Grado en Ingeniería Informática

Trabajo Fin de Grado

Reconocimiento de Facturas de Consumo Eléctrico haciendo uso de Redes Neuronales Recurrentes

Nassr Eddine Moussati Lamhamdi

Tutorizado por:

Javier Sánchez Pérez

Nelson Monzón López

Las Palmas de Gran Canaria

18 de diciembre de 2020

Agradecimientos

En primer lugar quiero agradecer a mis tutores D. Javier Sánchez y D. Nelson Monzón por darme la oportunidad de participar en este proyecto, y a Javier quien me guió a través de este proyecto para alcanzar unos resultados aceptables.

Por último, quiero agradecer a todos mis compañeros, amigos y a mi familia, por apoyarme aún cuando mis ánimos decaían y no sabía que hacer. En especial, quiero hacer mención de mi familia y amigos más cercanos, que siempre estuvieron ahí dando su apoyo incondicional aun cuando no entendían que es lo que hacía ni que hablaba.

Muchas gracias a todos por la paciencia que habéis tenido conmigo.

Resumen

El objetivo principal del proyecto es desarrollar una aplicación que permita leer facturas de distintas comercializadoras de energía en distintos formatos (PDF e imágenes), reconocer entidades en el texto generado y, posteriormente, generar un JSON de las entidades reconocidas.

Dada la variedad de facturas que nos encontramos, es difícil aplicar un método determinista que resuelva el problema, con lo que es conveniente aplicar técnicas de aprendizaje profundo. La aplicación se basa en herramientas que hacen uso de redes neuronales, principalmente recurrentes. Para esto se utiliza spaCy como librería principal para el desarrollo.

Una vez definido y entrenado nuestro modelo con herramientas como spaCy, hemos conseguido detectar la mayoría de las entidades en las facturas con una fiabilidad superior al 75 %.

Abstract

The main objective of the project is to develop an application that reads invoices from different energy marketing firms in different formats (PDF and images), recognize entities in the generated text and then generate a JSON of recognized entities.

Given the variety of invoices that we have, it is difficult to apply a deterministic method that solves the problem, so it is convenient to apply deep learning techniques. The application is based on tools that use mainly recurrent neural networks. We use spaCy as the main library for development.

After defining and training our model with tools such as spaCy, we were able to detect most of the entities in the invoices with an accuracy of 75 % or above.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

Muchas empresas tienen la necesidad de extraer información de las facturas que reciben de sus proveedores o clientes. Esta información se suele almacenar en los sistemas de información de la organización con el fin de mejorar la gestión o para tener un mayor conocimiento sobre el mercado, lo que puede suponer una ventaja competitiva con respecto a la competencia.

En el caso de la empresa, y dentro del marco de este proyecto, las facturas que se quieren procesar son las de los clientes de otras compañías. El objetivo que se persigue es el de automatizar la extracción de la información para facilitar la tarea de los comerciales. A partir del conocimiento de las facturas, la organización puede ofrecer, a los potenciales clientes, mejoras en el servicio. Por otro lado, también va a permitir explotar los datos con el fin de analizar el comportamiento del mercado y poder implementar políticas que puedan contribuir a mejorar su acción comercial.

Actualmente, existen aplicaciones comerciales genéricas que permiten extraer el contenido de estos ficheros, pero en la mayoría de los casos requieren intervención manual para determinar la zona en la que se encuentran los datos, o los datos que se extraen no son correctos, o no están bien estructurados. Esto significa que cada vez que cambia el formato de una factura es necesario volver a marcar las zonas con la información que se desea extraer.

En muchos casos, son los comerciales de la empresa los encargados de extraer manualmente la información consultando directamente las facturas. Esto requiere un esfuerzo considerable y la información que se extrae puede contener errores. Se persigue desarrollar un sistema automático que consiga extraer la información de las facturas de consumo eléctrico de forma rápida y fiable. Por lo tanto, el objetivo principal del proyecto es desarrollar un módulo que permita extraer información de las facturas de energía en distintos formatos (PDF e imágenes) y obtener el valor de una serie de campos de forma automática.

La aplicación será capaz de leer facturas de comercializadoras de energía de distintas compañías y podrá extraer un conjunto de campos determinados a partir de éstas. Existe una serie de datos básicos que se repiten en todas las facturas, pero la organización y el texto difieren considerablemente dependiendo de cada empresa. Es complicado desarrollar un método determinista que consiga extraer esta información de forma fiable, con lo que es necesario desarrollar métodos que sean capaces de aprender a partir de ejemplos de facturas.

En este proyecto se desarrollará un método para el reconocimiento de facturas de consumo eléctrico haciendo uso de técnicas de aprendizaje automático. Estas resultan adecuadas para este tipo de problemas por dos razones principales: la primera es que si se hace uso de técnicas tradicionales resultaría en estar codificando distintas reglas, que incluso se podrían solapar, por cada factura nueva; la segunda razón es que al estar tratando con centenares de documentos, usar técnicas de aprendizaje automático resulta en soluciones eficaces para la gestión de problemas respecto a la escalabilidad.

Junto al desarrollo del proyecto en base a redes neuronales, también se genera un conjunto de datos para el entrenamiento mediante un proceso de simulación. Dicho proceso recibe una plantilla de factura y genera varios documentos, previamente siendo procesados para eliminar información irrelevante como símbolos y signos de puntuación, con la información relevante a reconocer. Teniendo el conjunto de entrenamiento, se evaluaron distintos enfoques y se llegó a la conclusión de que el problema se resolvería enfocándolo como un problema de reconocimiento de entidades nombradas. Por esta razón se optó por hacer uso de la librería spaCy, ya que está diseñada específicamente con el objetivo de ser útil para implementar sistemas listos para producción.

Se obtuvieron resultados interesantes para el tiempo invertido en el entrenamiento, con un grado de fiabilidad superior al 75 % en el peor de los casos y, en el mejor de los casos, superior al 90 %. Para obtener dichos resultados se crearon métricas propias para la evaluación tanto de la precisión como de la pérdida del modelo durante el proceso de entrenamiento.

1.1. Objetivos

El objetivo principal del proyecto es desarrollar una herramienta que permita extraer información de las facturas de comercializadoras y distribuidoras de energía eléctrica en distintos formatos (PDF e imágenes), esto implica procesos de limpieza de los datos como de la generación de los mismos mediante simulación de datos. Posteriormente se anotan las entidades a reconocer y se procede al entrenamiento del modelo, este proceso es facilitado gracias a la librería spaCy, la cual abstrae ciertos conceptos durante el desarrollo. El modelo es evaluado durante dicho proceso de entrenamiento para verificar que tan bien se generalizan

las facturas. Como último paso está el obtener el valor de una serie de campos de la factura con un grado de fiabilidad superior al 75 % y en un rango de tiempo aceptable usando el modelo entrenado.

Desde el punto de vista académico, se persigue adquirir conocimientos que no se han obtenido a lo largo de la carrera, y reforzar aquellos que no se han aplicado lo suficiente. A su vez que se profundiza en el uso de frameworks de aprendizaje automático y en campos como lo es el aprendizaje profundo con el uso de redes neuronales recurrentes.

1.2. Competencias específicas

Las competencias cubiertas en este proyecto son los siguientes:

IS01. Capacidad para desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la ingeniería del software.

En el desarrollo del proyecto se ha procurado que el código sea limpio y entendible y por lo tanto abierto a futuras implementaciones sin necesidad de tener muchas dependencias.

IS02. Capacidad para valorar las necesidades del cliente y especificar los requisitos software para satisfacer estas necesidades, reconciliando objetivos en conflicto mediante la búsqueda de compromisos aceptables dentro de las limitaciones derivadas del coste, del tiempo, de la existencia de sistemas ya desarrollados y de las propias organizaciones.

El D. Javier Sánchez Pérez actuó como el cliente principal del producto, aunque existiese por detrás una entidad interesada, el objetivo del proyecto estaba claro y en base a ello se fueron definiendo fechas y obteniendo unas conclusiones para cada una de las etapas del desarrollo.

IS03. Capacidad de dar solución a problemas de integración en función de las estrategias, estándares y tecnologías disponibles.

A lo largo de todo el proyecto se usaron tecnologías de más bajo nivel como pueden ser Keras y Tensorflow hasta herramientas de alto nivel como spaCy, e inclusive Anaconda como gestor principal de entornos, herramientas y librerías para el desarrollo.

IS04. Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un

conocimiento adecuado de las teorías, modelos y técnicas actuales.

Se ha seguido una serie de pautas que dan como resultado el fin del proyecto, y entre estas etapas con encontramos con validar distintas estrategias de aprendizaje profundo y evaluar cuales ofrecen mejor resultado, y en base a esta medida se profundiza en una herramienta más específica para obtener el mejor resultado.

1.3. Planificación

En el desarrollo del proyecto se tomó un enfoque ágil, siguiendo algunas ideas de la metodología *Scrum*, por lo que se han ido definiendo distintas fases que abarcan todo el proyecto de principio a fin. En la tabla 1.1 se muestran las distintas fases por las que ha pasado el proyecto junto a su duración.

Cuadro 1.1: Planificación del trabajo

Fases	**Duración estimada (horas)**	**Tareas**
Estudio del estado actual y toma de contacto con distintas herramientas	20	Análisis de la situación actual (Estimado 10 horas)
		Evaluar distintas herramientas y priorización de necesidades (Estimado 10 horas)
Sprint Zero	40	Estudiar el funcionamiento de las redes neuronales recurrentes. (Estimado 20 horas)
		Plantear un esquema conceptual del proyecto (Estimado 20 horas)
Release 1 (Sprint 1)	40	Preparar y reajustar el dataset de pruebas además de diseñar y configurar la red. Desarrollar una interfaz sencilla de visualización de resultados
Release 2 (Sprint 2 y 3)	140	Investigar, desarrollar, evaluar y entrenar de un modelo que interprete información a partir texto. Esto requiere automatizar el proceso de limpieza del dataset para tanto entrenar la red como predecir
Release 3 (Sprint 4)	20	Desarrollo de interfaz final.
Documentación / Presentación	40	Redacción de la memoria y la presentación

El proceso de desarrollo se realizó de forma iterativa y con periodos de desarrollo de una semana, estableciendo reuniones donde se informaba de todos los avances como complicaciones en el proceso durante los *Sprints* 1, 2, 3 y 4. Gracias a la existencia de la retroalimentación de *Sprints* anteriores, se conseguía evitar invertir mucho tiempo en procesos que no aportaban mucho valor frente al tiempo que se requería.

1.4. Aportaciones del trabajo

Este proyecto solventa un problema que tienen distintas empresas donde buscan extraer información específica de las facturas. Esta información se suele almacenar en los sistemas de información de las organizaciones con el objetivo de mejorar la gestión o para tener un mayor conocimiento sobre el mercado, lo que puede suponer una ventaja competitiva con respecto a la competencia. Desde el punto de vista de la empresa, busca automatizar la extracción de información para poder obtener conocimiento de las facturas y poder competir en el mercado con soluciones más competitivas.

En el mercado existen distintos servicios que prometen resolver el problema, pero en la mayoría de los casos implica un trabajo manual o los costes son muy altos. Actualmente, los comerciales de la empresa extraen manualmente la información consultando directamente las facturas. Esto requiere un esfuerzo considerable y la información que se extrae puede contener errores. Por lo que Se persigue desarrollar un sistema automático que consiga extraer la información de las facturas de consumo eléctrico de forma rápida y fiable.

