

Mecánica Computacional Vol XL, págs. 1137-1146 (artículo completo)  
Reconocimiento de Entidades en Relatos de Denuncias (A.S.)  
Mediante Aprendizaje Automático y Procesamiento de  
Lenguaje Natural  
Congreso de Ingeniería, 6-9 Noviembre 2023

ENTITY RECOGNITION IN THEFT REPORTS USING MACHINE LEARNING AND  
NATURAL LANGUAGE PROCESSING

Gustavo A. Rivera<sup>a</sup>, Edgar A. Rivera<sup>b</sup>, Eduardo Xamena<sup>b,c</sup> y Diego A. Rodríguez<sup>b</sup>

<sup>a</sup>Facultad de Ingeniería, Universidad Católica de Salta, Pellegrini 490, 4400 Salta, Argentina,  
riveramgustavo@gmail.com, <https://www.ucasal.edu.ar/>

<sup>b</sup>Departamento de Informática, Universidad Nacional de Salta, Av. Bolivia 5150, 4400 Salta,  
Argentina, riveramgustavo@gmail.com, riverae8080@gmail.com, examena@di.unsa.edu.ar,  
drodriguez@di.unsa.edu.ar, <http://www.unsa.edu.ar/>

<sup>c</sup>Inst. de Investigación en Cs. Soc. y Humanidades, Universidad Nacional de Salta - CONICET, Av.  
Bolivia 5150, 4400 Salta, Argentina, eduardoxamena@conicet.gov.ar, <http://www.icsoh.unsa.edu.ar/>

**Palabras clave:** Reconocimiento de Entidades Nombradas, Aprendizaje Automático,  
Procesamiento del Lenguaje Natural, Análisis de Denuncias.

**Resumen.** En el presente trabajo se muestran las fases del desarrollo de una plataforma de reconocimiento, clasificación y visualización de entidades en relatos de denuncias. Estos relatos fueron provistos por el Ministerio Público Fiscal de la provincia de Salta. Los cinco tipos de entidades identificadas y clasificadas son: Persona, Barrio, Documento Nacional de Identidad (DNI), Teléfono y Patente. Para el desarrollo del módulo de visualización se emplearon tecnologías web, y para las tareas de procesamiento del lenguaje natural se empleó la librería spaCy. Fueron entrenados distintos modelos machine learning para la tarea de extracción y clasificación de entidades sobre los relatos suministrados mediante el uso de spaCy. En cuanto a la performance del proceso de clasificación de entidades, los resultados obtenidos cumplen con los objetivos y criterios de éxito propuestos en las fases iniciales, por lo cual la implementación del sistema es factible, o al menos de los modelos entrenados, en fiscalías y dependencias policiales.

**Keywords:** Named Entities Recognition, Machine Learning, Natural Language Processing, Theft Reports Analysis.

**Abstract.** This work shows the different stages in the development of a Named Entities Recognition, Classification and Visualization platform over theft reports. These reports have been provided by Public Prosecutor's Office of Salta province, Argentina. The developed platform recognizes five entity types: Person, Neighborhood, National Identification Document (DNI), Phone and Car patent. A stack of web technologies was employed for visualization module development, and for Natural Language Processing tasks the SpaCy library was used. Several Machine Learning models were trained for the entities extraction and classification task. Regarding process performance for the recognition task, the obtained results fulfil the prior goals and criteria posed in early stages of the project. From this, the implementation of this platform or at least the trained models is feasible in prosecution offices or police stations.

## 1 INTRODUCCIÓN

En la actualidad, la gestión de la información relacionada con reportes de eventos delictivos es llevada a cabo mayormente de forma manual en las distintas dependencias de la justicia del ámbito salteño. Desde las búsquedas de información puntual acerca de un sospechoso, hasta la clasificación de un reporte por tipo de delito, requieren un esfuerzo cognitivo puramente humano. Al día de hoy, sin embargo, existen herramientas y metodologías que pueden ayudar a los agentes de la justicia en estas tareas, haciéndolas mucho más eficientes.

