange mhjabreel nperez sohrab	0.0000	0.9957	1.0000	0.9979
LOCATION	CALLE	lukas.lange	0.0031	0.9353	0.9443	0.9398
	CENTRO_SALUD	FSL jiangdehuan lsi2_uned lukas.lange mhjabreel	0.0001	1.0000	0.8333	0.9091
	HOSPITAL	FSL	0.0016	0.9672	0.9077	0.9365
	INSTITUCION	jiangdehuan	0.0036	0.6061	0.5970	0.6015
	PAIS	jiangdehuan	0.0004	0.9890	0.9917	0.9904
	TERRITORIO	lukas.lange	0.0035	0.9759	0.9728	0.9743
NAME	NOMBRE_PERSONAL_SANITARIO	lukas.lange	0.0003	0.9960	0.9960	0.9960
	NOMBRE_SUJETO_ASISTENCIA	jiangdehuan	0.0000	1.0000	1.0000	1.0000
OTHER	FAMILIARES_SUJETO_ASISTENCIA	lukas.lange	0.0017	0.8293	0.8395	0.8344
	OTROS_SUJETO_ASISTENCIA	nperez	0.0008	1.0000	0.1429	0.2500
	SEXO_SUJETO_ASISTENCIA	FSL	0.0004	0.9892	0.9935	0.9913
PROFESSION	PROFESION	lukas.lange	0.0004	1.0000	0.6667	0.8000

Tabla 1: Resultados subtarea 1 MEDDOCAN (por etiquetas)

Más adelante, en esta Memoria, mostraremos los resultados finales de cada subtarea para los mejores equipos, aunque la información de la tabla anterior es ya significativa de los PHI que podremos, con mayor o menor acierto, ser capaces de identificar.

3.7 Conclusión

CRF (**Conditional Random Field**), es un modelo estocástico usado para etiquetar y segmentar secuencias de datos o extraer información de documentos. Tanto el conjunto de variables de entrada y salida son aleatorias, y el problema de predicción del campo aleatorio condicional se adapta perfectamente a nuestra tarea. Además hemos constatado que una gran mayoría de participantes en MEDDOCAN basaron sus sistemas en este modelo.

Por ello finalmente, dados los resultados de los sistemas estudiados, y sobre qué métodos y modelos se apoyaron, **para el presente trabajo se decidió implementar inicialmente un sistema híbrido, apoyado en un clasificador CRF [13],** similar al último sistema comentado, de Yang y Garibaldi. Posteriormente se entrenó un nuevo modelo basado en **RNN**, que usará CRF en sus capas de salida.

4. Tecnologías empleadas

Como adelantamos en secciones previas, usaremos el lenguaje **Python** (en concreto versión **3.7.4**), por su simplicidad y porque realmente se adapta muy bien al ámbito de nuestro trabajo, contando con librerías suficientemente reconocidas para el PLN (procesamiento de lenguaje natural) y para entrenar CRFs. Entre dichas librerías podemos destacar **nltk** [10] y **spaCy** [11].

Ambas librerías cuentan con métodos para detectar y separar frases o palabras, saltar stopwords, etiquetar palabras gramaticalmente, etc. aunque **spaCy** es algo más sencilla de usar y por ello la elegimos. Además parece más eficiente calculando las **Entidades Nombradas**.

Respecto al modelo de aprendizaje automático basado en CRFs, tenemos a nuestra disposición la librería **CRFsuite** [12] la cual nos ayudará en la evaluación de los modelos de entrenamiento, facilitándonos el análisis de los resultados. Por último, usaremos **Keras** como librería API de redes neuronales, basado en **Tensorflow** (versión **2.6.0**).

5. Propuesta de Sistema

Veamos a continuación las características del sistema desarrollado para este trabajo, en qué se basa, detalles del corpus MEDDOCAN usado para los datos, junto con un análisis de la estructura. Por último estudiaremos el módulo responsable de procesar la información, y el algoritmo base empleado.

5.1 Arquitectura del Sistema

Nuestro sistema usará como algoritmo de aprendizaje automático supervisado CRF. Además, al ser híbrido, combinará el uso de reglas con Machine Learning. A alto nivel, el sistema cuenta con las siguientes partes:

- ***Pre-procesamiento***
- ***Detección de PHI***
- ***Evaluación***

En la etapa de **pre-procesamiento** se reciben los informes de la tarea **MEDDOCAN**, y se separa la información según la estructura (**cabecera, cuerpo y pie**) de los documentos del corpus. Dichos documentos presentan una estructura común, aunque no todos tendrán necesariamente los mismos campos en cada parte de la estructura. Finalmente, mediante técnicas de procesamiento de textos la información es gestionada para producir la entrada al siguiente nivel: las sentencias se **tokenizan** (se eliminan saltos de línea y caracteres especiales, para dividir cada sentencia en palabras únicamente), a continuación se normaliza para obtener un **diccionario** de esos tokens, y por último eliminamos las palabras vacías (**stopwords**) que carecen de significado.

La segunda parte del sistema se encarga de **detectar PHI**. Obviamente como cada apartado del documento tiene una estructura bien definida, pero diferente a las otras, se aplicaron diferentes maneras para dicha detección. En la ***cabecera*** sí se pueden extraer directamente los PHI, pero en el ***cuerpo*** fue necesario entrenar un modelo CRF con las categorías como base. Respecto al ***pie***, no funcionó bien ningún modelo CRF así como la extracción mediante diccionarios y reglas, por lo que se usaron **PHI de confianza**.

Por último, respecto a la ***evaluación***, ésta se ha realizado de forma **parcial** (cada parte de los módulos anteriores) y **completa** (todo el sistema). Los mejores resultados, como era de esperar, se dieron en la **cabecera**, por su mejor definición estructural, mientras que en el **cuerpo** y en el **pie** los resultados fueron más modestos.

5.2 Corpus MEDDOCAN

En la tarea **MEDDOCAN** [(*)] se puede localizar un corpus compuesto de **1.000 informes médicos en español,** al que se le han agregado expresiones PHI, realizadas por profesionales.

En total, el corpus cuenta con casi 500.000 palabras, y alrededor de 33.000 oraciones. Se distribuye en **texto plano** (**UTF-8**), y existe un fichero independiente para cada informe. Las anotaciones PHI se publican en formato **BRAT**, como se observa en la siguiente Figura:

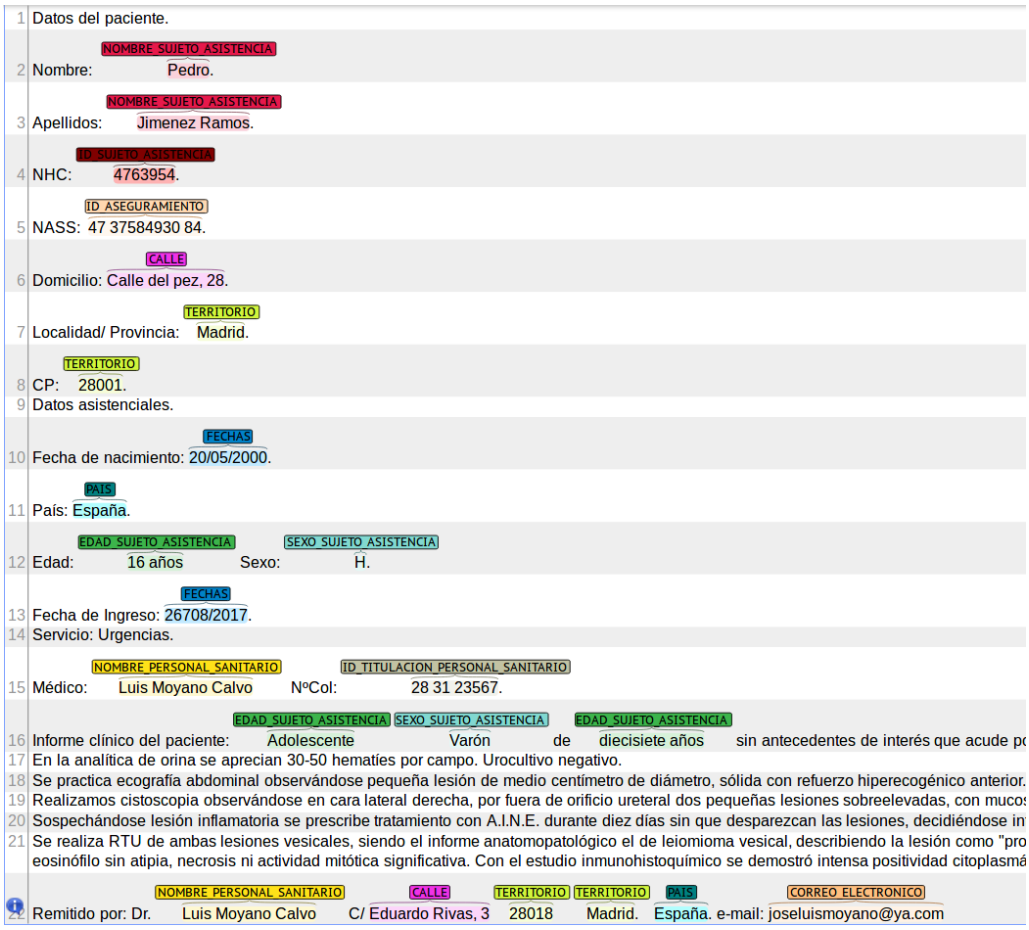

Figura 2: Ejemplo de anotación en formato BRAT

(*) Tarea MEDDOCAN: https://temu.bsc.es/meddocan

Cada documento presenta una **única plantilla compuesta por tres partes**, claramente diferenciadas: **cabecera, cuerpo y pie**, donde cada una de ellas muestra un listado de campos y valores diferentes y específicos, por lo que será necesario un completo análisis.

Respecto a las **categorías PHI**, encontramos **28** (nombres, números identificativos, municipios, etc.). En el Anexo I puede consultarse el listado completo.

5.3 Análisis **previo**

Es el momento de analizar las tres partes de cada informe médico de la tarea **MEDDOCAN**. Así podremos adaptar nuestro sistema de la mejor manera, aprovechando en lo posible la estructura bien conocida, y donde aparecen las PHI mayoritariamente.

5.3.1 Cabecera

Empíricamente se constata que es la parte del documento con una organización mejor definida respecto a los campos que se muestran. También el conjunto de PHI puede obtenerse fácilmente y de forma segura. Aquí un ejemplo:

```
Nombre:  Ignacio.
Apellidos: Rico Pedroza.
NHC: 5467980.
Domicilio: Av. Beniarda, 13.
Localidad/ Provincia: Valencia.
CP: 46271.
Datos asistenciales.
Fecha de nacimiento: 11/02/1970.
País: España.
Edad: 46 años Sexo: H.
Fecha de Ingreso: 28/05/2016.
Médico:  Ignacio Rubio Tortosa Servicio  N°Col: 46 28 52938.
_____________________________________________________________

T2  ID_TITULACION_PERSONAL_SANITARIO 318 329    46 28 52938
T10 NOMBRE_PERSONAL_SANITARIO 279 300   Ignacio Rubio Tortosa
T11 FECHAS 258 268  28/05/2016
T12 SEXO_SUJETO_ASISTENCIA 237 238  H
T13 EDAD_SUJETO_ASISTENCIA 223 230  46 años
T14 PAIS 209 215    España
T15 FECHAS 191 201  11/02/1970
T16 TERRITORIO 142 147  46271
T17 TERRITORIO 128 136  Valencia
T18 CALLE 88 104    Av. Beniarda, 13
T19 ID_SUJETO_ASISTENCIA 68 75  5467980
T20 NOMBRE_SUJETO_ASISTENCIA 49 61  Rico Pedroza
T21 NOMBRE_SUJETO_ASISTENCIA 29 36  Ignacio
```

Figura 3: Ejemplo de Cabecera (con anotaciones)

Junto con cada "título de línea" (Nombre, Apellidos, NHC, etc.), aparece separado por **dos puntos (:),** el valor correspondiente. Además, se observan un total de **25 indicadores distintos**, los cuales salvo contadas excepciones (posibles erratas), aparecen siempre unos detrás de otros en distintos documentos, siempre en el mismo orden.

5.3.2 Cuerpo

En el análisis realizado al cuerpo, sobre todos los informes, encontramos **18 PHI diferentes**, para un total de 2.891, entre los 475 documentos leídos correctamente para el análisis. Resultados demasiado modestos y que no aseguran que vayamos a encontrar PHI en el futuro de más documentos. Además, obviamente, no tenemos un esquema definido.

Por todo, se decide usar un **modelo de aprendizaje supervisado con CRFs** en esta parte del documento, pero además para que funcione correctamente, requeriremos un vector con las características de las palabras, siendo relevantes únicamente las oraciones que posean PHI.

Como sabemos, los **Campos Aleatorios Condicionales** (Conditional Random Fields, CRFs) son **distribuciones condicionales P(Y|X)** que tienen asociada una estructura de grafo, siendo en nuestro caso, "Y" quien representa la etiqueta asociada a cada palabra (**PHI / no-PHI**), mientras que la variable "X" es el vector con las características asociadas a una palabra.

5.3.3 Pie

Empíricamente se constata que existe en esta parte del documento normalmente información asociada al nombre del médico, centro hospitalario, dirección, teléfono, mail, etc.

Sin embargo el orden de dichos valores no se respetaba en todos los informes, ni tampoco aparecían todos los campos. Además no había una estructura definida en la que los valores pudieran obtenerse a partir de, o a continuación de algún separador ortográfico.

Por ello, no se recomendaría usar CRFs aquí, sino **diccionarios** para información geográfica (países, provincias, etc.) y **expresiones regulares** para el resto de campos.

Como algunas categorías que aparecieron en el pie, ya aparecían en la cabecera o cuerpo, se hará uso de las **PHIs de confianza**, es decir operativamente si se obtiene un PHI en la cabecera por ejemplo, el sistema clasificará ese mismo valor (caso que aparezca en el pie) dentro de la misma categoría. La única excepción será el campo "*Sexo*", que es más simple de gestionar, a partir de los únicos valores aceptados: "H" y "M".

5.4 Módulo de pre-procesamiento

Aquí procesaremos el texto de nuestros informes, generando la salida para las siguientes fases del proceso. Se crea una instancia de la clase ***MEDDOCAN_doc***, conteniendo el nombre del documento, texto instanciado, los PHI y las posiciones donde se localizan, con cada parte del documento separadamente.

Usaremos adicionalmente técnicas en las distintas partes de cada informe:

- **Cabecera**

Cada frase se separa en dos partes, **separadas por los dos puntos (:).** La primera parte se mantiene inalterable, y la segunda se divide en palabras para su tratamiento. Obtendremos un vector con todas las frases del documento.

- **Cuerpo**

Se extraen las características de cada palabra para entrenar al modelo CRF, mediante diversas técnicas de PLN:

- **Tokenización** (separar el texto en palabras o tokens)
- **Lematización** (obtener el lema de cada palabra)
- **Detección de signos de puntuación y palabras irrelevantes** (stopwords)
- **Detección de mayúsculas y caracteres no alfabéticos**
- **POS-tagging** (etiquetar palabras en función de su tipo: verbo, adjetivo, etc.)
- **NER (Named Entity Recognition)** (asociar palabras a entidades categorizadas)

En nuestro sistema crearemos un vector completo para cada palabra que contendrá: la propia palabra, lema, POS, si es o no alfanumérico, si contiene o no mayúsculas, si es un título o sea todas las letras son mayúsculas, si es un NE, si es principio o final de una frase, y la dependencia dentro de la propia frase. A dicho vector se le asocia la categoría correspondiente caso de ser PHI.

- **Pie**

Como se usarán diccionarios, expresiones regulares y PHIs de confianza, se necesitará localizar el texto completo en esta parte del informe. Para los municipios y Comunidades Autónomas se usará la base de datos de la Agencia Tributaria **[14]**

5.4.1 Módulo de detección

Se compone de tres partes, donde cada una está encargada de detectar PHI en cada sección de los informes.

En el **cuerpo**, la clave es el entrenamiento del modelo CRF. Como ya sabemos, este método aprende diferenciando entre distintas categorías PHI. También es importante el *tipo de etiquetado*, como se observa en los sistemas declarados en los artículos expuestos en el Capítulo 3: hay algoritmos que se entrenan distinguiendo la **organización de las propias palabras dentro del PHI**. Cuando un PHI contiene varias palabras, la primera de ellas será etiquetada como "*Comienzo PHI*" **(C)** y el resto "*Dentro PHI*" **(D)**. Las que representen valores "*Nominales*" serán identificadas como **(N),** mientras que si no dan mayor información serán consideradas "*Irrelevantes*" **(I)**.