1.5. Organización de la documentación

A lo largo de esta sección se expondrá una breve resumen por cada uno de los capítulos en orden cronológico. La organización del documento se divide en los siguientes capítulos:

- En el Capítulo 2, se dará una breve introducción al estado del arte en el campo del procesamiento de lenguaje natural, como los tipos de redes neuronales más usados en este tipo de tareas, el uso de los *Transformers* y su importancia en el campo y de las tecnologías que más se están usando para reconocimiento de entidades y extracción de información en facturas, contratos, diagnósticos médicos, etc.
- En el Capítulo 3, se explicará en detalle cómo se desarrolló el conjunto de datos, como el proceso de limpieza, la introducción al concepto de reconocimiento de entidades nombradas, más conocido en inglés como *Named Entity Recognition* (NER), y la necesidad de anotar de las entidades en el conjunto de datos.
- El Capítulo 4 entrará en detalle sobre el desarrollo del proyecto y el motivo del uso de *spaCy* como herramienta principal para abordar el proyecto.
- En el Capítulo 5, se mostrarán los resultados obtenidos del entrenamiento y deducciones.
- En el Capítulo 6 se detallan las conclusiones a las que se han llegado y el trabajo futuro del proyecto.

Capítulo 2

Estado del arte

Cada día es más importante para los analistas de negocio el poder encontrar u obtener información sobre empresas desde múltiples fuentes. La web se ha convertido en una fuente de información muy importante en todas las áreas, los analistas intentan obtener más información de los productos de la competencia como del propio negocio monitorizando distintas métricas. Sin embargo, las herramientas utilizadas, en primera instancia, tanto para extraer información como procesar las distintas fuentes, distan de ser triviales.

El campo de análisis de negocios como el *Business Intelligence* está en auge y en gran medida es gracias a los avances en campos como el Procesamiento de Lenguaje Natural (PLN), y se busca cómo recolectar todos los datos posibles sobre la clientela, pasando por analizar los propios datos recolectados y por último elaborar estrategias para aumentar las ganancias. Gracias a este tipo de perfiles que hacen uso de dichas herramientas, las empresas pueden lograr objetivos significativos como aumentar ventas, fidelizar a los clientes, reducir gastos, etc.

Un campo que también se ve muy beneficiado de estos avances es el de la salud, donde la adopción del PLN en la atención médica está aumentando debido a su potencial para buscar, analizar e interpretar cantidades gigantescas de conjuntos de datos de pacientes. El uso de algoritmos médicos avanzados, el aprendizaje automático en el cuidado de la salud y los servicios de tecnología de PLN tienen el potencial de aprovechar conocimientos y conceptos relevantes a partir de datos que antes se consideraban irrelevantes. El PLN en los medios de atención médica da una oportunidad a los datos no estructurados del universo de la atención médica, brindando una visión increíble sobre la calidad de la comprensión, la mejora de los métodos y mejores resultados para los pacientes. Véase el proyecto THYME [19].

Gracias a la popularización de las redes neuronales han surgido muchas herramientas que

aprovechan su potencial para desarrollar herramientas que facilitan y reducen el costo del desarrollo de tareas manuales y repetitivas. Otro problema que se abordó gracias a las soluciones existentes es la extracción de información de documentos de interés para las empresas, como pueden ser las nóminas, facturas, cargos, etc.

A continuación se presentarán las herramientas más populares que buscan dar una solución eficaz al problema y una introducción a una serie de definiciones sobre distintas herramientas y/o tecnologías que se encuentran en desarrollo dentro del campo del procesamiento de lenguaje natural.

2.1. Herramientas para procesado de facturas

Actualmente, en el mercado se encuentran varias herramientas para extraer información de facturas, documentos, tickets, etc. Sin embargo, muchos hacen uso de métodos tradicionales como el uso de expresiones regulares y esto los hace dependientes del formato de la factura. Por ello, han surgido otras herramientas que están aprovechando las funcionalidades de las redes neuronales, ya sea infiriendo información, clasificando documentos o extrayendo información. Existe variedad de herramientas que hacen uso de redes neuronales, se presentarán las más conocidas en el mercado actual.

2.1.1. Tradeshift

Este software [13] es interno de la empresa Tradeshift y forma parte de las herramientas que ofrecen en sus servicios, pero aunque sea privado ellos publicaron un artículo donde mostraban cómo estaba construido el software y se hacía hincapié en los modelos de atención. Uno de los problemas al que daban importancia era lo costoso que resulta crear el conjunto de datos con el que se entrenaba el modelo.

Tradeshift en comparación a su competencia ofrece un flujo de trabajo depurado en lo que respecta al escaneo y análisis de la información de los documentos. Permite procesar PDFs e imágenes y genera como respuesta archivos PDF, PDF/A y TIFF.

2.1.2. Nanonets-OCR

La empresa Nanonets ofrece otro punto de vista aunque también haga uso de modelos de atención, pero agregan una capa más que el *Visual Attention* [28] que está hecha en base de CRNN (*Convolutional Recurrent Neural Network*) [22, 18]

La gran ventaja que ofrece Nanonets-OCR es su flexibilidad en el flujo de trabajo, ya que sus distintas partes del proceso se encuentran desacopladas y se puede integrar un desarrollo

propio de algunas fases, además ofrece la posibilidad tanto de agregar como crear modelos de factura personalizados y anotar todas las entidades que se requieren.

2.1.3. Intellix

En Intellix [20] abordan el problema de una forma distinta. Tienen en común que hacen uso de herramientas OCR como Tesseract para convertir un archivo PDF o imagen a texto, sin embargo, para clasificar el texto entrenaron un modelo kNN (*k-nearest neighbours*), en español k vecinos más cercanos, donde k lo fijaron en 5. Usaron Apache Lucene como principal clasificador y sobre este modelo crearon en Lucene un ranking de palabras haciendo uso de TF-IDF (*Term Frequency/Inverse Document Frequency*). Para la extracción de información crearon internamente una serie de algoritmos para indexar las palabras según su posición más frecuente y, dependiendo de si la palabra extraída se correspondía con una expresión regular u otra, se cambiaba de estrategia.

2.1.4. Docparser

Docparser[1], empresa fundada en 2016, se especializa en la extracción de información exclusivamente de documentos PDF y ofrece la exportación de los mismos en distintos formatos como XLSX (formato propietario de Microsoft Excel), CSV, JSON y XML. Además ofrecen integraciones directas con distintos servicios de almacenamiento para guardar tanto las facturas y correspondiente exportación. Docparser no entra en detalles respecto a las implementaciones de sus servicios pero sí hacen uso de un sistema basado en plantillas OCR (*Template OCR*). Consiste en encuadrar y aislar distintos campos de la factura como tablas de costes, dirección, información del cliente, etc, para extraer la información de forma independiente y que posteriormente se unifica en un solo documento. Las principales desventajas del software que han expresado muchos cliente son los falsos positivos y que tienen que repetir dicho proceso por cada modificación en la plantilla de una factura.

2.1.5. Rossum

Rossum[2], empresa fundada en 2017, ofrecen soluciones de extracción de información en documentos PDF y/o imágenes utilizando técnicas de inteligencia artificial. Permiten la exportación de la información extraída en distintos formatos como XLSX, CSV, JSON y XML. Destacan la posibilidad de integrar sus servicios a distintas aplicaciones de terceros gracias a que exponen una API para desarrolladores. Las principales tecnologías que utilizan son Zonal

[1]Docparser: `https://docparser.com/`
[2]Rossum: `https://rossum.ai/`

OCR para agrupar campos y para la extracción de información de las tablas utilizan YOLO[3] para la detección de objetos, en su caso los campos en las tablas. Entre sus desventajas se encuentra que el precio del servicio puede verse incrementado según la complejidad de la factura.

2.1.6. CloudScan

CloudScan [14] fue creado con la premisa de ser un sistema preciso, capaz de reconocer facturas que nunca había visto, sin agregar plantillas y sin configurar nada, y con ello mantener una alta precisión. Inspirados en Intellix [20] e ITESOFT [17], desarrollaron su propio sistema que reconoce un número limitado de entidades diferentes, en su caso 8. Su modelo principal, según indica la figura 2.1, está basado en una Bi-LSTM en conjunto con una capa *embedding* preentrenada.

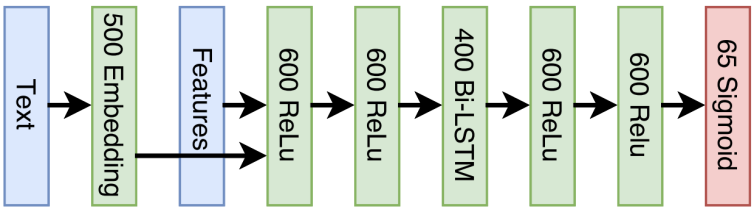


Figura 2.1: Modelo LSTM. Fuente: Technical University of Denmark - Tradeshift

CloudScan, a diferencia de las empresas mencionadas, no es una corporación en sí, pero sí es una herramienta que varias empresas utilizan o basaron sus implementaciones inspirados del modelo planteado por CloudScan.

2.2. Redes neuronales

Gracias al auge de las redes neuronales, se empezó a experimentar con ellas en todo tipo ámbitos y problemas. De entre todos los problemas, también beneficia los que concierne a este proyecto, en concreto en la extracción de información de facturas de luz. Analizando un poco en detalle las facturas, para una computadora, las facturas presentan una serie de datos no estructurados, por ejemplo archivos PDF e imágenes. Lo que resulta en que es mucho más complejo de analizar en comparación a tener hojas de cálculo o formularios, sin embargo, para los humanos no lo es. Ya que en un instante el ser humano, es capaz de comprender el significado de las palabras y los números, y todo esto se hace teniendo en cuenta la estructura y el diseño del documento. Realmente es un verdadero desafío crear un modelo de aprendizaje

[3]YOLO: `https://pjreddie.com/darknet/yolo/`

automático que pueda combinar el significado de las palabras y el diseño de los documentos de la misma manera. Sin embargo, las redes neuronales vienen a resolver este tipo problemas, aunque se requiera de una cantidad significativa de plantillas de facturas de luz únicas.

2.2.1. Redes neuronales recurrentes

Una red neuronal recurrente (RNN) [8] es un tipo de red neuronal artificial que utiliza datos secuenciales o datos de series temporales. Estos algoritmos de aprendizaje se utilizan comúnmente para problemas ordinales o temporales, como traducción de idiomas, procesamiento del lenguaje natural, reconocimiento de voz y subtítulos de imágenes; están incorporados en aplicaciones populares como asistentes por voz (Siri, Alexa, Google Assistant) o Google Translate. Pero destacan por su "memoria", ya que toman información de entradas anteriores para influir en la entrada y salida actuales. Mientras que las redes neuronales tradicionales asumen que las entradas y salidas son independientes entre sí, mientras que la salida de las redes neuronales recurrentes depende de los elementos anteriores dentro de la secuencia.

2.2.2. Long-Short Term Memory

Las RNN básicas tienen problemas con la memoria a corto plazo. Si una secuencia es lo suficientemente larga, tienen dificultades para transportar información de los primeros estados a los posteriores. Entonces, si se procesa un párrafo de texto para hacer predicciones, las RNN pueden omitir información importante que se encuentre al principio del texto. Por lo tanto, esto provoca la necesidad de una memoria a largo plazo. Una de las arquitecturas de RNN más conocidas, en inglés Long-Short Term Memory (LSTM) [5], es un tipo capaz de mantener las dependencias a largo plazo. Las redes LSTM tienen la capacidad de persistir la información durante largos períodos de tiempo [12]. Más recientemente ha aparecido otro tipo de red neuronal, conocida como Gated Recurrent Unit (GRU) [3], que tiene una arquitectura más sencilla que la anterior y obtiene resultados similares en general.

