UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Reconocimiento de entidades nombradas mediante técnicas de aprendizaje neuronal profundo en imágenes manuscritas

Trabajo Fin de Grado

Grado en Ciencia de Datos

AUTOR/A: Giner Perez de Lucia, Jose

Tutor/a: Sánchez Peiró, Joan Andreu

CURSO ACADÉMICO: 2021/2022

Resumen

El reconocimiento de entidades nombradas o NER tiene como misión extraer entidades específicas en grandes colecciones de textos y clasificarlas según su campo semántico. Las entidades más comunes incluyen personas, localizaciones y organizaciones, aunque pueden variar dependiendo de los requisitos de la tarea. Con el paso de los años, las tecnologías especializadas en NER se han desarrollado rápidamente para abordar nuevos retos y lograr unos resultados propios del estado del arte, pasando por sistemas basados en reglas creadas a mano hasta alcanzar los modelos más complejos de aprendizaje neuronal profundo. Estos últimos son la referencia de muchas investigaciones actuales por presentar unos mecanismos potentes capaces de aprender del contexto de las palabras y apoyarse en representaciones distribuidas de características en un espacio latente.

Mediante este trabajo final de grado, se presenta una arquitectura basada en el concepto de red neuronal para identificar las entidades nombradas en una colección antigua de licencias matrimoniales manuscritas en catalán. En concreto, se propone un red con memoria a corto y largo plazo bidireccional (Bi-LSTM) y un campo aleatorio condicional (CRF) en la capa final como decodificador de etiquetas. Los resultados obtenidos reflejan las buenas prestaciones de reconocimiento sobre las categorías semánticas y de estas junto con la persona asociada cuando las transcripciones no contienen fallos (errores del 2.05 % y 2.34 %, respectivamente). Por otra parte, se ha evaluado el rendimiento del modelo con unas transcripciones generadas por un proceso de reconocimiento de texto manuscrito que pueden presentar errores. Ante esta situación, las etiquetas de salida predichas también se ajustan adecuadamente a cada palabra.

Palabras clave: Reconocimiento de entidades nombradas, Aprendizaje neuronal profundo, Licencias matrimoniales, Bi-LSTM-CRF

Abstract

Named Entity Recognition (NER) attains to extract and classify specific entities on large text collections according to its semantic field. Most common entities include persons, places and organizations but may vary depending on the task requirements. Over the years, specialized NER technologies have rapidly developed to tackle upcoming challenges and achieve state-of-art results, covering rule-based systems that are created using hand-crafted rules until reaching more complex deep learning models. These last ones serve as a reference to many actual investigations as they present powerful mechanisms capable of learning from word context and rely on distributed feature representations in a latent space.

Through this final degree project, an architecture based on the neural network concept is presented to identify named entities in an old catalan collection of handwritten marriage licenses. Specifically, a bidirectional Long-Short Term Memory (Bi-LSTM) network with a Condition Random Field (CRF) in the final layer as a tag decoder is proposed. Results obtained reflect a good recognition performance on both semantic categories and these ones together with the associated person when transcripts do not contain mistakes (error rates of 2.05% and 2.34%, respectively). On the other hand, the model's performance has been tested with some transcripts generated by a handwritten text recognition process that can present errors. In this situation, the predicted output labels also fit appropiately for each word.

Key words: Named entity recognition, Deep learning, Marriage licenses, Bi-LSTM-CRF

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO 1
Introducción

Con la gran cantidad de información digital existente a día de hoy, son muchos los sistemas encargados de extraer conocimiento de los datos mediante la identificación de patrones relevantes para automatizar la toma de decisiones con la menor intervención humana posible. Una tarea para la cual se especializan estos sistemas es en el reconocimiento de entidades nombradas, comúnmente conocida como Named Entity Recognition (NER) [8], que es un campo importante del procesamiento del lenguaje natural (PLN). Implica la detección de entidades de interés en textos y su clasificación en diferentes categorías semánticas. Las entidades representan términos claramente distinguibles entre sí y son fundamentales para entender el significado de un mensaje, pudiendo ser nombres propios de personas, localizaciones y cantidades numéricas. Algunas aplicaciones de este problema se encuentran en la recuperación de la información en documentos legales [9, 10], financieros [11] y médicos [12, 13], sistemas de respuesta automática [14] o en sistemas de recomendación [15] donde el almacenamiento de entidades puede ayudar a revelar gustos y aficiones.

A pesar de las áreas de investigación más recientes en las que el reconocimiento de entidades nombradas proporciona grandes éxitos, su rendimiento es objeto de estudio en la extracción de información relevante presente en imágenes de texto manuscrito antiguo. A diferencia de los textos electrónicos encontrados en la web, los documentos históricos requieren técnicas de reconocimiento de texto manuscrito para obtener las transcripciones textuales que en ocasiones pueden contener fallos propios del proceso de reconocimiento.

1.1 Motivación

El reconocimiento de entidades nombradas permite dar una visión general de aquellos términos que son útiles para entender la temática principal del texto del que se dispone y poder clasificarlo según las entidades extraídas. Implica el estudio y comprensión del lenguaje natural por parte de un algoritmo inteligente que pueda aprender de la estructura narrativa común que caracteriza a los textos. Con un adecuado proceso de entrenamiento, el sistema tiene que detectar y categorizar entidades en textos que no ha manejado anteriormente. A pesar de que es posible identificar dichas entidades de forma manual con una simple lectura sobre el documento, esta opción no es factible cuando hay que tratar con grandes colecciones de textos como pueden ser los miles de comentarios o reseñas escritas por clientes acerca de un producto o servicio, donde la automatización en la detección de entidades relevantes ahorra un tiempo muy valioso. Dotando de las técnicas más modernas de la inteligencia artificial, en especial del aprendizaje neuronal profundo, se han logrado unos resultados de reconocimiento muy prometedores por los que merece la pena implementar e investigar estos sistemas en diferentes dominios.

En caso de procesar imágenes de documentos manuscritos, la transcripción final puede contener errores causados por el programa o modelo probabilístico de reconocimiento de texto. Por tanto, es interesante analizar cómo afecta la calidad de la transcripción a la detección de entidades nombradas cuando no se tiene exactamente el texto original de entrada, existiendo palabras eliminadas, mal escritas o erróneamente separadas.

La motivación profesional nace de la necesidad de mejorar los algoritmos existentes y proponer soluciones de última generación para presentar oportunidades en diversos negocios con el proceso de la transformación digital, ahorrando tiempos y costes. Cada vez más empresas optan por el uso de herramientas de procesamiento del lenguaje natural en sus actividades diarias, por lo que dotar de experiencia y de conocimientos en este campo de investigación puede ser beneficioso.

1.2 Objetivos

Este proyecto tiene como primer objetivo extraer y clasificar entidades nombradas mediante técnicas de procesamiento del lenguaje natural y aprendizaje neuronal profundo. Los textos empleados para la tarea son unos registros matrimoniales manuscritos del siglo XVII de un volumen perteneciente a una colección de los Libros de Esposalles, ubicados en los Archivos de la Catedral de Barcelona. A partir de los registros anotados, se tratará de diseñar una red neuronal para el reconocimiento de entidades nombradas y estudiar su rendimiento.

El segundo objetivo es poner el modelo a prueba con otra versión de las transcripciones que pueden presentar errores en ciertas palabras y con ello comprobar la posible caída de rendimiento en la detección de entidades. Dichas transcripciones son las hipótesis que mejor se adecuan a los textos manuscritos matrimoniales.

1.3 Impacto esperado

A través de este trabajo, se espera agilizar el proceso de búsqueda y recuperación de entidades relevantes en altos volúmenes de documentos de diferentes contenidos e idiomas gracias a las técnicas más modernas de la Inteligencia Artificial. En vez de etiquetar las entidades manualmente y guardar la información en una base de datos, se busca automatizar este trabajo realmente costoso. Al conseguir un modelo capaz de reconocer entidades nombradas entrenado con textos de una lengua y época específica, se espera que pueda ser evaluado en otros dominios de contexto similar como pueden ser en los registros matrimoniales localizados en el resto de libros de los Archivos de la Catedral de Barcelona. Con las frases etiquetadas por un sistema inteligente en transcripciones de textos antiguos, los historiadores pueden beneficiarse a la hora de consultar ciertas entidades cuya categoría pueda ser desconocida y clave para entender un hecho de interés. Además, apoyándose en las predicciones realizadas por el modelo, se espera que los textos o documentos puedan ser clasificados según las entidades que aparezcan en cada uno y encontrar relaciones entre ellas.

1.4 Metodología

Para obtener los resultados de clasificación y extraer conclusiones acerca del rendimiento del modelo, se ha seguido una metodología característica de cualquier tarea rela-

cionada con el procesamiento del lenguaje natural y aprendizaje automático, pero enfocada a la tarea en cuestión:

1. Leer y almacenar la información de los registros matrimoniales, que incluye la transcripción a nivel de palabra, etiquetas de categorías semánticas y de personas. Asociar a cada palabra su categoría semántica y persona relacionada como etiquetas de referencia.

2. Analizar la naturaleza de las transcripciones en cuanto a la estructura narrativa, vocabulario empleado y términos frecuentes. Esto ayudará a observar patrones comunes de palabras que pueden ser aprendidos por las redes neuronales. También, comprobar las distribuciones de longitudes de registros y frecuencias de etiquetas.

3. Codificar los caracteres, palabras y etiquetas de los registros para que puedan ser interpretados como entradas y salidas por el modelo de aprendizaje profundo. Para ello, se tienen que definir unas correspondencias con índices, permitiendo mostrar las palabras originales junto con sus etiquetas reales y predichas.

4. Fijar una longitud vectorial de palabras por registro para que ésta coincida con la dimensión de entrada esperada por la red neuronal. De esta forma, todos los registros tienen el mismo tamaño y en aquellos casos donde no se alcance la longitud establecida, se completa el vector con palabras de relleno que tienen un índice reservado. Lo mismo se aplica para los caracteres de una palabra.

5. Emplear la técnica de validación cruzada para entrenar y evaluar el modelo con 5 particiones distintas de registros (*k-fold Cross Validation*) para obtener una precisión y error final promedio. Así se consigue una estimación más precisa del error de predicción al reducir el sesgo de entrenamiento y validación, maximizando el ajuste del modelo.

6. Definir la composición de la red neuronal y evaluarla combinando todas las predicciones hechas sobre las particiones de validación. De este modo se consideran todas las muestras para determinar el rendimiento global.

7. Obtener una configuración final de hiperparámetros que den el mejor rendimiento posible.

Un proceso similar se sigue para obtener las predicciones de las mejores hipótesis generadas. La diferencia principal es que solo son utilizadas para evaluar el modelo ya que es entrenado con los registros auténticos. Debido a que pueden existir nuevos términos que no han sido vistos por el modelo, es necesario reservar otro índice para identificarlos.

1.5 Estructura de la memoria

El contenido del trabajo se segmenta de la siguiente manera:

- En el capítulo 2 se exponen algunos eventos cronológicos de NER surgidos a raíz de su creación y trabajos relacionados con distintos enfoques a la tarea. También se critica al estado del arte y se define la dirección que toma la propuesta para el trabajo en cuanto a los enfoques vistos.

- En el capítulo 3 se analiza con más detalle en qué consiste el reconocimiento de entidades nombradas o NER. Además, se valora el cumplimiento del margo legal

y ético en relación con la protección de los datos a tratar y posibles sesgos introducidos en el modelo. Luego se señalan algunas dificultades para NER en documentos históricos manuscritos como los que se disponen y se analizan las posibles soluciones para esta tarea relacionadas con los enfoques cubiertos en el capítulo 2, indicando la opción elegida.

- En el capítulo 4 se describe el origen y las propiedades del conjunto de datos utilizado, consistiendo en unas licencias matrimoniales para detectar las entidades nombradas. También se realizará un análisis exploratorio para entender mejor la colección de la que se dispone y revelar información sobre el vocabulario y etiquetas empleadas.

- En el capítulo 5 se introducen algunos conceptos fundamentales propios de cualquier sistema basado en el aprendizaje neuronal profundo como el que se ha desarrollado y al final se presentan algunas variantes de redes neuronales tradicionales muy populares que procesan estructuras de datos entrantes específicas para abordar distintas tareas.

- En el capítulo 6 se describen las componentes principales del modelo basado en NER. Primero se explica como se codifican las palabras y caracteres de cada registro junto a sus etiquetas, luego las representaciones distribuidas de características y su función en la red neuronal, el codificador de contexto, decodificador de etiquetas y finalmente se facilita una visión completa del modelo.

- En el capítulo 7 se analiza como parte de la experimentación el efecto y función de cada hiperparámetro del modelo además de señalar la configuración elegida. También se comenta con más detalle en que consiste la estrategia de entrenamiento y validación, la función de pérdida utilizada para ajustar los pesos en cada capa, las métricas de evaluación seleccionadas, los resultados finales sobre los registros matrimoniales y las mejores hipótesis.

- Por último, en el capítulo 8 se concluye el trabajo analizando la efectividad del modelo. Así mismo, se referencia el código utilizado para la tarea, se destaca la relación del trabajo con los estudios cursados, las competencias transversales más relevantes que se han puesto en práctica y se proponen trabajos futuros de desarrollo a partir del realizado.

CAPÍTULO 2
Contexto tecnológico

El término de "entidad nombrada" se introdujo por primera vez en la sexta edición de la Conferencia de Comprensión de Mensajes para la informática (MUC-6) [16] organizada en Columbia, Maryland (EE.UU) en 1995 para la evaluación del progreso en sistemas de recuperación de la información. Fue allí donde se establecieron los principales tipos de entidades nombradas incluyendo personas, organizaciones, lugares y ciertas cantidades numéricas. Desde entonces, han surgido múltiples eventos y avances en cuanto a los mecanismos diseñados para revelar entidades, siendo aplicados a textos de diferentes lenguas y contextos. Por ejemplo, los sistemas presentados por Curran y Clark [17] para evaluar el reconocimiento de entidades en el conjunto de datos CoNLL-2003 constando de textos en inglés y alemán, implementaron diferentes técnicas de aprendizaje automático como modelos de máxima entropía y modelos ocultos de Markov. El programa ACE [18] presentado en 2004 se centró en el reconocimiento de entidades, relaciones entre ellas y eventos en diferentes fuentes de información como imágenes, audios y textos en árabe y chino, además del inglés. Tareas de búsqueda de entidades en la web fueron realizadas en el taller de INEX 2009 [19], que consistió en clasificar páginas de la Wikipedia mediante la detección de entidades en sus textos, devolviendo un ranking con relevancia de cada página y se crearon algunos modelos generativos para detectar relaciones entre entidades. Otro evento similar fue el TREC 2010 Entity Track [20], donde los participantes tenían que encontrar las páginas con mayor coincidencia a diferentes consultas dadas en base a sus entidades nombradas presentes. La competición de extracción de información de FactRuEval llevada a cabo en 2016 [21] dio un paso adelante, al pedir a sus participantes no solo reconocer la entidades nombradas en textos sino que también devolver una lista con atributos asociados a cada entidad y reconocer relaciones de distintos tipos entre entidades como de ocupación y de pacto.

2.1 Enfoques para NER

A continuación, se exponen algunos trabajos de las principales metodologías que han surgido para afrontar el reto de NER. Estos son los algoritmos basados en reglas, en aprendizaje automático tradicional y aprendizaje neuronal profundo.

2.1.1. Enfoques basados en reglas

Los métodos basados en reglas requieren una definición de patrones hechos a mano, almacenándose en estructuras de datos como en un diccionario o nomenclátor, donde se registran diversas entidades organizadas por su categoría y así obtener información asociada. En relación con el procesamiento de textos históricos basados en reglas lingüísticas

necesarias para entender la naturaleza del dominio, se encuentra un trabajo de Grover et al. [22] donde se diseña un sistema para identificar entidades nombradas en registros digitalizados del parlamento británico entre los siglos XVII y XIX. Los resultados de reconocimiento indicaron que dicho sistema actúa mejor en clasificar a las personas más que a los lugares, especialmente al examinarlo sobre el conjunto de registros más antiguos, indicando que el software de procesamiento de imagen (OCR) para transcribir los documentos contribuye a la falta de acierto al no reconocer y distinguir bien ciertas palabras. Otro enfoque [23] adaptó una serie de reglas del lenguaje manuales sobre casos judiciales del siglo XVIII en EE.UU, en el que se comparten algunas dificultades asociadas con el reconocimiento de entidades causadas por variaciones en orden de las palabras o la aparición de términos no incluidos en el listado predefinido de reglas. Los autores en [24] propusieron otro sistema de reglas, utilizando un diccionario geográfico para el reconocimiento de entidades en un periódico estadounidense del siglo XIX que trataba temas de la Guerra Civil sucedida en aquella época. Con las transcripciones de las páginas junto con las etiquetas que hacen referencia a 10 tipos distintos de entidades, se obtuvo un buen resultado de reconocimiento en los lugares y fechas. Sin embargo, surgieron más problemas con la identificación de nombres de personas y de periódicos. Por último, Diez et al. [1] aplicaron NER a un conjunto de textos medievales españoles (entre los siglos XII y XV) de distintos géneros poéticos y legales, desarrollando una arquitectura modular que hace uso de analizadores léxicos creados manualmente para detectar patrones y expresiones regulares comunes en los textos. En la Figura 2.1, se ilustran las relaciones entre módulos del sistema, viéndose como las entidades encontradas en el módulo de análisis son almacenadas en un diccionario y en un nomenclátor mediante la interacción entre los módulos de preproceso y de generación de términos variantes. El módulo de análisis de dependencias detecta las posibles relaciones entre entidades como las familiares y de autoridad, y son representadas en un grafo creado con la sintaxis de XML-TEI.

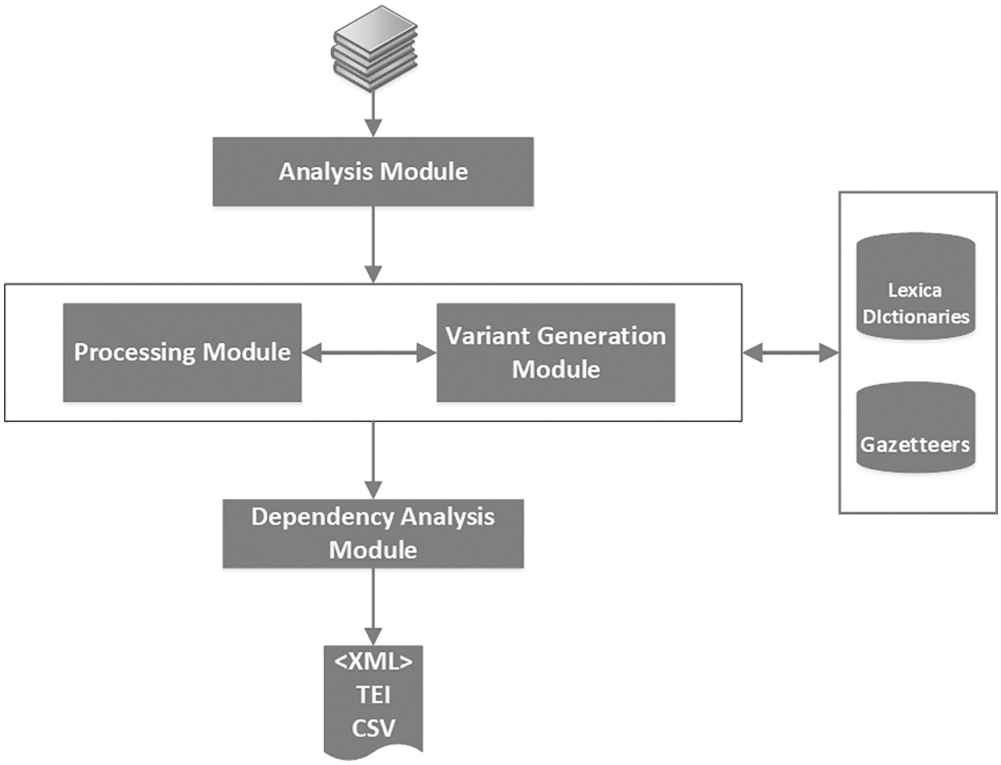


Figura 2.1: Arquitectura del sistema de reconocimiento de entidades nombradas basado en módulos y reglas léxicas presentado por Diez et al [1].

2.1.2. Enfoques basados en aprendizaje automático tradicional

Los algoritmos tradicionales de aprendizaje automático analizan los datos de entrada para realizar predicciones sobre las salidas, encontrándose aquí los algoritmos supervisados que hacen uso de las etiquetas de los datos para ajustar sus parámetros, y los no supervisados que solo se apoyan en los datos de entrada para tomar decisiones. Ehrmann et al. [25] utilizaron interfaces de programación de aplicaciones (APIs) y herramientas de reconocimiento implementadas con algoritmos supervisados de aprendizaje automático aplicados a publicaciones antiguas del diario suizo de Le Temps. Los clasificadores presentados se apoyan en modelos de lenguaje pre-entrenados para la tarea, evaluando su precisión sobre la identificación de personas y localizaciones en los archivos de distintos años. Los resultados mostraron una bajada generalizada de rendimiento en la detección de entidades (especialmente en personas) para las publicaciones más antiguas, debiéndose a la baja calidad de transcripción del software OCR y variabilidad del lenguaje. Packer et al. [26] diseñaron diferentes sistemas de extracción de entidades basados en modelos de *ensemble* (conjunto de modelos) formados por expresiones regulares, diccionarios de palabras y de modelos ocultos de Markov de máxima entropía, en una colección de documentos comerciales provenientes de diversas fuentes. El comportamiento fue comparado individualmente para cada modelo y para el conjunto, donde la estrategia de votación en los modelos conjuntos dio mejores resultados. Con respecto a entidades nombradas en informes médicos, Keretna et al. [12] plantearon un procedimiento de clasificación tomando muestras etiquetadas según el tratamiento médico, problema y test del paciente. Dicho procedimiento primero calcula vectores de palabras y de contexto para los informes como método de extracción de características, y luego diferentes clasificadores incluyendo árboles de decisión, vecinos más cercanos y CRF, fueron entrenados tomando las características de cada palabra como entrada para generar la predicción final en forma de etiquetas de entidades.

2.1.3. Enfoques basados en aprendizaje neuronal profundo

Los últimos avances en NER vienen del aprendizaje neuronal profundo, y de ahí se derivan los modelos más sofisticados que constituyen el estado del arte en tareas de reconocimiento. Al igual que los enfoques basados en aprendizaje automático tradicional, estos sistemas se refuerzan con representaciones vectoriales de palabras y de caracteres para cumplir con su objetivo, pero además poseen la habilidad de aprender del contexto de una entidad en función de las palabras cercanas debido a unidades de memoria que extraen la información pasada y futura. Arquitecturas neuronales recurrentes como las presentadas por Li et al. [27], hacen uso de estas funcionalidades debido al módulo bidireccional de memoria LSTM instalado en sus capas que les permite tener en cuenta la información anterior y posterior del paso temporal en la secuencia de entrada. También es común encontrar capas convolucionales para la extracción de características y una capa CRF final que modela las dependencias entre estados o etiquetas. Dicho esto, el modelo Bi-LSTM con capa CRF presentado por Huang et al. [28], utiliza las características propias del lenguaje natural y del contexto relacionado junto con información de diccionarios geográficos para mostrar el incremento de rendimiento producido al incluir estas fuentes adicionales, además de incorporar herramientas del aprendizaje neuronal profundo. Otro enfoque similar de Chiu y Nichols [2] fue presentado, en el que usó un modelo híbrido Bi-LSTM con una red convolucional (CNN), basado en representaciones numéricas a nivel de palabra y carácter sobre el conjunto de datos de CoNLL-2003. Esta arquitectura se ilustra en la Figura 2.2, donde primero se extraen las representaciones vectoriales de las palabras e información adicional del vocabulario, la CNN se encarga de procesar la información a nivel de carácter de cada palabra para crear vectores de una di-

mensión específica, que son concatenados e introducidos a la red Bi-LSTM para devolver las predicciones finales. A diferencia del resto de trabajos, Yang et al. [29], implementaron una red neuronal recurrente capaz de realizar múltiples tareas de procesamiento del lenguaje natural en diferentes idiomas, entre ellas el reconocimiento de entidades nombradas en textos, y sin hacer uso de representaciones numéricas de palabras. La red está compuesta de módulos jerárquicos de memoria GRU bidireccionales en vez de unidades LSTM, y de una capa CRF final. Mediante la información secuencial recibida de caracteres, se obtienen estados ocultos en ambas direcciones para cada posición en la palabra, y se concatenan las secuencias salientes de los módulos GRU para conseguir las representaciones completas de palabras. Recientemente, Wu et al. [30] presentaron un sistema Bi-LSTM-CRF con un mecanismo de atención para capturar dependencias a largo plazo de la información en la extracción de entidades nombradas en textos clínicos en chino. Dicho mecanismo calcula la similitud entre palabras, aprendiendo una matriz de pesos a partir de las salidas recibidas de la capa Bi-LSTM. El sistema también se apoya en características externas de cada palabra como su categoría gramatical para recoger información semántica. Finalmente, Devlin et al. [31] utilizaron un modelo de representación del lenguaje llamado BERT, un transformador bidireccional multicapa que usa modelos pre-entrenados para aprender de las expresiones propias del lenguaje natural.