El trabajo llevado a cabo busca aportar una herramienta de reconocimiento y visualización de entidades, dirigida en primera instancia al Ministerio Público Fiscal (MPF), pero con potencial utilidad para otras dependencias del ámbito judicial y policial de la provincia de Salta, Argentina. En el MPF, entre fiscalías y dependencias policiales, se reciben alrededor de 350 denuncias por día, resultando sumamente compleja la tarea de realizar un análisis de todas ellas de forma manual.

Se detalla en este trabajo la elaboración de un sistema de Reconocimiento de Entidades Nombradas (NER por sus siglas en inglés) para los siguientes tipos de entidades en denuncias policiales: personas, barrios, números de DNI/CUIL/CUIT, patentes de vehículos y números de teléfono. Luego de la fase de reconocimiento, el usuario puede realizar las correcciones que considere necesarias utilizando la herramienta de visualización desarrollada también como parte del presente trabajo. Además, la plataforma genera resúmenes y gráficos sobre la cantidad de entidades encontradas de cada tipo. Estos datos pueden ser utilizados como información que apoye a la toma de decisiones.

Una visualización bien lograda permite reducir el tiempo y el esfuerzo requeridos para identificar datos de interés. Por ello, se considera de suma importancia la existencia de herramientas de visualización flexibles que faciliten el trabajo de las personas.

## 2 CONCEPTOS PREVIOS

### 2.1 Metodología CRISP-DM

La metodología CRISP-DM, del inglés Cross-Industry Standard Process for Data Mining (Chapman et al., 2000) provee un ciclo de vida iterativo para procesos de minería de datos independientes del dominio de aplicación y las tecnologías empleadas. Esta metodología cubre las distintas etapas de un proyecto, sus tareas respectivas y las relaciones entre estas tareas a través de 6 fases identificadas: Comprensión del negocio, Comprensión de los datos, Preparación de los datos, Modelado, Evaluación e Implementación.

### 2.2 Aprendizaje automático

El aprendizaje automático consiste en diseñar algoritmos que extraigan automáticamente información valiosa de un conjunto de datos. El énfasis aquí está en lo "automático", es decir, el foco en las metodologías de propósito general que se pueden aplicar a muchos conjuntos de datos, mientras se produce algo significativo. Hay tres conceptos que están en el centro del aprendizaje automático: datos, un modelo y aprendizaje (Deisenroth, Faisal, and Ong, 2020, p. 11).

Dado que el aprendizaje automático se basa intrínsecamente en datos, estos constituyen su núcleo. Su objetivo principal es diseñar metodologías de propósito general para extraer patrones valiosos de los datos, idealmente sin mucho conocimiento sobre un dominio específico. Por ejemplo, dado un gran corpus de documentos como puede ser un volumen de libros en diversas bibliotecas, los métodos de aprendizaje automático se pueden usar para encontrar automáticamente temas relevantes que se comparten entre los mismos (Hoffman, Blei, and

Back, 2010). Para lograr este objetivo, se diseñaron modelos que suelen estar relacionados con el proceso que genera los datos, similares al conjunto de datos que se proporciona. Por ejemplo, en una configuración de regresión, el modelo describiría una función que mapea entradas a salidas de valor real. Parafraseando a Mitchell (1997): Se dice que un modelo aprende de los datos si su desempeño en una tarea dada mejora después de que se toman en cuenta los datos. El objetivo es encontrar buenos modelos que generalicen bien los datos aún no vistos, que pueden interesarnos en el futuro. El aprendizaje puede entenderse como una forma de encontrar automáticamente patrones y estructura en los datos optimizando los parámetros del modelo (Deisenroth, Faisal, and Ong, 2020, p. 11).

### 2.2.1 Aprendizaje supervisado

Hay tres clases principales de técnicas de aprendizaje automático: aprendizaje supervisado, aprendizaje no supervisado y aprendizaje reforzado. En el aprendizaje supervisado, el conjunto de entrenamiento consta de pares de entrada y salida, y el objetivo es aprender un mapeo entre los espacios de entrada y salida (Simeone, 2018, p. 2).