2.2.3. Mecanismos de Atención

Los modelos de atención [25] en el aprendizaje profundo, se puede interpretar como un vector de ponderaciones de importancia, por ejemplo, para predecir o inferir un elemento, como un píxel en una imagen o una palabra en una oración, estimamos usando el vector de atención qué tan fuerte es la correlación entre otros elementos y toma la suma de sus valores ponderados por el vector de atención como la aproximación del objetivo.

2.2.4. Word Embeddings

Las *Word Embeddings* [11] es una técnica del campo del procesamiento de lenguaje natural que consiste en asignar un vector a cada palabra. Este vector guarda información semántica, lo que permite que pueda ser asociado o disociado a otros vectores (palabras) según distintos contextos gramaticales. En este sentido, *Word Embeddings* se convierte en una solución efectiva para codificar tanto la semántica como la relación de las palabras entre sí. Dicha codificación es generalizable, lo que significa que el algoritmo creado puede ser utilizado para resolver distintos tipos de problemas, como la traducción, generación de textos, entre otros. Todo esto se sustenta gracias a la existencia de las capas *embedding*. Éstas se pueden utilizar para analizar texto. Esto requiere que los datos de entrada estén codificados en números enteros, de modo que cada palabra esté representada por un número entero único. Este paso de preparación de datos es conocido como *tokenize*. Este paso resulta importante ya que sin él no se puede alimentar la entrada de una capa *embedding*. La capa *embedding* se inicializa con pesos aleatorios y aprenderá a construir un *embedding* para todas las palabras en el conjunto de datos de entrenamiento. Es una capa flexible que se puede utilizar de diversas formas, como:

- Se puede usar solo para aprender una capa *embedding* que se puede guardar y usar en otro modelo más adelante.
- Se puede usar como parte de un modelo de aprendizaje profundo en el que el *embedding* se aprende junto con el modelo en sí.
- Se puede utilizar para cargar un modelo preentrenado de word *embedding*, un tipo de aprendizaje por transferencia.

A continuación vamos a revisar varias técnicas que se utilizan para generar *Word Embeddings* a partir de datos de texto.

Word2Vec

Word2Vec [11, 10], en español "de palabra a vector", es una técnica de procesamiento del lenguaje natural, fue desarrollado por un grupo de investigadores de Google liderados por Tomas Mikolov.

El algoritmo Word2Vec utiliza un modelo de red neuronal para aprender asociaciones de palabras de un gran corpus de texto. Una vez entrenado, dicho modelo puede detectar palabras sinónimas o sugerir palabras para completar la oración. Además, como su nombre lo indica, Word2Vec representa cada palabra distinta con una lista de números únicos. Los vectores se eligen cuidadosamente de modo que con una función matemática simple se pueda

conocer el nivel de similitud semántico entre las palabras representadas por esos vectores.

Global Vectors

Global Vectors (GloVe) [15], en español vectores globales, se creó con el objetivo de obtener representaciones vectoriales de palabras. Es un algoritmo de aprendizaje no supervisado desarrollado por Jeffrey Pennington, Richard Socher y Christopher D. Manning, en la universidad de Stanford, para generar *embeddings* de palabras agregando una matriz global de co-ocurrencia palabra-palabra de un corpus.

2.2.5. Transformer

El Transformer [25] es un modelo de aprendizaje profundo introducido en 2017, utilizado principalmente en el campo del procesamiento del lenguaje natural.

Al igual que las redes neuronales recurrentes, los Transformers están diseñados para manejar datos secuenciales, como el lenguaje natural, para tareas como la traducción y el resumen de texto. Sin embargo, a diferencia de los RNN, los Transformers no requieren que los datos secuenciales se procesen en orden. Por ejemplo, si los datos de entrada son una oración en lenguaje natural, el Transformer no necesita procesar el principio antes del final. Debido a esta característica, el Transformer permite mucha más paralelización que los RNN y, por lo tanto, reduce los tiempos de entrenamiento [25].

Desde su introducción, los Transformers se han convertido en el modelo de elección para abordar muchos problemas de PLN, reemplazando los modelos de redes neuronales recurrentes más antiguos, como las LSTM. Dado que el modelo Transformer facilita una mayor paralelización durante el entrenamiento, ha permitido el entrenamiento en conjuntos de datos más grandes de lo que era posible antes de su introducción. Esto ha llevado al desarrollo de sistemas previamente entrenados como BERT [4] y GPT [16], que han sido entrenados con enormes conjuntos de datos de lenguaje general y pueden ajustarse a tareas de lenguaje específicas.

BERT [4] es un modelo de aprendizaje profundo que ha proporcionado resultados de vanguardia en una amplia variedad de tareas de procesamiento del lenguaje natural. Sus siglas significan *Bidirectional Encoder Representations for Transformers*, en español Representación de Codificador Bidireccional de Transformadores. BERT es una técnica basada en redes neuronales para el pre-entrenamiento de modelos para el procesamiento de lenguaje natural y fue desarrollado por Google. Este modelo se entrenó usando el corpus de Wikipedia en inglés y BooksCorpus, y requiere de un ajuste fino, proceso en el que los parámetros del modelo deben ajustarse con precisión en base a las tareas a resolver.

XLNet [26] es un modelo de lenguaje autorregresivo que prioriza el orden de las palabras y el orden de predicción, frente a BERT, que la predicción se realiza en paralelo resultando en predicciones independientes del contexto que puede existir en la oración. El objetivo de su entrenamiento es calcular la probabilidad de un *token* de palabra condicionado a todas las permutaciones de tokens de palabras en una oración, en lugar de solo aquellas que están a la izquierda o solo de las que están a la derecha del *token* de destino.

Esto significa que es un modelo que calcula la probabilidad de que una palabra se encuentre en una oración. Por ello, un modelo que contiene la suficiente información para predecir lo que sigue en una oración puede aplicarse a otras tareas más útiles; por ejemplo, podría usarse para determinar a quién se menciona en el texto, qué acción se está tomando o si el texto tiene un sentimiento positivo o negativo. Por lo tanto, los modelos son pre-entrenados con el objetivo de modelar el lenguaje y posteriormente se ajustan los parámetros, conocido en inglés como *fine-tunning*, para resolver tareas más prácticas.

GPT-3 (Generative Pre-trained Transformer 3) [1] es un modelo de lenguaje autorregresivo que emplea aprendizaje profundo para generar texto similar a como lo haría un humano. Es la tercera generación de los modelos de predicción de lenguaje perteneciente a la serie GPT, creados por OpenAI, un laboratorio de investigación de inteligencia artificial con sede en San Francisco. La versión completa de GPT-3 [21] tiene una capacidad de 175 mil millones de parámetros de aprendizaje automatizado [1], lo cual supera la magnitud de su predecesor, GPT-2, y a toda su competencia a día de hoy. Cabe destacar que todos los modelos se entrenaron con 300 mil millones de tokens. Una peculiaridad de este modelo es que a medida que se aumentaba el número de parámetros de entrenamiento, también se aumentaba su *batch_size* y se disminuía el ratio de aprendizaje. Actualmente, los modelos basados en Transformers son parte de una tendencia en sistemas de PLN basados en 'representaciones de lenguaje pre-entrenadas' [2].

Capítulo 3

Facturas de consumo eléctrico

Las facturas son una parte crucial de todo el proyecto, ya que el objetivo final es conseguir un JSON como resultado de una serie de transformaciones de la factura. Uno de los casos que implica esto como un reto es que el modelo tiene que ser capaz de generalizar el concepto de factura a nivel semántico.

Esto es un problema, ya que las facturas no son textos que se puedan leer como una historia, sino que es un conjunto de datos distribuidos en tablas. Y lo que aumenta más la complejidad es que no existe un estándar de factura de consumo eléctrico, cierto es que hay una serie de campos que siempre tienen que estar, pero dichos campos pueden hacer referencia a lo mismo pero con nombres distintos. A continuación se realizará una breve explicación de los términos más importantes de una factura de luz.

3.1. Campos de una factura

La mayoría de las facturas tienen en común una información básica que mostrar en las facturas para los clientes y estas son los datos del cliente, datos de la factura, resumen, información del consumo y detalles del contrato.

- Datos de la factura: Está compuesto del importe de la factura que es el precio total a abonar, el número de factura que es un identificador de la misma, un código de referencia que identifica el contrato de suministro con el cliente, el periodo de consumo y la fecha de cargo.
- Datos del cliente: Está compuesto, generalmente, de la siguiente información: nombres y apellidos del contratante, dirección de la vivienda, código postal al que pertenece, población y provincia.

- Resumen de la factura: Se muestran los costes principales de los que está compuesto, el importe total de la factura, el coste de la potencia contratada, la energía consumida en el periodo, el alquiler de equipos (alquiler del contador) e impuestos.
- Información del consumo: Se muestra el consumo de energía eléctrica por mes y total en el periodo que comprende la factura.
- Información del contrato: En esta sección se detalla tanto el CUPS, que es un identificador de una vivienda con suministro de electricidad, como la distribuidora y la comercializadora contratada. También vienen definidos dos aspectos muy importantes: el peaje de acceso y la potencia contratada. El peaje de acceso puede resultar en casi la mitad del coste de la factura y está relacionado tanto con la energía consumida como la potencia contratada, que es la que limita cuánta energía máxima, en kilovatios, puedes consumir en un instante.

3.2. Formato de una factura de electricidad

Las facturas de consumo eléctrico son como cualquier otra factura, ya sea de gas o agua. Existen semejanzas, pero existen peculiaridades que hacen que las facturas de consumo eléctrico sean distintas del resto, e incluso entre las propias comercializadoras, la distribución de la información a lo largo de la factura y la terminología utilizada, puede haber diferencias significativas.

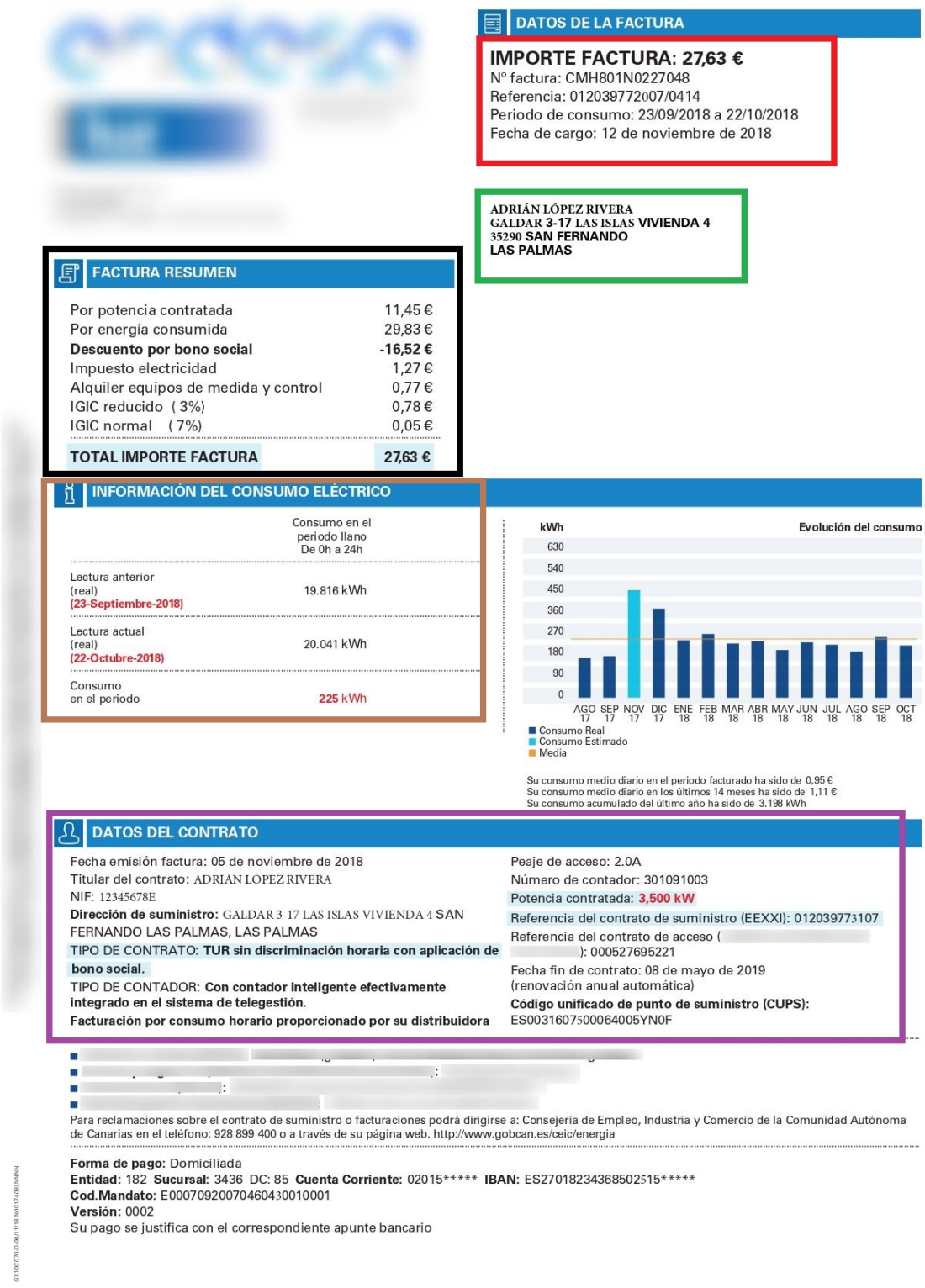
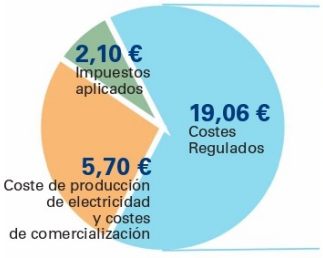