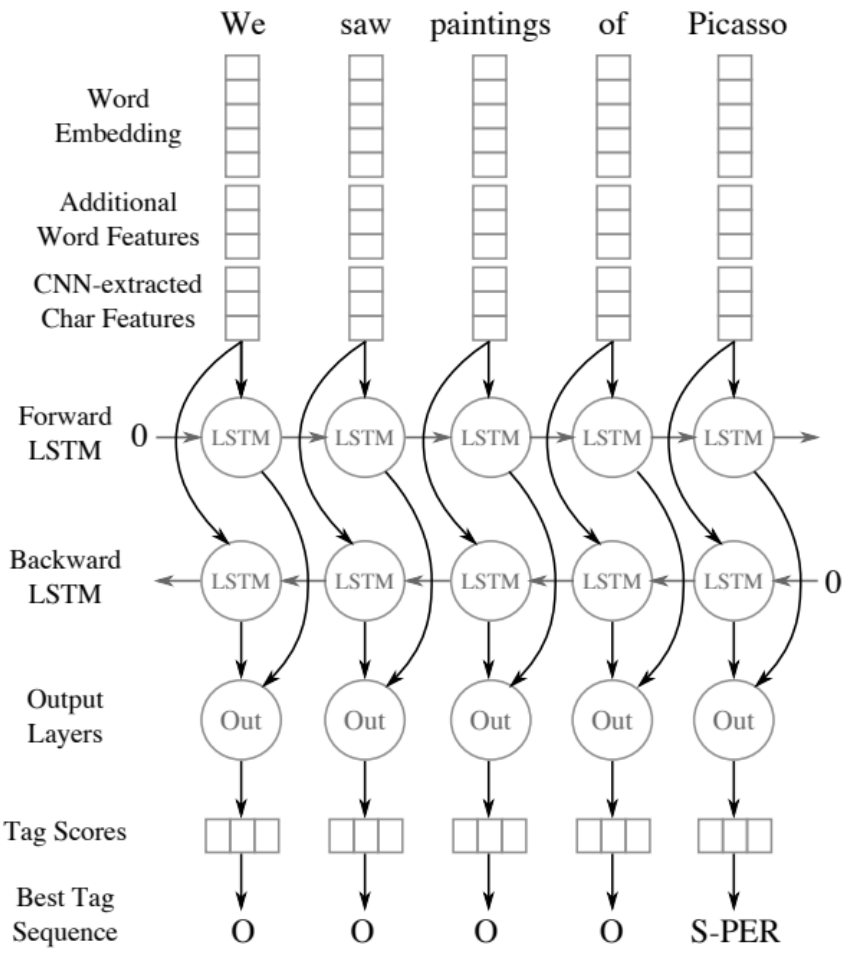


Figura 2.2: Arquitectura neuronal híbrida BiLSTM-CNN de Chiu y Nichols [2].

2.2 Crítica al estado del arte

En gran parte de las arquitecturas neuronales presentadas constituyendo al estado del arte en la tarea de NER, apenas se hace énfasis en el análisis descriptivo del propio del documento a tratar, sobre el cual se va a medir el rendimiento del sistema. Estudiar de antemano la composición de los textos e identificar términos frecuentes e irrelevantes que puedan introducir ruido al modelo, puede ser de gran ayuda de cara a mejorar su tasa de acierto.

Al igual que el propio vocabulario, no se estudia del todo las distribuciones de etiquetas. En casos de colecciones de documentos con categorías de entidades desequilibradas, el rendimiento del clasificador o de la red neuronal puede verse afectado al no dar la misma importancia a todos los tipos en el entrenamiento, por lo que en esta situación se puede disponer de estrategias de balanceo de clases.

2.3 Propuesta

El modelo neuronal planteado se asemeja a la arquitectura Bi-LSTM presentada por Chiu y Nichols [2], las diferencias principales son que en no se utiliza una red convolucional (CNN) para la extracción de información a nivel de carácter de cada palabra, sino que módulos LSTM y de esta forma obtener vectores reducidos que puedan ser fácilmente combinados con la información a nivel de palabra. Como se han presentado en múltiples investigaciones, se aprovecha la memoria sobre las representaciones de palabras pasadas y futuras mediante un codificador de contexto, y la decodificación de etiquetas para producir las salidas finales se realiza por medio de un campo aleatorio condicional (CRF) por lograr los mejores rendimientos de clasificación en muchas tareas de NER.

Gran parte de estos sistemas emplean representaciones tanto de palabras como de caracteres ya pre-entrenadas sobre largas colecciones de textos, siendo una ventaja puesto que se ha extraído conocimiento previo del lenguaje para ser adaptado a otra colección con un vocabulario similar. En este caso, al no disponer de modelos con representaciones distribuidas de palabras procedentes de una colección parecida a la que se ha de tratar, el modelo neuronal propuesto aprende a generar los valores numéricos en las representaciones, que son inicializadas con números aleatorios en un rango definido por una capa de incrustaciones de características o *embeddings*.

CAPÍTULO 3
Análisis del problema

La tarea de NER tiene como objetivo reconocer y clasificar entidades nombradas en amplias colecciones textos cuyos elementos son distinguibles entre sí por compartir una categoría semántica en común de interés. Extraer conocimiento de dichos elementos implica analizar el contexto en el que se dan y registrar similitudes con respecto a otros términos que puedan ayudar a interpretar sus significados. Con los diversos idiomas, variaciones ortográficas y rápida expansión del vocabulario, el procesamiento de entidades nombradas se ha convertido en un reto para muchos investigadores del campo que buscan incorporar las soluciones de última generación para hacer frente a estos problemas. Técnicas de NER se han aplicado a diferentes dominios que constan de una terminología específica tales como los médicos, jurídicos o informativos, en ocasiones otorgando buenas prestaciones de reconocimiento. En los últimos años, su estudio está siendo ampliamente desarrollado en textos manuscritos históricos para desvelar relaciones entre entidades y ayudar en la búsqueda y recuperación de documentos.

Un aspecto fundamental para desarrollar una arquitectura basada en NER son las anotaciones con etiquetas para categorizar los términos de los documentos. Normalmente varían según la colección y están enfocadas al contexto en el que transcurren los hechos pero algunas de las más comunes incluyen etiquetas de personas, lugares geográficos y organizaciones, pudiendo existir subcategorías más específicas dentro de cada una. En modelos de aprendizaje automático supervisados, las etiquetas les permiten realizar predicciones sobre los datos y en tareas como la de NER al igual que muchas otras, son esenciales para el correcto funcionamiento del sistema, en este caso haciendo referencia a grupos predefinidos dependiendo del significado de las palabras. Dicho esto, se trata de un problema de etiquetado, donde a partir de secuencias ordenadas de términos con una posición asignada, el sistema tiene que ser capaz de devolver otra secuencia con las clases (tipo de entidad nombrada) correspondientes a estos términos. Para ello, el sistema toma como referencia las etiquetas reales de las palabras permitiendo identificar patrones diferenciales que ayuden a predecir las etiquetas de nuevas secuencias que nunca ha visto.

3.1 Análisis del marco legal y ético

A continuación, se analizan aspectos legales y éticos que se tienen que cumplir para garantizar la protección de los datos que se van a utilizar y la toma de decisiones justa por el modelo de aprendizaje automático. Esto es importante puesto que hay que seguir las buenas prácticas en el tratamiento de la información que se posee e identificar posibles sesgos que puedan afectar a las predicciones finales.

3.1.1. Análisis de la protección de datos

La inteligencia artificial es el futuro para muchas industrias y comercios, cuyos modelos toman grandes cantidades de datos para sacar el máximo valor de ellos. Muchos de estos datos son de carácter personal y se tiene que mantener su privacidad a lo largo de todo el ciclo de vida, desde su creación hasta la retirada. Para ello, se han de seguir las indicaciones impuestas por el Reglamento General de Protección de Datos (RGPD) que recoge los derechos y principios fundamentales del tratamiento.

Los datos para este trabajo consistiendo en transcripciones etiquetadas de antiguos registros matrimoniales generados a partir del contenido auténtico manuscrito que incluye nombres, apellidos y lugares de residencia de personas, se han conservado de manera segura desde su recogida y han sido empleados únicamente para la finalidad acordada.

3.1.2. Ética

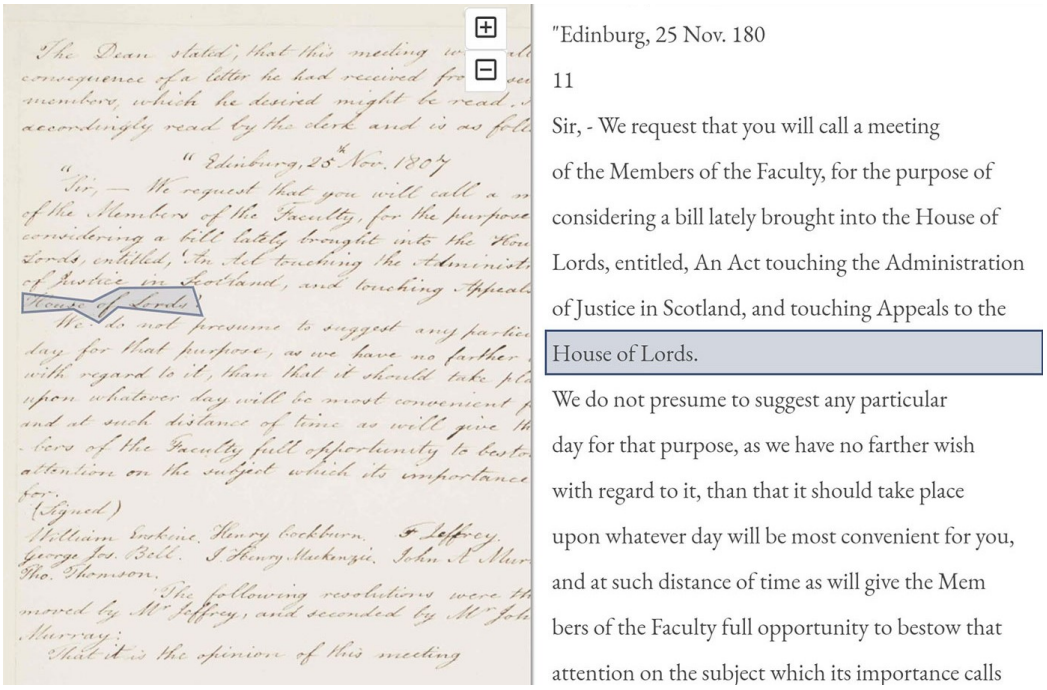
En el aprendizaje máquina, la correcta generalización de un algoritmo inteligente contribuye a eliminar los sesgos existentes en las decisiones finales. Estos sesgos pueden ser introducidos inconscientemente por la persona encargada de diseñar el sistema o estar presentes en los datos de entrada por su naturaleza, traduciéndose en predicciones con controversia. Es por ello que hay que analizar los patrones en los datos e identificar categorías o atributos que resulten más sensibles de ser confundidos.

Puede darse el caso de combinaciones de etiquetas muy poco frecuentes formadas por categorías semánticas y personas nombradas en los registros matrimoniales que puedan ser equivocadas por otra persona. Esto sucede por la falta de información que se tiene sobre estas etiquetas, decantando al modelo por extraer la categoría que comúnmente se da. Como se verá en el siguiente capítulo, ciertos atributos no ocurren por igual para los hombres que para las mujeres debiéndose al carácter narrativo de los registros, por lo que es importante tener en cuenta las distribuciones de frecuencias de cada etiqueta para evitar que se tomen decisiones sesgadas.

3.2 Dificultades en el reconocimiento de entidades nombradas en textos históricos

Este tipo de colecciones antiguas requiere un paso previo de procesamiento para extraer su contenido en un formato digital y poder ser entendido por una máquina. Habitualmente, la transcripción se realiza de forma automática por un software OCR de reconocimiento óptico o por técnicas de reconocimiento de texto manuscrito (HTR), y de esta forma poder ser tratados por los modelos de aprendizaje automático. El proceso de transcripción se ilustra en la Figura 3.1 sobre un texto manuscrito, donde el sistema genera el resultado del reconocimiento y señala las correspondencias entre líneas.

Una dificultad añadida en el procesamiento de documentos históricos es la cantidad de ruido en la transcripción originado por la falta de legibilidad de algunas palabras, causando errores de reconocimiento en el sistema. Por ejemplo, no detectando bien la separación entre palabras o confundir caracteres por la variabilidad de estilos o fuentes tipográficas. Como consecuencia, se producen términos ambiguos situados fuera de vocabulario que pueden afectar negativamente al rendimiento de clasificación, por lo que la calidad de la información textual juega un papel muy importante para reconocer las entidades nombradas. El estado de los documentos influye también en el proceso de digitalización ya que se tratan de colecciones de décadas o siglos de antigüedad y con el



"Edinburg, 25 Nov. 180

11

Sir, - We request that you will call a meeting of the Members of the Faculty, for the purpose of considering a bill lately brought into the House of Lords, entitled, An Act touching the Administration of Justice in Scotland, and touching Appeals to the House of Lords.

We do not presume to suggest any particular day for that purpose, as we have no farther wish with regard to it, than that it should take place upon whatever day will be most convenient for you, and at such distance of time as will give the Mem bers of the Faculty full opportunity to bestow that attention on the subject which its importance calls

Figura 3.1: Transcripción de un documento manuscrito histórico línea por línea. Fuente: [3].

tiempo pueden haberse deteriorado por factores ambientales (temperatura, polvo, humedad,...), biológicos (microorganismos, insectos,...) y factores humanos. La inconsistencia en la estructuración y segmentación de los textos manuscritos de la época son otro factor que influye en la calidad de la transcripción pues puede mezclarse el texto al darse casos de partes de un párrafo que no estén bien alineadas, palabras ubicadas entre dos lineas, aparecer dibujos en las páginas que no pueden ser interpretados y tachones de palabras.

En relación al proceso de reconocimiento de entidades nombradas, la falta de colecciones históricas relacionadas con anotaciones o de modelos de lenguaje específicos sobre los que se puede apoyar la red neuronal hacen aún más complicada esta tarea debido a la disparidad de dominios, periodos de tiempo e idiomas encontrados en estos textos. A esto hay que sumarle los cambios en el lenguaje y las variaciones en las expresiones gramaticales que han surgido en el tiempo, que no son triviales para un sistema basado en NER.

3.3 Identificación y análisis de soluciones posibles

Como se ha introducido en la sección 2.1, son diversas las estrategias para la aplicación de NER en textos de cualquier ámbito. La más clásica la forman los modelos basados en reglas gramaticales, que se crean mediante patrones comunes específicos del contexto y suelen respaldarse por recursos externos como diccionarios y nomenclátores, que contienen términos agrupados por su naturaleza semántica. Por ejemplo, constan de secciones con todos los nombres de ciudad, de personas, apellidos, organizaciones, etc. Estas reglas son frecuentemente creadas a mano y permiten capturar menciones de entidades con vocabularios muy extensos. Una ventaja de los modelos basados en reglas es que no requieren un conjunto de entrenamiento y validación al no constar de etiquetas anotadas, ahorrando el tiempo de ajuste necesario de un modelo tradicional de aprendizaje automático, que en ocasiones es considerable. Sin embargo, las reglas no pueden ser generalizadas a otros contextos ya que están condicionadas por el dominio del diccionario

o catálogo de nombres propios sobre las cuales han sido definidas. Esto significa que si se han creado un conjunto de reglas para reconocer y extraer nombres de ciudades estadounidenses, estas no funcionarán correctamente en detectar otras ciudades en documentos con alta presencia de localizaciones europeas, por ejemplo, luego el alcance del vocabulario es una gran limitación. A esto hay que sumarle el tiempo consumido y la dificultad en definirlas, pues requieren conocimiento del entorno en cuestión.

En el segundo bloque se encuentran los modelos de aprendizaje automático, que se dividen en no supervisados y supervisados. Los no supervisados no utilizan las etiquetas de las observaciones para aprender patrones y se apoyan en sus similitudes para extraer las entidades según los grupos definidos, como el *clustering* [32]. En cambio, los supervisados recogen la información de las etiquetas a predecir y en tareas del procesamiento del lenguaje natural se basan principalmente en el *feature engineering* o ingeniería de características [33], transformando los datos en crudo en representaciones vectoriales para ser tomadas por el modelo. Algunos de los modelos supervisados más conocidos que operan con dichas representaciones de características son los modelos ocultos de Markov, árboles de decisión, máquinas de vector y soporte, y los campos aleatorios condicionales (CRF), convirtiéndose estos últimos en una referencia para muchos sistemas de reconocimiento de entidades de hoy en día debido a que tienen en cuenta el contexto de cada palabra para predecir las etiquetas mejor adaptadas.

Sin embargo, en estos últimos años, los sistemas basados en aprendizaje automático han ido progresando, en especial aquellos de aprendizaje profundo como son las redes neuronales profundas, que han conseguido mejorar la eficiencia y precisión del reconocimiento de las entidades importantes, logrando los resultados óptimos del estado del arte. Estos modelos son capaces de identificar relaciones ocultas más complejas en los datos, donde se eleva el nivel de abstracción en el funcionamiento debido a la composición en capas y bloques de sus unidades entrenables. Otra ventaja de estos modelos es que también pueden aprender representaciones vectoriales de palabras y caracteres que registran características o propiedades en un espacio de múltiples dimensiones, permitiendo conocer similitudes en función de las distancias de cada componente. Aparte, las composiciones neuronales comúnmente utilizadas en NER son consideradas como redes neuronales recurrentes [34] ya que pueden operar con información secuencial y utilizar memoria de las palabras cercanas para analizar el contexto por posición o paso temporal, destapando dependencias entre etiquetas haciendo uso de modelos probabilísticos. En especial, estos son los modelos Bi-LSTM que constan de representaciones distribuidas de palabras y caracteres, y de una capa CRF de salida, siendo los más utilizados para esta tarea. Evidentemente existen múltiples variantes de esta arquitectura común, especialmente en la forma de extraer las representaciones vectoriales y operar con ellas, y en la decodificación de etiquetas pero el mecanismo principal de etiquetado sigue siendo el mismo.

3.4 Solución propuesta

Habiendo analizado los tres principales grupos de modelos, la arquitectura diseñada para este trabajo sigue el concepto de red neuronal, dominante en los últimos años por presentar resultados prometedores en la tarea de NER. En concreto, se propone un modelo neuronal recurrente con celdas de memoria a corto y largo plazo LSTM (Long Short Term Memory) bidireccional o Bi-LSTM para aprender del contexto de las palabras en una frase o secuencia. También se hace uso de las representaciones numéricas de palabras y adicionalmente de las representaciones a nivel de carácter, ambas aprendiéndose en la etapa de entrenamiento. La razón de utilizar representaciones de caracteres

es que el modelo puede inferir información de palabras que están fuera del vocabulario o que nunca ha visto debido a similitudes morfológicas. Por ejemplo, si durante la evaluación del modelo se presenta la palabra desconocida “viuda” y la misma palabra pero en masculino “viudo” ya ha sido manejada en el entrenamiento, basándose solo en las representaciones de palabras se puede inferir su significado solo por las demás palabras de alrededor, y con las representaciones de caracteres se puede extraer información de su composición, en este caso de gran utilidad por ser variación morfológica. Por último, se escogerá una capa CRF para decodificar las etiquetas ya que modela las probabilidades de transición entre etiquetas en función de las palabras de antes y de después, además de ser una opción muy popular para el etiquetado de secuencias sobre la que se obtienen buenos resultados de clasificación.

CAPÍTULO 4
Registros matrimoniales de Esposalles

Los textos utilizados para poner a prueba al modelo propuesto pertenecen a un subconjunto de registros matrimoniales manuscritos, ubicados en la base de datos de Esposalles [5], que ha sido recogida de una colección de libros de licencias matrimoniales y contiene 291 libros con registros escritos por varios autores entre los siglos XV y XX, conservados en los Archivos de la Catedral de Barcelona. En concreto, se van a examinar 968 registros repartidos en 100 páginas del volumen 69 escrito en el siglo XVII, que fueron empleados como conjunto de entrenamiento en la competición de Extracción de Información en Registros Históricos Manuscritos (IEHHR) organizada por la *International Conference on Document Analysis and Recognition* (ICDAR) y celebrada en 2017 [4].

Cada licencia matrimonial contiene principalmente información acerca del marido y de la mujer sobre sus nombres, apellidos, trabajos, lugares de residencia y estados civiles. Existen casos donde también se proporciona información de los padres y de otras personas relacionadas. En la Figura 4.1 se observa la composición de un registro de la colección escrito en catalán antiguo, donde se sigue un patrón de expresión bastante común, haciendo primero referencia al marido y a sus padres, y luego a la mujer junto a sus padres. Otra forma en la que pueden aparecer es mencionando primero al marido y luego a su mujer junto con su ex-marido, a veces sin dar información de los padres. Para facilitar la búsqueda de estas licencias, dentro de cada libro es posible encontrar al principio un índice con los apellidos del marido y la página en la que se encuentra su respectivo registro. Cabe destacar que si al acabar de escribir un registro no se completa su última linea, se introduce una linea recta hasta el final para dar paso al siguiente registro.

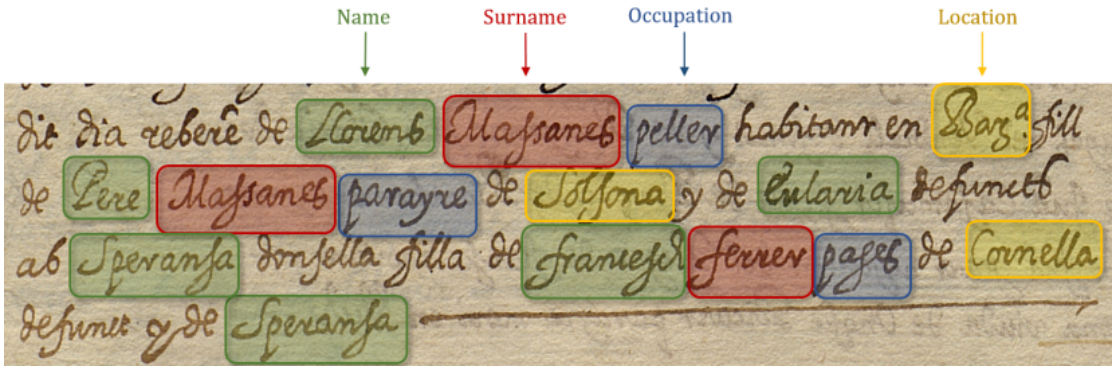


Figura 4.1: Ejemplo de un registro matrimonial señalando las categorías semánticas. Fuente: [4].

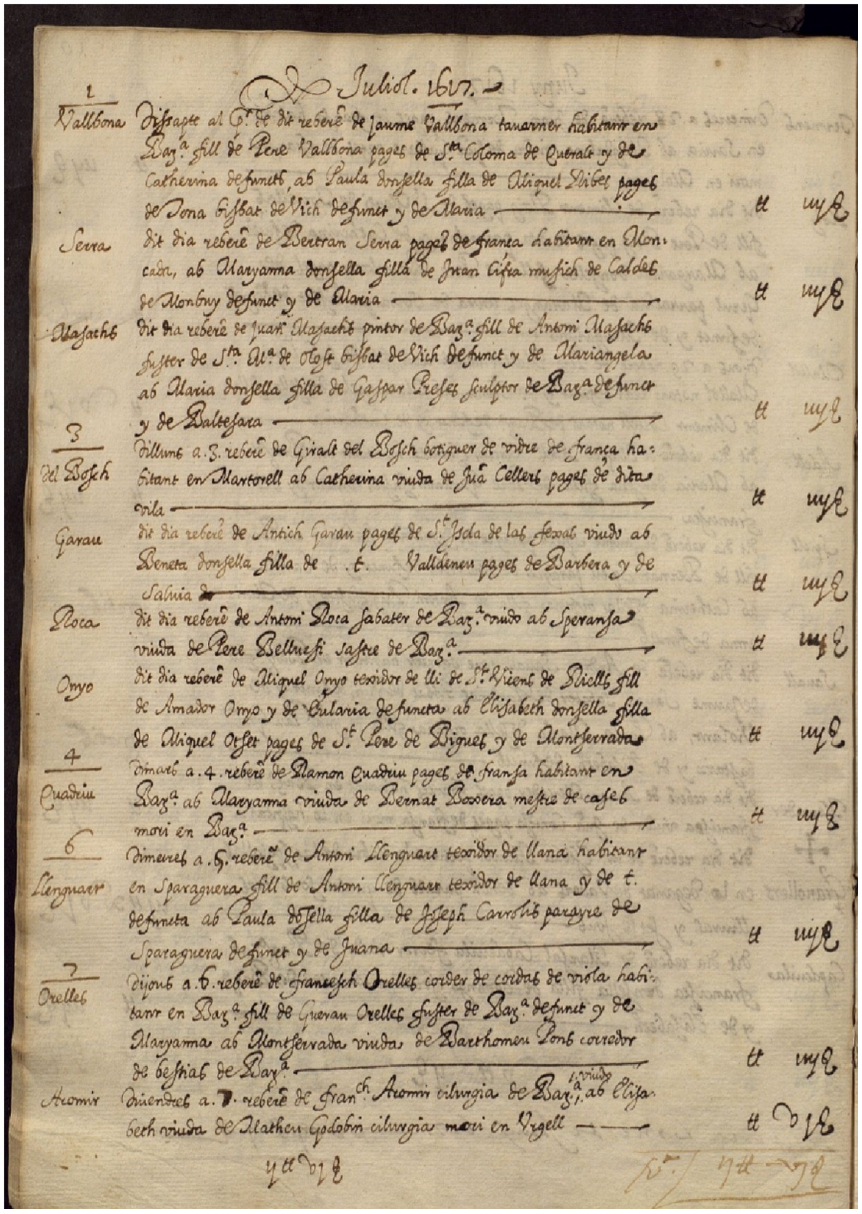


Figura 4.2: Ejemplo de una página de la colección de Esposalles con 11 registros matrimoniales. Fuente: [5].