Un ejemplo sería el vector *["Domicilio:", ["Calle","Larios"]]*, donde como indica el diccionario, después de Domicilio esperaremos encontrar una "Calle", así que el vector será etiquetado como "*Calle"*: **C**-CALLE", "*Larios"*: **D**-CALLE".

Por todo, existirán modelos basados en "combinaciones de estas siglas": **C, D, I**, y además simultáneamente si se está diferenciando, o no, entre categorías. En nuestro caso, probamos distintos modelos para analizar su rendimiento según la forma de etiquetado:

- **Etiquetado N-I (Nominal – Irrelevante)**, no diferenciando categorías
- **Etiquetado CDI (Comienzo, Dentro, Irrelevante),** no diferenciando categorías
- **Etiquetado N-I**, sí diferenciando categorías
- **Etiquetado CDI**, sí diferenciando categorías

Los mejores resultados se obtuvieron con el "*Etiquetado CDI, sí diferenciando categorías*".

Otro aspecto importante son los ***parámetros de entrada al modelo***, ya que la librería **CRFsuite** entrena los datos usando el método de "*Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS)*", el cual maximiza el logaritmo de la probabilidad sobre dichos datos, y usa unos términos **L1** y **L2** para regular. Si por ejemplo L1 no es cero, el algoritmo usará el método de "*Quasi-Newton Orthant-WiseLimited-memory (OWN-QN)*", el cual mejora los pesos de las características al inicio del entrenamiento poco a poco, y al final convergen los pesos óptimos. Tras varias simulaciones se eligió un valor **L1=0.01** y **L2=0.1**, junto con 5.000 iteraciones, para acercarnos al mejor resultado.

En la **cabecera** se creó un diccionario de Python para indicar el tipo de categoría asociada a cada palabra clave. La entrada a esta parte del módulo es *[indicador, tokens de palabras que siguen al indicador]*. El dividir en tokens es para usar el etiquetado CDI.

En el **pie** se definen las expresiones regulares usadas para localizar las categorías de *teléfono*, *correo electrónico* y *dirección*. Igualmente un diccionario para las categorías de territorio, así como reconocimiento de números de faxes, mediante el indicador literal "*Fax:".* Por último incluye una sección para obtener los PHI de confianza, después de procesarse la cabecera y el pie. Del resultado de las pruebas se decide considerar PHI de confianza, solo aquellos con una longitud de valor de al menos 3 caracteres.

En resumen, el proceso **ordenado** quedaría así:

1. **PHI de Confianza**
2. **Expresión regular para e-mail**
3. **Búsqueda de Fax por sentencia**
4. **Expresión regular para teléfonos**
5. **Diccionario de municipios**
6. **Expresión regular para categoría "CALLE"**

Como consecuencia, si una palabra es clasificada pero luego se detectase también en otro método, **prevalece el primero**. Así se solucionan casos concretos en los que nombres propios puedan coincidir con municipios, o valores de teléfonos y faxes, por ejemplo.

6. Evaluación del Sistema

Para medir los resultados de cada parte del sistema, así como de su totalidad, hemos diseñado un módulo dedicado, basado en la función de medida usada para entrenar el modelo CRF en ***CRFSuite***. Analizaremos los valores de **Precision**, **Recall**, **F_1-score**, que serán los **principales indicadores** usados para la evaluación para cada clase. Además mostraremos el valor del campo **Support**, con índole informativo, para conocer el volumen de muestras.

6.1 Medidas de evaluación

Partimos de los valores básicos siguientes:

- **TP (True Positive):** elemento de la clase, clasificado correctamente
- **TN (True Negative):** elemento de otra clase, clasificado correctamente
- **FP (False Positive):** elemento de la clase, clasificado incorrectamente
- **FN (False Negative):** elemento de otra clase, clasificado incorrectamente

A partir de ellos establecemos los siguientes valores, base de nuestra evaluación posterior:

PRECISIÓN	$\frac{TP}{TP+FP}$
RECALL	$\frac{TP}{TP+FN}$
F_1-SCORE	$2\frac{Precision \cdot Recall}{Precision+Recall}$
SUPPORT	Número de ocurrencias de cada clase

6.2 Resultados

Hemos tomado los mismos conjuntos que la tarea MEDDOCAN, uno representando el **75% del total de documentos** de la tarea **para el training** del clasificador CRF, y el resto **25% para tests**. Con pruebas realizadas en local, con una proporción 80-20%, o 70-30% se observaron resultados similares o ligeramente inferiores.

Los resultados se han ido midiendo en orden secuencial: primero en el **cuerpo**, a continuación en la detección PHI de la **cabecera**, luego usando expresiones regulares y diccionarios y PHI de confianza en el **pie**, y por último la **evaluación completa** del sistema.

6.2.1 Resultados Cuerpo – CRF

A continuación, en **[Tabla 2]**, **[Tabla 3]**, **[Tabla 4]** y **[Tabla 5]** podemos observar los resultados obtenidos respectivamente, al escoger etiquetado N-I sin diferenciación de clases, CDI sin diferenciación de clases, N-I con diferenciación de clases y CDI con diferenciación de clases.

CLASSIFICATION REPORT :

	precision	recall	f1-score	support
I	0.77	0.95	0.85	2844
N	0.98	0.92	0.95	9470
accuracy			0.92	12314
macro avg	0.88	0.93	0.90	12314
weighted avg	0.94	0.92	0.93	12314

Tabla 2: N-I sin diferenciación de clases

CLASSIFICATION REPORT :

	precision	recall	f1-score	support
I	0.78	0.94	0.85	3050
C-N	0.96	0.95	0.95	4583
D-N	0.96	0.84	0.90	4905
accuracy			0.91	12538
macro avg	0.90	0.91	0.90	12538
weighted avg	0.92	0.91	0.91	12538

Tabla 3: CDI sin diferenciación de clases

CLASSIFICATION REPORT :

	precision	recall	f1-score	support
N-CALLE	0.97	0.84	0.90	2121
N-CENT	0.00	0.00	0.00	33
N-DIRECT	0.00	0.00	0.00	0
N-EDADS	0.99	1.00	0.99	443
N-EMAIL	0.97	1.00	0.98	221
N-FAMS	0.00	0.00	0.00	0
N-FECH	1.00	1.00	1.00	459
N-HOSP	0.83	0.47	0.60	431
I	0.80	0.95	0.87	2764
N-IDASEG	1.00	0.99	1.00	552
N-IDCONT	1.00	1.00	1.00	35
N-IDDISP	0.00	0.00	0.00	0
N-IDEMP	0.00	0.00	0.00	0
N-IDOTRO	0.00	0.00	0.00	0
N-IDSUJ	1.00	1.00	1.00	230
N-IDTIT	1.00	1.00	1.00	648
N-IDVEH	0.00	0.00	0.00	0
N-INST	0.50	0.03	0.06	93
N-NOMP	1.00	0.99	0.99	1489
N-NOMS	1.00	1.00	1.00	709
N-NUMB	0.00	0.00	0.00	0
N-NUMF	0.72	0.72	0.72	18
N-NUMT	0.62	0.84	0.71	75
N-OTROS	0.00	0.00	0.00	0
N-PAIS	0.99	0.98	0.99	306
N-PROF	0.00	0.00	0.00	0
N-SEXOS	1.00	1.00	1.00	227
N-TER	0.89	0.93	0.91	915
N-URL	0.00	0.00	0.00	0
micro avg	0.92	0.92	0.92	11769
macro avg	0.56	0.54	0.54	11769
weighted avg	0.92	0.92	0.91	11769