Figura 3.1: Factura modificada con campos importantes remarcados - Página 1.

DESTINO DEL IMPORTE DE LA FACTURA

El destino del importe de su factura, **27,63 euros,** es el siguiente:



7,55 €	**Incentivos a las energías renovables, cogeneración y residuos**
7,62 €	**Coste de redes de transporte y distribución**
3,89 €	**Otros costes regulados** (incluida la anualidad del déficit)

A los importes indicados en el diagrama debe añadirse, en su caso, el importe del alquiler de los equipos de medida y control así como los conceptos no energéticos.

DETALLE DE LA FACTURA

Facturación por potencia contratada: Comprende dos conceptos: la facturación por peaje de acceso (resultado de multiplicar los kW contratados por el precio del término de potencia del peaje de acceso y el número de días del periodo de facturación) y la facturación por margen de comercialización fijo.

Importe por peaje de acceso:		
3,5 kW x 38,043426 Eur/kW y año x (29/365) días	10,58 €	
Importe del término fijo de los costes de comercialización:		
3,5 kW x 3,113 Eur/kW y año x (29/365) días	0,87 €	
		11,45 €

Facturación por energía consumida: Comprende dos conceptos: la facturación por peaje de acceso (resultado de multiplicar los kWh consumidos en el periodo de facturación por el precio del término de energía del peaje de acceso) y la facturación por coste de la energía (resultado de multiplicar los kWh consumidos por el precio del término del coste horario de energía del PVPC).

Importe por peaje de acceso:		
224,686 kWh x 0,044027 Eur/kWh	9,90 €	
Importe por coste de la energía (*):		
93,29 kWh x 0,092662 Eur/kWh (**)	8,64 €	
16,982 kWh x 0,085913 Eur/kWh (**)	1,46 €	
114,414 kWh x 0,085901 Eur/kWh (**)	9,83 €	
		29,83 €

Aplicación del Bono Social: A la facturación por potencia contratada y por **224,686 kWh** del total de la energía consumida, se le aplica un **40%** de descuento por Bono Social.

Descuento Potencia 11,45 Eur x 40 %	-4,59 €	
Descuento Energía (1) 29,83 Eur x 40 %	-11,93 €	
		-16,52 €

(1) Descuento Energía: El descuento por Bono Social se aplica al total del importe de "facturación por energía consumida", porque su consumo en esta factura no excede el límite con derecho a Bono Social:

Consumo total de su factura: 224,686 kWh
Consumo de su factura con descuento por Bono Social: 224,686 kWh (Límite de consumo con derecho a Bono Social según se establece en el RD 897/2017 y RDL 15/2018).
Consumo de su factura sin descuento por Bono Social: 0 kWh

Subtotal		**24,76 €**

Impuesto de electricidad: Impuesto especial al tipo del 5,11269632% sobre el producto de la facturación de la electricidad suministrada

Impuesto electricidad (24,76 X 5,11269632 %)	1,27 €	

Alquiler de equipos de medida y control. Precio establecido que se paga por el alquiler de equipos de medida y control.

Alquiler equipos de medida y control (29 días x 0,026552 Eur/día)	0,77 €	
Subtotal otros conceptos		**2,04 €**
Importe total	26,80 €	
IGIC: Impuesto General Indirecto Canario al tipo del 3% 7%		
IGIC reducido (3%) 3% s/ 26,03		0,78 €
IGIC normal (7%) 7% s/ 0,77		0,05 €
TOTAL IMPORTE FACTURA		**27,63 €**

El importe de su factura a PVPC previo a la aplicación del descuento por Bono Social, asciende a 45,51 €. De acuerdo a lo establecido en el artículo 12 del RD 897/2017, de 6 de Octubre, para los consumidores vulnerables severos en riesgo de exclusión social, el importe mínimo a financiar a efectos de lo establecido en el punto 1 del Artículo 12 del RD 897/2017, sería de 22,75 €.

Precios de los términos del peaje de acceso publicados en Orden ETU 1282/2017
PVPC calculado según Real Decreto RD 216/2014
Margen de comercialización fijo publicado en RD 469/2016. Orden ETU 1948/2016
Descuento del bono social regulado en RD 897/2017 y RDL 15/2018.
Precio del alquiler de los equipos de medida y control en Orden IET 1491/2013 de 3 de agosto

Figura 3.2: Factura modificada con campos importantes remarcados - Página 2.

En base a las dos figuras anteriores, 3.1 y 3.2, se puede apreciar que las facturas de consumo eléctrico tienen una serie de datos básicos que comparten todas las comercializadoras:

- Datos del cliente: En este apartado, que se encuadró con color verde en la ilustración 3.1, se encuentra el nombre completo y la dirección del domicilio.
- Datos de la factura: Marcado en color rojo en la ilustración 3.1, encontramos el importe total de la factura, el identificador de la factura como el periodo de consumo y fecha de cargo. Este apartado suele variar dependiendo de la comercializadora.
- Resumen de la factura: Marcado en color negro en la ilustración 3.1, se muestra una tabla con los costes que definen el importe total de la factura.
- Información de consumo eléctrico: Encuadrado en color marrón en la ilustración 3.1, se informa del consumo de energía eléctrica según el periodo de inicio y fin de la factura.
- Datos del contrato: Encuadrado en color morado en la ilustración 3.1, muestra una serie de campos relevantes para el cliente como sus datos, la fecha de emisión de la factura, tipo de contrato, tarifa de peaje de acceso, potencia contratada, CUPS y fecha de fin del contrato con la comercializadora.
- Detalle de la factura: Encuadrado en color naranja en la ilustración 3.2, se realiza un desglose de todos los apartados que componen una factura y de todos sus costes. Este apartado a veces es omitido por las comercializadoras.

3.3. Simulación de datos de una factura

Para el conjunto de entrenamiento se creó una serie de facturas en base a distintas plantillas de facturas de consumo eléctrico, por lo que se simularon los datos. Uno de los motivos principales que impulsó la idea de simular facturas es debido a las pocas facturas que se tenían y la dificultad y el tiempo que conllevaba anotar a mano todas las facturas.

La simulación se realizó con un programa propio teniendo en cuenta los campos importantes de la factura que fueron los que se modificaban, por ejemplo, CUPS, nombres, apellidos, calles, municipios, comercializadoras y distribuidoras. Las siglas CUPS son las iniciales de Código Universal de Punto de Suministro, y en España es un código único e identificador de un punto de suministro de energía, ya sea de electricidad o gas. Está compuesto por 20 o 22 dígitos alfanuméricos, es permanente e invariable. Sirve principalmente para identificar viviendas o locales que reciben suministro ya sea de electricidad como de gas. El CUPS identifica puntos de suministro en viviendas o locales, nunca personas. La estructura del CUPS es la siguiente:

- Los dos primeros caracteres representan al país, en este caso España es ES.
- Los cuatro siguientes caracteres son números e identifican a la empresa distribuidora, ya que todas están reconocidas por el Ministerio de Energía mediante un código.
- Por último, los doce números restantes son asignados por la propia distribuidora para reconocer el punto de suministro final. A estas cifras las acompañan dos o más letras que se usan para controlar y detectar errores en el suministro.

Se creó una serie de archivos con los nombres, apellidos, calles y municipios existentes en España y las comercializadoras y distribuidoras se obtuvieron de la página web de la Comisión Nacional de los Mercados y la Competencia (CNMC[1]). Para las entidades como el Número de Identificación Fiscal (NIF), número del contrato, potencia contratada, tarifa peaje de acceso, se realizaba una operación aleatoria donde escogía un número o letras de forma aleatoria de tal manera que se corresponda con el formato esperado para cada campo. Por ejemplo, en el NIF se espera una secuencia de 8 dígitos y un carácter alfabético, y en potencia contratada existe una lista de las distintas potencias que se pueden contratar. A continuación se mostrará cada uno de los archivos utilizados para realizar la simulación de las facturas:

```
1  Aarón       Aharón      Abel        Abelardo    Abigail     Abraham
2  Absalón     Adalberto   Adán        Adelardo    Adulfo      Adolfo
3  Adonis      Adrián      Adriano     Agamenón    Agapito     Agar
4  Agustín     Aitor       Alan        Albano      Alban       Alberto
5  Albino      Aldo        Alejandro   Alejo       Alex        Alexandro
6  Alf         Alfonso     Alfredo     Alonso      Alvar       . . . . .
```

Listing 1: Fichero spanish_names.txt

```
1  Acosta      Acuña       Aguilar     Aguirre     Agustín     Ahumada
2  Alanis      Alarcón     Alayón      Alcázar     Alcocer     Alfaro
3  Almendárez  Altamirano  Alvarez     Alzate      Amador      Anaya
4  Argüelles   Arjona      Arriaga     Arrollo     Ayala       Baca
5  Báez        Baños       Barba       Barrera     Bastida     . . . . .
```

Listing 2: Fichero spanish_surnames.txt

[1] https://www.cnmc.es

```
Callejon de Santa Eulalia (Madrid)
Calle de Santa Eva (Madrid)
Calle Santa Fe (Alcalá de Henares)
Calle de Santa Fe (Aranjuez)
Calle de Santa Fe (Madrid)
Calle Santa Fe (Pozuelo de Alarcón)
Calle de Santa Fe (San Martín de Valdeiglesias)
Calle de Santa Fe (Torrelaguna)
Calle de Santa Feliciana (Colmenar Viejo)
Calle de Santa Feliciana (Madrid)
...
```

Listing 3: Fichero spanish_streets.txt

```
Araba/Alava;Urkabustaiz;01449
Araba/Alava;Amurrio;01450
Araba/Alava;Zuia;01450
Araba/Alava;Urkabustaiz;01450
Albacete;Albacete;02001
Albacete;Albacete;02002
Albacete;Albacete;02099
Albacete;Tarazonade la Mancha;02100
Gipuzkoa;Lezo;02100
Albacete;Tarazona de la Mancha;02110
...
```