En cualquier página del libro como se observa en la Figura 4.2, el mes y año de la escritura se sitúan en la parte superior, y en la parte inferior la factura total de todos los registros encontrados en esa página. En el margen de la izquierda se encuentra el apellido del marido que identifica al registro y el día del mes, que solo es anotado en caso de ser la primera entrada en escribirse durante ese día. En el margen derecho se encuentra la factura individual por registro.

El texto manuscrito de los registros en el conjunto de datos, ha sido transcrito manualmente y el resultado de la transcripción se ha proporcionado en ficheros de texto tanto a nivel de linea, es decir, conservando el posicionamiento de las palabras en sus filas correspondientes del registro original, y a nivel de palabra, donde por cada fila se tiene una palabra. Cada registro viene acompañado por dos tipos de etiquetas: unas con las categorías semánticas de las palabras y otras con las personas a las que se hace referencia. Al igual que las transcripciones, las etiquetas son dadas en otros dos ficheros de texto a nivel de linea y a nivel de palabra, y las posiciones de cada etiqueta coinci-

den con la de las palabras. En resumen, cada registro tiene asociado una carpeta con su transcripción textual, categorías semánticas y personas, en los dos modos mencionados. Para este trabajo se han utilizado las transcripciones y etiquetas a nivel de palabra pues de esta forma se captura todo el contexto del registro en cada posición de la secuencia de entrada. El reconocimiento a nivel de linea implica analizar por separado las lineas que componen cada registro y en este caso el contexto a modelar por la red neuronal es más reducido. Con estos datos ya es posible realizar tareas de visualización y prepararlos para ser introducidos al modelo.

4.1 Análisis exploratorio de los registros

El análisis exploratorio de datos es un paso necesario para conocer mejor la colección de textos de la que se dispone. Implica la extracción de conocimiento previo al propio modelado de tal manera que se puedan identificar patrones relevantes mediante tablas y gráficos exploratorios. Con los registros etiquetados, algunos de los elementos a analizar incluyen el vocabulario de palabras, las distribuciones de longitudes de los registros y la distribución de etiquetas. De forma análoga, se analizará el vocabulario y longitudes de las transcripciones generadas por el proceso de reconocimiento de texto manuscrito, es decir, aquellas que pueden presentar errores en ciertas palabras.

En primer lugar, se ilustran algunos estadísticos básicos de las dos versiones de las transcripciones en la Tabla 4.1. Contando con la misma cantidad de registros, las transcripciones generadas por el reconocedor de texto manuscrito contienen en total más palabras, indicando que se han producido inserciones de nuevos términos, y constan de un vocabulario más reducido. También tienen una mayor cantidad y variedad de caracteres.

Tabla 4.1: Estadísticos de ambas versiones de las transcripciones.

Número de:	**Transcripciones sin fallos**	**Transcripciones con fallos**
Registros	968	968
Palabras	31501	33492
Palabras únicas	2430	1847
Caracteres	140614	148074
Caracteres únicos	60	66

Siguiendo con las frecuencias de los términos del vocabulario cuando se disponen de las transcripciones sin fallos, vemos en la Tabla 4.2 como la palabra “de” es la más dominante en la colección, formando el 20 % del total de palabras y apareciendo de media unas 6-7 veces por registro, debiéndose a que enlaza palabras de múltiples categorías semánticas e indica relaciones de pertenencia. Por ejemplo, es muy utilizada para destacar el lugar de residencia del marido y mujer, además de señalar múltiples relaciones familiares. El término más frecuente que hace referencia a una categoría semántica de interés es “pages” que significa campesino y tiene sentido que ocupe un lugar tan alto en la tabla pues en aquella época de crisis eran muchas las personas que pertenecían a una clase social modesta y se dedicaban a trabajar y cultivar la tierra. Otros términos a destacar son “Bara” y “donsella” que hacen referencia a la ciudad de Barcelona y a un estado civil, respectivamente.

Tabla 4.2: Palabras más frecuentes en el conjunto de datos de Esposalles.

Término w	**Rango** r	**Frecuencia** f	**Proporción** $P(w)$	$r \cdot f$
de	1	6355	0.2017	6355
y	2	1338	0.0425	2676
pages	3	977	0.0310	2931
ab	4	966	0.0307	3864
rebere	5	934	0.0296	4670
filla	6	729	0.0231	4374
Bara	7	718	0.0228	5026
donsella	8	646	0.0205	5168
dia	9	590	0.0187	5310
fill	10	570	0.0181	5700

En la misma tabla se ve como la frecuencia de cada término es inversamente proporcional a su rango, siguiendo la ley de Zipf [35] que confirma que son pocas las palabras que aparecen con bastante frecuencia mientras que la mayoría de palabras son mencionadas unas pocas veces. Esto significa que el producto del rango y la frecuencia de un término equivale a un valor constante, y como se puede ver en las constantes de la Tabla 4.2 para las palabras más frecuentes, se sitúa cerca de 5000. Dicha constante representa el gradiente de la distribución de frecuencias ordenadas de palabras en una escala logarítmica tanto para el rango como para la frecuencia (Figura 4.3).

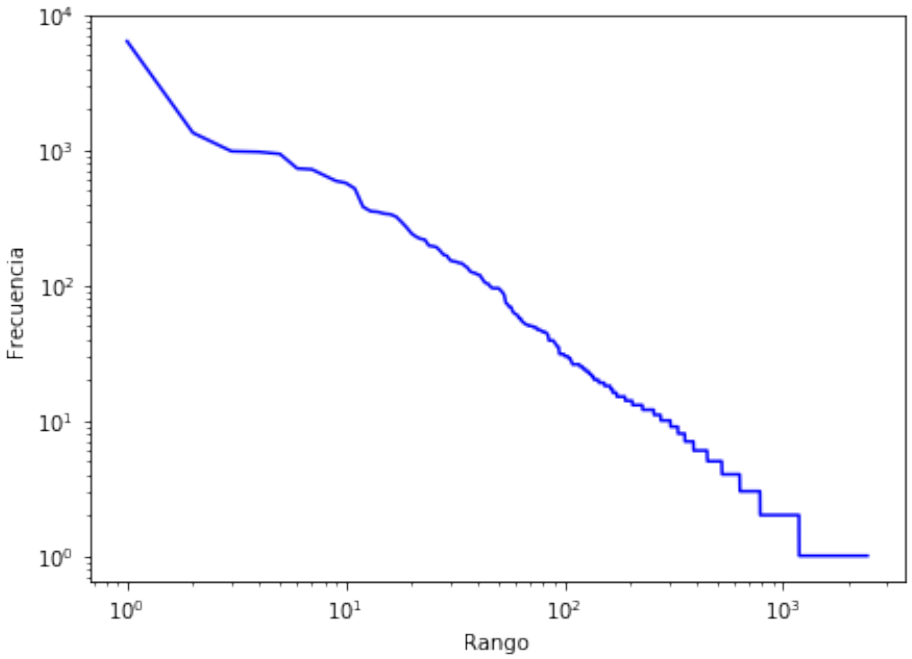


Figura 4.3: Frecuencia de cada término del vocabulario en relación con su rango en el conjunto de registros matrimoniales, sobre una escala logarítmica en ambos ejes. Se cumple la ley de Zipf, donde la frecuencia es inversamente proporcional al rango de la palabra.

Pasando a la distribución de longitudes de los registros definidas por la cantidad de palabras en cada uno, vemos a través de la Tabla 4.3 algunos estadísticos principales como la media, indicando que los registros constan en promedio de 32 palabras. También se ven otros como la desviación típica y valor mínimo y máximo. En la Figura 4.4 se

ilustra el histograma de longitudes de los registros. La distribución presente sigue una campana de Gauss, donde mayoritariamente las longitudes varían entre 20 y 50 palabras, y fuera de este intervalo hay menos observaciones, encontrándose un valor atípico en la cola derecha que corresponde a un registro con la longitud máxima de 77. Se trata pues de un registro en el que los apellidos de la familia del marido son más largos de lo habitual y la ocupación del padre de la mujer es un tanto especial al ser un governador de unos condados de la época.

En cuanto a la distribución de longitudes de palabras, estas tienen 4 letras en promedio y la más larga consta de 14 letras. Debido a que la red neuronal modela dependencias entre caracteres, es importante conocer de antemano las longitudes de las palabras además de los registros.

Tabla 4.3: Estadísticos de las longitudes de los registros y de palabras.

Estadístico	**Registros**	**Palabras**
Media	32.54	4.46
Desviación típica	6.69	2.47
Mínimo	13	1
Máximo	77	14

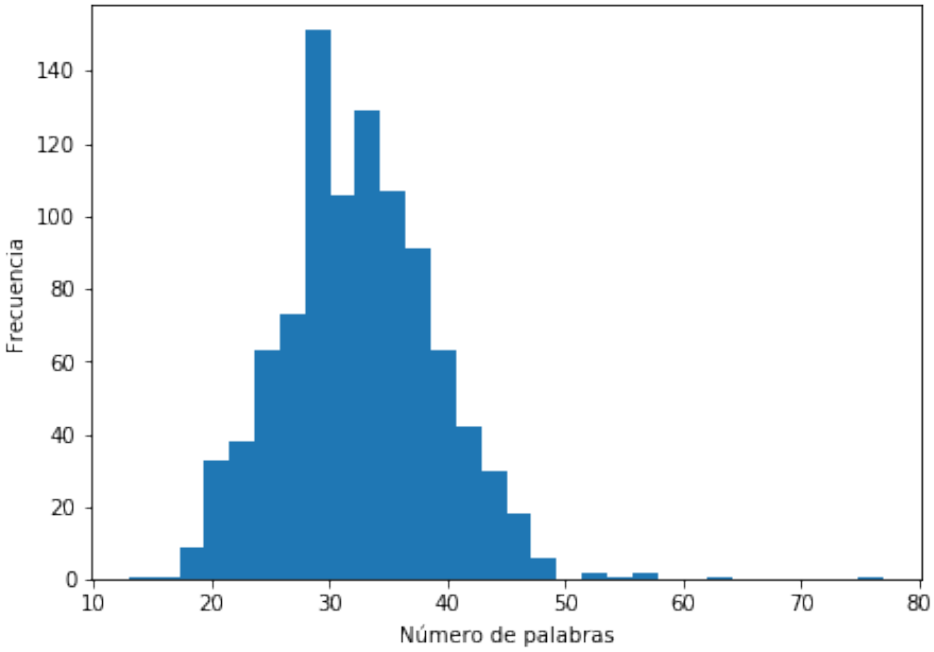


Figura 4.4: Histograma con la distribución de las longitudes de los registros.

Las etiquetas son las salidas a predecir por la red neuronal y por tanto conviene analizar su distribución ya que un número muy reducido de una categoría en concreto puede afectar negativamente a su detección por falta de aprendizaje. Disponiendo de todas las etiquetas, se muestra primero las frecuencias de las categorías semánticas en la Tabla 4.4, donde aproximadamente la mitad de palabras de la colección no pertenecen a una categoría de interés (“other”) y para el resto, las proporciones son más elevadas en los nombres y localizaciones. El estado civil (“state”) ocurre en menor medida porque suele describir principalmente a la condición matrimonial de la mujer, apareciendo de media una vez por registro, por lo que es una categoría que debe ser identificada en todas las

entradas al modelo. La frecuencia de los nombres es casi el doble que la de los apellidos porque en muchas ocasiones se omite el apellido de las mujeres.

Tabla 4.4: Frecuencias absolutas, relativas y promedio por registro de las categorías semánticas.

Categoría	Frecuencia	Proporción	Promedio por registro
other	15176	0.482	15.68
name	4997	0.159	5.16
surname	2667	0.085	2.76
occupation	3010	0.096	3.11
location	4509	0.143	4.66
state	1142	0.036	1.18

A continuación, se muestran las palabras más frecuentes de algunas categorías de interés en la tabla Tabla 4.5. Para las localizaciones se encuentran lugares geográficos como "Bara" y "Vich", además de otros relacionados con la iglesia y territorios. Otros términos como "de" o "St" actúan como conectores para indicar el nombre compuesto de una localización, y de ahí vienen sus elevadas frecuencias. Con respecto a las ocupaciones, la de "pages" o campesino es la más dominante, apareciendo también en los términos más frecuentes de toda la colección en la Tabla 4.2, y se encuentran otros como sastres, tejedores y zapateros. Los estados civiles constan de un vocabulario más específico donde solo se tienen doncellas, viudas y viudos, habiendo otras palabras menos frecuentes.

Tabla 4.5: Palabras más frecuentes por categoría de localización, ocupación y estado.

Localización	f	Ocupación	f	Estado	f
Bara	717	pages	972	donsella	640
de	641	de	249	viuda	223
St	230	parayre	217	viudo	194
bisbat	145	sastre	94	dosella	68
dita	144	texidor	88	sella	8
regne	137	sabater	62	don	4
frança	118	mestre	57	do	4
parrochia	90	llana	52	donzella	1
Vich	71	lli	49		
vila	66	hortola	47		

Las distribuciones de frecuencias de las etiquetas producidas al combinar la categoría semántica y persona se enseñan en la Tabla 4.6. Lo primero a destacar es que la etiqueta para la categoría "none-other" no se considera relevante pues son términos que no recogen información acerca de una entidad nombrada. De hecho, todas las palabras con la categoría semántica "other" vistas anteriormente no se corresponden a ninguna persona por lo que sus frecuencias en ambas tablas coinciden. Analizando las frecuencias de los nombres, estas son altas para los del marido y la mujer, apareciendo en promedio una vez por registro, pero más bajas para los padres, especialmente para los del marido, dando a entender que hay veces en las que solo se mencionan a los padres de la mujer. Como se comentaba antes, los apellidos se muestran principalmente para los hombres y en raras ocasiones para las mujeres. Con respecto a las localizaciones, se figuran mayoritariamente las asociadas con el marido, la mujer y la del padre de cada uno, llamando la atención que las propias del marido aparecen casi cinco veces más que las de la mujer. Esto puede explicarse por haber casos donde el matrimonio comparte lugar de origen por lo que escribir ambos es redundante. Algo similar pasa con las ocupaciones o traba-

jos, con un conteo de 1207 para el marido y 270 para la mujer, sin mencionarse para las madres. El comportamiento opuesto ocurre con los estados civiles, donde se señalan en casi todos los registros para la mujer y en pocos para el hombre. Para los ex-maridos que tienen etiqueta de "other_person" solo se destacan los nombres y apellidos, y en un caso excepcional un lugar relacionado.

Tabla 4.6: Frecuencias absolutas, relativas y promedio por registro de las etiquetas conjuntas de personas y categorías semánticas.

Persona y categoría	Frecuencia	Proporción	Promedio por registro
none-other	15176	0.4818	15.678
husband-location	2374	0.0754	2.452
husband-occupation	1207	0.0383	1.247
wifes_father-location	1170	0.0371	1.209
wife-name	1046	0.0332	1.081
husband-name	1025	0.0325	1.059
husband-surname	1010	0.0321	1.043
wife-state	931	0.0296	0.962
wifes_father-occupation	893	0.0283	0.923
wifes_father-surname	769	0.0244	0.794
wifes_father-name	760	0.0241	0.785
wifes_mother-name	737	0.0234	0.761
husbands_father-occupation	640	0.0203	0.661
husbands_father-name	618	0.0196	0.638
husbands_father-surname	586	0.0186	0.605
husbands_mother-name	582	0.0185	0.601
wife-location	490	0.0156	0.506
husbands_father-location	470	0.0149	0.486
wife-occupation	270	0.0086	0.279
other_person-surname	235	0.0075	0.243
other_person-name	229	0.0073	0.237
husband-state	211	0.0067	0.218
wife-surname	34	0.0011	0.035
wifes_mother-surname	17	0.0005	0.018
husbands_mother-surname	16	0.0005	0.017
husbands_mother-location	4	0.0001	0.004
other_person-location	1	3×10^{-5}	0.001

Considerando las transcripciones con fallos producidas por el reconocedor de texto manuscrito, vemos en la Tabla 4.7 los términos más frecuentes del nuevo vocabulario que se ha creado. En general, las frecuencias de los términos y proporciones son muy parecidas a las que se daban en las transcripciones reales de la Tabla 4.2, pero quizás lo que más llama la atención es la alta presencia de caracteres especiales como el guión ("-") y el punto (".") que no se han percibido anteriormente. De nuevo, se cumple la ley de Zipf en esta nueva colección, como se ilustra en la Figura 4.5. Un aspecto a tener en cuenta es que el nuevo vocabulario es más reducido (1847 palabras frente a 2430 en las transcripciones sin fallos), viéndose una ligera disminución del rango máximo en el eje de abscisas de la distribución.

Tabla 4.7: Palabras más frecuentes en las mejores hipótesis generadas por el proceso de reconocimiento de texto manuscrito.

Término w	**Rango** r	**Frecuencia** f	**Proporción** $P(w)$	$r \cdot f$
de	1	6382	0.1906	6382
y	2	1367	0.0408	2734
pages	3	977	0.0292	2931
ab	4	966	0.0288	3864
-	5	959	0.0286	4795
rebere	6	925	0.0276	5550
.	7	889	0.0265	6223
Bara	8	723	0.0216	5784
filla	9	721	0.0215	6489
donsella	10	656	0.0196	6560

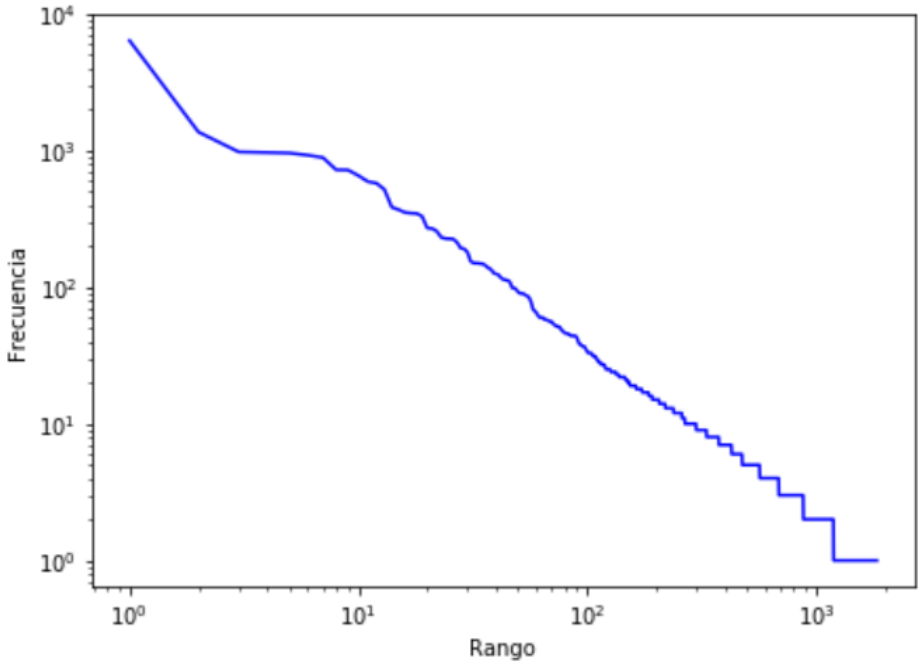


Figura 4.5: Frecuencia de cada término del vocabulario en relación con su rango en las mejores hipótesis generadas por el proceso de reconocimiento de texto manuscrito, sobre una escala logarítmica en ambos ejes.

En el histograma de la Figura 4.6, se ve como las longitudes en esta versión de las transcripciones también siguen una distribución normal, donde la mayoría tienen entre 20 y 50 palabras, con mayor presencia en el intervalo que va de 30 a 40 palabras. Como se observa en la Tabla 4.8, la longitud media ha incrementado ligeramente.

Tabla 4.8: Estadísticos de las longitudes de las mejores hipótesis y de sus palabras.

Estadístico	**Registros**	**Palabras**
Media	34.60	4.42
Desviación típica	7.07	2.61
Mínimo	12	1
Máximo	77	13

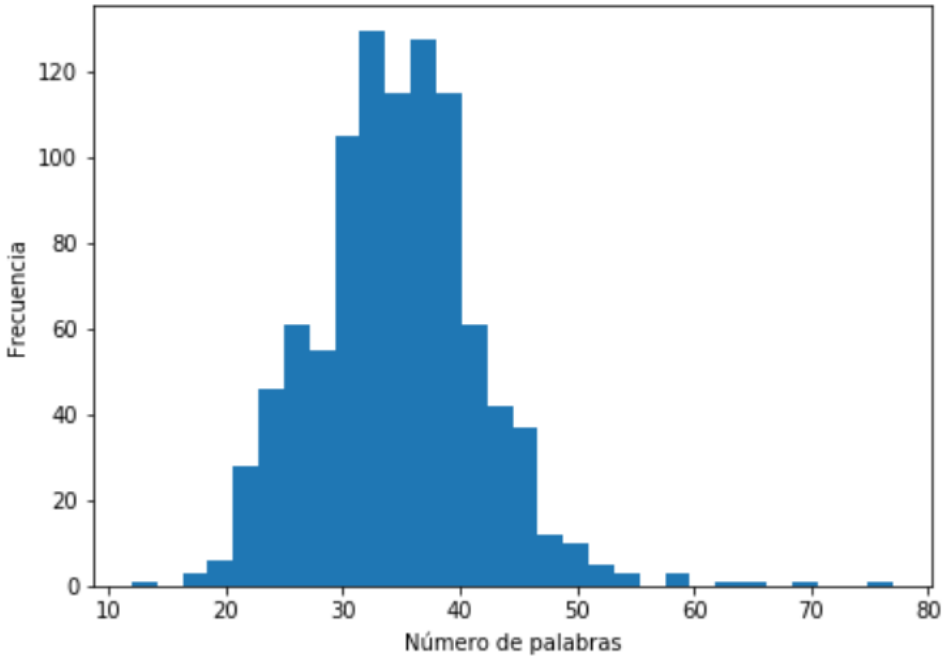


Figura 4.6: Histograma con la distribución de las longitudes de las mejores hipótesis .

Mediante este análisis exploratorio de los registros matrimoniales, se han mostrado algunos parámetros característicos de las colecciones de textos que se disponen. La distribución de frecuencias de los términos en ambas versiones de las transcripciones sigue una ley de potencias de Zipf en la que hay muy pocas palabras que forman la mayor parte de la colección como “de” ,“y”, “ab” y “pages”, pero son muchas las palabras que aparecen un número reducido de veces. Además, se han analizado algunos estadísticos propios de las distribuciones de longitudes de los registros y palabras, detectándose un registro más largo de lo habitual. La abundante aparición de símbolos especiales en las mejores hipótesis junto al incremento en la longitud media en las transcripciones da la impresión de que estos caracteres se han insertado erróneamente durante el proceso de reconocimiento del texto.

También se han estudiado las frecuencias de las etiquetas de categorías semánticas, donde son cerca de la mitad del total de palabras las que no tienen asociadas una categoría real y se han visto ejemplos de palabras frecuentes con etiquetas de lugares, trabajos y estados civiles. A pesar de que existen fallos en la transcripción de algunas palabras, el modelo se apoya también en la información de los caracteres que las componen por lo que si la palabra mal recogida y por tanto con pocas ocurrencias se asemeja a su forma correcta, es posible determinar su significado en base a la estructura morfológica. Por último, se han presentado las distribuciones de frecuencias por etiqueta conjunta de persona y categoría semántica, señalando algunos detalles interesantes como la escasa mención de los apellidos de las mujeres y madres, y la diferencia entre las apariciones de los trabajos y localizaciones del marido con respecto a la mujer.

CAPÍTULO 5
Aprendizaje neuronal profundo

En este capítulo se introducen algunos conceptos del aprendizaje neuronal profundo importantes para entender el funcionamiento de esta tecnología que se ha convertido en el foco de atención de los sistemas más modernos para NER. También se presentan las variantes de arquitecturas más conocidas en este campo, incluyendo las redes neuronales recurrentes, ampliamente utilizadas para el procesamiento de datos secuenciales como pueden ser las palabras que conforman una frase o párrafo.

El aprendizaje profundo o *deep learning* es una rama del aprendizaje automático cuyos algoritmos están inspirados en la propia composición neuronal del cerebro humano. Concretamente, estas son las redes neuronales y tienen como objetivo imitar al comportamiento característico de cualquier ser humano a la hora de aprender de una situación totalmente nueva. Las redes neuronales manejan y analizan grandes cantidades de datos para acelerar procesos de toma de decisiones. Algunas de las tareas para las cuales se utiliza el aprendizaje profundo son para la detección de correos no deseados, clasificación y reconocimiento en imágenes, predicción de precios de acciones, reconocimiento de entidades nombradas en textos como es en este caso o incluso en la implementación de vehículos autónomos. En comparación con los algoritmos de aprendizaje automático básicos, las redes neuronales son capaces de extraer patrones que cuestan más de identificar y tienen un método inteligente de aprender de sus propios errores y ajustar sus parámetros.

5.1 Composición de una red neuronal

Las unidades básicas que componen las redes neuronales son las neuronas o simplemente nodos, y están organizados en capas que componen la arquitectura principal [36]. Los tres tipos distintos de capas son: la capa de entrada, la capa oculta y la capa de salida. La capa de entrada es donde se reciben los datos a partir de los cuales se va a obtener la predicción. La capa oculta se encarga de almacenar una serie de características desconocidas de los datos entrantes que son de utilidad para la obtención de patrones relevantes y complejos. Por último, la capa de salida contiene en cada nodo el valor resultante de aplicar todas las operaciones necesarias sobre la información entrante. Como se observa en la Figura 5.1, las unidades de cada capa están densamente conectadas con las unidades de la siguiente capa por un conjunto de enlaces. Normalmente, suele haber más de una capa oculta para capturar relaciones adicionales que requieren un mayor procesamiento y este tipo de red se denomina red multicapa. Los enlaces tienen unos pesos asociados que determinan los valores que ocuparán los nodos en las capas ocultas y final, siendo los parámetros que se tienen que ajustar durante el entrenamiento. Una red neuronal es

por tanto un modelo matemático que en cada iteración recibe un conjunto de muestras y va modificando sus parámetros internos a partir de ellas.