Tabla 4: N-I con diferenciación de clases

CLASSIFICATION REPORT :

	precision	recall	f1-score	support
C-CALLE	0.99	0.84	0.91	403
D-CALLE	0.96	0.78	0.86	1821
C-CENT	0.00	0.00	0.00	1
D-CENT	0.00	0.00	0.00	5
C-DIRECT	0.00	0.00	0.00	0
D-DIRECT	0.00	0.00	0.00	0
C-EDADS	0.98	1.00	0.99	233
D-EDADS	0.99	1.00	0.99	217
C-EMAIL	0.99	0.99	0.99	227
D-EMAIL	0.00	0.00	0.00	1
C-FAMS	0.00	0.00	0.00	0
D-FAMS	0.00	0.00	0.00	0
C-FECH	0.99	0.99	0.99	475
D-FECH	1.00	0.17	0.29	6
C-HOSP	0.87	0.38	0.53	119
D-HOSP	0.74	0.39	0.51	379
I	0.76	0.95	0.84	2975
C-IDASEG	0.99	0.98	0.99	191
D-IDASEG	1.00	0.98	0.99	374
C-IDCONT	1.00	1.00	1.00	34
D-IDCONT	0.00	0.00	0.00	0
C-IDDISP	0.00	0.00	0.00	0
D-IDDISP	0.00	0.00	0.00	0
C-IDEMP	0.00	0.00	0.00	1
D-IDEMP	0.00	0.00	0.00	4
C-IDOTRO	0.00	0.00	0.00	0
D-IDOTRO	0.00	0.00	0.00	0
C-IDSUJ	1.00	0.99	1.00	239
D-IDSUJ	1.00	1.00	1.00	2
C-IDTIT	0.99	1.00	1.00	223
D-IDTIT	0.99	1.00	1.00	429
C-IDVEH	0.00	0.00	0.00	0
D-IDVEH	0.00	0.00	0.00	0
C-INST	0.00	0.00	0.00	25
D-INST	0.00	0.00	0.00	72
C-NOMP	0.99	0.97	0.98	472
D-NOMP	0.99	0.97	0.98	1061
C-NOMS	1.00	1.00	1.00	475
D-NOMS	1.00	1.00	1.00	259
C-NUMB	0.00	0.00	0.00	0
D-NUMB	0.00	0.00	0.00	0
C-NUMF	0.50	0.62	0.56	8
D-NUMF	0.20	0.10	0.13	10
C-NUMT	0.00	0.00	0.00	24
D-NUMT	0.42	0.93	0.58	29
C-OTROS	0.00	0.00	0.00	0
D-OTROS	0.00	0.00	0.00	0
C-PAIS	0.98	0.98	0.98	325
D-PAIS	1.00	0.50	0.67	4
C-PROF	0.00	0.00	0.00	0
D-PROF	0.00	0.00	0.00	0
C-SEXOS	1.00	1.00	1.00	232
D-SEXOS	0.00	0.00	0.00	0
C-TER	0.85	0.94	0.89	873
D-TER	0.98	0.40	0.57	133
C-URL	0.00	0.00	0.00	0
D-URL	0.00	0.00	0.00	0
micro avg	0.90	0.90	0.90	12361
macro avg	0.46	0.42	0.42	12361
weighted avg	0.90	0.90	0.89	12361

Tabla 5: CDI con diferenciación de clases

Conclusiones:

- CDI diferenciando por clases es ligeramente la mejor opción

- CRF ofrece buenos resultados, dependiendo de los valores de la etiqueta “I”

Por último hemos recalculado las medidas de precisión, excluyendo ahora la clase “I”. El resultado es aceptable y curiosamente se mantiene con un nivel de eficacia similar a los datos anteriores, como se observa en la [Tabla 6]

Las categorías con un rendimiento menor han sido, como era de esperar, aquellas de las que teníamos pocos datos anotados.

En el lado opuesto, las categorías mejor valoradas fueron porque han tenido suficientes datos anotados, o como en el caso de “C-FECH”, correspondiente a fechas, debido su intrínseco formato bien conocido.

CLASSIFICATION REPORT :

	precision	recall	f1-score	support
C-IDSUJ	0.99	0.99	0.99	239
D-PROF	0.00	0.00	0.00	0
D-NUMF	0.00	0.00	0.00	9
D-EMAIL	0.00	0.00	0.00	9
D-IDDISP	0.00	0.00	0.00	0
C-OTROS	0.00	0.00	0.00	0
C-IDDISP	0.00	0.00	0.00	0
D-IDOTRO	0.00	0.00	0.00	0
C-NUMF	0.29	0.50	0.36	4
C-NUMT	0.00	0.00	0.00	20
D-PAIS	0.00	0.00	0.00	2
C-INST	0.00	0.00	0.00	28
C-PROF	0.00	0.00	0.00	0
D-FECH	1.00	0.33	0.50	3
D-TER	0.98	0.37	0.54	160
C-CALLE	0.99	0.85	0.91	411
D-URL	0.00	0.00	0.00	0
D-DIRECT	0.00	0.00	0.00	0
C-NOMP	1.00	0.96	0.98	478
D-FAMS	0.00	0.00	0.00	0
D-NOMS	1.00	1.00	1.00	261
D-NUMB	0.00	0.00	0.00	0
C-DIRECT	0.00	0.00	0.00	0
C-FECH	1.00	1.00	1.00	481
D-SEXOS	0.00	0.00	0.00	0
C-IDOTRO	0.00	0.00	0.00	0
D-IDCONT	0.00	0.00	0.00	0
D-HOSP	0.77	0.40	0.53	371
C-IDVEH	0.00	0.00	0.00	0
C-TER	0.83	0.93	0.88	889
D-INST	0.00	0.00	0.00	86
D-IDSUJ	0.00	0.00	0.00	0
D-NOMP	1.00	0.96	0.98	1083
D-OTROS	0.00	0.00	0.00	0
C-URL	0.00	0.00	0.00	0
D-CENT	0.00	0.00	0.00	12
C-CENT	0.00	0.00	0.00	3
D-IDEMP	0.00	0.00	0.00	0
C-IDTIT	1.00	1.00	1.00	223
C-PAIS	0.97	0.98	0.97	331
D-CALLE	0.96	0.78	0.86	1897
C-EDADS	0.99	1.00	1.00	237
D-IDTIT	1.00	1.00	1.00	438
C-FAMS	0.00	0.00	0.00	0
D-EDADS	0.99	1.00	1.00	226
D-IDVEH	0.00	0.00	0.00	0
C-SEXOS	1.00	1.00	1.00	237
C-NOMS	1.00	1.00	1.00	480
C-NUMB	0.00	0.00	0.00	0
C-IDCONT	1.00	1.00	1.00	29
C-IDASEG	0.99	0.98	0.99	186
C-IDEMP	0.00	0.00	0.00	0
D-IDASEG	1.00	0.99	0.99	358
C-HOSP	0.85	0.39	0.54	114
D-NUMT	0.47	1.00	0.64	32
C-EMAIL	0.99	0.98	0.98	232
micro avg	0.95	0.87	0.91	9569
macro avg	0.41	0.38	0.39	9569
weighted avg	0.94	0.87	0.90	9569

Tabla 6: CDI sin incluir clase I

6.2.2 Resultados Cabecera - Detección PHI según estructura

En la siguiente [Tabla 7] se muestran los resultados obtenidos tras procesar los registros con todos los datos.

En líneas generales funciona bastante bien, salvo casos muy concretos de erratas en las anotaciones o en el propio formato del campo. También encontramos tabulaciones o saltos de líneas inesperados lo cual "desencajaba" el formato esperado.

Por último, se confirma que el porcentaje de datos PHI es considerablemente mayor en la cabecera que en el cuerpo, como se esperaba.

CLASSIFICATION REPORT :

	precision	recall	f1-score	support
C-CALLE	1.00	1.00	1.00	236
D-CALLE	1.00	1.00	1.00	967
C-EDADS	0.99	1.00	1.00	234
D-EDADS	0.99	1.00	1.00	224
C-FECH	1.00	1.00	1.00	474
D-FECH	1.00	1.00	1.00	1
I	1.00	1.00	1.00	111
C-IDASEG	1.00	0.99	0.99	198
D-IDASEG	1.00	0.99	0.99	392
C-IDCONT	1.00	1.00	1.00	35
C-IDSUJ	1.00	1.00	1.00	235
D-IDSUJ	1.00	1.00	1.00	4
C-IDTIT	0.99	1.00	1.00	220
D-IDTIT	0.99	1.00	1.00	427
C-NOMP	1.00	1.00	1.00	234
D-NOMP	1.00	1.00	1.00	526
C-NOMS	1.00	1.00	1.00	474
D-NOMS	1.00	1.00	1.00	259
C-PAIS	1.00	0.99	0.99	238
D-PAIS	0.00	0.00	0.00	2
C-SEXOS	1.00	1.00	1.00	231
C-TER	1.00	1.00	1.00	472
D-TER	1.00	1.00	1.00	57
micro avg	1.00	1.00	1.00	6251
macro avg	0.95	0.96	0.95	6251
weighted avg	1.00	1.00	1.00	6251
samples avg	1.00	1.00	1.00	6251

Tabla 7: Resultados Cabecera

6.2.3 Resultados Pie - Detección PHI + Diccionarios

En la siguiente [Tabla 8] se muestran los resultados obtenidos tras procesar los registros PHI de confianza que obtuvimos del Cuerpo y Cabecera, tras tomar como base el modelo CRF ya entrenado, es decir, basados en los datos de TEST.