En este trabajo se realizan tareas de reconocimiento de entidades. Para este tipo de tareas suele emplearse aprendizaje supervisado, indicándole al modelo cuales son las entidades que debería reconocer (salida) en un texto dado (entrada). Los pares de entrada y salida deseada se obtienen mediante el proceso de anotación y luego se suministran al modelo como lotes de entrenamiento y validación para que automáticamente modifique sus parámetros y mejore sus predicciones (Figura 1).

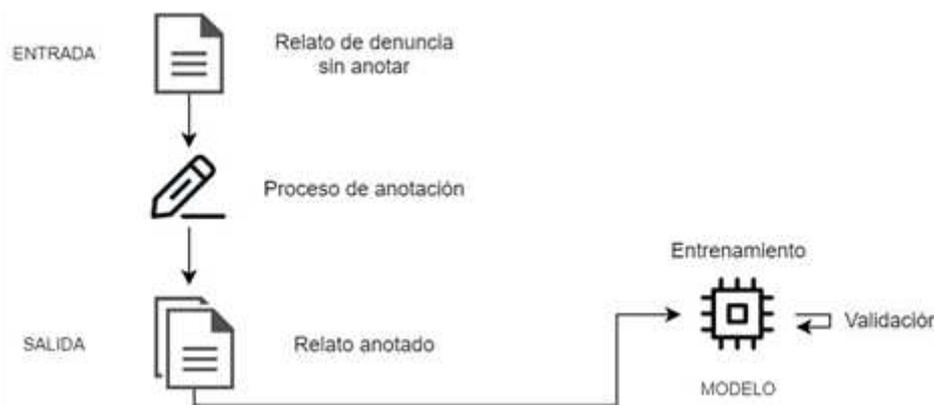


Figura 1: Proceso de anotación, entrenamiento y validación – Aprendizaje Supervisado.

### 2.2.2 Redes Neuronales

Una red neuronal es un tipo particular de modelo de aprendizaje automático. Las redes neuronales son especiales debido a su alta flexibilidad, esto significa, que pueden resolver una gama inusualmente amplia de problemas con solo encontrar los pesos correctos (Howard and Gugger, 2020). Los pesos o parámetros son variables cuyos valores se definen en una instancia previa al entrenamiento del modelo. Por cada valor diferente que se asigne a cada peso, se obtiene un modelo distinto a los demás, cuyas predicciones, una vez entrenado, pueden variar significativamente. Generalmente estos pesos se inicializan con un valor aleatorio y luego se van ajustando automáticamente a través de una técnica matemática, conocida como descenso de gradiente estocástico (SGD).

### 2.2.3 Aprendizaje Profundo

El Aprendizaje profundo es una especialidad dentro del aprendizaje automático que utiliza redes

neuronales con múltiples capas. La clasificación de imágenes es un ejemplo representativo (también conocido como reconocimiento de imágenes). Comenzamos con datos etiquetados: un conjunto de imágenes para las que hemos asignado una etiqueta a cada imagen, indicando lo que representa. Nuestro objetivo es producir un programa, llamado modelo, que, dada una nueva imagen, hará una predicción precisa sobre lo que representa esa nueva imagen (Howard and Gugger, 2020).

#### 2.2.4 Evaluación

A la hora de evaluar los modelos de Aprendizaje Automático, incluidos los modelos de Reconocimiento de Entidades que se utilizan en este proyecto, se pueden utilizar diferentes métricas que ayuden a tener una noción de cómo se está desempeñando el modelo entrenado. Dentro de las métricas más populares encontramos cuatro: accuracy, precision, recall y F1 score.

Para una tarea de clasificación, como el reconocimiento de entidades donde se busca predecir si una palabra pertenece o no a una determinada clase, podemos distinguir las siguientes cuatro situaciones: Los verdaderos positivos (TP) y los verdaderos negativos (TN) son clasificaciones correctas. Un falso positivo (FP) es un resultado que se predice incorrectamente como positivo cuando en realidad es negativo. Un falso negativo (FN) representa la situación en que el resultado se predice incorrectamente como negativo cuando en realidad es positivo. (Witten et al., 2017, p. 180).

### 2.3 Procesamiento del Lenguaje Natural

El procesamiento del lenguaje natural (PLN) es un campo en la intersección de la Informática, la Inteligencia Artificial y la Lingüística. Se ocupa del desarrollo de sistemas que puedan analizar, modelar y comprender el lenguaje humano. Toda aplicación inteligente que involucre lenguaje humano tiene algo de PLN detrás (Vajjala et al., 2020, pp. 11,27,28).