Listing 4: Fichero spanish_villages.txt

```
A33591611;R1-008;HIDROCANTÁBRICO DISTRIBUCIÓN ELÉCTRICA S.A.U
A17451733;R1-015;BASSOLS ENERGÍA, S.A
B64744642;R1-363;ELECTRA DEL LLOBREGAT ENERGÍA, S.L
B24291833;R1-239;SALTOS DEL CABRERA,S L
B24373367;R1-187;HIDROELECTRICA DEL CABRERA,S L
B17011404;R1-231;ELECTRICA CUROS, SL
B24014151;R1-149;HIJOS DE FELIPE GARCIA ALVAREZ SL
B25799172;R1-020;PEUSA DISTRIBUCIO SLU
B54862602;R1-033;Distribución Eléctrica Crevillent, S.L.U.
...
```

Listing 5: Fichero spanish_distributors.txt

```
R2-661;ZAIGLOBAL GESTIÓN, S.L.;CAMINO ZORROZGOITI 66, 2a PLANTA;
    48013;BILBAO;Vizcaya
R2-662;ACSOL ENERGÍA GLOBAL, S.A.;RAMBLA DEL GARRAF, 76;8812;
    SANT PERE DE RIBES;Barcelona
R2-663;ECONOMICLUZ, S.L.;C/ CAMPANAR Nº41 POL. IND. BOCH;3330;
    CREVILLENT;Alicante
R2-767;GAIA GLOBAL ENERGY SOCIEDAD LIMITADA;C/ DE LA REINA, 48;
    46800;XÀTIVA;VALENCIA
R2-768;TERUGAS ENERGY SL;PASEO DE LA CASTELLANA 141;28046;MADRID;
    MADRID
R2-769;LA CORRIENTE SOCIEDAD COOPERATIVA;CALLE EMBAJADORES
    Nº 41 LOCAL 3;28012;MADRID;MADRID
R2-770;INGEBAU SOLUCIONES DE MEDIDA SL;C/ BARTOLOMÉ DE MEDINA 1,
    2D5;41004;SEVILLA;SEVILLA
R2-771;OVO ENERGY SPAIN S.L.;CALLE MUNTANER 328 ENTRESUELO 1A;
    8021;BARCELONA;BARCELONA
R2-772;BULB ENERGIA IBERICA SL;PASEO DE LA CASTELLANA 43;28046;
    MADRID;MADRID
R2-773;GIROA S.A.;CAMINO PORTUETXE  53A OFICINA 201;20018;
    DONOSTIA - SAN SEBASTIÁN;GIPUZKOA
...
```

Listing 6: Fichero spanish_marketers.txt

3.4. Generación de un conjunto de entrenamiento

El proceso de simular una factura implica tener previamente una plantilla de factura anotada con etiquetas específicas donde se sustituirían los valores existentes por otros generados aleatoriamente en base a los archivos de simulación y el formato del campo. Esto resulta en un archivo JSON con las facturas y sus entidades anotadas.

```
 [('VX10C020-3-16/01/19 #eol N0017620ANNNN #eol ALFREDA GAMBOA GAVILÁN'
 '#eol Calle de Ladera de los Almendros (Madrid) #eol 14889 Luque '
 'Córdoba #eol ELECTRICA VINALESA, S.L.U Inscrita #eol en #eol'
 'el #eol Registro #eol Mercantil #eol de #eol Madrid. #eol Tomo '
 '#eol 12.797, #eol Libro #eol 0, #eol Folio #eol 208, #eol Sección'
...
{'entities': [ (44, 66, 'Nombre'),
   (72, 113, 'Direccion'),
   (119, 124, 'CP'),
   (125, 130, 'Poblacion'),
   (136, 143, 'Provincia'),
   (149, 174, 'Comercializadora'),
   (500, 525, 'Comercializadora'),
   (645, 657, 'NumeroFactura'),
   ...
   ]
}),
('Facturación #eol Producto: TARIFA GENERAL #eol Concepto Cálculos '
  'Importes(€) #eol FACT. ENER ENTRE REALES 495 KWH x 0,147879  '
  'EUR/KWH 73,20 (01) #eol ABONO CONSUMO ESTIMADO 243 KWH x '
  '0,149218 EUR/KWH -36,26 #eol Potencia 2.0 KW x 32 x 0,059817 '
  'EUR/KW Y DIA 40,38 #eol Ajuste precios 4º T 2011 1,58 #eol '
  'Ajuste precios 1º T 2012 11,70 #eol Impto. Electricidad 60,75 '
  'EUR x 1,05113 x 4,864 \% 6,20 #eol ALQUILER DE EQUIPOS ELECTR.'
  ' 0,97 #eol __________ #eol Subtotal 64,43 #eol IGIC REDUC 3 '
  '\% de 49,90 1,50 #eol IGIC REDUC 2 \% de 13,96 0,28 (*) #eol '
  'IGIC NORMA 7 \% de 0,57 0,04 #eol Total Factura: 315,72 € #eol'
  ' Consumo eléctrico #eol Desglose de consumos ')
  ...,
  {'entities': [(220, 223, 'PotenciaContratada'),
    (256, 261, 'ImportePotencia'),
    (388, 392, 'ImporteImpuestos'),
    (426, 430, 'ImporteAlquiler'),
    (593, 599, 'ImporteTotal'),
    (1476, 1482, 'ImporteTotal'),
    ...
    ]})]
```

Listing 7: Ejemplo de fichero de facturas generado a partir de datos simulados

Observando el fichero de facturas se puede apreciar el formato siguiente: el archivo es una lista de facturas y cada factura está compuesta por dos campos principales, el cuerpo y las entidades. El primer campo es el cuerpo, que representa toda la factura en texto y en una

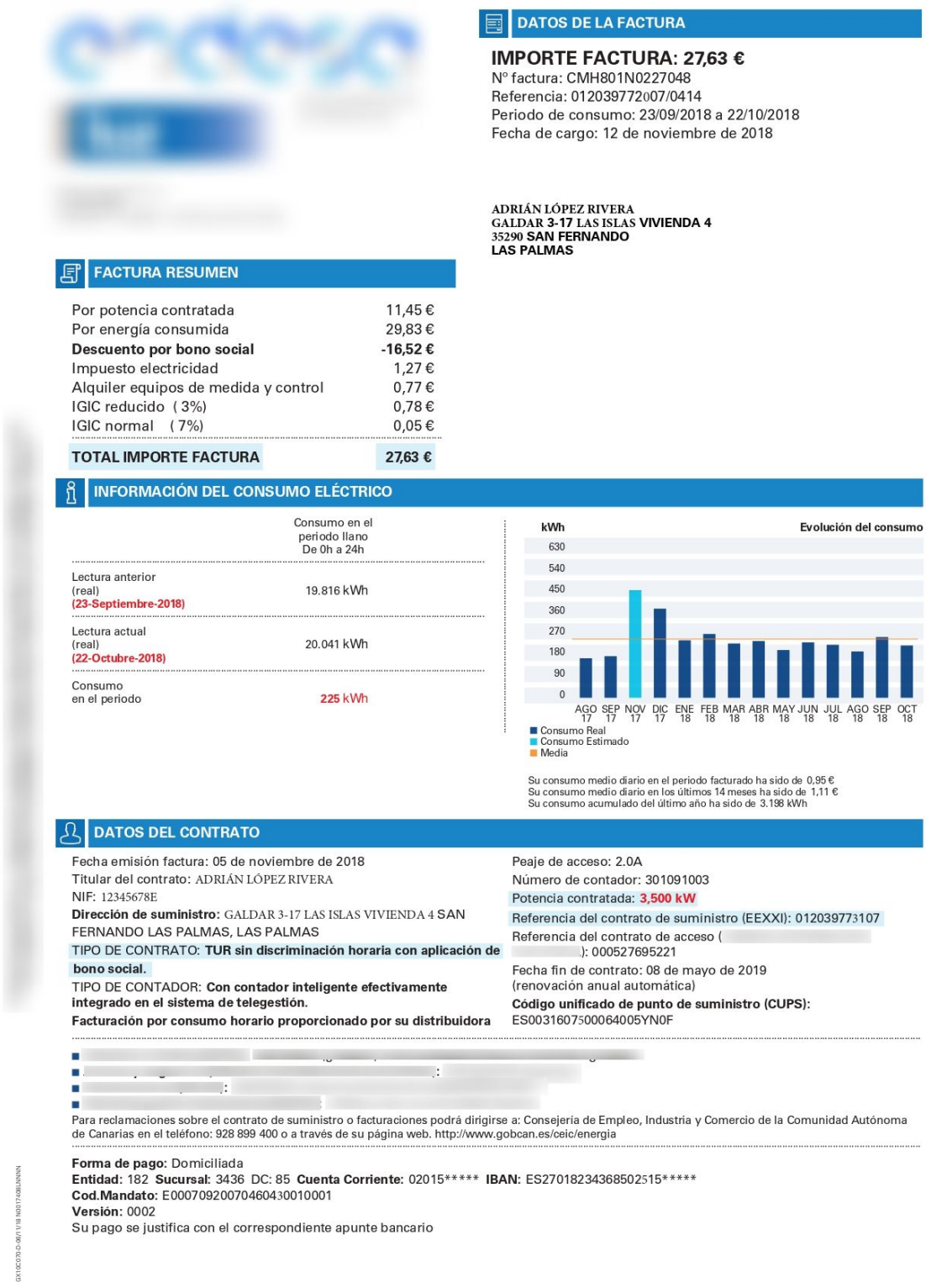
sola línea. El segundo es una lista de entidades, cada elemento de la lista representa una entidad en la factura con la posición de inicio y fin del texto correspondiente.

En base a este proceso de simulación de los datos, se procede a crear tres conjuntos: uno pequeño, otro de tamaño mediano y el último grande. Cada conjunto de datos contiene 10 plantillas y las variaciones de estas. El pequeño contiene 50 facturas por plantilla, el mediano 150 facturas por plantilla y el grande 500 facturas por plantilla. Todas estas facturas están anotadas con sus respectivas identidades. Durante el proceso de entrenamiento solo se llegó a utilizar el conjunto pequeño y mediano, ya que como se verá en el capítulo 5 no se tenía el tiempo suficiente ni aumentar el número de muestras era necesario. En el proyecto las anotaciones se generaron durante el proceso de simulación, sin embargo existen soluciones software para dicho fin, por ejemplo, Doccano[2]. Doccano es una herramienta de software libre que facilita la tarea de etiquetar texto, ya sea para clasificación de texto como etiquetar secuencias.

Uno de los aspecto más importantes en el momento de entrenar un sistema para detectar patrones, es tener un buen conjunto de datos, ya que si se alimenta al modelo con una mala entrada, muy probablemente la salida sea deficiente. Así que a continuación se mostrará una factura simulada y sus entidades en formato JSON.

Prosiguiendo con el proceso de generar el conjunto de entrenamiento, se mostrará un ejemplo de factura y su transformación a lo largo del proceso haciendo uso de Doccano.