Como dichos registros guardaban como campo clave el "documento" asociado (historial clínico) es posible recuperar dichos valores de Test para cada documento concreto. Recordemos que estos PHI de confianza son "exclusivos" para el documento, y no serán válidos para detectar PHI en ningún otro documento.

Conclusiones:

- Especialmente la detección de categorías como "NOMS" (Nombre del sujeto) funcionaron excepcionalmente, con F_1-score rondando o superando 0.99

- Sin embargo las clases basadas en expresiones regulares no fueron igual de efectivas. Como ejemplo los datos de dirección (calle, etc.), números de teléfono y fax (por el formato)

- El diccionario usado para detectar la clase "TER" (territorios, como municipios, ciudades y códigos postales) tampoco fue demasiado óptimo, a pesar de usar una base de datos consolidada. Aquí también el motivo más repetido fue que no estaba escrito correctamente, o en alguna otra forma desconocida, etc.

- En general, constatamos que el Pie tiene una estructura no tan definida como en el caso del cuerpo o cabeza.

CLASSIFICATION REPORT :

	precision	recall	f1-score	support
C-CALLE	0.99	0.84	0.91	391
D-CALLE	0.95	0.80	0.87	1731
C-CENT	0.00	0.00	0.00	3
D-CENT	0.00	0.00	0.00	10
C-EDADS	0.99	1.00	1.00	232
D-EDADS	1.00	1.00	1.00	217
C-EMAIL	0.99	1.00	0.99	216
D-EMAIL	0.00	0.00	0.00	1
C-FECH	1.00	1.00	1.00	469
D-FECH	1.00	1.00	1.00	1
C-HOSP	0.83	0.41	0.55	111
D-HOSP	0.74	0.39	0.51	348
I	0.64	0.95	0.76	2879
C-IDASEG	0.99	0.98	0.99	185
D-IDASEG	1.00	0.99	0.99	365
C-IDCONT	1.00	1.00	1.00	35
C-IDSUJ	0.99	0.99	0.99	234
D-IDSUJ	1.00	1.00	1.00	3
C-IDTIT	1.00	1.00	1.00	219
D-IDTIT	1.00	1.00	1.00	429
C-INST	0.00	0.00	0.00	36
D-INST	0.00	0.00	0.00	95
C-NOMP	1.00	0.50	0.67	467
D-NOMP	1.00	0.49	0.66	1078
C-NOMS	1.00	1.00	1.00	470
D-NOMS	1.00	1.00	1.00	253
C-NUMF	0.31	0.80	0.44	5
D-NUMF	0.20	0.14	0.17	7
C-NUMT	0.00	0.00	0.00	24
D-NUMT	0.46	0.84	0.59	32
C-PAIS	0.99	0.75	0.85	311
D-PAIS	0.00	0.00	0.00	1
C-SEXOS	1.00	1.00	1.00	230
C-TER	0.79	0.93	0.86	845
D-TER	1.00	0.44	0.62	99
micro avg	0.83	0.83	0.83	12032
macro avg	0.71	0.66	0.67	12032
weighted avg	0.86	0.83	0.83	12032
samples avg	0.83	0.83	0.83	12032

Tabla 8: Resultados Pie

6.2.4 Análisis de errores

Turno de **recopilar** los principales **errores encontrados en los distintos resultados** (Cuerpo, Cabecera y Pie), así como **ejemplos** concretos localizados, que ayudarán a mejorar nuestro sistema en el futuro:

Tanto en el **Cuerpo** como en el **Pie** (la Cabecera en general funcionó mejor, dado su formato conocido), los principales problemas surgen al intentar identificar PHI basados en **expresiones regulares.** Nuestro código del proyecto incluyó las siguientes:

```
#Patrones Expresiones Regulares
PATTERN_EMAIL = '[\w\.-]+@[\w\.-]+'
PATTERN_SEPARATOR = '[\:\s\.\-\(\)\/]*'
PATTERN_TEL1 = PATTERN_SEPARATOR.join(['[69]\d{2}', '\d{2}', '\d{2}', '\d{2}', '\d{2}'])
PATTERN_TEL2 = PATTERN_SEPARATOR.join(['[69]\d{1}', '\d{3}', '\d{3}', '\d{2}', '\d{2}'])
PATTERN_TEL3 = PATTERN_SEPARATOR.join(['[69]\d{2}', '\d{3}', '\d{3}'])
PATTERN_TEL4 = PATTERN_SEPARATOR.join(['[69]\d{1}', '\d{2}', '\d{3}', '\d{2}'])
PATTERN_TEL5 = PATTERN_SEPARATOR.join(['[69]\d{3}', '\d{3}', '\d{3}'])
PATTERN_TEL6 = PATTERN_SEPARATOR.join(['\d{3}', '\d{2}', '\d{2}', '\d{2}'])
PATTERN_PREFIJO = '(?:00\d{2}|(?:\+\s*?\d{2,4}?))'
PATTERN_TELEFONO = PATTERN_PREFIJO + '?' + PATTERN_SEPARATOR + '(?:(?:' + PATTERN_TEL1 + ')|(?:' + PATTERN_TEL2 + ')|(?:' \
                + PATTERN_TEL3 + ')|(?:' + PATTERN_TEL4 + ')|(?:' + PATTERN_TEL5 + ')|(?:' + PATTERN_TEL6 + '))'
PATTERN_CALLE = '(?:(?:Avenida)|(?:Calle)|(?:avenida)|(?:calle)|(?:c\/)|(?:C\/)|(?:Carretera)|(?:carretera)|(?:Av\.)' \
          '|(?:av\.)|(?:avda\.)|(?:Avda\.)|(?:Carrer)|(?:carrer)|(?:Travesía)|(?:travesía)|(?:Paseo)|(?:paseo))'
PATTERN_HOSPITAL = '(?:(?:Hospital)|(?:Centro Penitenciario)|(?:Centro de))'
PATTERN_INSTITUCION = '(?:(?:Facultad)|(?:Universidad)|(?:Escuela universitaria)|(?:Clínica)|(?:Instituto))'
PATTERN_CP = '(?:0[1-9]\d{3}|[1-4]\d{4}|5[0-2]\d{3})'
patterns = {'calle': PATTERN_CALLE, 'hospital': PATTERN_HOSPITAL, 'institucion': PATTERN_INSTITUCION, 'CP': PATTERN_CP}
```

Figura 4: Expresiones Regulares (configuración)

Especialmente los valores de *HOSPITAL* e *INSTITUCIÓN* no fueron bien reconocidos en el **Cuerpo**. Tampoco los nombres de *CALLE* en el **Pie** tuvieron un buen rendimiento. Y de forma **general**, la identificación de *TERRITORIOS*, *NÚMEROS DE TELÉFONO* y *NÚMEROS DE FAX* fue dificultosa.

Algunos **ejemplos** detectados (en rojo el valor literal que aparece en el informe):

HOSPITAL:

Centro de Salud Talavera C/ José Luis Gallo

(*Centro*: C-HOSP, *de*: D-HOSP, *Salud*: D-HOSP, *Talavera*: C-TER)

INSTITUCION:

Unidad de Gestión Clínica-Cirugía General

(*Unidad*: C-HOSP, *de*: D-HOSP, *Gestión*: D-HOSP,
Clínica: C-INST, -: D-INST, *Cirugía*: D-INST, *General*: D-INST)

CALLE:

Avda. de Córdoba s/n

(*Avda.*: C-CALLE, *de*: D-CALLE, *Córdoba*: C-TER, *s/n*: I)

TERRITORIO:

C/ Altos de Nava, s/n 24071

(*C/*: C-CALLE, *Altos*: C-TER, *Nava*: C-TER)

NUMERO_TELEFONO:

Correos 20134 48080

(*Correos*: C-NUMT, *20134 48080*: D-NUMT)

NUMERO_FAX:

Fax: 987 23 73 92 e-mail

(*987*: C-NUMF, *23*: C-NUMF, *73*: C-NUMF, *92*: C-NUMF, e: D-NUMF)

6.2.5 Resultado total del sistema

Finalmente en la siguiente [Tabla 9] podemos encontrar los resultados obtenidos tras evaluar todo el sistema, con el 75% de documentos para training de la tarea, y el 25% restante para tests.