#### 2.4 Extracción de Información

Muchos documentos breves describen un tipo particular de objeto o evento, combinando entidades en un compuesto de nivel superior que representa todo el contenido del documento. La tarea de identificar la estructura compuesta, que a menudo se puede representar como una plantilla con espacios que se llenan con piezas individuales de información estructurada, se denomina extracción de información (EI). Una vez que se han encontrado las entidades, se analiza el texto para determinar las relaciones entre ellas. (Witten et al., 2017, p. 513).

#### 2.5 Conceptos de SpaCy

SpaCy es una biblioteca de software de código abierto para procesamiento avanzado de lenguaje natural en Python (Explosion, 2021). Un concepto esencial en spaCy es el de *token*. Este se menciona por primera vez en el proceso de *tokenización*, el cual se define como la tarea de dividir un texto en segmentos con significado llamados tokens. Esto se consigue aplicando reglas específicas a cada lenguaje.

### Fórmula de Aprendizaje Profundo para modelos de NLP

Algunos modelos de SpaCy utilizan aprendizaje profundo en un proceso de 4 pasos: *embed*, *encode*, *attend*, *predict*. Las representaciones de palabras incrustadas (*word embeddings*),

también conocidas como "vectores de palabras", son una de las tecnologías de procesamiento del lenguaje natural más utilizadas. Las incrustaciones de palabras permiten tratar palabras individuales como unidades de significado relacionadas, en lugar de identificadores completamente distintos. Sin embargo, la mayoría de los problemas de PLN requieren la comprensión de tramos de texto (*spans*) más largos, no solo palabras individuales. Existe para ello una solución simple y flexible que está logrando un rendimiento excelente en una amplia gama de problemas. Después de incrustar el texto en una secuencia de vectores, se utilizan Redes Neuronales Recurrentes (RNN) bidireccionales, un tipo de Red Neuronal eficiente para tareas de PLN, para codificar los vectores en una matriz de frases. Las filas de esta matriz pueden entenderse como vectores de tokens: son sensibles al contexto de oración del token. La siguiente fase en el proceso se llama mecanismo de atención, herramienta fundamental en la arquitectura transformer para el intercambio de información a lo largo de las representaciones de datos. Con este pipeline de procesos se logran representaciones útiles para la fase de predicción (Honnibal, 2016).

En este proyecto se utilizan dos componentes de spaCy: Tok2Vec y NER. Estos son modelos de redes neuronales que siguen una estructura similar a la descrita por Honnibal. Puntualmente, dentro de la documentación de spaCy, las 5 arquitecturas de redes neuronales que aquí se emplean son:

- *TransitionBasedParser.v2*: Crea un modelo analizador basado en transiciones. Se puede aplicar a NER o análisis de dependencia. El análisis sintáctico basado en transiciones es un enfoque de la predicción estructurada en el que la tarea de predecir la estructura se asigna a una serie de transiciones de estado. Está compuesta por 3 sub redes neuronales que reciben los nombres de *tok2vec*, *lower* y *upper*.
- *Tok2VecListener.v1*: Un oyente se utiliza como subcapa dentro de un componente como *EntityRecognizer* (el que se utiliza para la tarea de NER). Por lo general, tendrá varios oyentes que se conectarán a un único componente Tok2Vec ascendente que se encuentra anteriormente en el flujo. Las capas de escucha actúan como *proxies*, pasando las predicciones del componente Tok2Vec a componentes descendentes y comunicando gradientes ascendentes. En este caso solamente se utiliza un componente adicional que nos permite realizar la tarea de NER, por lo cual solo se emplea un oyente en dicho componente.
- *Tok2Vec.v2*: Construye un modelo tok2vec a partir de dos subredes: una para incrustar (*embedding*) y otra para codificar (*encoding*).
- *MultiHashEmbed.v2*: Construye una capa de incrustación (*embedding*) que incrusta por separado una serie de atributos léxicos mediante la incrustación de hash, concatena los resultados y los pasa a través de una subred de avance para crear una representación mixta.
- *MaxoutWindowEncoder.v2*: Codifica el contexto utilizando convoluciones con activación máxima (*maxout activation*), normalización de capas y conexiones residuales. Estos últimos son conceptos normalmente utilizados en un tipo de red neuronal conocida como Redes Neuronales Convolucionales.