[2]Repositorio Doccano: `https://github.com/doccano/doccano`



DATOS DE LA FACTURA

IMPORTE FACTURA: 27,63 €
Nº factura: CMH801N0227048
Referencia: 012039772007/0414
Periodo de consumo: 23/09/2018 a 22/10/2018
Fecha de cargo: 12 de noviembre de 2018

ADRIÁN LÓPEZ RIVERA
GALDAR 3-17 LAS ISLAS VIVIENDA 4
35290 SAN FERNANDO
LAS PALMAS

FACTURA RESUMEN

Por potencia contratada	11,45 €
Por energía consumida	29,83 €
Descuento por bono social	**-16,52 €**
Impuesto electricidad	1,27 €
Alquiler equipos de medida y control	0,77 €
IGIC reducido (3%)	0,78 €
IGIC normal (7%)	0,05 €
TOTAL IMPORTE FACTURA	**27,63 €**

INFORMACIÓN DEL CONSUMO ELÉCTRICO

	Consumo en el periodo llano De 0h a 24h
Lectura anterior (real) (23-Septiembre-2018)	19.816 kWh
Lectura actual (real) (22-Octubre-2018)	20.041 kWh
Consumo en el periodo	225 kWh

Su consumo medio diario en el periodo facturado ha sido de 0,95 €
Su consumo medio diario en los últimos 14 meses ha sido de 1,11 €
Su consumo acumulado del último año ha sido de 3.198 kWh

DATOS DEL CONTRATO

Fecha emisión factura: 05 de noviembre de 2018
Titular del contrato: ADRIÁN LÓPEZ RIVERA
NIF: 12345678E
Dirección de suministro: GALDAR 3-17 LAS ISLAS VIVIENDA 4 SAN FERNANDO LAS PALMAS, LAS PALMAS
TIPO DE CONTRATO: **TUR sin discriminación horaria con aplicación de bono social.**
TIPO DE CONTADOR: **Con contador inteligente efectivamente integrado en el sistema de telegestión.**
Facturación por consumo horario proporcionado por su distribuidora

Peaje de acceso: 2.0A
Número de contador: 301091003
Potencia contratada: **3,500 kW**
Referencia del contrato de suministro (EEXXI): 012039773107
Referencia del contrato de acceso ( ): 000527695221
Fecha fin de contrato: 08 de mayo de 2019 (renovación anual automática)
Código unificado de punto de suministro (CUPS): ES0031607500064005YN0F

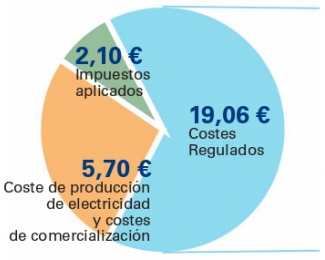
Para reclamaciones sobre el contrato de suministro o facturaciones podrá dirigirse a: Consejería de Empleo, Industria y Comercio de la Comunidad Autónoma de Canarias en el teléfono: 928 899 400 o a través de su página web. http://www.gobcan.es/ceic/energia

Forma de pago: Domiciliada
Entidad: 182 **Sucursal:** 3436 DC: 85 **Cuenta Corriente:** 02015***** **IBAN:** ES2701823436850251*****
Cod.Mandato: E000709200704604300010001
Versión: 0002
Su pago se justifica con el correspondiente apunte bancario

Figura 3.3: Plantilla de factura sin etiquetar - Página 1.

DESTINO DEL IMPORTE DE LA FACTURA

El destino del importe de su factura, **27,63 euros,** es el siguiente:



7,55 €	**Incentivos a las energías renovables, cogeneración y residuos**
7,62 €	**Coste de redes de transporte y distribución**
3,89 €	**Otros costes regulados** (incluida la anualidad del déficit)

A los importes indicados en el diagrama debe añadirse, en su caso, el importe del alquiler de los equipos de medida y control así como los conceptos no energéticos.

DETALLE DE LA FACTURA

Facturación por potencia contratada: Comprende dos conceptos: la facturación por peaje de acceso (resultado de multiplicar los kW contratados por el precio del término de potencia del peaje de acceso y el número de días del periodo de facturación) y la facturación por margen de comercialización fijo.

Importe por peaje de acceso:		
3,5 kW x 38,043426 Eur/kW y año x (29/365) días	10,58 €	
Importe del término fijo de los costes de comercialización:		
3,5 kW x 3,113 Eur/kW y año x (29/365) días	0,87 €	
		11,45 €

Facturación por energía consumida: Comprende dos conceptos: la facturación por peaje de acceso (resultado de multiplicar los kWh consumidos en el periodo de facturación por el precio del término de energía del peaje de acceso) y la facturación por coste de la energía (resultado de multiplicar los kWh consumidos por el precio del término del coste horario de energía del PVPC).

Importe por peaje de acceso:		
224,686 kWh x 0,044027 Eur/kWh	9,90 €	
Importe por coste de la energía (*):		
93,29 kWh x 0,092662 Eur/kWh (**)	8,64 €	
16,982 kWh x 0,085913 Eur/kWh (**)	1,46 €	
114,414 kWh x 0,085901 Eur/kWh (**)	9,83 €	
		29,83 €

Aplicación del Bono Social: A la facturación por potencia contratada y por **224,686 kWh** del total de la energía consumida, se le aplica un **40%** de descuento por Bono Social.

Descuento Potencia 11,45 Eur x 40 %	-4,59 €	
Descuento Energía [1] 29,83 Eur x 40 %	-11,93 €	
		-16,52 €

[1] Descuento Energía: El descuento por Bono Social se aplica al total del importe de "facturación por energía consumida", porque su consumo en esta factura no excede el límite con derecho a Bono Social:

Consumo total de su factura: 224,686 kWh
Consumo de su factura con descuento por Bono Social: 224,686 kWh (Límite de consumo con derecho a Bono Social según se establece en el RD 897/2017 y RDL 15/2018).
Consumo de su factura sin descuento por Bono Social: 0 kWh

Subtotal	**24,76 €**

Impuesto de electricidad: Impuesto especial al tipo del 5,11269632% sobre el producto de la facturación de la electricidad suministrada

Impuesto electricidad (24,76 X 5,11269632 %)	1,27 €

Alquiler de equipos de medida y control. Precio establecido que se paga por el alquiler de equipos de medida y control.

Alquiler equipos de medida y control (29 días x 0,026552 Eur/día)	0,77 €
Subtotal otros conceptos	**2,04 €**

Importe total	26,80 €	
IGIC: Impuesto General Indirecto Canario al tipo del 3% 7%		
IGIC reducido (3%) 3% s/ 26,03		0,78 €
IGIC normal (7%) 7% s/ 0,77		0,05 €

TOTAL IMPORTE FACTURA	**27,63 €**

El importe de su factura a PVPC previo a la aplicación del descuento por Bono Social, asciende a 45,51 €. De acuerdo a lo establecido en el artículo 12 del RD 897/2017, de 6 de Octubre, para los consumidores vulnerables severos en riesgo de exclusión social, el importe mínimo a financiar a efectos de lo establecido en el punto 1 del Artículo 12 del RD 897/2017, sería de 22,75 €.

Precios de los términos del peaje de acceso publicados en Orden ETU 1282/2017
PVPC calculado según Real Decreto RD 216/2014
Margen de comercialización fijo publicado en RD 469/2016. Orden ETU 1948/2016
Descuento del bono social regulado en RD 897/2017 y RDL 15/2018.
Precio del alquiler de los equipos de medida y control en Orden IET 1491/2013 de 3 de agosto

Figura 3.4: Plantilla de factura sin etiquetar - Página 2.

Vista la factura, el siguiente paso es convertir el formato de la factura, ya sea PDF o imagen, a texto. Para dicha tarea se utilizan principalmente herramientas de reconocimiento óptico de caracteres, más conocidas en inglés con el acrónimo OCR. Esta tarea de reconocer caracteres es un proceso dirigido a la digitalización de textos, los cuales identifican símbolos o caracteres que pertenecen a un determinado alfabeto, para luego almacenarlos en archivos de texto. Ejemplos de herramientas OCR utilizadas en el proyecto, Tesseract-OCR [23] o pdftotext. Tras el procesado del archivo, el texto resultante sería el siguiente.

DATOS DE LA FACTURA
IMPORTE FACTURA: 27,63 € Nº factura: CMH801N0227048 Referencia: 012039772007/0414 Periodo de consumo: 23/09/2018 a 22/10/2018 Fecha de cargo: 12 de noviembre de 2018
Endesa Energía XXI S.L.U. Cif: B82846825 C/Albareda nº 38 35008 - Las Palmas de Gran Canaria
ADRIÁN LÓPEZ RIVERA GALDAR 3-17 LAS ISLAS VIVIENDA 4 35290 SAN FERNANDO LAS PALMAS
Por potencia contratada Por energía consumida Descuento por bono social Impuesto electricidad Alquiler equipos de medida y control IGIC reducido (3 %) IGIC normal (7 %)
11,45 € 29,83 € -16,52 € 1,27 € 0,77 € 0,78 € 0,05 €
.....
TOTAL IMPORTE FACTURA
27,63 €
INFORMACIÓN DEL CONSUMO ELÉCTRICO Consumo en el periodo llano De 0h a 24h
.....
Lectura anterior (real) (23-Septiembre-2018)
19.816 kWh
Lectura actual (real) (22-Octubre-2018)
20.041 kWh
.....
Endesa Energía XXI, S.L. Unipersonal. Inscrita en el Registro Mercantil de Madrid. Tomo 160.957, Libro 0, Folio 121, Sección 8ª, Hoja 272.593, CIF B-82846825. Domicilio Social: C/Ribera del Loira, nº60 28042 - Madrid.
. . . .

En este fragmento, se puede observar como hay varios caracteres que no aportan valor. Por lo tanto, se procede a la limpieza del texto y agrupar todo el texto en una sola línea. El motivo por el que se hace eso, se explicará en detalle más adelante, pero se llegó a la conclusión de que entrenando el modelo con una factura entera y sin separar por líneas, aportaba contexto a las entidades que se buscan reconocer.

El primer paso que se sigue para limpiar el texto, es eliminar todos los puntos y sustituyendo los saltos de línea por etiquetas como ' #eol '. El texto quedaría tal que así.

DATOS DE LA FACTURA #eol #eol IMPORTE FACTURA: 27,63 € #eol Nº factura: CMH801N0227048 #eol Referencia: 012039772007/0414 #eol Periodo de consumo: 23/09/2018 a 22/10/2018 #eol Fecha de cargo: 12 de noviembre de 2018 #eol #eol Endesa Energía XXI S.L.U. #eol Cif: B82846825 #eol C/Albareda nº 38 35008 - Las Palmas de Gran Canaria #eol #eol ADRIÁN LÓPEZ RIVERA #eol GALDAR 3-17 LAS ISLAS VIVIENDA 4 #eol 35290 SAN FERNANDO #eol LAS PALMAS #eol #eol Por potencia contratada #eol Por energía consumida #eol Descuento por bono social #eol Impuesto electricidad #eol Alquiler equipos de medida y control #eol IGIC reducido (3 %) #eol IGIC normal (7 %) #eol #eol 11,45 € #eol 29,83 € #eol -16,52 € #eol 1,27 € #eol 0,77 € #eol 0,78 € #eol 0,05 € #eol #eol #eol #eol TOTAL IMPORTE FACTURA #eol #eol 27,63 € #eol #eol INFORMACIÓN DEL CONSUMO ELÉCTRICO #eol Consumo en el #eol periodo llano #eol De 0h a 24h #eol #eol . #eol #eol Lectura anterior #eol (real) #eol (23-Septiembre-2018) #eol #eol 19.816 kWh #eol #eol Lectura actual #eol (real) #eol (22-Octubre-2018) #eol #eol 20.041 kWh #eol #eol #eol . #eol #eol Endesa Energía XXI, S.L. Unipersonal. Inscrita en el Registro Mercantil de Madrid. Tomo 160.957, Libro 0, Folio #eol 121, Sección 8ª, Hoja 272.593, CIF B-82846825. Domicilio Social: C/Ribera del Loira, nº60 28042 - Madrid. #eol #eol . #eol . #eol . #eol .