De aquel 25% (250 documentos) se ha obtenido el cuerpo para predecir las etiquetas introduciéndolos en el modelo de CRF entrenado, se ha cogido a continuación la cabecera para detectar los PHI según sus estructuras, y al final, obtuvimos los PHI de confianza, que junto a la gestión del pie, nos permite evaluar todo en su conjunto. Como ya dijimos, hemos usado **CRFsuite** para calcular las medidas de evaluación.

CLASSIFICATION REPORT :

	precision	recall	f1-score	support
C-CALLE	0.99	0.86	0.92	1594
D-CALLE	0.96	0.79	0.87	7296
C-CENT	0.00	0.00	0.00	12
D-CENT	0.00	0.00	0.00	49
C-EDADS	0.99	1.00	0.99	933
D-EDADS	0.99	1.00	1.00	877
C-EMAIL	0.99	0.99	0.99	889
D-EMAIL	0.09	0.17	0.12	12
C-FECH	1.00	1.00	1.00	1894
D-FECH	1.00	0.25	0.40	12
C-HOSP	0.86	0.45	0.59	435
D-HOSP	0.76	0.43	0.55	1399
I	0.77	0.95	0.85	11370
C-IDASEG	1.00	0.99	0.99	744
D-IDASEG	1.00	0.99	1.00	1451
C-IDCONT	0.97	1.00	0.99	136
C-IDEMP	0.00	0.00	0.00	1
D-IDEMP	0.00	0.00	0.00	4
C-IDSUJ	0.99	0.99	0.99	943
D-IDSUJ	0.41	1.00	0.58	7
C-IDTIT	1.00	1.00	1.00	883
D-IDTIT	1.00	1.00	1.00	1709
C-INST	0.14	0.01	0.02	105
D-INST	0.11	0.01	0.01	327
C-NOMP	1.00	0.98	0.99	1875
D-NOMP	0.99	0.98	0.99	4257
C-NOMS	1.00	1.00	1.00	1893
D-NOMS	1.00	1.00	1.00	1043
C-NUMF	0.38	0.80	0.52	25
D-NUMF	0.15	0.07	0.09	45
C-NUMT	0.00	0.00	0.00	91
D-NUMT	0.49	0.89	0.63	141
C-PAIS	0.98	0.98	0.98	1267
D-PAIS	1.00	0.17	0.29	12
C-SEXOS	1.00	1.00	1.00	929
C-TER	0.86	0.95	0.90	3424
D-TER	0.98	0.41	0.58	532
micro avg	0.90	0.90	0.90	48616
macro avg	0.70	0.65	0.64	48616
weighted avg	0.90	0.90	0.90	48616
samples avg	0.90	0.90	0.90	48616

Tabla 9: Resultado Sistema Completo

Conclusiones:

- Las conclusiones van en la misma línea, que en cada sección independiente
- Categorías con pocas o ninguna anotación no suelen ser bien reconocidas
- Gracias a los resultados de la cabecera, las puntuaciones finales son mejores

En la [Tabla 7] se veían los resultados de la cabecera introduciendo todos los documentos. Se observa que más de la mitad de los PHI se encuentran en el encabezado de los informes y, como el método propuesto para la detección de esta parte funciona muy bien, hace que tenga bastante peso en los resultados globales del sistema.

Recordemos que los valores "**samples avg**" representan la relación entre aciertos y etiquetas introducidas ("**support**"), incluyendo "I". Las *medias* de valores obtenidas tras sucesivas simulaciones para nuestra tarea fueron:

precision	recall	F1-score
0.9171	0.9077	0.9091

Como comparativa, mostraremos los **mejores resultados** obtenidos en la **tarea original**, por los sistemas presentados en el desafío. Las evaluaciones están basadas en cálculos propios de dicha tarea, y que corresponden a nuestros valores "**micro avg**".

En la [Tabla 10] se observan los valores más altos globales, para cada una de las **dos subtareas MEDDOCAN**:

(1) → Identificación y clasificación de entidades
(2) → Detección de texto sensible (**2a**:estándar ; **2b**:mejorado con uso de fusiones)

subtarea	precision	recall	F1-score
(1)	0.97191	0.96944	0.96961
(2a)	0.97747	0.97474	0.97491
(2b)	0.98749	0.98335	0.98530

Tabla 10: Mejores resultados MEDDOCAN (subtareas)

En la [Tabla 11] encontramos los mejores resultados finales obtenidos por **equipo**, así como el sistema empleado por sus componentes, en cada subtarea:

Subtarea	Organización	País	Usuario	Sistema	precision	recall	F1-score
(1)	Bosch Center for AI	Alemania	lukas.lange	NLNDE	0.96978	0.96944	0.96961
(1)	Univ. Rovira i Virgili, CRISES group	España	Fadi	ReCRF	0.96991	0.95672	0.96327
(1)	Vicomtech	España	nperez	CRF + XGBoost + (CNN+LSTM)	0.96403	0.95637	0.96018
(2a)	Bosch Center for AI	Alemania	lukas.lange	NLNDE	0.97508	0.97474	0.97491
(2a)	Univ. Rovira i Virgili, CRISES group	España	Fadi	ReCRF	0.97529	0.96202	0.96861
(2a)	Vicomtech	España	nperez	CRF + XGBoost + (CNN+LSTM)	0.97187	0.96414	0.96799
(2b)	Bosch Center for AI	Alemania	lukas.lange	NLNDE	0.98749	0.98311	0.98530
(2b)	Harbin Institute of Technology	China	jiangdehuan	Deep Learning (BERT+CRF,Flair)	0.98033	0.98335	0.98184
(2b)	Vicomtech	España	nperez	CRF + XGBoost + (CNN+LSTM)	0.97954	0.97235	0.97593

Tabla 11: Mejores resultados MEDDOCAN (equipos)

Por todo, los resultados obtenidos en nuestro presente trabajo pueden entenderse como aceptables, con un *F1-score medio de 0.91*, usando como base clasificadores basados en redes neuronales y CRF.

Los participantes en la tarea usaron un corpus desglosado así: *50% para training* (500 informes), *25% para desarrollo* (250 informes) y *25% para tests* (informes). Nosotros usamos los **mismos conjuntos** exactamente, mas incluimos la parte de desarrollo al de training, obteniendo 2 conjuntos: 75% para training (750 informes) y 25% para tests (250 informes).

Al haber usado los mismos conjuntos referencia de informes que en MEDDOCAN, sí podemos considerar como aceptable la referencia comparativa con nuestro proyecto, y evaluar así nuestra solución.

6.2.6 Clasificador adicional basado en GRU

Además de CRF vamos a incluir finalmente un nuevo clasificador, basado en redes neuronales recurrentes (RNN), más en concreto en una versión evolucionada, como es **GRU**. El desarrollo en Python, mediante **Tensorflow** y **Keras**, fue sumamente sencillo y simple. La configuración de la red apenas necesitó unos pocos parámetros para tener rápidamente los datos listos para ser procesados.

Vamos a repasar las características del clasificador, y por último, compararemos el rendimiento respecto al modelo anterior basado en CRF.

En primer lugar definimos a nivel de código unos valores *constantes*, que representarán el número de unidades (**units**) de la red neuronal, así como un valor booleano para controlar la ejecución de forma que podamos llamar a uno u otro clasificador fácilmente.