### 3 PROPUESTA

Se propone la generación de un modelo que permita identificar las entidades existentes en relatos de denuncias y aplicar técnicas de Reconocimiento de Entidades y visualización, para que, mediante un sistema web gestionado por el Ministerio Público Fiscal de la Provincia de Salta, el usuario pueda acceder a información que ayude a la toma de decisiones.

#### 3.1 Metodología

El trabajo fue desarrollado siguiendo las pautas establecidas por la metodología CRISP-DM. A continuación, se presentan los aspectos más relevantes de las distintas fases.

##### 3.1.1 Comprensión del negocio

Para cumplir con los objetivos pautados, se estableció un plan de proyecto que se resume en la Figura 2.

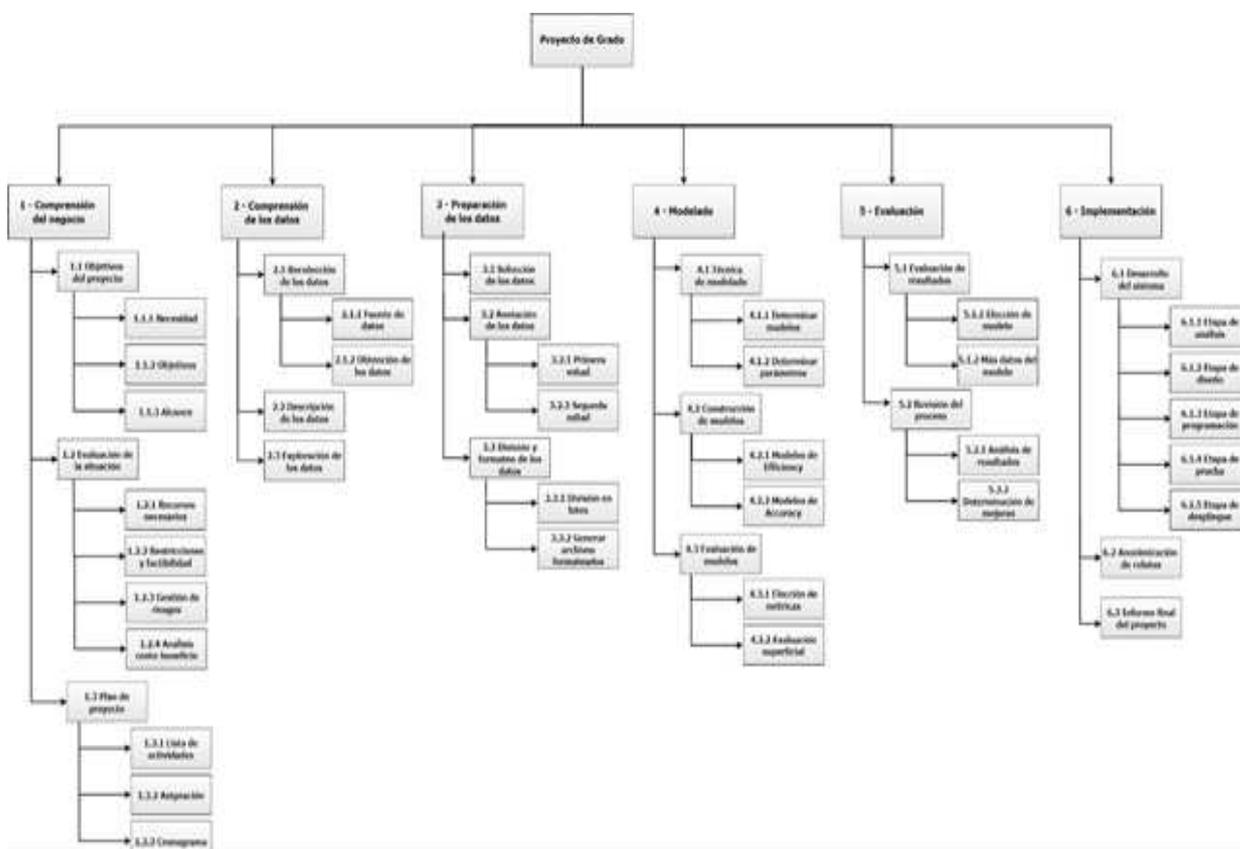


Figura 2: Estructura de trabajo para el proyecto.