El siguiente paso es anotar las entidades, ya que la idea es que se reconozcan las entidades sobre dicho texto. Una de las herramientas para anotar las facturas es Doccano. A continuación se muestra un ejemplo de una factura etiquetada y su formato tras exportar los datos.

Figura 3.5: Factura etiquetada. Fuente: Hecho con la herramienta Doccano

Tener la factura en este formato resulta ideal ya que es muy similar al formato que es requerido posteriormente para el entrenamiento con spaCy, y que se utilizará más adelante para entrenar el NER (Named Entity Recognition), en español reconocimiento de entidades nombradas. El formato requerido por spaCy es el siguiente:

```
train_data = [
        ("Text to use in training process using spaCy",
            {
                "entities": [ (38, 42, "Tool"), ...]
            }
        ), ( . . . . . ), ( . . . . . )
]
```

Listing 9: Formato del conjunto de datos para entrenar con spaCy

```
{"id": 51,
"text": ('DATOS DE LA FACTURA  IMPORTE FACTURA: 27,63 € Nº factura:'
        'CMH801N0227048 Referencia: 012039772007/0414 Periodo de '
        'consumo: 23/09/2018 a 22/10/2018 Fecha de cargo: 12 de '
        'noviembre de 2018  Endesa Energía XXI S.L.U. Cif: B82846825'
        ' C/Albareda nº 38 35008 - Las Palmas de Gran Canaria  '
        'ADRIÁN LÓPEZ RIVERA GALDAR 3-17 LAS ISLAS VIVIENDA 4 35290'
        'SAN FERNANDO LAS PALMAS  Por potencia contratada Por energía'
        'consumida Descuento por bono social Impuesto electricidad '
        'Alquiler equipos de medida y control IGIC reducido ( 3\%)'
        'IGIC normal ( 7\%)  11,45 € 29,83 € -16,52 € 1,27 € 0,77 € '
        '0,78 € 0,05 €  .  TOTAL IMPORTE FACTURA  27,63 €  INFORMACIÓN'
        'DEL CONSUMO ELÉCTRICO Consumo en el periodo llano De 0h a 24h'
        '.  Lectura anterior (real) (23-Septiembre-2018)  19.816 kWh'
        ' Lectura actual (real) (22-Octubre-2018)  20.041 kWh  .  .  '
        'Consumo en el periodo  225 kWh    Endesa Energía XXI, S.L. '
        'Unipersonal. Inscrita en el Registro Mercantil de Madrid.'
        ' Tomo 160.957, Libro 0, Folio 121, Sección 8ª, Hoja 272.593,'
        ' CIF B-82846825. Domicilio Social: C/Ribera del Loira, '
        'nº60 28042 - Madrid,'),
"meta": {},
"annotation_approver": "admin",
"labels": [
    [38,43,"ImporteTotal"],
    [58,72,"NumeroFactura"],
    [123,133,"FechaInicio"],
    [136,146,"FechaFin"],
    [163,186,"FechaEmision"],
    [282,301,"Nombre"],
    [536,541,"ImportePotencia"],
    [544,549,"ImporteEnergia"],
    [561,565,"ImporteImpuestos"],
    [568,572,"ImporteAlquiler"],
    [575,579,"ImporteImpuestos"],
    [582,586,"ImporteImpuestos"],
    [616,621,"ImporteTotal"],
    [749,755,"ConsumoEnergia"],
    [85,102,"NumeroContrato"],
    [302,334,"Direccion"],
    [335,340,"CP"],
    [341,364,"Poblacion"]
]}
```

Listing 8: JSON generado por Doccano

Capítulo 4

Método para la extracción de información

Antes de continuar con el proceso de extracción de información, es necesario introducir un concepto sobre el que se centra el proyecto, el reconocimiento de entidades nombradas, o más conocido por su término anglosajón *Named Entity Recognition* (NER). El reconocimiento de entidades nombradas es la tarea de identificar y categorizar información clave en el texto que se realiza mediante entidades. Una entidad puede ser una palabra o una serie de palabras que hagan referencia al mismo concepto. Cada entidad detectada se clasifica en una categoría determinada. Por ejemplo, el modelo de NER podría detectar 'Google' en un texto y clasificarlo como 'ORG', es decir, organización, y lo mismo pasaría con palabras como 'Microsoft' o 'Amazon'. El reconocimiento de entidades nombras está compuesto por dos procesos principales:

1. Detectar la entidad. Implica detectar una palabra o serie de palabras y cada una de las palabras es representada con un token.
2. Categorizar la entidad. Creación de las categorías de entidades.

Una vez definida la lista de entidades, se puede proceder a etiquetar el conjunto de datos. En este proyecto las entidades a reconocer son las mostradas en la tabla 4.1.

Cuadro 4.1: Entidades y descripciones

Entidad	**Descripción**
CP	Código postal.
CUPS	Código Universal de Punto de Suministro.
Comercializadora	Compañía que ofrece el servicio de contratar energía eléctrica. No se responsabiliza del suministro.
ConsumoEnergia	Cantidad de energía utilizada.
Direccion	Dirección de facturación.
Distribuidora	Compañía encargada de suministrar correctamente la energía eléctrica a los domicilios o locales.
FechaCargo	Fecha en la que se realiza el cobro.
FechaEmision	Fecha en la que se emitió la factura.
FechaInicio	Fecha en la que empieza a contabilizar el consumo.
FechaFin	Fecha en la que se termina de contabilizar el consumo.
FinContrato	Fecha en la que finaliza el contrato.
ImporteAlquiler	Importe del alquiler del contador.
ImporteEnergia	Importe de la energía consumida.
ImporteImpuestos	Importe de los impuestos correspondientes al consumo.
ImportePotencia	Importe de la potencia contratada.
ImporteTotal	Importe total de la factura.
NIF	Número de identificación fiscal.
Nombre	Nombres y apellidos del contratante.
NumeroContrato	Identificador del contrato.
NumeroFactura	Identificador de la factura.
PeajeAcceso	Tarifa impuesta por el gobierno que garantiza la infraestructura y su mantenimiento para la distribución de energía. Depende directamente del consumo y la potencia contratada.
Poblacion	Ciudad o núcleo rural.
PotenciaContratada	Unidad que limita el número de aparatos eléctricos que podemos conectar a la vez a la red eléctrica.
Provincia	División administrativa territorial.

4.1. Librería spaCy

La herramienta spaCy [7] es una librería de código abierto para el procesamiento de lenguaje natural desarrollada en Python y Cython. Está diseñada específicamente para su uso en producción y facilita la creación de aplicaciones para procesar y 'comprender' grandes volúmenes de texto. Es principalmente utilizada para construir sistemas de extracción de información, comprensión del lenguaje natural o preprocesamiento de texto para aprendizaje

profundo [6].

Actualmente no existe ningún artículo oficial sobre el desarrollo de la herramienta, ya que el desarrollo de la arquitectura es privado. Aunque el desarrollador principal, Matthew Honnibal, explicó que spaCy no se desarrolló con un objetivo académico, afirmó que el enfoque de la librería estaba basado en un artículo[1] publicado por Emma Strubell, con título *Fast and Accurate Entity Recognition with Iterated Dilated Convolutions* [24]. Existen diferencias, relacionadas con el uso de un método de *embeddings* distinto y que en vez de usar una red neuronal convolutiva dilatada (DilatedNet), como se indica en el artículo, en spaCy utilizan una red neuronal residual (ResNet).

```
TRAIN_DATA = [
        ("Uber blew through $1 million a week",
            {"entities": [ (0, 4, "ORG")]}),
        ("Google rebrands its business apps",
            {"entities": [ (0, 6, "ORG")]})
        ]

nlp = spacy.blank("en")
optimizer = nlp.begin_training()
for i in range (20):
    random.shuffle(TRAIN_DATA)
    for text, annotations in TRAIN_DATA:
        nlp.update([text], [annotations], sgd=optimizer)
nlp.to_disk("/model")
```

Listing 10: Ejemplo de entrenamiento de un modelo. Fuente: spaCy

Como se ha mencionado anteriormente, la herramienta principal a utilizar para realizar el proceso de extracción de información es spaCy. Sin embargo los modelos preentrenados ofrecidos por spaCy no sirven ya que no están entrenados con las entidades que se buscan reconocer, por lo que se decidió crear un modelo desde cero y entrenar con las entidades mencionadas en la tabla 4.1.

4.2. Arquitectura de spaCy

El reconocimiento de entidades nombradas haciendo uso de spaCy, ofrece una respuesta rápida ya que se basa en sistemas estadísticos donde se busca reconocer grupos de tokens

[1] https://github.com/explosion/spaCy/issues/2107

contiguos como entidades. Bajo la nomenclatura de spaCy, Token es la representación de una palabra, símbolo u otro carácter, y grupo de tokens se conoce como «Span». La estructura de datos principal de spaCy son los Doc y Vocab. El objeto Doc contiene una secuencia de tokens con todas sus anotaciones y el Vocab contiene una serie de tablas de toda la información común entre los documentos. Las anotaciones están diseñadas para que solo permitan una sola fuente: el Doc tiene el contenido, y tanto Span como Token son formas distintas de visualizar el Doc. Un componente muy importante dentro del pipeline de spaCy y el encargado de convertir un texto a un objeto Doc, es el Tokenizer. Este componente es el primero en ejecutarse en el pipeline de procesos y se encarga de convertir en tokens cada una de las palabras contenidas en el texto pasado. Por lo tanto, el objeto Doc se construye a raíz del Tokenizer y posteriormente modificado o no por los distintos procesos integrados en el pipeline. El pipeline por defecto incluye tres procesos principales, aunque también se pueden agregar procesos hechos a medida, y son los siguientes;

- Tagger, encargado de asignar etiquetas gramaticales al texto, tal que nombre, verbo, adjetivo, etc.
- DependencyParser, define las relaciones y dependencias entre distintas etiquetas.
- EntityRecognizer, encargado de detectar y etiquetar las entidades nombradas.

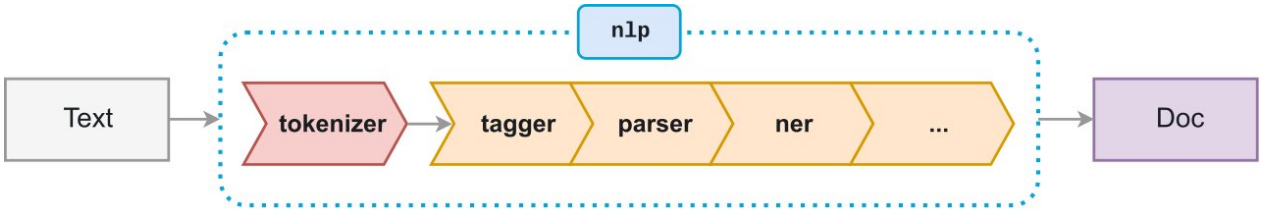


Figura 4.1: Tokenizer de spaCy. Fuente: spacy.io

Como se muestra en la figura 4.2, la clase Language es la que coordina todos los componentes. Esta clase es creada cuando se carga un modelo y contiene el vocabulario compartido y los procesos a ser ejecutados en orden. Por lo tanto, toma texto sin formato y lo envía a través del pipeline de procesos, devolviendo un documento anotado. También organiza el entrenamiento y la serialización del modelo.