En la siguiente captura incluimos el código Python usado para **crear, compilar, entrenar y evaluar** el modelo**:**

```
gru_model = Sequential()
gru_model.add(Embedding(size_input, size_output))
gru_model.add(GRU(units=GRU_units
                  ,activation = 'tanh'
                  ,recurrent_activation = 'sigmoid'
                  ,return_sequences = True
                  ,input_shape = (size_input, size_output)
                 ))
gru_model.add(Dropout(0.2))
gru_model.add(Dense(units=1, activation='sigmoid'))

METRICS = ['acc', precision_m, recall_m, f1_m]
gru_model.compile(loss='binary_crossentropy', optimizer=adam_v2.Adam(lr=0.005), metrics=METRICS)

print("\n*****************************************\n")
print(gru_model.summary())

#Training
gru_model.fit(X_train, y_train, epochs=num_epochs, verbose=1)

#Evaluamos el modelo
scores = gru_model.evaluate(X_test, y_test, verbose=1)
print("-------------------------")
print("Accuracy: %.2f%%" % (scores[1]*100))
print("Precision: %.2f%%" % (scores[2]*100))
print("Recall: %.2f%%" % (scores[3]*100))
print("F1-Score: %.2f%%" % (scores[4]*100))
```

Figura 5: Modelo GRU (configuración)

Ahora podemos observar la configuración, donde establecimos, tras algunas pruebas afinando los parámetros, un valor de **10 unidades** en la red neuronal, y un número de **epochs=15**, con los cuales hemos obtenido buenos resultados.

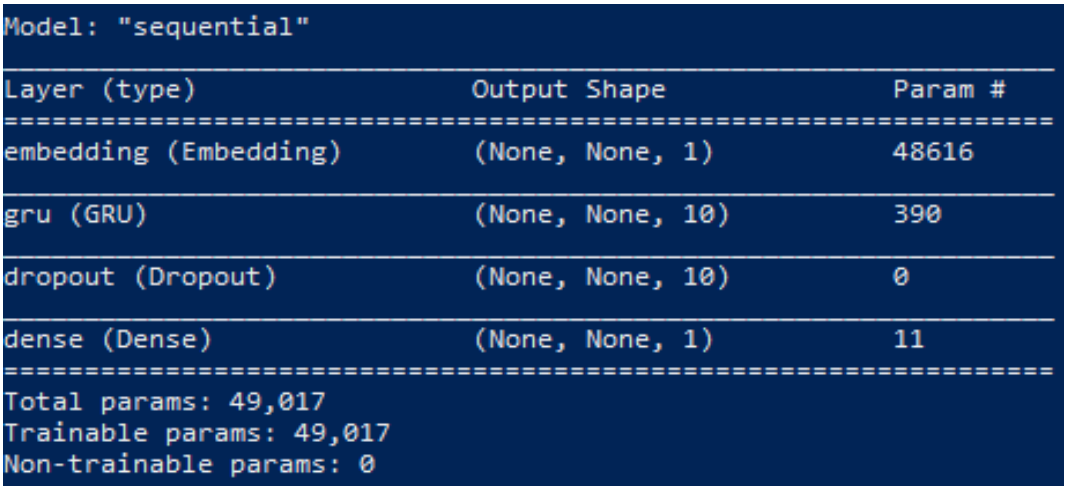

Figura 6: Modelo GRU (summary)

Hemos declarado un modelo **secuencial**, que consta de una capa "**Embedding**", a continuación una capa **GRU**, luego una "**Dropout**" y finalmente una capa "**Dense**".

La capa "*Embedding*" es un método de clustering que agrupa palabras similares en grupos homogéneos, de forma que los grupos entre sí sean lo más heterogéneos posibles.

La capa "*GRU*" es la que usará tantas unidades internas (units) como hayamos declarado, para controlar el flujo de los datos a través de las puertas. Esta red usará la función "*tanh*" (por defecto) para el paso de activación, y la función "*sigmoid*" (por defecto) para el paso recurrente. En nuestro caso hemos configurado **True** el parámetro "*return_sequences*" para tomar la secuencia completa como la salida (en caso contrario solo se devolvería la última salida del estado oculto.

La capa "*Dropout*" es usada para reducir el overfitting, mientras que "*Dense*" sirve para establecer el número de neuronas (en nuestro caso "1") como dimensión de salida.

Compilamos el modelo para evaluar las métricas "**accuracy**", "**precision**", "**recall**" y "**F1-score**".

Los resultados para dicha evaluación muestran un rendimiento mejorado respecto al anterior clasificador, con porcentajes de **"accuracy", por ejemplo,** entre **0.93-0.96 (de media)**, en las pruebas realizadas.

En la siguiente captura podemos localizar algunos valores en una de las simulaciones realizadas:

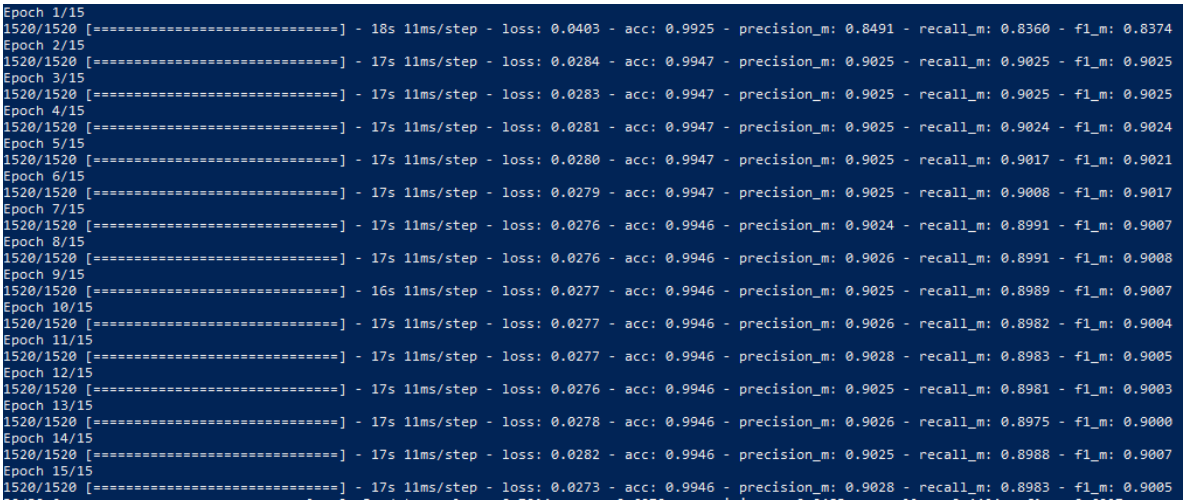

Figura 7: Modelo GRU (evaluación)

Igualmente las métricas "**precision**" y "**recall**" superaron de media **0.91**; mientras que para "**F1-score**" se ha llegado a alcanzar valores superiores a **0.92**, en diversas ejecuciones del modelo GRU.

A modo resumen mostremos por último en la siguiente tabla los resultados unificados de los sistemas implementados en este proyecto, junto con los mejores obtenidos en la tarea MEDDOCAN. Valores no muy distantes pero algo alejados, por lo que para el futuro se recomendaría mejorar el rendimiento de la red neuronal o incluso combinarlas:

Subtarea	Organización / Usuario	País	Usuario	Sistema	precision	recall	F1-score
(1)	Bosch Center for AI	Alemania	lukas.lange	NLNDE	0.96978	0.96944	0.96961
(1)	Univ. Rovira i Virgili, CRISES group	España	Fadi	ReCRF	0.96991	0.95672	0.96327
(1)	Vicomtech	España	nperez	CRF + XGBoost + (CNN+LSTM)	0.96403	0.95637	0.96018
(2a)	Bosch Center for AI	Alemania	lukas.lange	NLNDE	0.97508	0.97474	0.97491
(2a)	Univ. Rovira i Virgili, CRISES group	España	Fadi	ReCRF	0.97529	0.96202	0.96861
(2a)	Vicomtech	España	nperez	CRF + XGBoost + (CNN+LSTM)	0.97187	0.96414	0.96799
(2b)	Bosch Center for AI	Alemania	lukas.lange	NLNDE	0.98749	0.98311	0.98530
(2b)	Harbin Institute of Technology	China	jiangdehuan	Deep Learning (BERT+CRF,Flair)	0.98033	0.98335	0.98184
(2b)	Vicomtech	España	nperez	CRF + XGBoost + (CNN+LSTM)	0.97954	0.97235	0.97593
TFM (avg)	José Antonio Gaitán Rivas	España	jagr	CRF + GRU	0.91710	0.90770	0.90910

Tabla 12: Resultados comparativa MEDDOCAN vs presente trabajo

7. Conclusiones

En nuestro trabajo hemos estudiado diversos métodos existentes para anonimizar textos contenidos en informes médicos. Se han evaluado todos ellos, aunque nos consta que existen multitud de otras formas en las que poder haber afrontado el problema, las cuales serían igualmente válidas.