##### 3.1.2 Comprensión de los datos

Previo acuerdo de confidencialidad con personal de Informática del Ministerio Público Fiscal de la Provincia de Salta, se obtuvo de ellos un lote de 35628 relatos de denuncias, todas ellas con fecha de registro en el año 2016. Cada registro presentaba los siguientes atributos: Id de denuncia, Año, Id de Modalidad, Modalidad, Id de Tipo de Lugar, Tipo de Lugar y Relato.

### 3.1.3 Preparación de los datos

Del total de 35628 denuncias, se seleccionaron 450 para las etapas de entrenamiento, validación y prueba de modelos, y otras 70 las cuales pasaron por un proceso de anonimización a fin de ser cargadas en el Sistema.

### 3.1.4 Modelado

La técnica de modelado consistió en realizar distintos tipos de entrenamientos con los relatos anotados, organizados en lotes de dos tamaños diferentes: 150 y 450. Partiendo de estos datos la técnica consistió en entrenar un total de 12 modelos: 2 de ellos para las pruebas con 150 relatos y 10 con 450 relatos. En la Tabla 1 se resumen las distintas configuraciones de modelos entrenados y validados. Se trata de un total de 12 modelos, 5 optimizados para exactitud y 7 para eficiencia. Para evaluar los modelos se utilizó la métrica F1 Score, Precision y Recall.

| Optimizado para                | Modelo | Cantidad de Relatos | Archivo config.cfg                            | Separación en lotes             |
|--------------------------------|--------|---------------------|---|---------------------------------|
| <b>Exactitud (Accuracy)</b>    | 1      | 150 (96, 24, 30)    | Con valores por defecto                       | Al azar                         |
|                                | 2      | 450 (350, 50, 50)   | Con valores por defecto                       | Al azar                         |
|                                | 3      | 450 (350, 50, 50)   | Con valores por defecto                       | Balanceo de entidades           |
|                                | 4      | 450 (350, 50, 50)   | Con valores por defecto                       | Balanceo de entidades con pesos |
|                                | 5      | 450 (350, 50, 50)   | Con valores por defecto<br>Learn rate: 0.0003 | Balanceo de entidades con pesos |
| <b>Eficiencia (Efficiency)</b> | 6      | 150 (96, 24, 30)    | Con valores por defecto                       | Al azar                         |
|                                | 7      | 450 (350, 50, 50)   | Con valores por defecto                       | Al azar                         |
|                                | 8      | 450 (350, 50, 50)   | Con valores por defecto                       | Balanceo de entidades           |
|                                | 9      | 450 (350, 50, 50)   | Con valores por defecto                       | Balanceo de entidades con pesos |
|                                | 10     | 450 (350, 50, 50)   | Con valores por defecto<br>Learn rate: 0.001  | Balanceo de entidades con pesos |
|                                | 11     | 450 (350, 50, 50)   | Con valores por defecto<br>Learn rate: 0.003  | Balanceo de entidades con pesos |
|                                | 12     | 450 (350, 50, 50)   | Con valores por defecto<br>Learn rate: 0.0003 | Balanceo de entidades con pesos |

Tabla 1: Configuraciones de modelos entrenados.

### 3.1.5 Fase 5: Evaluación

En la Tabla 2 se resumen los puntajes obtenidos para cada uno de los 12 modelos, ordenados del mayor puntaje al menor. El modelo con mejor puntaje en la métrica F1 Score y Recall es el número 5. Para la métrica Precision el mejor es el modelo 4, pero el que le sigue es el 5 nuevamente.