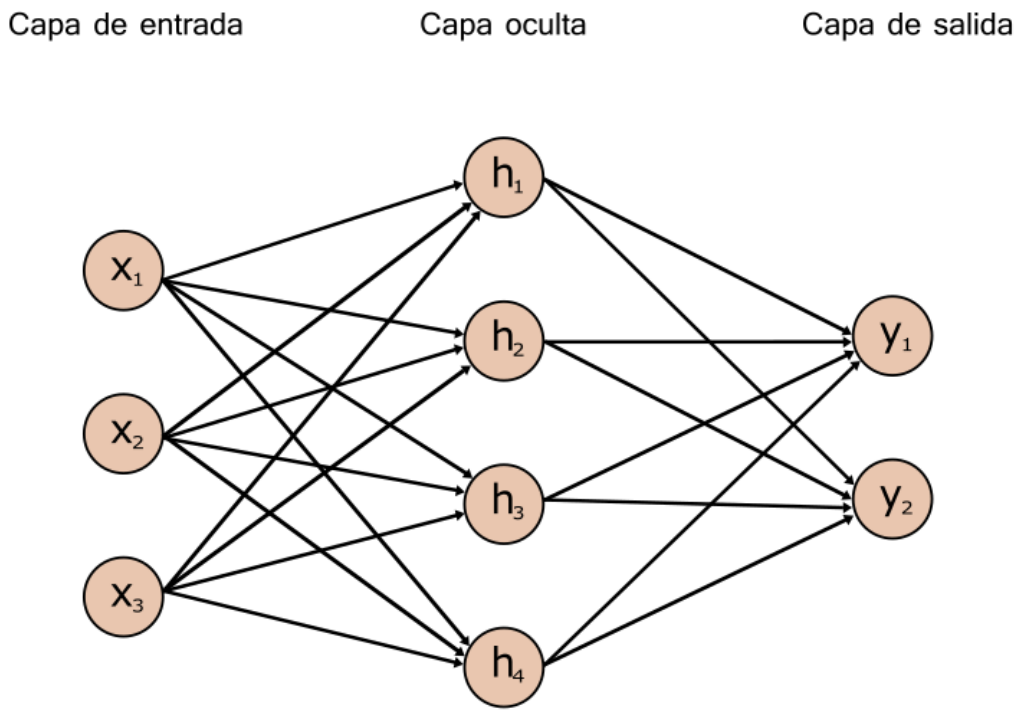


Figura 5.1: Composición clásica de una red neuronal en su capa de entrada, oculta y de salida.

5.2 Propagación hacia adelante

Una vez se introducen unos valores de entrada a la red neuronal, se inicializan los nodos de la capa de entrada junto a los pesos de todas las conexiones y el proceso de cálculo hacia adelante comienza, que consiste en determinar secuencialmente las salidas de todos los nodos en cada capa hasta obtener los valores de la capa de salida. Para ello se llevan a cabo multiplicaciones matriciales entre las salidas de cada capa y los pesos que las conectan a la siguiente capa. Es decir, cuando un nodo oculto o de salida recibe las entradas de la capa anterior, estas son multiplicadas por sus respectivos pesos y se suman. Además, a esta suma se le aplica la ponderación de un término de sesgo o de *bias* cuyo valor es constante. Los resultados pasan por una función de activación que decide si son o no relevantes para contribuir en la predicción. Actúa como una función de transformación que normaliza las salidas de los nodos de una capa y ayuda en el proceso de descenso por gradiente comentado en la siguiente sección. Las más comunes son la lineal, la *ReLu*, que actúa como una lineal si detecta un valor positivo pero los negativos los convierte en ceros, la sigmoide que es una función no lineal que normaliza los valores de entrada entre cero y uno, y la *softmax* que transforma las salidas de una capa en probabilidades y se utiliza a menudo para problemas de clasificación con múltiples categorías.

La salida s_i^n de un nodo i de la capa n , con $1 \leq i \leq M_n$ y $1 < n \leq N$, donde M_n es la cantidad de nodos en la capa n y N es el total de capas, se calcula de la siguiente forma:

$$s_i^n = g(\sum_{j=1}^{M_{n-1}} \Theta_{i,j}^n \cdot s_j^{n-1}) \text{ , donde:}$$

- $g(x)$ denota una función de activación sobre x,

- $1 \leq j \leq M_{n-1}$ y M_{n-1} es la cantidad de nodos de la capa $n-1$,
- $\Theta_{i,j}^{n}$ es el peso del enlace que conecta al nodo j con el nodo i de la capa n,
- s_j^{n-1} es la salida del nodo j de la capa $n-1$.

Cuando se obtienen resultados para los nodos de la capa de salida, se genera una predicción para los datos de entrada. Dependiendo del problema, ya sea de clasificación o de regresión, la cantidad de salidas devueltas varían. Por ejemplo, la presencia de múltiples valores suele indicar la probabilidad de pertenecer a una clase u otra, eligiendo aquella más alta como decisión final, y en redes neuronales con solo un valor de salida devuelto pueden hacer referencia a una clasificación binaria o a una predicción numérica.

5.3 Función de pérdida

Un factor importante para entrenar una red neuronal es la función de pérdida seleccionada para evaluar las salidas predichas. La función de pérdida cuantifica como de distintas son la salidas esperadas o reales y las predicciones generadas por el proceso de propagación hacia adelante, calculándose un error de predicción. En cada ciclo donde todas las muestras de entrenamiento pasan por la red, se guarda el promedio de la pérdida para observar su evolución. Al igual que en cualquier problema de optimización, se trata de minimizar o maximizar el valor de una función objetivo y conseguir el conjunto de variables de decisión que mejor se sitúen para la resolución del problema. En el contexto de las redes neuronales, se busca obtener aquella composición de pesos que producen el menor error posible.

En problemas de clasificación donde se busca establecer unas probabilidades de pertenencia a cada clase, se utilizan a menudo funciones de pérdida de entropía cruzada que indican como difieren las distribuciones de probabilidad predichas con las auténticas. En una clasificación binaria, la entropía cruzada binaria (BCE) para una muestra x se determina por:

$$BCE(x) = -y(x) \cdot \log p(x) - (1 - y(x)) \cdot \log(1 - p(x)) \text{ , donde:}$$

- $y(x)$ indica la clase real a la que pertenece x (0 o 1),
- $p(x)$ es la probabilidad de que x pertenezca a la clase 1,
- $1 - p(x)$ es la probabilidad de que x pertenezca a la clase 0.

Para problemas de clasificación con más de dos clases, se utiliza la entropía cruzada categórica, que calcula por separado la pérdida en cada clase y se suman. Dada una muestra x, su entropía cruzada categórica CCE se calcula de la siguiente manera:

$$CCE(x) = -\sum_{i=1}^{M} y_i(x) \cdot \log p_i(x) \text{ , donde:}$$

- M es el número total de clases,
- $y_i(x)$ indica si x pertenece a la clase i (1) o no (0),
- $p_i(x)$ es la probabilidad de que x pertenezca a la clase i.

Como se puede ver en la fórmula, bajas probabilidades predichas para la clase correcta incrementan más el error al tratarse de una función logarítmica que penaliza grandes diferencias entre $p_i(x)$ y $y_i(x)$.

El error cuadrático medio (MSE) y el error absoluto medio (MAE) suelen emplearse para medir el error de pronóstico en problemas de regresión que precisan una cantidad numérica. Las fórmulas para ambas funciones de pérdida son las siguientes:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

Donde:

- N es total de muestras,
- y_i es el valor real de la muestra i,
- $\hat{y}_i$ es el valor predicho de la muestra i.

Además de las funciones de pérdida mencionas, otras que son utilizadas habitualmente son la pérdida de Hinge, exponencial y la divergencia de Kullback-Leibler para clasificación, y la pérdida de Huber para regresión.

5.4 Propagación hacia atrás

Durante el proceso de aprendizaje, la red neuronal debe actualizar sus pesos cada vez que obtenga unas predicciones sobre los datos de entrada, cuyos cambios vienen dados por la función de pérdida. A este procedimiento se llama propagación hacia detrás o retropropagación del error e indica la dirección en la que se tiene que mover los pesos de la red para aproximarse a un valor mínimo de pérdida utilizando una técnica conocida como descenso por gradiente [37]. El objetivo es ajustar estos pesos utilizando como referencia el valor de la función de pérdida para que en las próximas iteraciones de entrenamiento las salidas de la red sean más cercanas a las salidas reales, teniendo que calcular el error en todos los nodos menos en la capa de entrada.

Los pesos se actualizan por capas en orden inverso, es decir, desde la de salida a la de entrada. Primero la red calcula el error de predicción y la derivada de la función de activación para cada nodo de la capa de salida. Multiplicando ambos resultados, se establece el error en el nodo i, δ_i^2 (suponiendo un perceptrón multicapa de una capa de entrada, una oculta y una de salida):

$$\delta_i^2 = g'(s_i^2)(t_i - s_i^2) \text{ , donde:}$$

- $g'(s_i^2)$ es la derivada de la función de activación en la capa de salida sobre la salida en el nodo i, s_i^2,
- t_i es el valor real en el nodo i.

Con estos errores, las salidas de los nodos de la capa oculta anterior s_j^1 y un factor de aprendizaje p, se determina el cambio que se produce en cada peso Θ_{ij}^2 :

$$\Delta\Theta_{ij}^2 = p\delta_i^2 s_j^1$$

Por tanto, el nuevo peso Θ_{ij}^2 será:

$$\Theta_{ij}^2 = \Delta\Theta_{ij}^2 + \Theta_{ij}^2$$

El cálculo de los nuevos pesos de la capa oculta Θ^1_{ij} sigue un proceso parecido. En este caso, los errores de los nodos ocultos δ^1_i se calculan multiplicando sus derivadas en la función de activación por los errores de la capa de salida δ^2_r por nodo r previamente calculados, junto a los pesos Θ^2_{ri}:

$$\delta^1_i = g'(s^1_i) \sum_{r=1}^{M_2} \delta^2_r \cdot \Theta^2_{ri} \text{ ,donde:}$$

- $g'(s^1_i)$ es la derivada de la función de activación en la capa oculta sobre la salida oculta s^1_i,
- M_2 es la cantidad de nodos de la capa de salida.

De nuevo, se consideran las salidas de la capa anterior que son los datos de entrada x_j, y el mismo factor de aprendizaje p para actualizar los pesos:

$$\Delta\Theta^1_{ij} = p\delta^1_i x_j$$

Entonces, los nuevos pesos son:

$$\Theta^1_{ij} = \Delta\Theta^1_{ij} + \Theta^1_{ij}$$

Con estos cálculos ya se han actualizado todos los parámetros de un perceptrón multicapa. A pesar de que la mejora del error con esta nueva configuración de parámetros posiblemente no sea del todo significativa, la red va repitiendo el proceso de retropropagación con nuevos datos que recibe y en más iteraciones. Un factor importante a controlar en este paso es el sobreajuste de pesos, que ocurre cuando la red neuronal presta demasiada atención al conjunto de muestras de entrenamiento que se le presentan y por tanto no es capaz de generalizar bien para otros datos que nunca antes ha visto. Para hacer frente a este problema, se define un conjunto de muestras validación sobre las que únicamente se realiza la propagación hacia adelante para obtener sus predicciones y medir el error, pero no se propaga para ajustar pesos. Signos de sobreajuste del modelo aparecen cuando la evolución del error de entrenamiento progresa adecuadamente mientras se queda estancada o no mejora para el conjunto de validación.

5.5 Tipos de redes neuronales

Visto la composición y los mecanismos clásicos de aprendizaje de las redes neuronales, se presentarán otras variantes populares que son creadas para tareas más específicas y que en comparación con un simple perceptrón multicapa, suelen obtener mejores prestaciones a la hora de evaluar otras estructuras de atributos como pueden ser imágenes o secuencias de datos. Los tipos de redes a destacar son las redes neuronales convolucionales y las redes neuronales recurrentes.

5.5.1. Redes neuronales convolucionales

Las redes neuronales convolucionales (CNNs) se especializan en tareas de procesamiento y extracción de características de imágenes o en la visión por computador, aunque son también utilizadas en el reconocimiento de voz y en la traducción automática de idiomas. A diferencia de las redes neuronales clásicas que operan con vectores unidimensionales de entrada de datos, las CNNs reciben datos o píxeles en tres dimensiones, cada una haciendo referencia a la altura, anchura y número de canales (uno para escala de grises y tres para RGB).

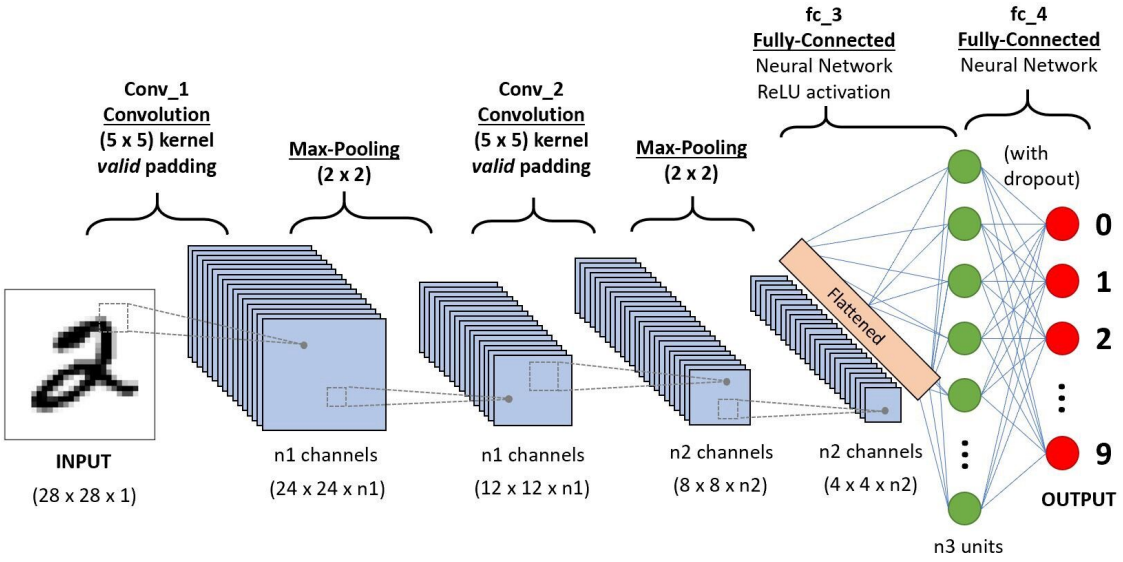


Figura 5.2: Arquitectura de una red neuronal convolucional para la clasificación de imágenes. Fuente: [6].

Dada una imágen de entrada, la red convolucional pasa una serie de filtros sobre ella que capturan distintas propiedades. Como se observa en la Figura 5.2, el objetivo es agregar la información recibida de forma que se reduzca la cantidad de elementos con los que se tienen que operar y a la vez ayudar a clasificar el contenido de la imagen. Cada capa en la red se ocupa de procesar ciertos tipos de características, desde las de bajo nivel como bordes o esquinas, hasta las de más alto nivel que incluyen objetos y formas. Los tres tipos de capas más habituales son la capa convolucional, la capa de submuestreo o *Pooling* y la capa densamente conectada.

5.5.2. Redes neuronales recurrentes

Las redes neuronales recurrentes (RNNs) son un tipo de red neuronal que procesan datos temporales o secuenciales, siendo muy utilizadas especialmente para tareas relacionadas con el procesamiento del lenguaje natural como el análisis de sentimiento, traducción automática, generación de discurso y el etiquetado de texto, incluyendo el reconocimiento de entidades nombradas. Una red neuronal clásica recibe entradas que son independientes entre sí y que corresponden a muestras del conjunto de datos con el propósito de devolver unas predicciones asociadas a cada una, pero en ocasiones pueden existir dependencias entre estas observaciones que no es capaz de modelar, dando lugar a las redes recurrentes. Lo que les hacen distintas a esta variantes de redes son que tienen memoria de información previa que ayuda a determinar las predicciones sobre datos en posiciones posteriores dentro de la secuencia. En relación a esto, las RNNs también pueden pronosticar el comportamiento de series temporales donde existen tendencias y patrones cíclicos identificables.

La esencia de cualquier RNN es almacenar una serie de estados ocultos asociados a cada paso temporal para no utilizarse únicamente como salida de esa posición, sino que además servir como entrada de cara a calcular el próximo estado oculto del paso de la secuencia (Figura 5.3). Esto significa que se toma una decisión combinando los valores de entrada del paso actual y la información arrastrada de entradas anteriores. Como en las redes neuronales artificiales, todas las entradas y salidas son ponderadas por matrices de pesos pero ahora estas son las mismas en todas las capas temporales. Los pesos son

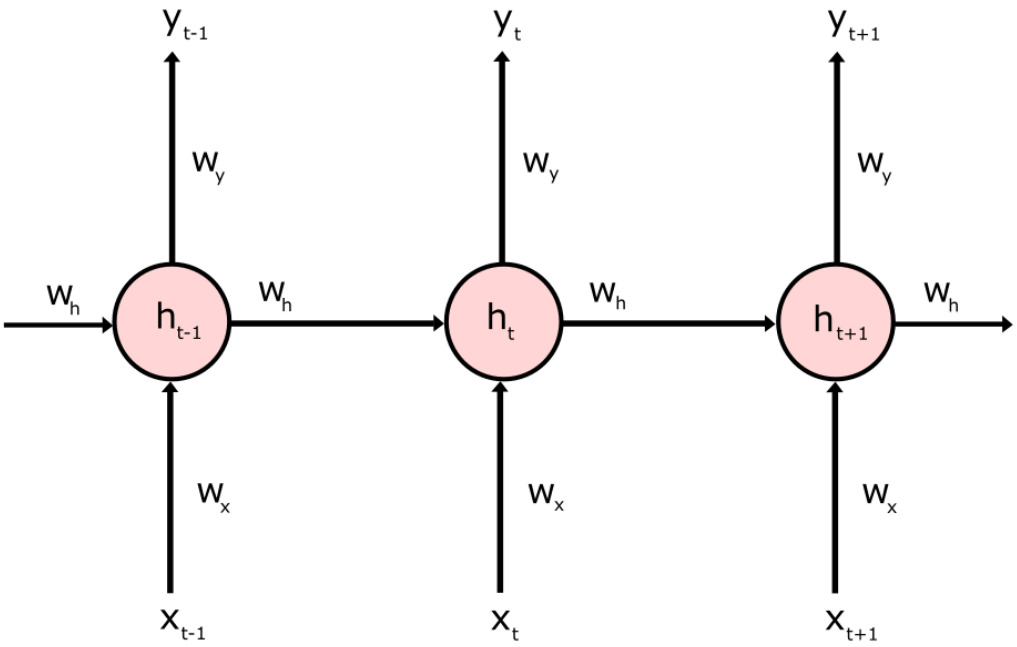


Figura 5.3: Ejemplo de una composición muchos a muchos de una RNN.

ajustados durante el entrenamiento por un proceso de retropropagación en el tiempo (BPTT) basado en el descenso por gradiente.

Retropropagación en el tiempo

El mecanismo de actualización de pesos en una RNN funciona igual que en una red neuronal artificial, donde primero se realiza el paso hacia adelante en la red para obtener unas salidas y un error de predicción en cada posición de la secuencia, y luego se hace el paso hacia detrás para determinar como se tienen que modificar los pesos en función de la derivada del error. La diferencia está en que se consideran dependencias temporales previas y el error se acumula por cada paso en la secuencia. Suponiendo que solo existe una capa de estados ocultos, son tres las matrices de pesos a considerar:

- W_x: Representa la matriz de pesos que conecta las entradas en cada paso x_t con la capa de estados ocultos,
- W_y: Representa la matriz de pesos que conecta a cada estado oculto h_t con su salida y_t,
- W_h: Representa la matriz de pesos que conecta al estado oculto de un paso al estado del paso siguiente.

Antes de ver el funcionamiento de la retropropagación en el tiempo, hay que comprobar como se determina cada salida y_t de la red recurrente junto a sus errores E_t. En primer lugar, el estado oculto h_t se calcula aplicando una función de activación sobre la suma de la entrada del paso actual x_t y el estado oculto anterior h_{t-1}, ponderados por sus respectivas matrices:

$$h_t = g(x_t \cdot W_x + h_{t-1} \cdot W_h)$$

La salida y_t es entonces el producto del vector resultante en h_t con la matriz de pesos W_y, donde también se puede aplicar una función de activación sobre el resultado:

$$y_t = h_t \cdot W_y$$

Con y_t y la salida real o esperada d_t, se determina el error E_t dependiendo de la función de pérdida seleccionada. Por ejemplo con la diferencia cuadrática se calcula como:

$$E_t = (d_t - y_t)^2$$

Para actualizar cada matriz, es necesario encontrar las derivadas parciales de la función de pérdida en un paso temporal concreto con respecto a la propia matriz de pesos. Se consideran los anteriores pasos temporales al modificar cada matriz mediante el descenso por gradiente. Dicho esto, el cálculo para ajustar W_y es el siguiente:

$$\frac{\mathrm{d}E_t}{\mathrm{d}W_y} = \frac{\mathrm{d}E_t}{\mathrm{d}y_t} \cdot \frac{\mathrm{d}y_t}{\mathrm{d}W_y}$$

En caso de W_h, hay que considerar todos los estados ocultos que contribuyen a la salida y_t y por tanto se acumulan sus gradientes:

$$\frac{\mathrm{d}E_t}{\mathrm{d}W_h} = \sum_{i=1}^{t} \frac{\mathrm{d}E_t}{\mathrm{d}y_t} \cdot \frac{\mathrm{d}y_t}{\mathrm{d}h_i} \cdot \frac{\mathrm{d}h_i}{\mathrm{d}W_h}$$

Al igual que en W_h, para ajustar W_x se necesita considerar todas las contribuciones de estados ocultos a la salida y_t:

$$\frac{\mathrm{d}E_t}{\mathrm{d}W_x} = \sum_{i=1}^{t} \frac{\mathrm{d}E_t}{\mathrm{d}y_t} \cdot \frac{\mathrm{d}y_t}{\mathrm{d}h_i} \cdot \frac{\mathrm{d}h_i}{\mathrm{d}W_x}$$

Un problema de este método de ajuste de pesos es que el gradiente del error puede crecer de forma incontrolada sobre la red como consecuencia de existir múltiples estados ocultos sobre los que se calculan sus derivadas (problema conocido como la explosión de gradiente [38]). Ante esta situación, lo que se suele hacer es establecer un umbral máximo para el gradiente y si es superado se aplica una normalización. De manera contraria, si se propaga el error en muchos pasos temporales, su gradiente acumulado puede ser muy pequeño y la contribución de pasos posteriores se pierde conforme se avanza en la secuencia. Este es el problema llamado desvanecimiento del gradiente [39], que es muy común en cualquier RNN y puede causar la pérdida de dependencias temporales a largo plazo. Las memorias a corto y largo plazo o LSTM se diseñaron específicamente para solucionar este problema, siendo utilizadas en el modelo NER diseñado y se hablarán de ellas con mayor detalle en el próximo capítulo.

CAPÍTULO 6
Descripción del modelo

En este capítulo se describirán las componentes de la arquitectura neuronal propuesta y la interacción entre ellas para lograr el reconocimiento y clasificación de las entidades nombradas en la colección de registros matrimoniales. Las componentes que forman la red son: 1) Las representaciones distribuidas de características formadas por vectores n-dimensionales de números reales, tanto para las palabras como para los caracteres, 2) El codificador de contexto que utiliza las representaciones de cada palabra y de los caracteres que las componen para extraer conocimiento basándose en las demás palabras cercanas en la secuencia de entrada, que lo forma una LSTM bidireccional, y 3) el decodificador de etiquetas para obtener la mejor secuencia de salida, utilizando un campo aleatorio condicional (CRF).

6.1 Representación de los registros y etiquetas

Antes de explicar el funcionamiento de cada componente, es conveniente señalar cómo se representa la información textual de entrada al modelo y las etiquetas a predecir. Primero, a cada palabra distinta en la colección de registros matrimoniales, se le asigna un valor numérico o índice único para ser codificada, y estas correspondencias se almacenan en un diccionario. Esta operación es necesaria puesto que las redes neuronales operan siempre con valores numéricos y no con cadenas de palabras por lo tanto omitir este paso resultará en un error durante el entrenamiento. Una vez establecido el diccionario, se aplican los índices a las palabras de todos los registros para obtener secuencias numéricas.

Puesto que los registros tienen números de palabras diferentes y la capa de entrada a la red es de una longitud fija, todas las secuencias han de tener obligatoriamente la misma longitud y coincidente con la dimensión de entrada esperada, por lo que tiene que establecerse una longitud común para todas. Dicha longitud es un hiperparámetro del modelo que se tiene que decidir, aunque en el gráfico anterior de distribuciones (Figura 4.4) se ha comprobado que la mayoría de registros constan de una longitud como muy alta de 50. Para aquellas secuencias que no alcancen el tamaño acordado, se utiliza una técnica habitual llamada *padding* que consiste en rellenarlas con un índice reservado (normalmente el cero) que no representa ningún término concreto. En la Figura 6.1 se ilustra la aplicación de todo el proceso de representación sobre un registro de la colección.