Siempre es sumamente complejo el tratamiento de texto plano, escrito en lenguaje natural, y si a la vez contiene información clínica, aún más. Para solventarlo de la mejor manera posible hemos acudido a librerías estándar que facilitan el PLN y a la vez nos aportan información sobre dichos textos.

En el gran abanico de sistemas a ser aplicados, optamos por uno híbrido, basado en el modelo de aprendizaje automático sobre CRFs y redes neuronales GRU, que junto con una gestión de diccionarios, expresiones regulares y PHIs de confianza, nos han ayudado a construir y evaluar nuestro sistema.

7.1 Ventajas y desventajas

- El modelo de aprendizaje automático basado en CRF, especialmente combinado su uso con redes neuronales, demostró ser muy buen candidato para anonimizar datos. Sin embargo, necesita cuantas más anotaciones mejor, para aprender aceptablemente.
- Las expresiones regulares son siempre recomendables en clases bien definidas. En caso contrario pueden convertirse en un elemento muy costoso y candidato a errores.
- Igual ocurre con los diccionarios. En ocasiones se hace necesario un gran volumen de entradas para gestionar ciertos valores.
- El uso de PHI de confianzas, aunque en general dan buen resultado, tienen el riesgo de clasificar incorrectamente ciertos valores, los cuales necesitan revisión manual.
- En el cuerpo se usó un único modelo de aprendizaje automático por simplicidad, lo cual limita en cierto modo los resultados. Tendría un mayor valor haberlo comparado con otros modelos, pero eso inicialmente estaba fuera del alcance de este trabajo.

7.2 Trabajo futuro

Hemos constatado que nuestro sistema es correcto, y consigue el objetivo marcado en el trabajo, pero por supuesto es mejorable. A continuación enumeramos algunos puntos en los que se podría enmarcar una nueva implementación basada en nuestro sistema:

- Una recomendación sencilla, aunque llevaría tiempo de implementación, sería la creación de un módulo adicional encargado de actualizar los PHI detectados en cada nuevo informe gestionado.

- El procesamiento del pie del documento es bastante mejorable. Se podría intentar mediante expresiones regulares más óptimas o el uso de nuevos diccionarios.

- En el cuerpo podrían mejorarse las clases identificadas mediante algún otro método de aprendizaje automático.

- Añadir o eliminar características a las palabras antes de entrenarlas con CRF y comparar los resultados con los actuales.

- Implementación mediante el uso de otras variantes de redes neuronales, de forma independiente o combinado, para analizar y comparar sus resultados, identificando qué topología se adapta mejor al alcance de nuestra tarea.

Bibliografía

[1] El Emam, K. y Arbuckle L. (2013): "Anonymizing Health Data", O'Reilly.

[2] Sweeney, L. (1996): "Replacing Personally-Identifying Information in Medical Records, the Scrub System"

[3] Bernan, J. (2003): "Concept-Match medical data scrubbing. How pathology text can be used in research"

[4] Wellner, B. y otros (2007): "Rapidly Retargetable Approaches to Deidentification in Medical Records"

[5] Szarvas y otros (2007): "Anonymization of Medical Records Using an Iterative Machine Learning Framework"

[6] Lafferty, J. (2001): "Conditional random fields: probabilistic models for segmenting and sequence data"

[7] Hara, K. (2006): "Applying a SVM based chunker and a text classifier to the Deid Challenge"

[8] Garner, J y Xiong Li (2008): "HIDE: An Integrated System for Health Information Deidentification"

[9] Yang H. y Garibaldi J. (2015): "Automatic Detection of Protected Health Information from Clinic Narratives"

[10] Bird S., Klein E., Loper E. (2009): "Natural Language Processing with Python." http://www.nltk.org/book_1ed

[11] spaCy 101: "Everything you need to know" https://spacy.io/usage/spacy-101

[12] CRFsuite - Documentation. http://www.chokkan.org/software/crfsuite/manual

[13] CRF - Wikipedia https://es.wikipedia.org/wiki/Campo_aleatorio_condicional

[14] Agencia Tributaria. Tabla de municipios

https://www.agenciatributaria.es/AEAT.internet/Inicio/Ayuda/Tablas_auxiliares_de_domicilios__provincias__municipios____/Tabla_de_Municipios/Tabla_de_Municipios.shtml

[15] NLNDE: http://ceur-ws.org/Vol-2421/MEDDOCAN_paper_5.pdf

[16] ReCRF: http://ceur-ws.org/Vol-2421/MEDDOCAN_paper_12.pdf

[17] Vicomtech at MEDDOCAN: http://ceur-ws.org/Vol-2421/MEDDOCAN_paper_8.pdf

[18] A Deep Learning-Based System: http://ceur-ws.org/Vol-2421/MEDDOCAN_paper_16.pdf

Anexos.

Anexo I: Valores PHI

’NOMBRE_SUJETO_ASISTENCIA’: "NOMS", (Nombre del paciente)

’EDAD_SUJETO_ASISTENCIA’: "EDADS", (Edad del paciente)

’SEXO_SUJETO_ASISTENCIA’: "SEXOS", (Sexo del paciente: “H” o “M”)

’FAMILIARES_SUJETO_ASISTENCIA’: "FAMS", (Parentescos)

’NOMBRE_PERSONAL_SANITARIO’: "NOMP", (Médico, Enfermero, etc.)

’FECHAS’: "FECH", (Fecha)

’PROFESION’: "PROF", (Profesión)

’CENTRO_SALUD’: "CENT", (Centro de Salud, sin incluir Hospitales)

’HOSPITAL’: "HOSP", (Hospital)

’INSTITUCION’: "INST", (Instituciones: Universidades, Clínicas, etc.)

’ID_TITULACION_PERSONAL_SANITARIO’: "IDTIT", (Id. numérico del personal sanitario)

’ID_EMPLEO_PERSONAL_SANITARIO’: "IDEMP", (Id. numérico del empleo)

’IDENTIF_VEHICULOS_NRSERIE_PLACAS’: "IDVEH", (Id. numérico de vehículos)

’IDENTIF_DISPOSITIVOS_NRSERIE’: " IDDISP", (Número de serie de dispositivos)

’CALLE’: "CALLE", (Nombre de direcciones postales)

’TERRITORIO’: "TER", (Municipios, Localidades y Código Postal)

’PAIS’: "PAIS", (Países)

’NUMERO_TELEFONO’: "NUMT", (Número de teléfono)

’NUMERO_FAX’: "NUMF", (Número de fax)

’CORREO_ELECTRONICO’: "EMAIL", (Dirección de correo electrónico)

’ID_SUJETO_ASISTENCIA’: "IDSUJ", (Identificador del paciente)

’ID_CONTACTO_ASISTENCIAL’: "IDCONT", (desconocido)

’NUMERO_BENEF_PLAN_SALUD’: "NUMB", (desconocido)

’ID_ASEGURAMIENTO’: "IDASEG", (desconocido)

’URL_WEB’: "URL", (Dirección página Web)

’DIREC_PROT_INTERNET’: "DIRECT", (desconocido)

’OTRO_NUMERO_IDENTIF’: "IDOTRO", (Id. distinto a los anteriores)

’OTROS_SUJETO_ASISTENCIA’: "OTROS" (desconocido)

Anexo II: Totales PHI (cuerpo)

’C−SEXOS’: 406,
’C−EDADS’: 509,
’D−EDADS’: 534,
’C−FAMS’: 236,
’D−FAMS’: 134,
’C−FECH’: 193,
’D−FECH’: 221,
’C−NOMP’: 15,
’D−NOMP’: 30,
’C−IDSUJ’: 41,
’D−IDSUJ’: 27,
’C−INST’: 50,
’D−INST’: 77,
’C−TER’: 59,
’C−PAIS’: 43,
’C−PROF’: 24,
’C−NOMS’: 11,
’D−SEXOS’: 2,
’D−TER’: 10,
’D−PROF’: 28,
’C−OTROS’: 9,
’D−OTROS’: 23,
’D−PAIS’: 4,
’C−HOSP’: 26,
’D−HOSP’: 81,
’C−IDTIT’: 4,
’D−IDTIT’: 7,
’C−CALLE’: 9,
’D−CALLE’: 45,
’C−NUMT’: 5,
’D−NUMT’: 8,
’C−EMAIL’: 7,
’D−EMAIL’: 1,
’D−NOMS’: 3,
’C−IDASEG’: 1,
’D−IDASEG’: 2