### 3.1.6 Fase 6: Implementación

Las herramientas que se utilizaron para el desarrollo del trabajo fueron las siguientes: Python como lenguaje de programación, utilizando el framework Flask para el backend y la librería spaCy para procesamiento avanzado de lenguaje natural. Para desplegar y administrar la base de datos se utilizó MongoDB. Para el frontend se utilizó React, con Recharts como biblioteca de gráficos.

| Modelo    | Precision    | Recall       | F1 Score     |
|-----------|--------------|--------------|--------------|
| <b>5</b>  | 89.05        | <b>81.88</b> | <b>85.31</b> |
| <b>4</b>  | <b>90.08</b> | 73.15        | 80.74        |
| <b>12</b> | 82.01        | 76.51        | 79.17        |
| <b>2</b>  | 79.87        | 77.44        | 78.64        |
| <b>9</b>  | 88.89        | 69.80        | 78.20        |
| <b>11</b> | 83.47        | 67.79        | 74.81        |
| <b>3</b>  | 82.64        | 68.03        | 74.63        |
| <b>7</b>  | 78.32        | 68.29        | 72.96        |
| <b>10</b> | 78.57        | 66.44        | 72.00        |
| <b>8</b>  | 83.64        | 62.59        | 71.60        |
| <b>1</b>  | 84.38        | 48.21        | 61.36        |
| <b>6</b>  | 50.00        | 32.14        | 39.13        |

Tabla 2: Resumen de puntajes obtenidos para los 12 modelos entrenados.

## 4 RESULTADOS

Del resumen de puntajes obtenidos por los distintos modelos que se presentaron en la sección anterior, se observa que el modelo con mejor puntaje en la métrica F1 Score y Recall es el número 5. Para la métrica Precision el mejor es el modelo 4, pero el que le sigue es el 5 nuevamente. De aquí podemos concluir que los mejores resultados según estas métricas los obtenemos con el modelo 5.

Para los modelos 4, 5 y 12 se utilizó el lote de 50 relatos de prueba con los cuales se generó una matriz de confusión. La Figura 3 presenta la matriz de confusión del modelo 5 a partir de 50 relatos. En los ejes de la matriz se presentan los 5 tipos de entidades más la entidad "O", la cual representa los tokens que no indican presencia de ningún tipo de entidad, "Outer" por su equivalente en inglés. Por lo tanto, si a una palabra se la etiqueta como "O" es equivalente a decir que el modelo consideró que esa palabra no pertenece a ninguno de los 5 tipos de entidades analizados.

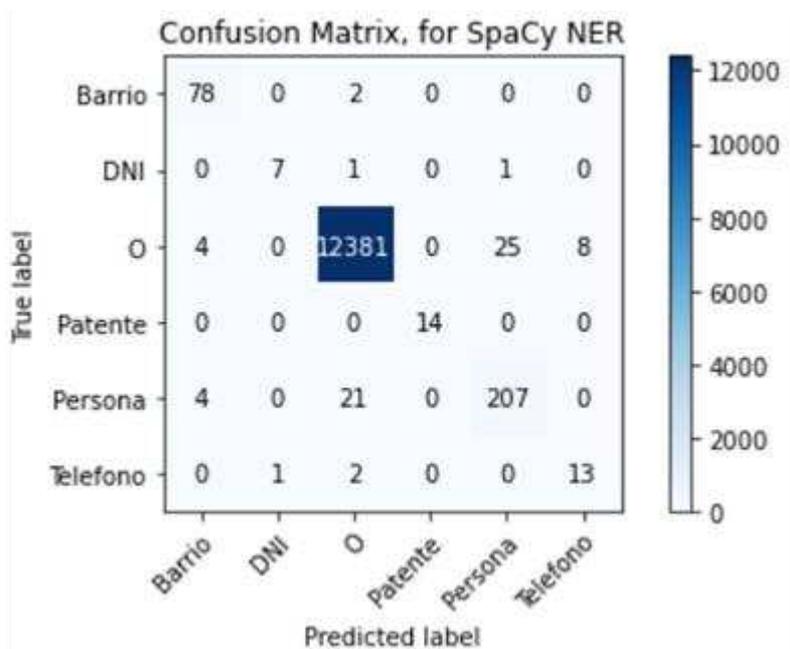


Figura 3: Matriz de confusión del Modelo 5.