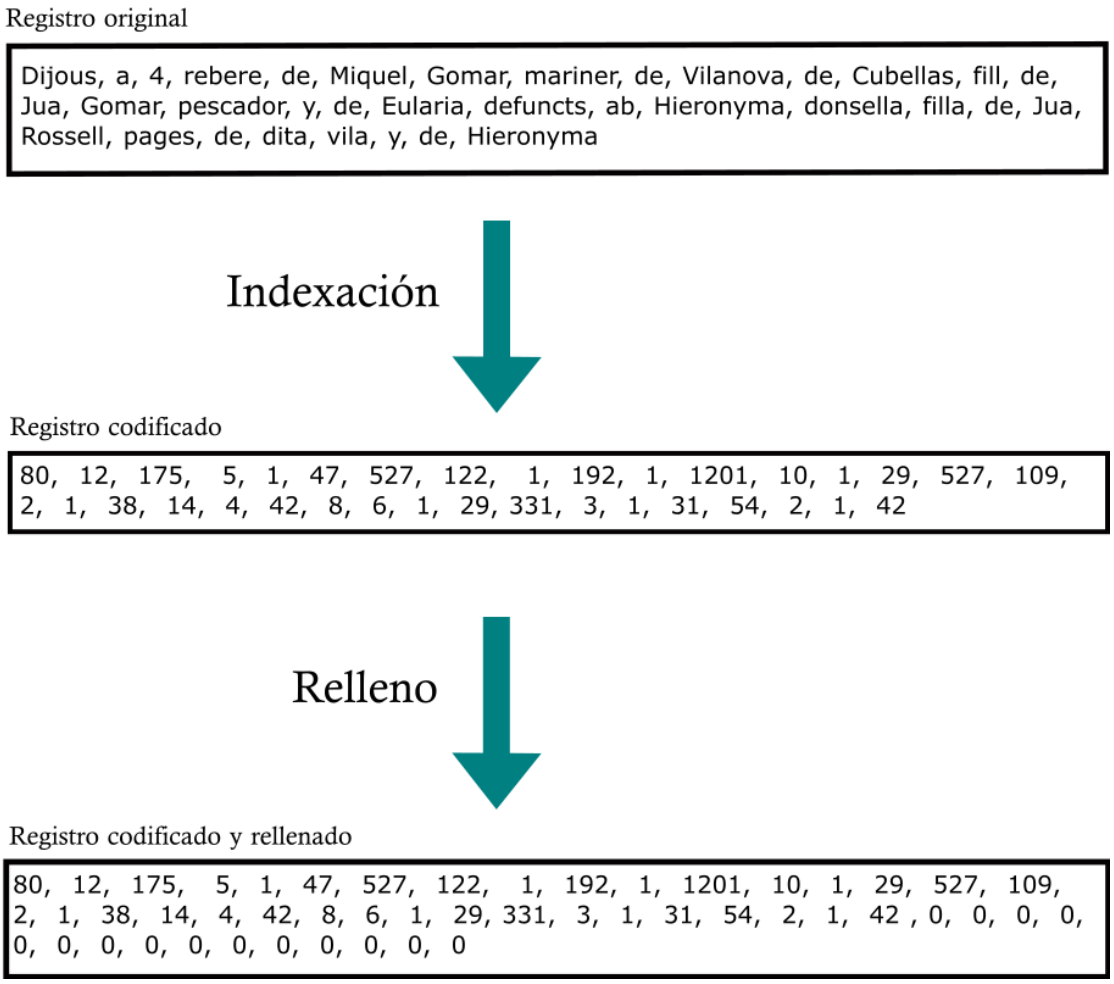


Figura 6.1: Proceso de representación de un registro arbitrario. Primero se indexan los términos y luego se rellena la secuencia hasta alcanzar la longitud deseada. El relleno se puede hacer alternativamente al principio de la secuencia.

El proceso de representación de los caracteres es análogo al de palabras, donde también se almacenan las correspondencias entre carácter e índice, y se fija una longitud de secuencia menor. Lo que cambia es la forma de representación, que ya no la componen vectores unidimensionales, sino matrices (una por registro) que contienen los caracteres codificados de cada palabra en sus filas, es decir, su dimensión es de longitud de secuencia de palabras por longitud de la secuencia de caracteres. El relleno se aplica para aquellas palabras que no superen la longitud de caracteres máxima. En la Tabla 6.1, se ilustran las secuencias de caracteres asociadas a algunas de las palabras del registro utilizado en la Figura 6.1.

Las etiquetas también son codificadas de la misma forma que las palabras, con tantos índices como etiquetas distintas existan, y con una longitud de secuencia equivalente a la propia de las palabras. El índice de relleno seleccionado es el mismo que el de las etiquetas de los términos que no representan entidades (del tipo “other” o “none-other”), aunque no importa demasiado cuál sea ya que los índices de relleno en las secuencias de entrada se pueden enmascarar. El enmascaramiento consiste en indicar a las capas encargadas de procesar la información secuencial que ciertos pasos temporales, aquellos con el índice reservado cero, faltan y que por tanto no deben ser tenidos en cuenta al ajustar sus representaciones distribuidas ni sus vectores de contexto.

Tabla 6.1: Codificación a nivel de carácter para las palabras de un registro.

Palabra	Codificación									
Dijous	28	9	42	3	18	11	0	0	0	0
a	4	0	0	0	0	0	0	0	0	0
4	44	0	0	0	0	0	0	0	0	0
rebere	2	8	15	8	2	8	0	0	0	0
...	...									
vila	33	9	5	4	0	0	0	0	0	0
y	13	0	0	0	0	0	0	0	0	0
de	7	8	0	0	0	0	0	0	0	0
Hieronyma	24	9	8	2	3	14	13	12	4	0

6.2 Representaciones distribuidas de características

En el aprendizaje profundo aplicado a NER, es bastante frecuente disponer de representaciones distribuidas de los elementos del lenguaje, que recogen información semántica y sintáctica de cada uno. Una representación distribuida es un vector denso de números reales asociado a un elemento, por ejemplo a una palabra, donde cada dimensión hace referencia a una propiedad oculta. La idea principal detrás de este concepto es determinar la proximidad entre observaciones en un espacio de n componentes, cuyas distancias dependen de la similitud entre estas n características numéricas y de esta forma determinar grupos de interés. Durante el entrenamiento en cada iteración, los vectores se irán ajustando a medida que el modelo aprenda a conocer nuevas relaciones, siendo las representaciones a nivel de palabra y de carácter.

A diferencia de las representaciones distribuidas, las representaciones locales o *one-hot* tienen una longitud equivalente al tamaño del vocabulario $|V|$, donde cada dimensión indica la presencia de un término definido en el vocabulario. Un ejemplo de esta representación se muestra en la Figura 6.2. Para un término en concreto, su vector asociado contendrá tantos ceros como términos distintos a este existan en el vocabulario ($|V| - 1$), y un uno para su posición correspondiente. Esto significa que por muy similares que sean dos palabras, serán ortogonales en el espacio vectorial al tener representaciones totalmente diferentes, causando que las dependencias contextuales no se capturen. Otra desventaja es que requiere mucha memoria para tamaños grandes de vocabulario puesto que el número de dimensiones crece de forma lineal. Por estos motivos no se ha optado por esta representación.

perro	1	0	0	0
gato	0	1	0	0
pájaro	0	0	1	0
pato	0	0	0	1

Figura 6.2: Representación *one-hot* para los términos de un vocabulario V= {perro, gato, pájaro, pato}. Fuente: [7].

6.2.1. Representaciones a nivel de palabra

El primer tipo de representaciones distribuidas que aprende la red neuronal son para las palabras. Muchos sistemas presentados en otros trabajos implementan representaciones distribuidas ajustadas de palabras por modelos como el *Word2Vec* basado en redes neuronales multicapa entrenadas con con dos posibles arquitecturas: el modelo de *skip-gram* o de omisión gramatical, donde para una palabra central, se intenta predecir las palabras que la rodean en base a los vectores de palabras, y el *continuous bag of words* (CBOW), que para una secuencia de palabras contexto, se predice la palabra central.

En la arquitectura propuesta, las representaciones de palabras son almacenadas en una capa de incrustación o *embedding*, que a partir de las secuencias entrantes de palabras codificadas, busca los vectores reales de longitud fija a través de los índices recibidos. Antes de entrenar la red, las representaciones son inicializadas con números aleatorios y a medida que se aprende del lenguaje, los pesos van siendo ajustados. El funcionamiento en esta capa se asemeja a la de una tabla de búsqueda que asocia los índices de cada término con sus respectivos vectores de características. La capa de incrustación es en realidad la primera capa oculta del modelo y requiere de tres principales parámetros para su implementación:

- El tamaño del vocabulario o longitud del diccionario con las correspondencias entre palabras e índices, incluida la de las palabras de relleno.
- La dimensión de los vectores de representación que definen la cantidad de características que se quieren obtener de cada palabra.
- La longitud de las secuencias codificadas de entrada al modelo.

Adicionalmente, se incluye un parámetro booleano opcional de enmascaramiento (conocido como *mask zero*) para informar al modelo que faltan algunas palabras en la secuencia de entrada y que estas son indexadas con el valor de cero. Es importante especificar este parámetro ya que indica a las capas superiores de la red que se han de ignorar estos índices puesto que no interesa conocer las predicciones de palabras inexistentes cuya única función es cuadrar la longitud de las secuencias de enteros con la dimensión de entrada al modelo.

La salida de la capa es una matriz que contiene las representaciones vectoriales, con tantas filas como palabras existan en el vocabulario y con tantas columnas como dimensiones de características se quieran considerar. Con esta matriz se seleccionan las representaciones de las palabras en la secuencia entrante, que serán pasadas a la unidad de memoria LSTM bidireccional junto con las codificaciones de caracteres, en caso de operar con ellos. En la Figura 6.3 se visualizan las representaciones extraídas de la capa de incrustaciones de palabras una vez finalizado el ajuste de parámetros, sobre las dos primeras dimensiones de algunas palabras de los registros matrimoniales. En el gráfico se ve como palabras de la misma categoría semántica tienden a estar en la misma área de representación. Es interesante ver que el término "viudo" está en una posición más distante a las demás palabras de su categoría, especialmente con su forma en femenino, aunque solo se compruebe su similitud en dos dimensiones.

6.2.2. Representaciones a nivel de carácter

Las incrustaciones de palabras han llevado a producir buenos resultados de clasificación de entidades nombradas sobre distintas colecciones anotadas de textos, al encontrar asociaciones entre palabras en función de su significado. Sin embargo, un estudio reciente

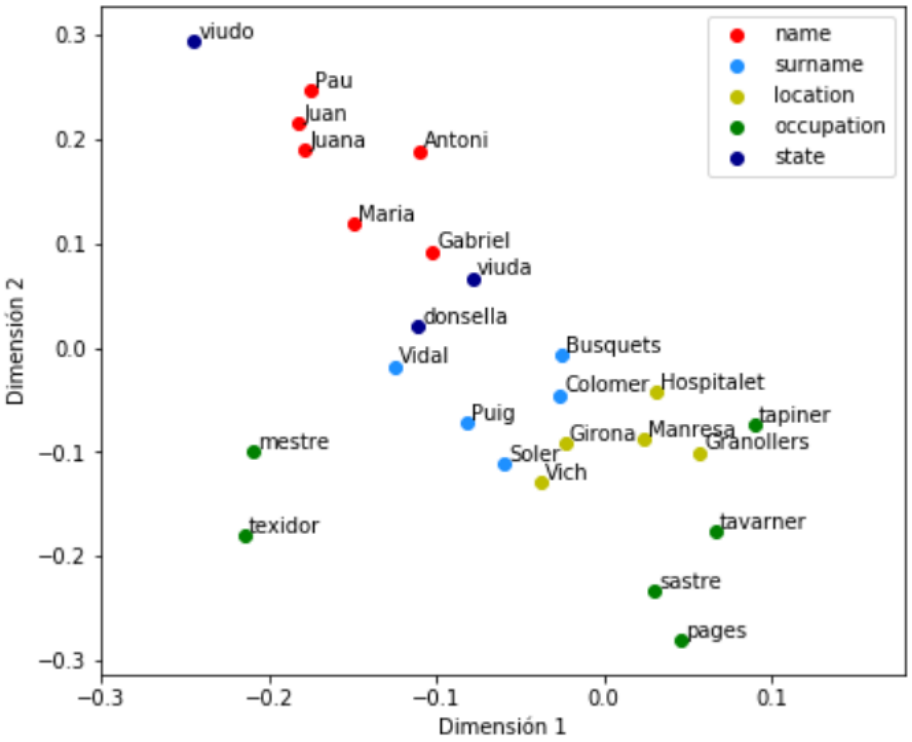


Figura 6.3: Representaciones distribuidas en las dos primeras componentes sobre términos que aparecen en la colección de registros matrimoniales.

de Ha et al. [40] destaca la importancia que los caracteres tienen en determinar similitudes sintácticas y por tanto mejorar el rendimiento de los modelos. Combinar las representaciones distribuidas de palabras con las de los caracteres implica reconocer palabras que no han sido vistas pues sus incrustaciones por sí solas están limitadas al vocabulario de la colección. Por ejemplo, al poner a prueba el clasificador, una palabra mal pronunciada o variante de otra palabra encontrada posiblemente no tenga un vector numérico asociado en la capa de incrustaciones de palabras, y es aquí cuando puede utilizarse su estructura interna para dar información útil acerca de su significado gracias a los vectores de representaciones de caracteres.

Al igual que las palabras, las representaciones de caracteres pueden estar ya preentrenadas a partir de modelos de lenguaje como el *char2vec*, pero de nuevo, estas serán aprendidas por la red ya que las dependencias encontradas en estos modelos no pueden ser generalizadas a la colección de registros matrimoniales con un vocabulario tan específico de términos. Los vectores numéricos de cada letra y símbolo son inicializados con números aleatorios en un rango definido y almacenados en otra capa de incrustaciones, que recibirá de entrada las matrices de carácteres codificados y con los índices selecciona las representaciones correspondientes. Los parámetros a introducir para la creación de la capa hacen referencia a los mismos elementos que antes pero relativos a los caracteres y son los siguientes:

- La cantidad de caracteres a representar, que es la longitud del diccionario con las correspondencias entre índices y caracteres. De nuevo, se debe considerar la inclusión de los caracteres de relleno.

- La dimensión de los vectores de representación que definen la cantidad de propiedades que se quieren obtener de cada carácter. Esta dimensión suele ser menor a la de los vectores de representaciones de palabras.

- La longitud de las secuencias de caracteres codificadas de entrada al modelo, referente a las palabras.

También se añade la condición de enmascaramiento sobre los caracteres de relleno y de esta forma ignorar sus representaciones. En este caso, la salida de la capa de incrustaciones de caracteres es un vector tridimensional que contiene las representaciones distribuidas de caracteres para cada palabra de la secuencia entrante.

Una vez entrenada esta capa, se han ilustrado en la Figura 6.4 las representaciones para los distintos símbolos y letras (minúsculas y en mayúsculas) presentes en los registros matrimoniales sobre las dos primeras dimensiones. Se aprecia una correlación entre las dos dimensiones, donde los dígitos se sitúan en la parte inferior derecha del gráfico al obtener valores altos para la primera dimensión y bajos para la segunda. El comportamiento opuesto ocurre con la mayoría de caracteres alfabéticos, aunque es cierto que algunos como la "b", "d" y "e" se han acoplado en la zona donde se ubican los dígitos, que indica algún tipo de relación con ellos. Por último, llama la atención que las versiones en mayúscula y en minúscula de cada letra no están cercanas entre sí en este espacio de representación.

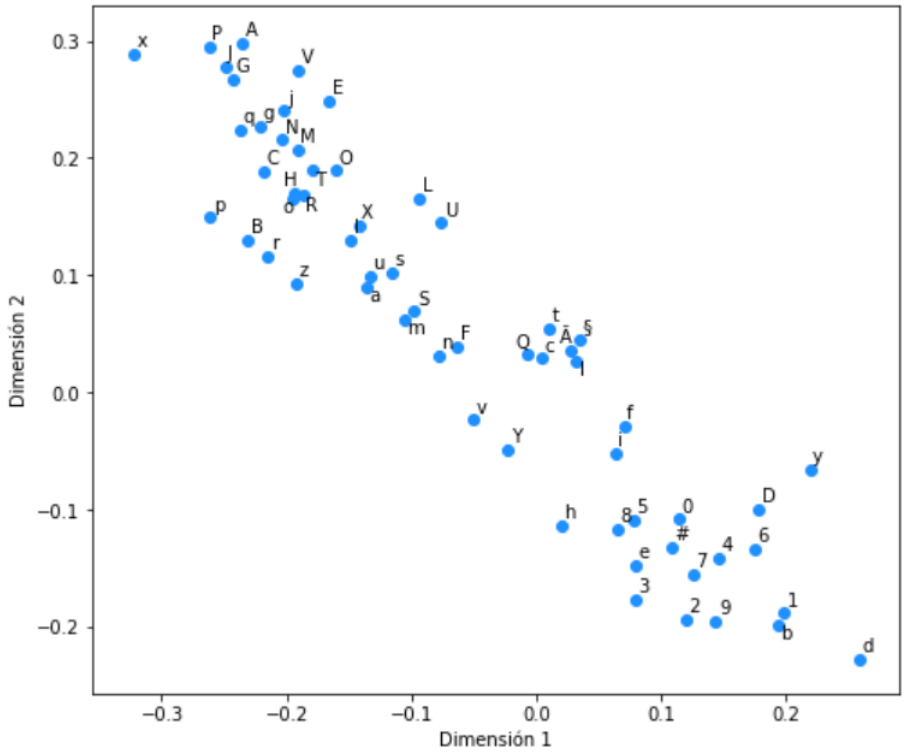


Figura 6.4: Representaciones distribuidas en las dos primeras componentes sobre caracteres que aparecen en la colección de registros matrimoniales.

A continuación, las representaciones de caracteres se tienen que combinar con las representaciones de palabras y poder usarse conjuntamente por el codificador de contexto. Puesto que tienen dimensiones diferentes, la información a nivel de carácter por palabra se pasa a una unidad de memoria LSTM (no la bidireccional) para codificar la palabra a un vector de dimensión reducida. Esta operación se asimila a una función de aplanamiento sobre un matriz, que en este caso son las representaciones de los caracteres de una palabra, la diferencia es que no se conserva su cantidad de elementos. Un método común y alternativo a la LSTM para extraer las representaciones a nivel de carácter y convertirlas en un único vector, es utilizar una red neuronal convolucional (CNN). La CNN

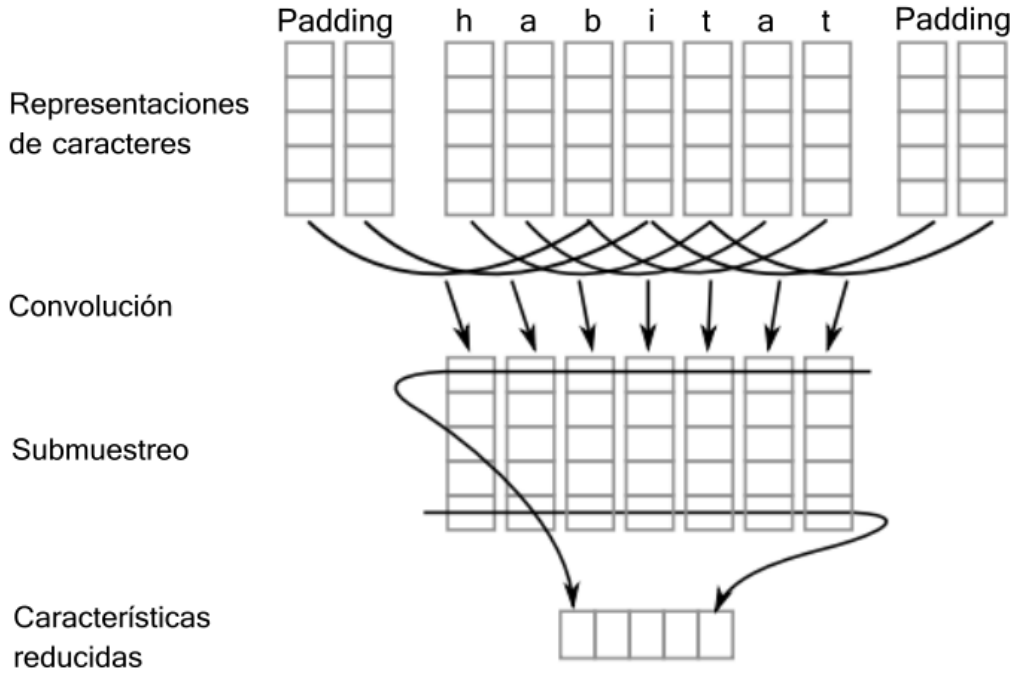


Figura 6.5: CNN que extrae las representaciones distribuidas de los caracteres de una palabra y crea un vector reducido de características. Imagen adaptada del artículo de Chiu y Nichols [2].

captura propiedades ortográficas de la palabra pasando diferentes filtros sobre las representaciones distribuidas de sus caracteres y aplica un submuestreo de tipo *max pooling* para reducir la dimensión de salida. Este proceso se ilustra en la Figura 6.5.

Antes de pasar estos vectores al codificador de contexto, se aplica una función de eliminación de valores (conocida como *spatial dropout*) con cierta probabilidad para reducir el posible sobreajuste del modelo. El propósito es saber cómo actuaría la red en caso de no disponer de representaciones de palabras completas.

6.3 Codificador de contexto

Una vez concatenados los vectores transformados de caracteres con las representaciones distribuidas de palabras aprendidas, el siguiente paso es introducirlos al codificador de contexto, un bloque con unidades de memoria encargado de capturar información de palabras en base a su contexto. Antes de explicar su funcionamiento, es necesario introducir qué son las unidades de memoria a corto y largo plazo (LSTM) pues sus salidas determinan la representación final de cada palabra.

6.3.1. Memoria a corto y largo plazo (LSTM)

Las redes neuronales recurrentes (RNN) tienen el inconveniente de que durante el proceso de retropropagación, el gradiente del error que ajusta los pesos se va desvaneciendo o acercando a cero a medida que incrementa los pasos temporales (problema del desvanecimiento del gradiente). Si el gradiente es muy pequeño, significa que no se está aprendiendo del todo del error y por tanto el efecto de aprendizaje es nulo sobre todo en las capas correspondientes a los primeros pasos de la secuencia. Las LSTM son una par-

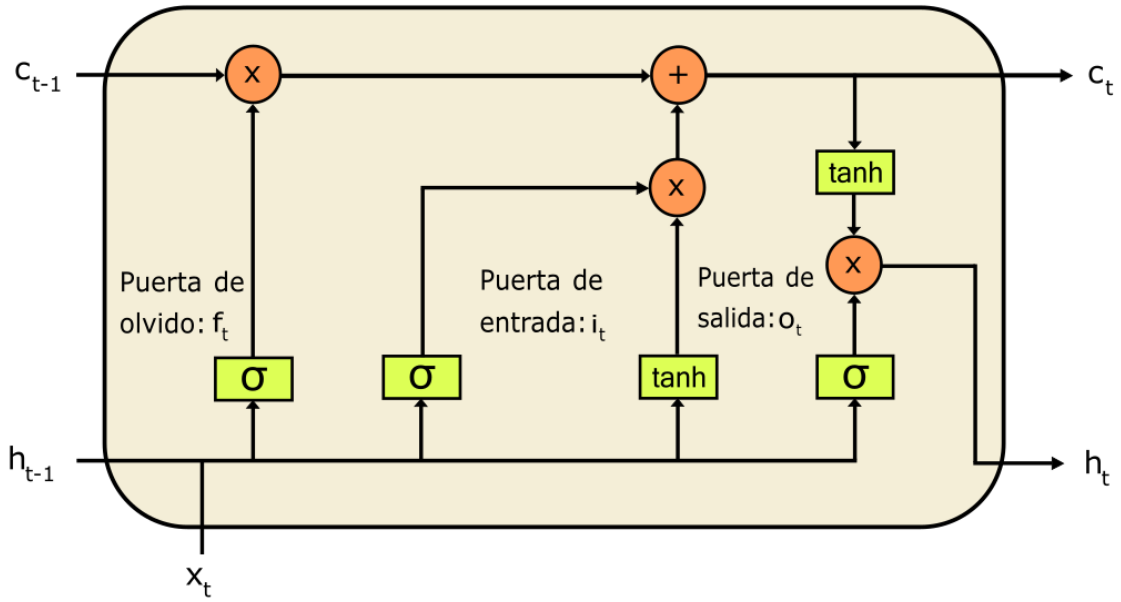


Figura 6.6: Composición interna de una LSTM. Dados el estado oculto del paso previo h_{t-1} , la entrada en el paso actual x_t y el estado celular del paso previo c_{t-1}, se calcula el nuevo estado oculto h_t y se actualiza el estado celular c_t.

ticularidad de las RNNs a la hora de tratar con datos secuenciales de un orden específico y dentro del ámbito del procesamiento del lenguaje natural es muy usada por su capacidad de recordar información a corto y largo plazo, creadas como solución al problema de desvanecimiento del gradiente visto en las RNNs.

Como se observa en la Figura 6.6, una LSTM está formada por tres puertas fundamentales que aprenden cuál es la información relevante en el paso actual para almacenar o aquella no tan importante para ignorar, un estado celular encargado de almacenar la información relevante a lo largo de la secuencia, la entrada en el paso temporal concreto, y los estados ocultos del instante previo y actual. La clave para retener u olvidar esta información son las funciones de sigmoide y tangente hiperbólica de activación que están presentes en las puertas, siendo estas de olvido, de entrada y de salida, además de los operadores de multiplicación y adición elemento a elemento. En las siguientes subsecciones se señalan las operaciones realizadas por cada puerta y la actualización del estado celular.