En la Tabla 3 se muestran los resultados del modelo 5, el cuál obtuvo los mejores valores en las diferentes métricas empleadas. En particular, la Tabla 3 exhibe los resultados desagregados por tipo de entidad. Las métricas reportadas son nuevamente Precision, Recall y F1 Score, en este caso para cada tipo de entidad por separado.

| Tipo de entidad | Precision | Recall | F1-Score |
|-----------------|-----------|--------|----------|
| <b>Persona</b>  | 0,888     | 0,892  | 0,890    |
| <b>Barrio</b>   | 0,907     | 0,975  | 0,940    |
| <b>DNI</b>      | 0,875     | 0,778  | 0,824    |
| <b>Patente</b>  | 1,000     | 1,000  | 1,000    |
| <b>Teléfono</b> | 0,812     | 0,812  | 0,812    |

Tabla 3: Resumen de puntajes obtenidos para el modelo 5, por tipo de entidad.

## 5 CONCLUSIONES

Los grandes avances en investigaciones de Aprendizaje Automático han causado que cada día más aplicaciones y software en general, implementen funcionalidades “inteligentes”. En este trabajo se desarrolló un sistema en el cual se implementa un modelo de detección y clasificación de entidades nombradas para asistir al personal de distintas reparticiones judiciales de la provincia de Salta, Argentina. Se obtuvieron muy buenos resultados para uno de los modelos entrenados en particular, lo cual es un resultado muy importante para el futuro de esta tarea en el contexto mencionado. Los valores en diferentes métricas superan en muchos casos el 90% de desempeño, mostrando que existen patrones marcados en el lenguaje del personal policial a

la hora de emitir reportes de denuncias, y que puede obtenerse un aumento considerable en la eficiencia del trabajo cotidiano para estos agentes.

A futuro se pretende realizar las gestiones para obtener una mayor cantidad de datos, aplicar técnicas de aumentación de datos para incrementar el lote de entrenamiento y validación e incluso el de prueba y utilizar otras librerías de aprendizaje automático, como lo son Scikit-Learn , TensorFlow o PyTorch en lugar de SpaCy.

## 6 AGRADECIMIENTOS

Este trabajo fue realizado con fondos del Proyecto CIUNSa B-2825/0 y Proyecto PUE 22920160100056CO (CONICET - Proyecto de Unidad Ejecutora,ICSOH).

## REFERENCIAS

Chapman, P. et al. (2000) *CRISP-DM 1.0: Step-by-step data mining guide*. SPSS Inc.

Deisenroth, M. P., Faisal, A. A., and Ong, C. S. (2020). *Mathematics for Machine Learning*. Cambridge: Cambridge University Press.

Explosion (2016). *spaCy 101: Everything you need to know*. Available at: <https://spacy.io/usage/spacy-101#whats-spacy> (Accessed: 24 August 2023).

Hoffman, M. D., Blei, D. M., and Back, F. (2010). *Online learning for Latent Dirichlet Allocation*. *Advances in Neural Information Processing Systems*. Curran Associates Inc.

Available at:

[https://proceedings.neurips.cc/paper\\_files/paper/2010/file/71f6278d140af599e06ad9bf1ba03cb0-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2010/file/71f6278d140af599e06ad9bf1ba03cb0-Paper.pdf) (Accessed: 24 August 2023).

Honnibal, M. (2016). *Embed, encode, attend, predict: The new deep learning formula for state-of-the-art NLP models*. Available at: <https://explosion.ai/blog/deep-learning-formula-nlp> (Accessed: 24 August 2023).

Howard, J., and Gugger, S. (2020). *Deep Learning for coders with FastAI & PyTorch AI applications without a PhD*. Sebastopol, CA: O'Reilly Media.

Simeone, O. (2018). A very brief introduction to Machine Learning with applications to Communication Systems. *IEEE*.

Vajjala, S. et al. (2020). *Practical Natural Language Processing a comprehensive guide to building real-world NLP systems*. Sebastopol, CA: O'Reilly Media.

Witten, I. H. et al. (2017) *Data Mining practical Machine Learning tools and techniques*. 4th edn. Cambridge: Morgan Kaufmann.