Puerta de olvido

Esta puerta es la encargada de decidir qué información debe ser descartada del estado celular c_{t-1}. La información del anterior estado oculto h_{t-1}, y de la entrada correspondiente al paso temporal actual x_t se combina y pasa por una función sigmoide de activación, creando un vector de olvido f_t:

$$f_t = \sigma(U_f \cdot h_{t-1} + W_f \cdot x_t + b_f) \text{ , donde:}$$

- $\sigma(x)$ denota la función sigmoide sobre x,
- U_f,W_f son matrices de pesos asociadas al estado oculto en el paso previo y entrada del paso actual, respectivamente,
- b_f es un vector de términos independientes.

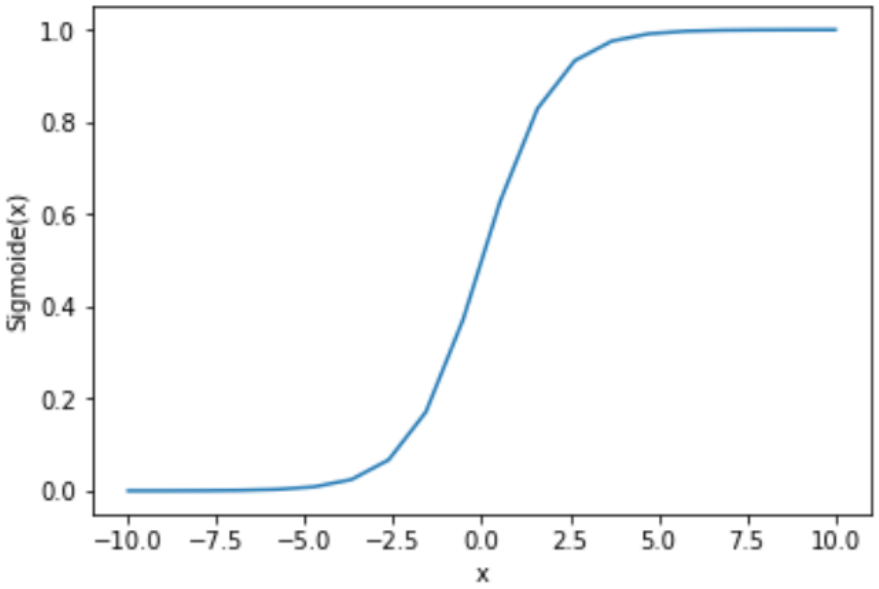


Figura 6.7: Función sigmoide para valores de entrada entre -10 y 10.

En la Figura 6.7 se ilustra la función sigmoide en un rango acotado entre 0 y 1 para cualquier número real. Un valor más cercano a 1 significa que la característica es guardada y más cercano a 0 implica que se descarta.

Puerta de entrada

La puerta de entrada decide qué información debe ser guardada en el estado celular. Al igual que en la puerta de olvido, la información del estado anterior y de la entrada actual pasan de nuevo por otra función sigmoide para crear el vector de entrada i_t y decidir qué valores siguen siendo guardados y cuáles no. La diferencia es que ahora se aplica por otro lado una función de tangente hiperbólica para normalizar los valores entre -1 y 1, creando un vector de posibles candidatos $\bar{c}_t$ a añadir al estado celular. Este resultado luego se multiplica por la salida de la función sigmoide para actualizar el estado celular. A continuación, se muestran las operaciones involucradas para determinar i_t y $\bar{c}_t$:

$$i_t = \sigma(U_i \cdot h_{t-1} + W_i \cdot x_t + b_i),$$

$$\bar{c}_t = \tanh(U_c \cdot h_{t-1} + W_c \cdot x_t + b_c)$$

Donde:

- $\tanh(x)$ denota la función de la tangente hiperbólica sobre x,
- U_i, W_i, U_c, W_c son matrices de pesos en las distintas operaciones asociadas al estado oculto en el paso previo y entrada del paso actual.
- b_i, b_c son vectores de términos independientes.

En la Figura 6.8 se representa la función de la tangente hiperbólica, presentando la misma forma que la función sigmoide, la diferencia es que se conserva el signo del valor de entrada tras aplicarse.

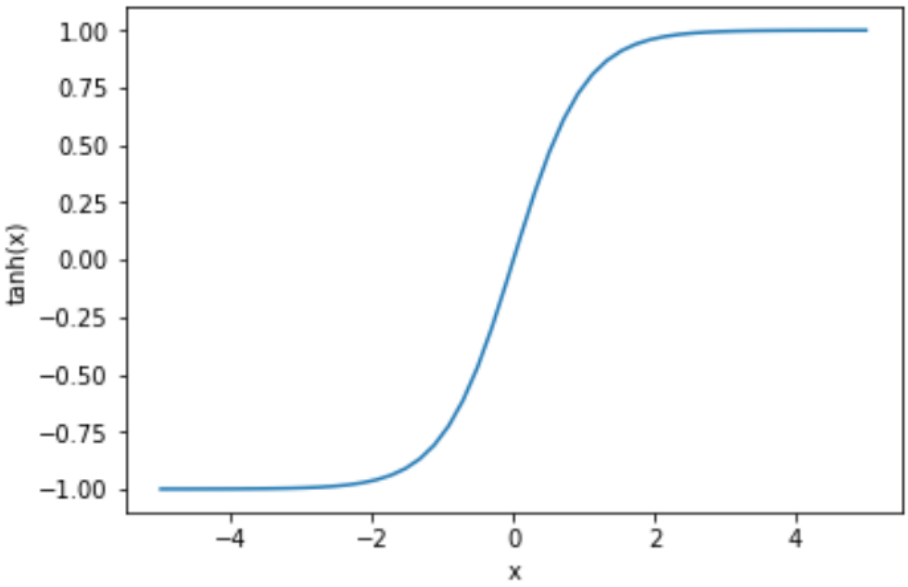


Figura 6.8: Tangente hiperbólica para valores de entrada entre -5 y 5.

Actualización del estado celular

Una vez detectada la información relevante y la que se ha de ignorar, el siguiente paso es realizar los cambios en el estado celular, que transporta información relevante encontrada en cada paso a lo largo de la secuencia. Se actualiza de tal forma que se multiplica su contenido del anterior paso temporal c_{t-1} por el vector de olvido f_t para descartar valores en el paso actual y luego se hace una operación de suma con $i_t * \bar{c}_t$, que contiene la información de los nuevos candidatos escalada por su relevancia. Con esto se obtiene el nuevo estado celular c_t:

$$c_t = f_t * c_{t-1} + i_t * \bar{c}_t$$

Puerta de salida

Por último, la puerta de salida define cuál será el estado oculto h_t a pasar a la siguiente unidad de memoria. Para conseguir el estado final, por un lado se aplica una función sigmoide al estado oculto previo h_{t-1} combinado con la entrada actual x_t para obtener el vector de salida o_t, y por otro se aplica la tangente hiperbólica al estado celular actualizado c_t. Estos dos resultados de las funciones son multiplicados para definir el nuevo estado oculto, que será pasado junto con el estado celular a la siguiente etapa temporal. Los operaciones realizadas por esta puerta son las siguientes:

$$o_t = \sigma(U_o \cdot h_{t-1} + W_o \cdot x_t + b_o),$$

$$h_t = o_t * \tanh(c_t)$$

Donde:

- U_o, W_o son matrices de pesos asociadas al estado oculto en el paso previo y entrada del paso actual, respectivamente,
- b_o es un vector de términos independientes.

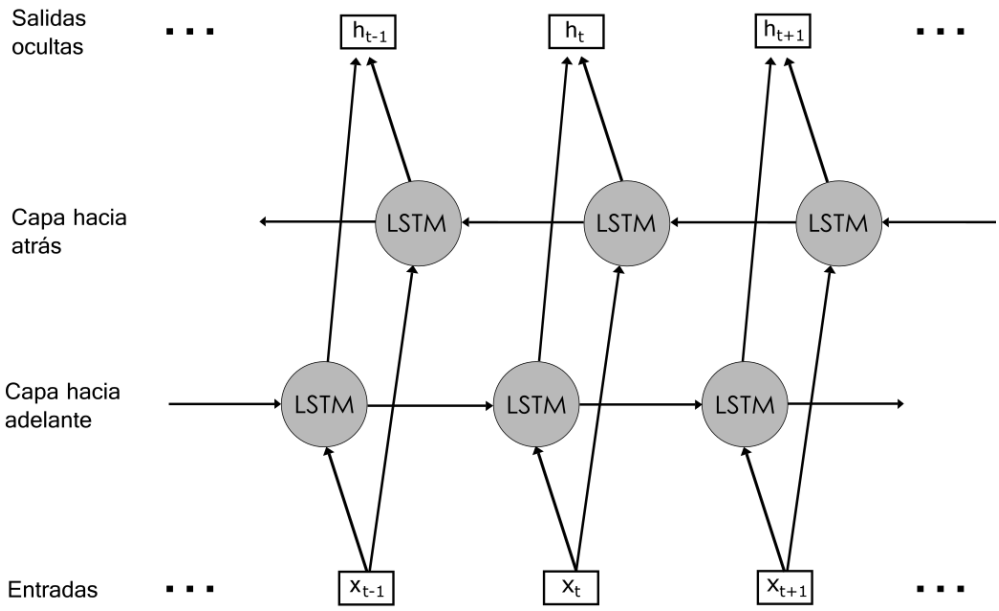


Figura 6.9: Composición de una Bi-LSTM. Los módulos de memoria en ambas capas reciben como entradas los datos secuenciales además del estado oculto anterior o posterior. Las salidas de cada par de módulos son concatenadas para producir las representaciones finales.

6.3.2. LSTM bidireccional

A pesar de que las LSTM son capaces de recordar información a largo y corto plazo, los estados ocultos de salida en cada paso temporal se basan en la información almacenada de estados previos, significando que solo se cuenta con entradas del pasado, sin considerar el flujo de información que proviene de pasos posteriores. En una tarea como la de NER, las palabras posteriores a una entidad nombrada pueden dar una pista importante acerca de su significado y categoría semántica, además de las previas, por lo que instalar una capa única de módulos LSTM no será suficiente. Por esta razón se utilizan dos capas de LSTM para procesar la información en ambos sentidos de la secuencia, formando el codificador de contexto de la red neuronal.

En un paso temporal concreto, una capa procesa las entradas recibidas a la izquierda o previas, y la otra procesa las representaciones de palabras a la derecha. A esta composición de las LSTM se le llama Bi-LSTM o LSTM bidireccional por referir a la información en ambos lados de cada término para precisar su contexto. En la Figura 6.9 se ilustra la estructura común de una Bi-LSTM, donde los vectores con las representaciones de palabras se introducen en las dos LSTM asociadas al paso correspondiente, y producen un vector contextual de la palabra, creado a partir de una operación de concatenación sobre los estados ocultos de cada módulo. Destacar que los vectores contextuales no son las salidas finales ya que han de pasar por el decodificador de etiquetas, por lo que se marcan como ocultos y ya pueden utilizarse para etiquetar la secuencia entrante.

6.4 Decodificador de etiquetas

El último bloque del modelo está formado por el decodificador de etiquetas. Como su propio nombre indica, produce una secuencia de etiquetas a partir de los vectores contextuales extraídos de la Bi-LSTM, que se corresponden a la secuencia de palabras de entrada. Existen dos arquitecturas muy comunes para obtener las etiquetas finales, una de ellas y la más directa es un perceptrón multicapa que extrae las probabilidades de

pertenencia a cada categoría, aplicando una función de activación *softmax* sobre los nodos de la capa de salida y la otra son los campos aleatorios condicionales o CRFs (*Conditional Random Fields*), siendo esta la opción elegida para extraer las etiquetas finales de cada palabra.

6.4.1. Campo aleatorio condicional

El campo aleatorio condicional (CRF) se introdujo en 2001 por Lafferty et al. [41] y es un modelo estadístico basado en un grafo no dirigido, cuyo objetivo es determinar una secuencia Y de variables desconocidas dado una secuencia de entrada X de observaciones mediante probabilidades condicionales $P(Y|X)$. Con la información contextual de cada entrada x_i ($1 < i < n$, donde n es la longitud de la secuencia), se trata de predecir cuál es la etiqueta y_i más conveniente. Para eso el modelo se tiene que entrenar con diferentes entradas para ajustar sus pesos y maximizar la distribución de probabilidad condicional. Si existe una aparición de un término en una secuencia con un contexto que habitualmente no es el que le rodea, se puede utilizar el conocimiento previo de otras instancias para predecir su etiqueta en base a sus probabilidades condicionales. El CRF se representa como un grafo $G = (V, E)$, con V siendo el conjunto de nodos formado por todas las observaciones de entrada y etiquetas semánticas, y E el conjunto de aristas que consideran las probabilidades de emisión de etiquetas. Los nodos están contenidos en dos conjuntos disjuntos A y B, con $A = \{x_1, x_2, ..., x_n\}$ y $B = \{y_1, y_2, ..., y_n\}$ que contienen cada elemento de las secuencias de entrada y de salida respectivamente. En la Figura 6.10 se ilustra la estructura del CRF.

A diferencia de los modelos ocultos de Markov (HMM), un CRF es un modelo discriminativo. Esto significa que modela las probabilidades condicionales en vez de las conjuntas $p(x, y)$ y las distribuciones de observaciones $p(x)$ para realizar predicciones. Además, un CRF no presenta las suposiciones de independencia de los HMM por lo que se considera el contexto de cada observación como información adicional y las dependencias no solo se calculan con respecto al estado previo sino con cualquier estado presente en la secuencia, incluso con el del final. También, los CRF suelen rendir mejor que los HMM a la hora de generar la secuencia final de estados ocultos.

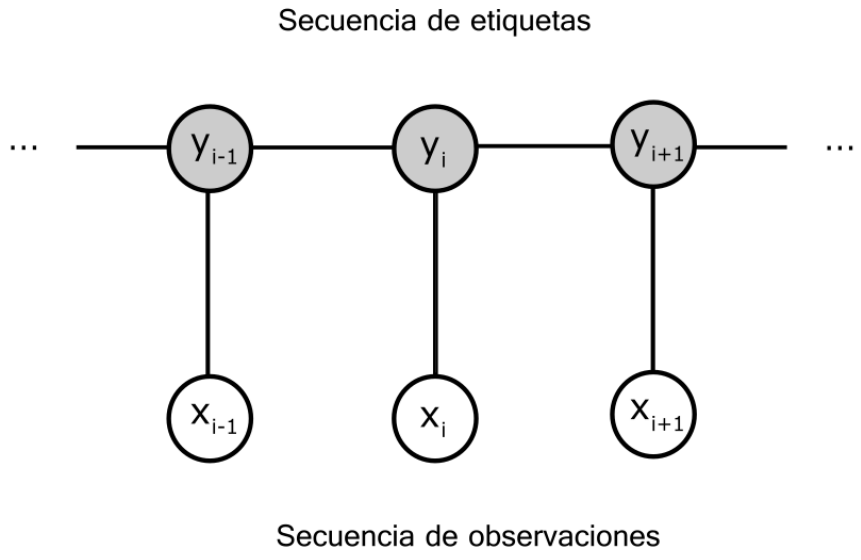


Figura 6.10: Modelo CRF que asocia a cada entrada x_i su correspondiente etiqueta y_i.

Los CRFs son muy utilizados en tareas como la de NER para etiquetar secuencias de palabras de una cierta longitud. El hecho de que la predicción para una palabra dependa de las predicciones de palabras vecinas hace que el CRF sea una herramienta muy valiosa para modelar probabilidades condicionales. Si no existiesen dependencias entre las palabras, la probabilidad de emitir su secuencia de etiquetas sería el producto de las probabilidades de emisión de cada estado individual. Entonces, la probabilidad de generar la secuencia de estados Y dado unas observaciones X de longitud T considera adicionalmente unas probabilidades de transición entre etiquetas y se denota por:

$$P(Y|X) = \prod_{t=1}^{T} P(y_t|x_t) \cdot P(y_t|y_{t-1}), \text{ donde:}$$

- $P(y_t|x_t)$ es la probabilidad de emisión del estado y dado la entrada x en el paso t,
- $P(y_t|y_{t-1})$ es la probabilidad de transición al estado y_t desde y_{t-1}.

Destacar que las entradas x_t que recibe la capa CRF son los vectores contextuales que salen de la LSTM bidireccional en cada paso.

6.4.2. Algoritmo de Viterbi

Para obtener la secuencia final predicha, se emplea el algoritmo de Viterbi [42] que determina el camino de estados con mayor probabilidad mediante programación dinámica. En cada paso t se calcula la probabilidad máxima de generar un prefijo de observaciones de longitud t atravesando una secuencia de estados que acaba en s. Esta operación es equivalente a seleccionar la mejor probabilidad asociada al camino que emite el prefijo de longitud $t-1$ acabando en el estado s' y multiplicarla por la transición a s y por la emisión de x_t en s. De todas los estados s' evaluados, se escoge aquel que otorga la máxima probabilidad:

$$V(s,t) = \max_{s'} \quad V(s', t-1) \cdot T(s', s) \cdot U(s, x_t), \text{ donde:}$$

- $V(s,t)$ es la probabilidad de la secuencia de estados más probable de longitud t acabando en el estado s,
- $T(s', s)$ es la probabilidad de transición desde el estado s' a s,
- $U(s, x_t)$ es la probabilidad de emitir la observación x_t en s.

Cuando $t = 1$, se considera la probabilidad inicial $I(s)$ de cada estado y la probabilidad de emisión de la primera observación:

$$V(s,1) = I(s) \cdot U(s, x_1)$$

Finalmente, la probabilidad de la secuencia completa se determina por la máxima probabilidad entre todos los caminos que producen la cadena de observaciones de longitud T acabados en cada estado s:

$$\hat{P}(Y|X) = \max_{s} \quad V(s,T)$$

6.5 Modelo Bi-LSTM-CRF

Una vez definidas todas las componentes de la red, su estructura completa se ilustra en la Figura 6.11. Se trata de un modelo Bi-LSTM-CRF que toma por un lado las secuencias de registros con sus palabras codificadas y por otro, los caracteres codificados de cada palabra, para extraer las representaciones distribuidas de longitudes específicas. Las representaciones de los caracteres son pasadas por la LSTM que se encarga de codificarlas para concatenarse con las representaciones distribuidas de sus palabras correspondientes y dar información sobre su composición morfológica. Los vectores resultantes son introducidos a la LSTM bidireccional, creando vectores contextuales. Por último, el CRF predice las etiquetas en cada paso temporal con el algoritmo de Viterbi.

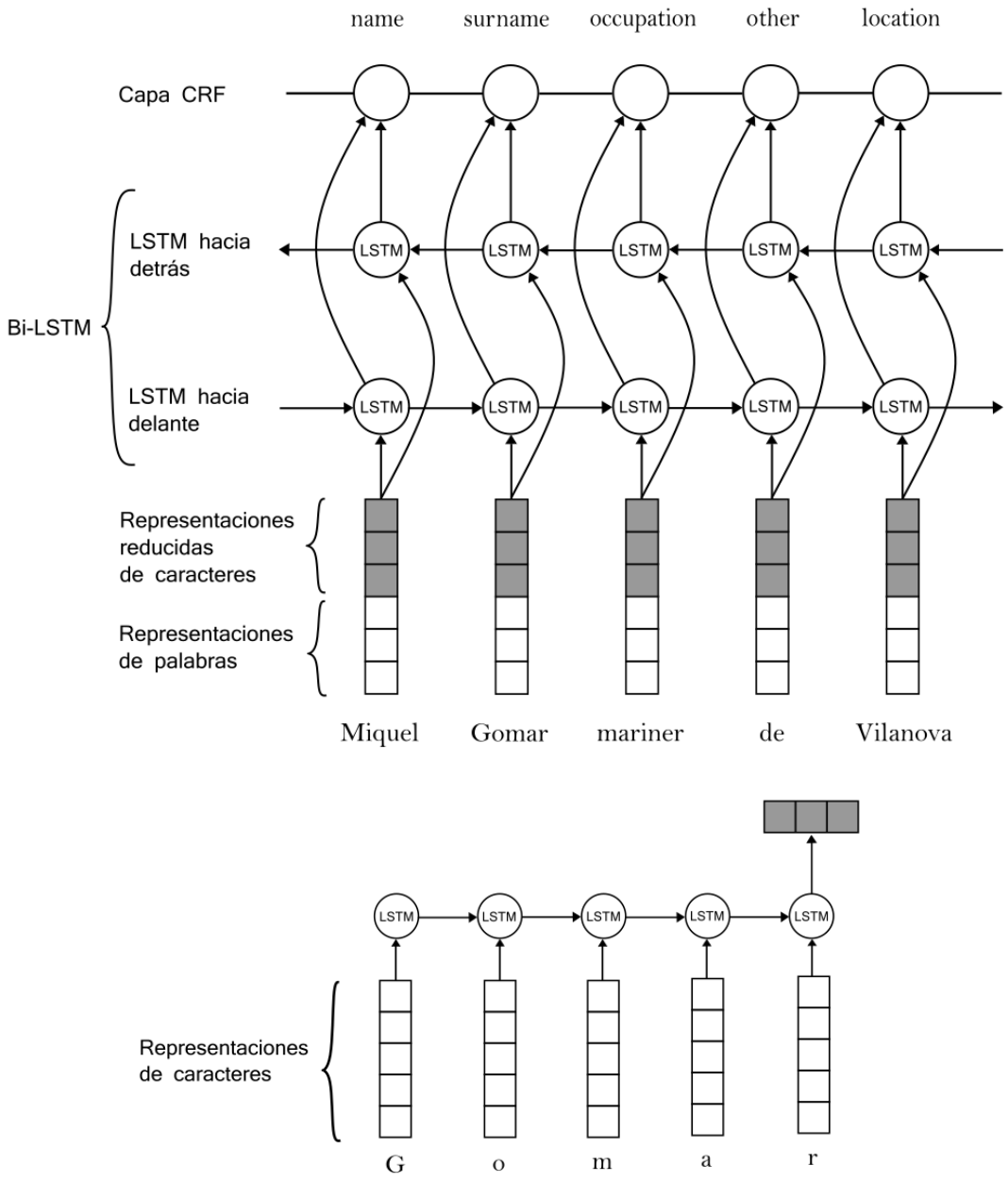


Figura 6.11: Modelo completo Bi-LSTM-CRF para predecir las etiquetas semánticas de una secuencia de palabras dada. En la parte inferior, la LSTM extrae las representaciones reducidas a nivel de carácter y son concatenadas con las incrustaciones o representaciones de palabras para extraer el contexto en la Bi-LSTM y finalmente decodificar las etiquetas en la capa CRF.

CAPÍTULO 7
Experimentación y resultados

Con el modelo para el reconocimiento de entidades nombradas creado, el siguiente paso es entrenarlo y evaluarlo con el conjunto de registros matrimoniales de la base de datos de Esposalles. Puesto que cada palabra consta de dos tipos etiquetas, la experimentación se separa por un lado en comprobar la actuación del modelo en reconocer y clasificar únicamente las categorías semánticas (nombres, apellidos, localizaciones, ocupaciones y estado civil) y por otro se añade un poco más de dificultad al tratar de reconocer a estas junto con la persona a la cual hacen referencia (por ejemplo, el nombre del marido, el apellido de la mujer, la ocupación del padre de la mujer, etc). Bajo este escenario, se parte del supuesto de que las transcripciones de palabras representan exactamente el contenido de los textos manuscritos originales y que por tanto no contienen fallos. Sin embargo, resulta de interés analizar el rendimiento de la red neuronal al recibir unas transcripciones que han sido generadas a partir de un proceso de reconocimiento de texto manuscrito, es decir aquellas transcripciones con mayor probabilidad a posteriori de producirse y que por tanto pueden contener errores de reconocimiento. Es por ello que se medirán las prestaciones de la red ante esta situación, para la cual no ha sido del todo entrenada.

7.1 Selección de hiperparámetros

Los hiperparámetros propios del modelo como característicos del proceso de entrenamiento constituyen en gran medida el éxito de clasificación del sistema. En comparación con los parámetros del modelo cuyos valores pueden ser aprendidos por los datos de entrada como los pesos de una red neuronal, los hiperparámetros son externos al modelo, significando que no se estiman con los datos sino que han de ser manualmente seleccionados para controlar y optimizar el proceso de aprendizaje. Una mala decisión en alguno de ellos puede condicionar a los resultados obtenidos independientemente de la complejidad de la arquitectura neuronal. Así pues, en esta sección se analiza el efecto que tienen en la evolución de aprendizaje, indicando la configuración final empleada.

7.1.1. Factor de aprendizaje y optimizador

Visto en el proceso de retropropagación del capítulo 5, durante el descenso por gradiente que obtiene aquel conjunto de parámetros de la red que minimiza el valor de la función de pérdida en el entrenamiento, el error producido por cada peso se escala por un factor de aprendizaje. El factor de aprendizaje posiblemente sea el hiperparámetro más importante de cualquier modelo de aprendizaje profundo. Está acotado entre 0 y 1, determina el tamaño de la modificación de pesos, donde un valor muy próximo a 0 indica una

convergencia lenta del error causada por el escaso ajuste de parámetros y por tanto se necesitarán muchas iteraciones o pasos de entrenamiento para llegar a un mínimo. Si por el contrario el factor de aprendizaje es muy alto, el error converge rápidamente y tiende a oscilar sobre el mínimo ya que la actualización de los pesos es notable, incluso puede aumentar el valor de la función de pérdida y desbordarse si los cambios son muy bruscos. No hay un método que especifique cual es el factor de aprendizaje óptimo, aunque existen múltiples técnicas como la exploración en rejilla o *grid search* que determinan el mejor valor para un rango dado. El problema de estos métodos es que requieren mucho tiempo de ajuste al entrenar cada vez el modelo con una configuración distinta de hiperparámetros y la mejora producida puede no ser del todo significativa. En muchas ocasiones el valor por defecto suele ser de 0.01 o 0.1 y es definido junto al optimizador, que es un algoritmo que define una estrategia de actualización de parámetros para reducir el error en el modelo. Existen dos grupos principales de optimizadores, los basados en el descenso por gradiente y los adaptativos, que modifican o adaptan el factor de aprendizaje en cada iteración según el cambio producido en los parámetros. Los optimizadores adaptativos tienen ventaja puesto que en conjuntos de datos con mucha variabilidad de atributos, puede que se tenga que modificar en mayor magnitud el valor de algunos parámetros en concreto y considerar un factor de aprendizaje dinámico es una buena solución para cambiar el ritmo de entrenamiento.

Poniendo a prueba distintos optimizadores y factores de aprendizaje, se ha elegido el optimizador adaptativo de Adam con un factor de aprendizaje (α) de 0.01. Generalmente, Adam consigue muy buenos resultados en el aprendizaje neuronal profundo por su capacidad de rápida convergencia hacia el mínimo y de tratar gradientes más dispersos provocados por las diferencias en los datos. Sin embargo, existen otros como el RMSProp y AdaDelta que rinden de forma parecida en algunas características pero Adam se considera en muchos de los casos el más efectivo. Además, consta de otros dos parámetros que modelan la tasa de reducción del factor de aprendizaje (β_1 y β_2) y de un épsilon (ϵ) para evitar divisiones por cero en los cálculos. Estos parámetros adicionales se han mantenido en sus valores por defecto.

7.1.2. Tamaño del lote

El lote o *batch* recoge una cantidad de muestras para procesar antes de que los parámetros del modelo sean actualizados. En vez de ir muestra por muestra para realizar los pasos hacia delante y hacia detrás en la red que puede ser muy costoso en grandes colecciones de datos, se trabajan con grupos de tamaño predefinido de muestras para acelerar estos procesos y el error de predicción por lote se calcula promediando o acumulando los errores individuales de cada muestra para luego propagar el error y actualizar pesos. Es muy frecuente que el último lote no acabe rellenándose al completo debido a que el número total de muestras no es divisible por el tamaño del lote.

La elección del tamaño de estos grupos es un hiperparámetro importante a considerar en cualquier modelo de aprendizaje profundo. Un tamaño excesivamente grande puede causar pérdida de generalización en el modelo al ajustar sus parámetros en base a un error muy resumido y requiere mayor tiempo en completar cada paso pero obtienen mejores estimaciones de gradientes. Por otro lado, utilizar tamaños muy pequeños mejora la velocidad de convergencia pero no está garantizado alcanzarla, necesitándose un factor de aprendizaje más reducido por presentarse gradientes muy dispersos. Debido a cuestiones de almacenamiento en memoria, los tamaños de los lotes suelen ser potencias de 2, siendo los más utilizados tamaños pequeños de 16, 32 y 64 que han presentado los mejores resultados en muchas tareas de reconocimiento [43, 44] aunque depende en gran

parte de la cantidad total de muestras y del modelo. En este caso, se ha optado por un tamaño de lote de 32 muestras.

7.1.3. Número de ciclos

En contraste con el tamaño del lote, el número de ciclos o *epochs* es un hiperparámetro que determina las veces que el modelo recibe a todo el conjunto de muestras de entrenamiento para actualizar sus parámetros. Durante un ciclo de entrenamiento, se pasan uno a uno todos los lotes de muestras por la red neuronal, hacia adelante y hacia detrás. En términos de programación, esto se puede pensar como un bucle anidado, donde en cada ciclo se recorren los lotes que forman el conjunto de datos y termina cuando se procesan los lotes en el último ciclo. No existe una formulación para establecer una cantidad de ciclos concreta aunque es posible que se produzca sobreajuste en el modelo al entrenarse con demasiados ciclos. El número de ciclos utilizados para entrenar al modelo neuronal es 30, punto en el que la mejora del error ya no es tan notable.

7.1.4. Longitud de las secuencias

Empezando con los parámetros específicos del modelo Bi-LSTM-CRF, la longitud de las secuencias es un hiperparámetro a determinar, que indica la cantidad de palabras o *tokens* a considerar para extraer sus etiquetas y por tanto es también la dimensión de entrada del modelo.

Como se ha visto anteriormente en el análisis exploratorio de los registros matrimoniales, las longitudes varían desde 13 hasta 77 palabras siguiendo una distribución normal. Debido a esta variabilidad, la longitud ideal es aquella que recoja la máxima cantidad de información posible de todos los registros sin añadir un excesivo relleno para aquellos que no la alcancen pues implica entrenar el modelo con posiciones adicionales que no van a tenerse en cuenta para generar sus etiquetas. Si la longitud se fija a 20, quedarían muchas secuencias incompletas ya que los registros tienen de media 32 palabras, significando que se omitiría mucha información relevante que la red puede aprovechar. Si por el contrario se fija a 77 que es la longitud máxima de la colección, solo se completaría la secuencia para un registro y al ser un caso extremo, para el resto de entradas se tendría que añadir bastante relleno como consecuencia, en ocasiones más que sus palabras totales. Considerando esto, la longitud de las secuencias se ha establecido a 50, siendo muy pocas las veces en las que se supera (de hecho solo 7 registros de los 968 iniciales tienen una longitud mayor). Bajar progresivamente de dicha cantidad conlleva desaprovechar términos que pueden ser útiles para determinar el contexto de ciertas palabras.

Para seleccionar la longitud de las secuencias de caracteres necesarias para extraer información morfosintáctica de las palabras, se ha seguido la misma reflexión, seleccionando un tamaño de 10 al ver mediante la Tabla 4.3 que la media de caracteres por palabra son unos 4 o 5 y un valor máximo de 14. Raramente se pasa de las 9 letras.

7.1.5. Dimensión de los vectores de representaciones

El siguiente hiperparámetro a determinar es el tamaño o dimensión de los vectores de representación de las palabras y caracteres, definido en la capa de incrustaciones. Estas representaciones numéricas agrupan elementos por su significado y el número de componentes a considerar depende sobre todo del tamaño del vocabulario pues cuanto más grande sea, más diversidad de términos o símbolos existen, implicando más propiedades que se tienen que capturar. El problema de los espacios de altas dimensiones es que se

pierde la correlación entre observaciones al considerar componentes que pueden introducir ruido al modelo e incrementa la cantidad de parámetros a entrenar, mientras que con espacios muy pequeños se comprime mucho la información léxica. Habitualmente, este hiperparámetro es seleccionado por prueba y error, aunque un punto de arranque común viene definido por la raiz a la cuarta del tamaño del vocabulario [45].

Debido al vocabulario reducido de los registros matrimoniales (2430 palabras distintas), las representaciones de palabras se componen de 20 dimensiones por ser suficientes a modelizar. En caso de los caracteres, al constar de un tamaño de vocabulario menor, sus dimensiones se han bajado a 10.

7.1.6. Unidades de salida en la LSTM

Cada capa LSTM tiene que devolver un vector de salida de un tamaño específico para utilizarse en las capas posteriores. El modelo diseñado tiene dos principales bloques LSTM para recordar información a corto y largo plazo: Uno que procesa las representaciones de caracteres y otro que es el codificador de contexto formado LSTM bidireccional.

El objetivo del primer bloque es producir representaciones más compactas de caracteres que tienen que ser fusionadas con las propias representaciones de palabras a las cuales se refieren. Como ya se tienen representaciones de palabras de dimensión 20, el número de salidas en este bloque se sitúa también a 20 y así obtener vectores resultantes de tamaño 40 que consideren por partes iguales características del significado y de la composición interna de la palabra.

La LSTM bidireccional consta de dos capas LSTM para almacenar datos relevantes en sentidos opuestos de la secuencia y las salidas de cada una también se tienen que combinar para generar los vectores contextuales según la palabra. En este punto, es interesante que el tamaño de dichas salidas sea lo suficientemente grande como para registrar todas los atributos que relacionan a palabras entre sí, pero a la vez introducir las unidades justas para limitar la cantidad de operaciones involucradas en calcular los estados ocultos y actualizar el estado celular de las LSTM. Por ello, se han precisado 50 unidades de salida en cada capa.

7.1.7. Probabilidad de abandono

En el aprendizaje automático, el abandono o *dropout* es una técnica para reducir el sobreajuste del modelo al hacer que neuronas o nodos arbitrarios en una cierta capa de la red se desactiven o ignoren. La idea detrás de esto es que existe una dependencia en las salidas de cada nodo que hace que los patrones detectados no puedan ser correctamente generalizados a otras muestras que no se han utilizado en el entrenamiento. Si algunos nodos no se tienen en cuenta durante los procesos hacia adelante y de retropropagación en la red, se ignoran todas sus conexiones que salen o llegan hasta ellos y por tanto son menos los parámetros a actualizar. Con esto se consigue romper las posibles dependencias entre pesos y hacer que los nodos restantes carguen también los valores de salida de los nodos desactivados. El hiperparámetro es entonces la probabilidad de omisión de nodos en una capa específica de la red.

Con respecto a redes recurrentes, se omiten el estado celular y el estado oculto que será pasado a la siguiente posición temporal con cierta probabilidad. Concretamente en el modelo propuesto, esto tiene lugar en las distintas puertas que componen las LSTM, donde las operaciones matriciales y de activación propias de cada una son ignoradas. En el trabajo de investigación de Ghal y Ghahramani [46], se presentó otra alternativa para aplicar la técnica de *dropout* en redes recurrentes que consiste en enmascarar o igno-

rar todas las unidades de pasos temporales aleatorios en la secuencia, en vez de estados ocultos y celulares arbitrarios. La probabilidad de omisión es de 0.3 en los dos bloques LSTM del modelo, significando que aproximadamente un tercio de las conexiones recurrentes no se consideran para determinar los estados ocultos en cada paso. Un proceso similar pero sobre un espacio de dos dimensiones (*spatial dropout*) se aplica para los vectores combinados con las representaciones de palabras. La probabilidad es también de 0.3 con la intención de eliminar dependencias entre propiedades de cada término.

7.2 Estrategia de entrenamiento y validación del modelo

Los registros originales se tienen que dividir en conjuntos de entrenamiento y validación para ajustar los parámetros del modelo y a la vez comprobar su actuación sobre datos que nunca antes ha visto. Esto es necesario para monitorizar el posible sobreajuste producido por la dependencia de los datos sobre los cuales ha sido entrenado. No hay que olvidar que el propósito final de cualquier modelo de aprendizaje automático es enfrentarse ante nuevas situaciones desconocidas y utilizar lo aprendido para obtener el mejor rendimiento posible. Así pues, un conjunto de validación simula estas situaciones y los resultados obtenidos sobre las muestras que lo componen son los que verdaderamente indican el potencial en la tarea.

El problema con estas particiones es que pueden estar sesgadas, significando que el comportamiento del modelo está condicionado por los datos que componen cada conjunto. Es decir, si solo se evalúa el modelo con un 20 % del total de registros y el 80 % restante se utiliza para entrenar, los resultados finales dependen solo de una parte de la colección total, que no podrían reflejar la verdadera capacidad predictiva de la red neuronal. Como solución, se emplea el método de la validación cruzada [47] que consiste en dividir los datos de entrada aleatoriamente en K particiones de un tamaño fijo y se va iterando de forma que cada vez se ajusta un nuevo modelo sobre $K-1$ particiones y con la restante se evalúa su comportamiento. Esto garantiza que todas las muestras sean puestas a prueba. Los resultados de cada modelo sobre cada partición distinta son almacenados para extraer un promedio, que elimina los sesgos de selección. Por esta razón, dicha técnica ha sido usada, donde se han definido 5 particiones y con ellas valorar el rendimiento del modelo. Dado que se busca conseguir unos resultados de referencia a ilustrar, las predicciones y etiquetas reales de cada partición examinada son reunidas para aplicar las métricas de evaluación conjuntamente sobre todos los registros.

Por otro lado, para generar las predicciones de las mejores hipótesis de los registros, el modelo ha sido entrenado y validado con las transcripciones sin fallos etiquetadas, con particiones de entrenamiento, validación y test. En la Tabla 7.1 se ilustran los tamaños de cada partición. Así pues, la validación cruzada no se ha utilizado en este caso, sin tener efecto en esta segunda experimentación. Debido a esta partición de muestras, se ha comprobado únicamente el rendimiento de reconocimiento sobre las hipótesis cuyas transcripciones sin errores están contenidas en el conjunto de test. Si por el contrario se evalúa modelo con todas las hipótesis, gran parte de ellas habrán sido empleadas para ajustar parámetros y por tanto los resultados finales obtenidos no serán fiables.

Tabla 7.1: Distribución de registros en las diferentes particiones para entrenar y evaluar el modelo de reconocimiento de entidades nombradas sobre las mejores hipótesis.

Partición	**Registros totales**
Entrenamiento	580
Validación	194
Test	194

7.3 Función de pérdida

La función de pérdida que utiliza el modelo para actualizar sus parámetros es la pérdida CRF propia del campo aleatorio condicional en la capa de decodificación de etiquetas. El objetivo es que la probabilidad estimada de la secuencia real de etiquetas se aproxime lo máximo posible a uno para minimizar la pérdida o el error. En el cómputo de esta función objetivo se calcula el logaritmo negativo de verosimilitud de la secuencia de salida esperada considerando todas las combinaciones de caminos posibles de secuencias. Como se indica por Arnaud Stiegler [48], esto es:

$$\text{loss} = -\log\Big(\frac{\exp(\sum_{t=1}^{n} U(y_t|x_t) + \sum_{t=1}^{n} T(y_t|y_{t-1}))}{\sum_{y'} \exp(\sum_{t=1}^{n} U(y'_t|x_t) + \sum_{t=1}^{n} T(y'_t|y'_{t-1}))}\Big)$$

El numerador del logaritmo hace referencia a la probabilidad de obtener la secuencia esperada a partir de las probabilidades de emisión U y de transición T entre estados. Las probabilidades de todas las combinaciones posibles de estados se acumulan en el denominador de forma exponencial con la longitud de la secuencia. Por tanto, la pérdida será menor cuanto más destacable sea la probabilidad de la secuencia esperada con respecto a las demás.

7.4 Métricas de evaluación

La tasa de acierto es la métrica más popular y referente en cualquier clasificador pues indica la proporción de muestras u observaciones que han sido correctamente asignadas a sus respectivas categorías. En NER también es importante mostrar el porcentaje de etiquetas de palabras reconocidas correctamente por el modelo. Sin embargo, cuando una clase es mayoritaria como sucede en los registros matrimoniales con la categoría "other" o "none-other" para palabras que no representan ninguna categoría de interés o persona, la tasa de acierto ya no es la mejor métrica a considerar porque no refleja la calidad de reconocimiento en categorías más relevantes como en nombres, trabajos o estados civiles. Por ello también se han tomado la precisión, cobertura o *recall* y el valor F1 o *F1-score* específicos de cada tipo de etiqueta.

La precisión de una etiqueta específica las veces que ha sido reconocida correctamente sobre el total de predicciones hechas para esa misma etiqueta en todo el conjunto. Por ejemplo, si se predicen 500 apellidos y 450 de esas predicciones corresponden a apellidos reales, su precisión es de 0.9. Por otra parte, la cobertura indica cuantas etiquetas de un tipo en concreto han sido reconocidas con acierto sobre el total de etiquetas de ese tipo. De nuevo, si hay 600 apellidos en el conjunto de validación y 450 son los identificados sin error, su cobertura es de 0.75. Finalmente, la medida F1 es una función que toma balance de la precisión y cobertura. Concretamente se calcula como:

$$\text{Valor F1} = \frac{\text{Precisión} \times \text{Cobertura}}{\text{Precisión} + \text{Cobertura}}$$

A partir de las métricas relativas a cada clase, se obtiene el promedio teniendo en cuenta todos los tipos de etiquetas de la colección. Los promedios de precisión, cobertura y del valor F1 junto a la tasa de acierto global evidencian el rendimiento generalizado del modelo. Para obtener estas métricas de evaluación, es necesario que la colección de registros matrimoniales tenga etiquetas asociadas a sus palabras, como ocurre con las transcripciones que no contienen fallos. Sin embargo, cuando se disponen de las transcripciones resultantes de un experimento de reconocimiento, pueden existir variaciones

en las longitudes originales causadas por inserciones y borrados algunas palabras. Como se menciona en la siguiente sección, la distancia de edición a nivel de palabra es empleada para medir la similitud entre las etiquetas de salida del modelo y las de referencia.

7.5 Resultados

En esta sección se muestran los distintos resultados de reconocimiento para las transcripciones reales y mejores hipótesis producidas. En ambos casos, se evalúa la capacidad de predicción de categorías semánticas y luego se suman las personas asociadas.

Como se puede ver en la Tabla 7.2, los resultados reunidos de cada partición en la validación cruzada indican buenas prestaciones de reconocimiento en las transcripciones auténticas. Cuando se tienen solo las categorías semánticas, la precisión, cobertura y valor F1 promedio son mejores debido a que estas etiquetas están más balanceadas y el modelo tiene suficiente información de cada una. Esto no pasa cuando se consideran también las etiquetas de personas pues como se analizó previamente, existen combinaciones que raramente se observan en la colección de registros y son más difíciles de identificar, afectando a sus métricas individuales y en especial a la cobertura. Los resultados promedios de todas las clases son alterados en consecuencia, bajando el valor F1 de 0.97 conseguido con las categorías semánticas a 0.83 por haber más variedad de etiquetas a predecir que no aparecen por igual. La tasa de acierto en las dos tareas es muy alta y parecida, indicando que la red neuronal es capaz de utilizar el significado y analizar el contexto de las palabras para asignar sus etiquetas correctamente en la mayoría de casos. En muchas de las veces donde no se ha devuelto bien las categorías predichas, el término "de" catalogado como localización u ocupación cuando une a dos términos que hacen referencia a trabajos y lugares de residencia, se identifica incorrectamente como otra categoría. Esto ocurre porque es un término muy abundante en los registros matrimoniales y se emplea en diferentes contextos, aunque prácticamente todos son conocidos por la red neuronal.

Tabla 7.2: Resultados de evaluación globales en el reconocimiento de categorías semánticas y combinación de categorías semánticas con personas en las transcripciones sin fallos.

Tipo de etiqueta	Precisión	Cobertura	Valor F1	Tasa de error
Categoría semántica	0.9759	0.9761	0.9759	2.05 %
Categoría semántica + persona	0.9280	0.8261	0.8333	2.34 %

Los resultados individuales de evaluación de cada categoría junto a su frecuencia se ilustran en las tablas 7.3 y 7.4. En caso de las categorías semánticas, las métricas resultantes son muy similares y señalan que se han distinguido bien todos los grupos de interés aunque es cierto que las localizaciones, ocupaciones y apellidos han costado un poco más de reconocer al tener una cobertura ligeramente menor que el resto. También se confunden otras categorías con localizaciones, apellidos e incluso estados civiles, prueba de ello las precisiones más reducidas en cada una.

Con las etiquetas a predecir en forma de categoría semántica y persona, se destaca la falta de acierto en las combinaciones poco comunes como son la localización y apellidos de las madres, y en menor medida en el apellido de la mujer, cuyas métricas individuales contribuyen negativamente a los promedios de la Tabla 7.2. Quitando de esas clases, se aprecia un buen rendimiento de detección en las demás al darse valores F1 superiores a 0.9. Quizás el error es más notable en el lugar de residencia o relacionado con la mujer para la cantidad de términos que poseen esta etiqueta, a pesar de que la misma

para el marido presenta mejores métricas. Además, hay que resaltar que los nombres de cualquier persona son reconocidos en general con más acierto que los apellidos, que en ocasiones pueden ser más complejos de identificar. Para aquellos registros que nombran a un exmarido (categorizado como "other_person") de la mujer, su nombre y apellido se detectan con bastante precisión.

Tabla 7.3: Resultados de evaluación específicos de cada categoría semántica en las transcripciones sin fallos.

Categoría	Precisión	Cobertura	Valor F1	Frecuencia
location	0.9580	0.9676	0.9628	4507
name	0.9905	0.9856	0.9881	4992
occupation	0.9877	0.9584	0.9728	3008
other	0.9842	0.9868	0.9855	15132
state	0.9709	0.9939	0.9823	1142
surname	0.9639	0.9642	0.9641	2657

Tabla 7.4: Resultados de evaluación específicos de cada categoría semántica y persona asociada en las transcripciones sin fallos.

Categoría combinada	Precisión	Cobertura	Valor F1	Frecuencia
husband-location	0.9677	0.9730	0.9704	2372
husband-name	0.9882	0.9843	0.9863	1021
husband-occupation	0.9837	0.9535	0.9684	1205
husband-state	0.9747	0.9147	0.9438	211
husband-surname	0.9703	0.9761	0.9732	1004
husbands_father-location	0.9146	0.9574	0.9356	470
husbands_father-name	0.9837	0.9789	0.9813	617
husbands_father-occupation	0.9856	0.9609	0.9731	640
husbands_father-surname	0.9429	0.9639	0.9533	582
husbands_mother-location	1.0000	0.0000	0.0000	4
husbands_mother-name	0.9812	0.9845	0.9828	582
husbands_mother-surname	0.2222	0.1250	0.1600	16
none-other	0.9866	0.9857	0.9861	15132
other_person-location	1.0000	0.0000	0.0000	1
other_person-name	0.9571	0.9738	0.9654	229
other_person-surname	0.9339	0.9617	0.9476	235
wife-location	0.8729	0.9388	0.9046	490
wife-name	0.9923	0.9866	0.9895	1046
wife-occupation	0.9847	0.9556	0.9699	270
wife-state	0.9486	0.9914	0.9695	931
wife-surname	0.7857	0.6471	0.7097	34
wifes_father-location	0.9495	0.9641	0.9567	1170
wifes_father-name	0.9868	0.9829	0.9848	760
wifes_father-occupation	0.9848	0.9418	0.9628	893
wifes_father-surname	0.9665	0.9766	0.9715	769
wifes_mother-name	0.9905	0.9905	0.9905	737
wifes_mother-surname	0.8000	0.2353	0.3636	17

Mediante estos resultados, se puede decir que el modelo neuronal utiliza las incrustaciones o representaciones aprendidas para determinar el significado de las palabras y

con el codificador de contexto se aprovecha el conocimiento de palabras vecinas en la secuencia que resulta de gran utilidad para esta tarea en concreto puesto que los registros matrimoniales siguen un patrón común de narración. Por ejemplo, para las categorías semánticas, saber que detrás de un nombre casi siempre viene un apellido, y que posiblemente se acompañen de un trabajo y ubicación en relación a una persona, es una información muy valiosa para el modelo. Si a esto se le suma un decodificador de etiquetas que modela probabilidades condicionales para averiguar la secuencia de salida correcta a partir de unas observaciones como un campo aleatorio condicional, se consigue un modelo completo para el reconocimiento de entidades nombradas. Finalmente, con una buena elección de los hiperparámetros de entrenamiento, los resultados pueden ser muy prometedores.

Antes de pasar a los resultados obtenidos para las transcripciones generadas por el proceso de reconocimiento de texto manuscrito, hay que mencionar que estas han sido obtenidas a partir del sistema presentado en [49]. La tasa de error de palabras o *Word Error Rate* (WER), definido como el número mínimo de inserciones, borrados y sustituciones de palabras para convertir las transcripciones de salida del reconocedor en las propias de referencia, dividido por el total de palabras en esta, es de un 10 %. Como son transcripciones que no ha visto el modelo, seguramente existan palabras y símbolos fuera de vocabulario que no tienen vectores de representación propios y sus entradas no serán reconocidas. Por ello, se ha reservado un índice especifico para toda la información desconocida, tanto de palabras como de caracteres y de esta forma asociar un vector de características particular. También hay que recordar que el modelo empleado para predecir las etiquetas de las mejores hipótesis conserva la misma configuración de hiperparámetros, incluidos el factor de aprendizaje, optimizador, número de ciclos de entrenamiento y tamaño del lote.

El principal problema con estas transcripciones es que no se tienen sus etiquetas exactas por cada palabra para utilizarse de referencia a la hora de evaluar las predicciones. En cambio, se disponen de las etiquetas de los registros originales sobre los que se basan pero estas pueden no tener una correspondencia uno a uno. Durante la generación de la nueva transcripción por el reconocedor de texto manuscrito, términos y especialmente símbolos adicionales como puntos y comas que no están presentes en la transcripción original se producen de forma errónea, significando que incrementa la longitud de la secuencia codificada. Como se observa en la Figura 7.1, es frecuente que el grafo interprete espacios en blanco inexistentes, causando la separación de una misma palabra en dos partes. En menos ocasiones, algunas palabras de pocas letras en la transcripción real no se generan por completo. Todo esto hace que la posición original de algunos términos en la secuencia de entrada al modelo se vea modificada y por tanto la posición de sus etiquetas.

	Mejor hipótesis	Registro original	Etiqueta
1)	Dit	Dit	none-other
2)	dia	dia	none-other
3)	rebere$	rebere	none-other
4)	de	de	none-other
5)	Pau	Pau	husband-name
6)	Riba	Ribafort	husband-surname
7)	font	sabater	husband-occupation
8)	sabater	de	none-other
9)	de	Palau	husband-location
10)	Palau	solitar	husband-location
11)	solitar	viudo	husband-state
12)	viudo	ab	none-other
13)	ab	Catherina	wife-name
14)	Catherina	viuda	wife-state
15)	viuda	de	none-other
16)	de	Agusti	other_person-name
17)	Agusti	Alsina	other_person-surname
18)	Alsina	pages	wife-occupation
19)	pages	de	none-other
20)	de	llissa	wife-location
21)	llissa	demunt	wife-location
22)	demunt		
23)	-		

Figura 7.1: Comparación a nivel de palabra de un registro original y su mejor hipótesis. La separación indeseada de una palabra en dos partes cambia de posición las palabras siguientes y las etiquetas pierden la referencia.

Por lo comentado anteriormente, se ha calculado la distancia de edición o de Levenshtein [50] entre las secuencias de etiquetas predichas y reales. Una vez se tienen las predicciones, los resultados se ilustran en la Tabla 7.5, donde se aprecia un bajo error por parte del modelo neuronal considerando que las transcripciones tienen de media unas 32 palabras. Al añadir las etiquetas de personas, las predicciones son ligeramente peores.

A pesar del ruido introducido en el vocabulario como signos de puntuación y algunos caracteres especiales pegados a términos existentes de las transcripciones auténticas, el modelo reconoce bastante bien las entidades nombradas. Esto se debe en gran parte a que ha aprendido del contexto de los registros auténticos en el entrenamiento, detectando dependencias entre palabras, y ha utilizado el significado de las palabras no solo a través de las incrustaciones propias de cada una sino también con las de caracteres para identificar similitudes dentro de un espacio multidimensional de propiedades. Puesto que son muchos los términos nuevos creados con estas hipótesis, las representaciones distribuidas a nivel de palabra no han podido ser aprovechadas en estos casos pero sí a nivel de carácter. Hay una cantidad considerable de palabras que son desconocidas y ante ello el modelo puede utilizar la información de la composición de caracteres para revelar parte de su significado dado la LSTM que memoriza el orden de aparición de símbolos y extrae las características reducidas.

Tabla 7.5: Resultados de evaluación sobre las predicciones de las mejores hipótesis tomando como referencia la distancia de edición promedia con respecto a las etiquetas originales.

Tipo de etiqueta	**Distancia de edición**
Categoría semántica	4.1
Categoría semántica + persona	4.8

Unos resultados recientes publicados por Cheikh et al. [51] sobre las transcripciones producidas por un reconocedor de texto manuscrito en la misma colección de registros matrimoniales, indican que su sistema de etiquetado basado en transformadores es capaz de reconocer con mucho acierto las palabras de las imágenes con texto manuscrito. En vez de calcular la distancia de edición entre las etiquetas predichas y las reales como métrica de evaluación, se ha determinado esta distancia (en la competición de IEHRR [4] se le conoce como CER o la tasa de error de caracteres) para aquellas palabras en la transcripción generada cuyas etiquetas son acertadas. En la tarea de reconocimiento de categorías semánticas, el sistema propuesto consiguió un acierto promedio (1 - CER) de 96.25, y para el reconocimiento de categorías semánticas junto a personas un acierto de 95.54. Puesto que las transcripciones resultantes del proceso de reconocimiento han sido dadas para este trabajo y no se ha diseñado un sistema para reconocer texto manuscrito, no se ha comprobado la calidad de la transcripción ya que el enfoque principal es determinar las entidades nombradas en ellas con el modelo neuronal creado y ver lo que se pierde con respecto a las etiquetas base. Es por ello que los resultados no son comparables con los propios de los autores.

Como conclusión de los resultados obtenidos en ambas experimentaciones, cada capa del modelo neuronal contribuye al éxito de reconocimiento de entidades nombradas en los textos matrimoniales mediante el procesamiento de datos secuenciales para extraer un vector de etiquetas final. Con esta arquitectura de muchos a muchos característica de las redes neuronales recurrentes, se ha conseguido un buen rendimiento de clasificación de entidades en las transcripciones originales de los registros para identificar categorías semánticas y personas. Las clases que no son tan frecuentes no se detectan del todo bien por proporcionar muy poca información a la red e influyen negativamente en los promedios de precisión, cobertura y valor F1. Aun así, resultados individuales son prometedores para las etiquetas con un número considerable de muestras.

En relación a las mejores hipótesis generadas por el sistema de reconocimiento de texto manuscrito, es cierto que se tienen las etiquetas de las transcripciones auténticas, pero en muchas ocasiones estas no encajan con el nuevo contenido creado pues existen inconsistencias en las posiciones por palabras mal separadas y símbolos adicionales que desplazan el texto de lugar. Por tanto, la distancia media de edición ha sido evaluada para medir la similitud en las etiquetas predichas y etiquetas reales, indicando que se necesitan muy pocas operaciones en promedio para que estas sean iguales. Los resultados son levemente peores que los obtenidos en las transcripciones sin fallos, que era de esperar. Al haber inserciones y borrados de palabras causadas por fallos en el proceso de reconocimiento de texto, la distancia de edición sube aun siendo perfectas las predicciones. Las salidas generadas se ajustan mucho a cada palabra como resultado de aprender unas representaciones de características y del contexto a partir de los registros originales. También se aprovecha la composición morfológica del vocabulario desconocido y las probabilidades de emisión y transición entre estados aprendidas en el campo aleatorio condicional a la hora de decodificar las etiquetas.

CAPÍTULO 8
Conclusiones

Con los resultados obtenidos y analizados, el objetivo principal de este trabajo consistiendo en el diseño de un modelo neuronal Bi-LSTM-CRF capaz de identificar las entidades nombradas de interés en los registros matrimoniales manuscritos de la colección de libros de Esposalles, se ha cumplido satisfactoriamente. Al disponer de las transcripciones etiquetadas, la red neuronal ha conseguido clasificar con mucho acierto tanto las categorías semánticas por sí solas como estas junto a las personas en cuestión de los registros auténticos, pudiendo confirmar que las palabras recibidas son etiquetadas de acuerdo a su contexto en la oración. Para mejorar la detección de aquellas etiquetas que no son tan frecuentes en los registros, se podrían recoger más transcripciones diferentes con palabras de la misma categoría y de esta forma balancear las clases. Mediante una composición de múltiples capas que procesan datos secuenciales con cada una cumpliendo con una tarea distinta, se ha conseguido crear un sistema eficiente de etiquetado gramatical para reconocer entidades en documentos o licencias antiguas de matrimonio. Las dependencias entre palabras a corto y largo plazo en la secuencia de entrada son modeladas con éxito por la LSTM bidireccional y las representaciones distribuidas de características aprendidas contribuyen en la distinción entre palabras según un conjunto de propiedades numéricas. Además, dotar de un modelo probabilístico que no consta de suposiciones de independencia estrictas en los datos de entrada como es un CRF, es una ventaja a la hora de decodificar las etiquetas pues implica asignar probabilidades condicionales de forma que finalmente se emita la secuencia de mayor verosimilitud dados unos vectores contextuales que contienen información de palabras vecinas. Por parte de las mejores hipótesis de los registros originales, se ha comprobado que existe una ligera caída de rendimiento en función de las distancias de edición entre etiquetas, necesitándose unos pocos cambios en las secuencias de salida. Sin embargo, la calidad de reconocimiento sigue siendo elevada porque las predicciones generadas se asemejan bastante a las etiquetas reales de las transcripciones que no tienen fallos.

8.1 Reproducibilidad

El modelo creado para reconocer entidades se encuentra en mi repositorio de GitHub [52] y puede ser adaptado a otros textos etiquetados de forma distinta y con idiomas distintos. Escenarios alternativos donde se le puede sacar partido incluyen en textos médicos y jurídicos para detectar personas o material de interés de forma automatizada, en motores de búsqueda para la indexación de documentos según las entidades en consultas de los usuarios, en la extracción de datos de carácter financiero que son costosos de determinar en grandes colecciones de documentos no estructurados y en la selección de

candidatos para un puesto de trabajo en base a la detección de entidades relevantes en los currículos como empresas en las que se han trabajado, universidades y conocimientos.

8.2 Relación del trabajo desarrollado con los estudios cursados

A lo largo de la carrera, son muchas las tecnologías que se han cubierto y conocimientos adquiridos en diferentes materias. De todos ellos, los que más se han puesto en práctica durante la realización del trabajo son la programación en Python vista en múltiples asignaturas puesto que es el lenguaje elegido para abordar el reconocimiento de entidades nombradas cuyos conceptos base han sido esenciales para el correcto desarrollo del modelo propuesto, la programación de modelos de aprendizaje automático enseñada también en el lenguaje de R y sobre todo la programación con las APIs o las bibliotecas de código abierto de Keras y Tensorflow para crear redes neuronales personalizadas. Estas últimas se han estudiado e implementado en los últimos años de la titulación pero debido a la gran variedad de funcionalidades y tipos de arquitecturas de modelos existentes adaptadas para tareas específicas, se han requerido nuevas técnicas que no han sido enseñadas, como es habitual en muchos proyectos.

Por otro lado, los conceptos relacionados con el procesamiento del lenguaje natural introducidos y puestos en práctica en el tercer curso también han sido muy útiles como las representaciones vectoriales de palabras, la ley de Zipf, y el etiquetado morfológico o *POS tagging* que también trata colecciones de documentos anotados. Además, los procesos iterativos de optimización de parámetros enseñados incluyendo el descenso por gradiente empleado en el algoritmo de retropropagación de la red neuronal han resultado muy provechosos. Finalmente, destacar la importancia de los modelos probabilísticos basados en predicciones estructuradas estudiados como los modelos ocultos de Markov pues determinan las secuencias de etiquetas para unas observaciones dadas con el algoritmo de Viterbi, donde en este caso se utiliza un campo aleatorio condicional que es también un modelo gráfico popular que modela las dependencias entre estados.

Las competencias transversales más significativas que se han necesitado para el desarrollo del trabajo son:

1. **CT2 – Aplicación y pensamiento práctico**: Se han identificado las ventajas y desventajas de todos los enfoques en relación con el reconocimiento de entidades nombradas. La opción seleccionada para cumplir con los objetivos propuestos es un modelo neuronal recurrente que proporciona uno de los mejores rendimientos de clasificación que constituye al estado del arte en muchas colecciones de datos, por ello se considera que la propuesta es la más idónea tras haber analizado resultados de referencia de otros sistemas. Aunque en los modelos NER de aprendizaje profundo es cierto que hay muchas variaciones en cuanto a la composición final, se han elegido las técnicas y elementos más populares en el campo de investigación, mostrando con métricas de evaluación sus prestaciones sobre unos datos específicos.

2. **CT3 – Análisis y resolución de problemas**: La solución presentada ha pasado por muchas fases de corrección partiendo de una idea inicial. A medida que se han ido explorando y conociendo nuevas alternativas que son bien valoradas por la comunidad científica, el modelo se ha expandido con componentes adicionales hasta conseguir una serie de resultados deseados. Con todas las piezas puestas en común, se ha conseguido crear un sistema potente para reconocer entidades en textos. La metodología también se ha visto alterada tanto en la fase de entrenamiento como en la de evaluación, apostando por la técnica de la validación cruzada y considerando

nuevas métricas de rendimiento como la precisión, cobertura, valor F1 y distancia de edición.

3. **CT11 – Aprendizaje permanente**: Muchos conceptos relacionados con la red neuronal y en especial con el procesamiento de datos secuenciales eran desconocidos hasta el momento. Los métodos y técnicas relacionadas aprendidas a lo largo del grado no son suficiente para cumplir con éxito esta tarea por lo que se necesita invertir tiempo en investigación y lectura de otras propuestas si se desea conseguir la mejor red neuronal posible. Para solucionar errores de programación que han surgido por el camino, la consulta en foros de ayuda ha sido en ocasiones clave y nuevas funciones de programación han sido aprendidas.

4. **CT12 – Planificación y gestión del tiempo**: En un trabajo de semejantes características como es un TFG, la organización del tiempo es un factor muy importante para cumplir con la correcta finalización del mismo. Esto implica constancia en su dedicación y seguir un esquema de elaboración de subtareas. Al principio, ha sido una tarea difícil compaginar los estudios con la investigación sobre el contenido incluyendo tiempo de lectura sobre trabajos relacionados e implementación de código. Sin embargo, conforme se han ido acabando las asignaturas del grado, el desarrollo del proyecto ha sido mayor y se han mantenido más reuniones de seguimiento.

8.3 Trabajos futuros

Como el tiempo para realizar este trabajo es limitado, se proponen futuras direcciones a explorar partiendo del estudio completado en la detección de entidades nombradas en textos manuscritos. En base a los objetivos originales propuestos, se plantea la posibilidad de evaluar y comparar el rendimiento del modelo con registros matrimoniales pertenecientes a otros libros o volúmenes de la colección que presentan una estructura similar, siempre y cuando estén disponibles. Puesto que son libros que se han escrito en diferentes siglos y por diferentes autores, se podría estudiar si existe una variación en el lenguaje y con ello comprobar de que forma afecta a las predicciones finales de clasificación. También se puede analizar el comportamiento de la red neuronal en detectar entidades en las n-mejores transcripciones de cada registro, agrupándose según la posición que ocupan con respecto a sus probabilidades condicionales. La idea sería ver si la distancia de edición promedia crece conforme se crean hipótesis de menor probabilidad a posteriori.

Bibliografía

[1] M. L. Díez Platas, S. Ros Munoz, E. González-Blanco, P. Ruiz Fabo, and E. Alvarez Mellado, "Medieval spanish (12th–15th centuries) named entity recognition and attribute annotation system based on contextual information," *Journal of the Association for Information Science and Technology*, vol. 72, no. 2, pp. 224–238, 2021.

[2] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," *Transactions of the association for computational linguistics*, vol. 4, pp. 357–370, 2016.

[3] "READ-COOP SCE transkribus." https://readcoop.eu/transkribus/. [Accessed: 2022-05-18].

[4] A. Fornés, V. Romero, A. Baró, J. I. Toledo, J. A. Sánchez, E. Vidal, and J. Lladós, "Icdar2017 competition on information extraction in historical handwritten records," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 1389–1394, IEEE, 2017.

[5] V. Romero, A. Fornés, N. Serrano, J. A. Sánchez, A. H. Toselli, V. Frinken, E. Vidal, and J. Lladós, "The esposalles database: An ancient marriage license corpus for off-line handwriting recognition," *Pattern Recognition*, vol. 46, no. 6, pp. 1658–1669, 2013.

[6] Datapeaker, "Arquitectura de red neuronal convolucional." https://datapeaker.com/big-data/arquitectura-de-red-neuronal-convolucional-arquitectura-cnn/, 2020. [Accessed: 2022-05-27].

[7] M. Esteve, "Word embeddings." Retrieved from https://marescas.medium.com/word-embeddings-8e6efd145e2e, 2019. [Accessed: 2022-05-22].

[8] M. Marrero, J. Urbano, S. Sánchez-Cuadrado, J. Morato, and J. M. Gómez-Berbís, "Named entity recognition: fallacies, challenges and opportunities," *Computer Standards & Interfaces*, vol. 35, no. 5, pp. 482–489, 2013.

[9] C. Dozier, R. Kondadadi, M. Light, A. Vachher, S. Veeramachaneni, and R. Wudali, "Named entity recognition and resolution in legal text," in *Semantic Processing of Legal Texts*, pp. 27–43, Springer, 2010.

[10] E. Leitner, G. Rehm, and J. Moreno-Schneider, "Fine-grained named entity recognition in legal documents," in *International Conference on Semantic Systems*, pp. 272–287, Springer, 2019.

[11] J. C. S. Alvarado, K. Verspoor, and T. Baldwin, "Domain adaption of named entity recognition to support credit risk assessment," in *Proceedings of the Australasian Language Technology Association Workshop 2015*, pp. 84–90, 2015.

[12] S. Keretna, C. P. Lim, D. Creighton, and K. B. Shaban, "Enhancing medical named entity recognition with an extended segment representation technique," *Computer methods and programs in biomedicine*, vol. 119, no. 2, pp. 88–100, 2015.

[13] K. Xu, Z. Zhou, T. Hao, and W. Liu, "A bidirectional lstm and conditional random fields approach to medical named entity recognition," in *International Conference on Advanced Intelligent Systems and Informatics*, pp. 355–365, Springer, 2017.

[14] A. Jiao, "An intelligent chatbot system based on entity extraction using rasa nlu and neural network," in *Journal of Physics: Conference Series*, vol. 1487, p. 012014, IOP Publishing, 2020.

[15] N. Dingwall and V. R. Gao, "Enhancing gazetteers for named entity recognition in conversational recommender systems," in *Proceedings of the Joint KaRS & ComplexRec Workshop. CEUR-WS*, 2021.

[16] B. M. Sundheim, "Overview of results of the muc-6 evaluation," 1995.

[17] J. R. Curran and S. Clark, "Language independent ner using a maximum entropy tagger," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, pp. 164–167, 2003.

[18] G. R. Doddington, A. Mitchell, M. A. Przybocki, L. A. Ramshaw, S. M. Strassel, and R. M. Weischedel, "The automatic content extraction (ace) program-tasks, data, and evaluation.," in *Lrec*, pp. 837–840, Lisbon, 2004.

[19] G. Demartini, T. Iofciu, and A. P. d. Vries, "Overview of the inex 2009 entity ranking track," in *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pp. 254–264, Springer, 2009.

[20] K. Balog, P. Serdyukov, and A. P. d. Vries, "Overview of the trec 2010 entity track," tech. rep., NORWEGIAN UNIV OF SCIENCE AND TECHNOLOGY TRONDHEIM, 2010.

[21] A. S. Starostin, V. V. Bocharov, S. V. Alexeeva, A. A. Bodrova, A. S. Chuchunkov, S. Dzhumaev, I. V. Efimenko, D. V. Granovsky, V. F. Khoroshevsky, I. V. Krylova, *et al.*, "Factrueval 2016: evaluation of named entity recognition and fact extraction systems for russian," 2016.

[22] C. Grover, S. Givon, R. Tobin, and J. Ball, "Named entity recognition for digitised historical texts," in *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, 2008.

[23] K. Bontcheva, D. Maynard, H. Cunningham, and H. Saggion, "Using human language technology for automatic annotation and indexing of digital library content," in *International Conference on Theory and Practice of Digital Libraries*, pp. 613–625, Springer, 2002.

[24] G. Crane and A. Jones, "The challenge of virginia banks: an evaluation of named entity analysis in a 19th-century newspaper collection," in *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries*, pp. 31–40, 2006.

[25] M. Ehrmann, G. Colavizza, Y. Rochat, and F. Kaplan, "Diachronic evaluation of ner systems on old newspapers," in *Proceedings of the 13th Conference on Natural Language Processing (KONVENS 2016)*, no. CONF, pp. 97–107, Bochumer Linguistische Arbeitsberichte, 2016.

[26] T. L. Packer, J. F. Lutes, A. P. Stewart, D. W. Embley, E. K. Ringger, K. D. Seppi, and L. S. Jensen, "Extracting person names from diverse and noisy ocr text," in *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, pp. 19–26, 2010.

[27] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1, pp. 50–70, 2022.

[28] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[29] Z. Yang, R. Salakhutdinov, and W. Cohen, "Multi-task cross-lingual sequence tagging from scratch," *arXiv preprint arXiv:1603.06270*, 2016.

[30] G. Wu, G. Tang, Z. Wang, Z. Zhang, and Z. Wang, "An attention-based bilstm-crf model for chinese clinic named entity recognition," *Ieee Access*, vol. 7, pp. 113942–113949, 2019.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[32] X. Han, C. K. Kwoh, and J.-j. Kim, "Clustering based active learning for biomedical named entity recognition," in *2016 International joint conference on neural networks (IJCNN)*, pp. 1253–1260, IEEE, 2016.

[33] J. Heaton, "An empirical analysis of feature engineering for predictive modeling," in *SoutheastCon 2016*, pp. 1–6, IEEE, 2016.

[34] H.-g. Lee, G. Park, and H. Kim, "Effective integration of morphological analysis and named entity recognition based on a recurrent neural network," *Pattern Recognition Letters*, vol. 112, pp. 361–365, 2018.

[35] S. T. Piantadosi, "Zipf's word frequency law in natural language: A critical review and future directions," *Psychonomic bulletin & review*, vol. 21, no. 5, pp. 1112–1130, 2014.

[36] G. Panchal, A. Ganatra, Y. Kosta, and D. Panchal, "Behaviour analysis of multilayer perceptrons with multiple hidden neurons and hidden layers," *International Journal of Computer Theory and Engineering*, vol. 3, no. 2, pp. 332–337, 2011.

[37] S. Du, J. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," in *International conference on machine learning*, pp. 1675–1685, PMLR, 2019.

[38] S. Kanai, Y. Fujiwara, and S. Iwamura, "Preventing gradient explosions in gated recurrent units," *Advances in neural information processing systems*, vol. 30, 2017.

[39] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, no. 02, pp. 107–116, 1998.

[40] P. Ha, S. Zhang, N. Djuric, and S. Vucetic, "Improving word embeddings through iterative refinement of word-and character-level models," in *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 1204–1213, 2020.

[41] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[42] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[43] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, 05 2020.

[44] Y. Yao, J. Wang, P. Long, M. Xie, and J. Wang, "Small-batch-size convolutional neural network based fault diagnosis system for nuclear energy production safety with big-data environment," *International Journal of Energy Research*, vol. 44, no. 7, pp. 5841–5855, 2020.

[45] TensorFlow, "Introducing tensorflow feature columns." https://developers.googleblog.com/2017/11/introducing-tensorflow-feature-columns.html, 2017. [Accessed: 2022-06-30].

[46] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[47] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation.," *Encyclopedia of database systems*, vol. 5, pp. 532–538, 2009.

[48] A. Stiegler, "Exploring conditional random fields for nlp applications." https://hyperscience.com/tech-blog/exploring-crfs-for-nlp-applications/, 2021. [Accessed: 2022-06-02].

[49] V. Romero, A. Fornés, E. Granell, E. Vidal, and J. A. Sánchez, "Information extraction in handwritten marriage licenses books," in *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, pp. 66–71, 2019.

[50] R. Haldar and D. Mukhopadhyay, "Levenshtein distance technique in dictionary lookup methods: An improved approach," *arXiv preprint arXiv:1101.1232*, 2011.

[51] A. C. Rouhou, M. Dhiaf, Y. Kessentini, and S. B. Salem, "Transformer-based approach for joint handwriting and named entity recognition in historical document," *Pattern Recognition Letters*, vol. 155, pp. 128–134, 2022.

[52] J. Giner, "Ner with bi-lstm-crf model." https://github.com/JoseGiner67/TFG, 2022. [Accessed: 2022-06-30].

ANEXO

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

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

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

Los objetivos de desarrollo sostenible de la Agenda 2030 dan lugar a acciones que todos juntos como una comunidad comprometida debemos de seguir por el bien de la población y del planeta. Con respecto al TFG realizado, algunos de los objetivos están ligeramente relacionados con la finalidad del sistema de reconocimiento de entidades nombradas y otros no proceden del todo. A continuación, se indican los de mayor conexión y en qué aspectos se cubren.

Un clasificador de entidades puede contribuir a garantizar una educación de calidad a los niños y jóvenes a la hora de aprender e identificar campos semánticos de palabras con mínima intervención de profesores. Con unas categorías de etiquetas a reconocer según el contenido textual, un algoritmo para NER entrenado revela aquellos grupos de palabras que comparten un significado en común y de esta forma aprovechar el conocimiento extraído con técnicas de aprendizaje profundo y procesamiento del lenguaje natural durante el proceso habitual de inferencia humana. Por ejemplo, para enseñar a poner en contexto por primera vez conceptos básicos como los colores, vehículos o ropa, además de otros grupos de palabras más complejos que pertenecen a un campo más técnico y específico.

La detección automática de información relevante en grandes volúmenes de documentos es una innovación tecnológica que aumenta la productividad de varios negocios. Un ejemplo es en la segmentación de las valoraciones y reseñas acerca de un producto o servicio dependiendo de lugares y personas mencionadas, ayudando a la empresa a identificar puntos débiles en los que tiene que mejorar para cumplir con las expectativas del cliente. La inteligencia artificial está siendo cada vez más implementada para agilizar los procesos de negocio y con ello generar ventajas competitivas en el mercado. Extraer información de interés en textos no estructurados resulta en el ahorro de tiempo en la lectura requerida pues el programa devuelve aquellos términos que distinguen el contenido actual del resto y disponer de este recurso supone un adelanto en el sector.

Los textos manuscritos necesitan papel o cualquier otro apoyo flexible para ser escritos y la información presente en ellos supone una dificultad añadida al clasificador de entidades nombradas puesto que se tienen que emplear técnicas de reconocimiento de imagen o de escritura para generar unas transcripciones que por varios motivos pueden ser imprecisas y afectar a las salidas predichas. El proceso de fabricación del papel implica la destrucción de árboles y bosques que son un pulmón de oxígeno para afrontar el cambio climático. Si a esto se le suma las altas emisiones de CO_2 como consecuencia de la deforestación y de los procesos de elaboración, la situación es aun más preocupante. Por ello, el sistema diseñado impulsa la digitalización de la información como solución al problema por un lado para dar prestaciones de mejor calidad y por otro para reducir el consumo de papel incluido en la elaboración de los textos a extraer entidades con el fin de proteger al medio ambiente.

En conclusión, se han destacado algunos objetivos del desarrollo sostenible relacionados con este trabajo. Aunque no existe un enlace directo con muchos de ellos, el reconocimiento de entidades nombradas es una herramienta poderosa para extraer aquello de mayor provecho en documentos sin una estructura común que supone un avance en la manera de esquematizar la información y contribuye principalmente en el objetivo de la innovación tecnológica para dar eficiencia de búsqueda en colecciones masivas de textos.

ETS Enginyeria Informàtica
Camí de Vera, s/n, 46022, València
T +34 963 877 210
F +34 963 877 219
etsinf@upvnet.upv.es - www.inf.upv.es