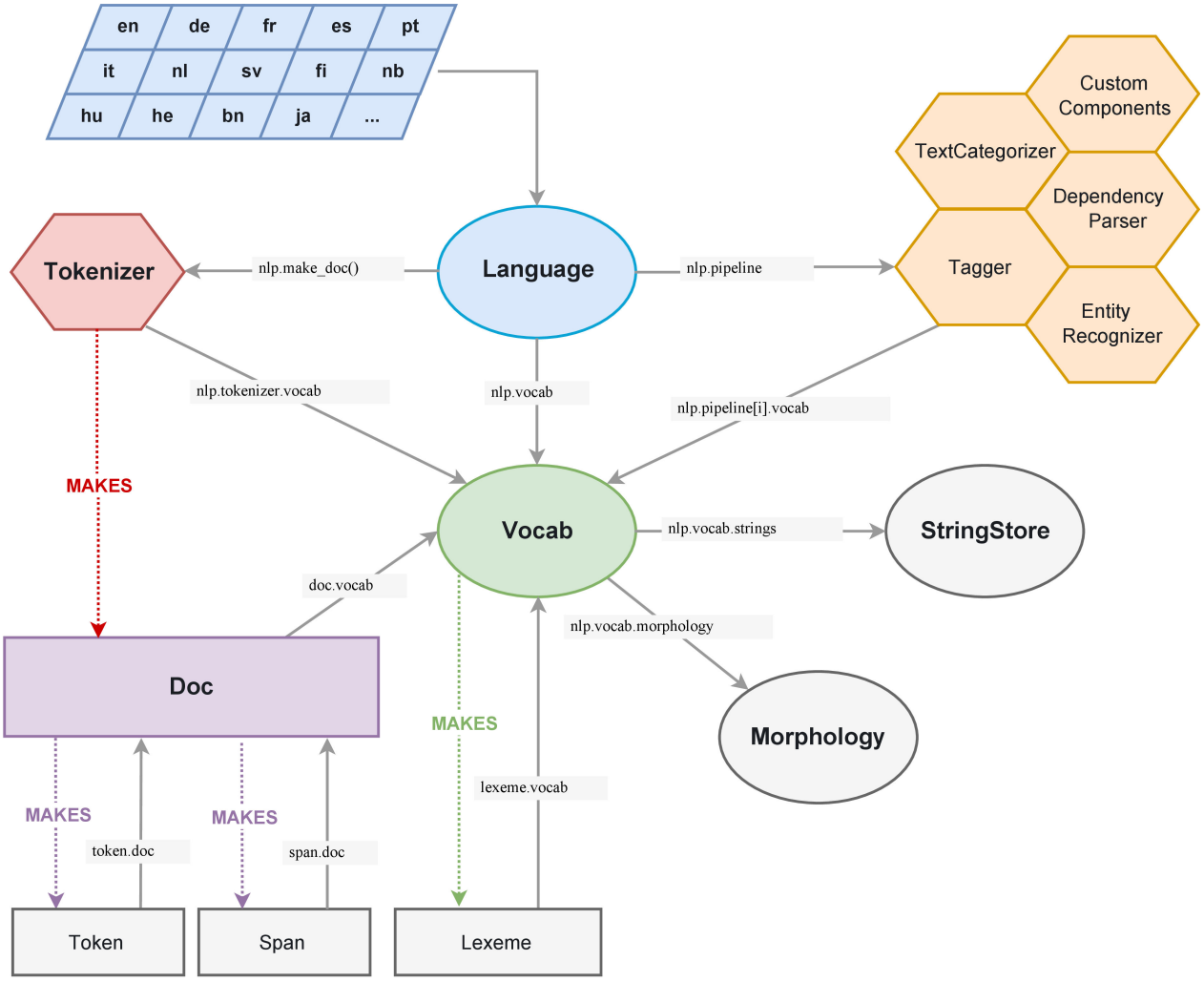


Figura 4.2: Arquitectura de spaCy. Fuente: spacy.io

spaCy tiene y utiliza su propia librería de aprendizaje profundo conocida como Thinc. Esta librería se creo con el objetivo de acelerar el proceso de crear modelos y aportar flexibilidad en la implementación de nuevos modelos. Soporta distintos *frameworks* como PyTorch, TensorFlow y MXNet.

4.3. Entrenamiento y extracción de información

Para hacer uso de las propiedades de spaCy se requiere descargar e importarlo al proyecto. A continuación, se crea un modelo vacío y se declara el idioma con el que será entrenado el NER, en este caso es el español. Luego se agrega al pipeline principal el NER, puesto que el objetivo es entrenarlo con nuestras entidades y por último deshabilitados cualquier tarea del pipeline que no sea nuestro NER. Cargamos las entidades que deseamos reconocer y el

conjunto de datos anotado tanto para el entrenamiento como la validación. Por último se inicia el entrenamiento.

```
other_pipes = [pipe for pipe in nlp.pipe_names if pipe != 'ner']
with nlp.disable_pipes(*other_pipes):  # only train NER
    for itn in range(n_iter):
        random.shuffle(TRAIN_DATA)
        losses = {}
        batches = minibatch(TRAIN_DATA,
                            size=compounding(4., 32., 1.001))
        for batch in batches:
            texts, annotations = zip(*batch)
            # Updating the weights
            nlp.update(texts, annotations, sgd=optimizer,
                       drop=0.5, losses=losses)
        print('Losses', losses)
```

Listing 11: Entrenar un NER personalizado

Guardamos el modelo en disco y lo probamos para revisar si reconoce las entidades correctamente. A continuación se muestra un ejemplo de una factura preprocesada, la cual se usará como prueba para detectar las entidades.

```
import spacy
import json
from spacy import displacy

output = []
def getNER(text=None):
    nlp = spacy.load("./models/spacy_ner")
    doc = nlp(text)
    displacy.render(doc, style="ent",jupyter=True)

def cleanInv(path=None):
    bill_lines = open(path,"r",encoding="utf-8").readlines()
    bill_lines = ''.join(bill_lines)
    bill_lines = bill_lines.replace("..", "")
    bill_lines = bill_lines.replace("\n", " ")
    return bill_lines

text = cleanInv("./factura.txt")

getNER(text)
```

Listing 12: Prueba del NER post entrenamiento

Como se puede observar en el código, primero nos aseguramos de eliminar ciertos caracteres de la factura y después procedemos a detectar las entidades y mostrar el resultado.

Figura 4.3: Factura con las entidades detectadas.

Como se puede observar en la ilustración 4.3, una gran parte de las entidades detectadas resultan ser correctas, a excepción de algunas, como por ejemplo, «COMERCIALIZADORA» y «DISTRIBUIDORA», el resultado es aceptable, pero mejorable. Cabe recalcar que aunque los valores de la factura son distintos del conjunto de datos, la plantilla es similar en algunos campos, ya que comparten similitudes entre distintas facturas sobre la disposición de la información. Esto puede ser bueno o no, ya que puede derivar en varios problemas si no se logra generalizar la información. Y como último paso generamos el JSON con las entidades reconocidas.

```
{
    "CP": "16708",
    "CUPS": "ES8452530888875446LESY",
    "Comercializadora": "ELÉCTRICA VAQUER
                         ENERGIA, S.A  Inscrita",
    "ConsumoEnergia": "816",
    "Direccion": "Calle de Jorge Manrique (Robledo de Chavela)",
    "Distribuidora": "EMPRESA MUNICIPAL DE DISTRIBUCIÓ
                      DENERGIA ELÉCTRICA DE PONTS, SL
                      Actividad económica (CNAE): 9820 Viviendas
                      Particulares. Primera Vivienda  Fecha final",
    "FechaCargo": "",
    "FechaEmision": "24/08/2015",
    "FechaFin": "22/08/2015",
    "FechaInicio": "23/06/2015",
    "FinContrato": "22/08/2015",
    "ImporteAlquiler": "0,50",
    "ImporteEnergia": "107,27",
    "ImporteImpuestos": "9,99",
    "ImportePotencia": "28,89",
    "ImporteTotal": "152,67",
    "NIF": "36548959U",
    "Nombre": "NAZARIO MENA HUERTA",
    "NumeroContrato": "4088757003928",
    "NumeroFactura": "FI1015412605",
    "PeajeAcceso": "20DHA",
    "Poblacion": "Casas de Guijarro",
    "PotenciaContratada": "4.0",
    "Provincia": "Cuenca"
}
```


Listing 13: Prueba del NER post entrenamiento

4.4. Optimización y métricas

Al final del proyecto se decidió entrenar el modelo de dos formas distintas: la primera fue entrenar un modelo desde cero, donde el conjunto de entrenamiento se mezclaba y se separaba una porción para la validación del mismo; en el segundo, el entrenamiento se hace separando el conjunto de datos en varios subgrupos, donde en cada subgrupo tenía un conjunto de entrenamiento y otro de validación. El objetivo de realizar el entrenamiento de estas dos maneras era para evaluar cómo aprendía el modelo cuando se le alimentaba con facturas

similares pero con datos distintos, en el capítulo chapter 5 se entrará en más detalle.

Durante el desarrollo y entrenamiento del modelo, existe la necesidad de visualizar distintas métricas para poder evaluar el avance y el aprendizaje durante el entrenamiento. Sin embargo, spaCy solo ofrece una métrica que es el cálculo del error del NER usando la función de pérdida logarítmica, en inglés *log loss*. Esta función mide la incertidumbre que ha tenido nuestro modelo frente a la etiqueta real, y se penalizas las clasificaciones falsas. Lo que se consigue minimizando la pérdida logarítmica es equivalente a maximizar la precisión del modelo. Por lo tanto, es preferible que esta métrica tienda a 0, ya que a más cerca se encuentre de 0, mayor es la precisión.

La pérdida logarítmica ofrecida por spaCy resulta insuficiente para poder obtener conclusiones del estado del modelo en el proceso de entrenamiento, ya que puede existir un sobreajuste o subajuste y no verse reflejado en la métrica. Por lo tanto se procedió a desarrollar métricas propias para el cálculo de la precisión y el error tanto en el conjunto de entrenamiento, como en el conjunto de validación.

La función utilizada para calcular la precisión y el error del modelo ha sido el Índice Jaccard (Eq. 4.1) o Coeficiente Jaccard [9]. Se realizó una implementación propia de la función para calcular la similitud entre el campo detectado y el que correspondería. Cuanto más cerca esté el coeficiente de ser 1, esto significa que la similitud entre los dos campos es alta, y a menor sea el coeficiente, menor es la similitud.

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{4.1}$$

Este cálculo se realiza cada vez que finaliza una época, evaluando tanto con el conjunto de entrenamiento como el de validación. Se adjunta un ejemplo en código de la implementación de la función Jaccard utilizado para el cálculo de la precisión y el error.

```python
# Calculate the accuracy
def jaccard(str1, str2):
    a = set(str1.lower().split(" "))
    b = set(str2.lower().split(" "))
    c = a.intersection(b)
    if len(a) + len(b) == len(c):
        return 0.0
    return float(len(c)) / (len(a) + len(b) - len(c))

# Calculate the loss
def jaccard_loss(str1, str2, smooth=100):
    a = set(str1.lower().split(" "))
    b = set(str2.lower().split(" "))
    c = a.intersection(b)
    jac = (float(len(c)) + smooth) / (len(a) + len(b) - len(c) + smooth)
    return (1 - jac) * smooth
```

Listing 14: Implementación de la función Jaccard

Se optó principalmente por una implementación propia de dicha función, porque se tenía mas control sobre cómo se realizan las comparaciones con cada una de las palabras, ya que lo que se requería era comparar cada una de las palabras reconocidas como una entidad. Esto puede resultar muy útil porque si se reconocen la mayoría de las palabras de una entidad y no se reconociera alguna palabra o agregara alguna de más, esto no penalizaría tanto la métrica de error.

Capítulo 5

Resultados experimentales

A lo largo del capítulo se expondrán distintas métricas y resultados tras el entrenamiento del modelo. Gracias a las métricas podemos dotar de cierto sentido qué tal está aprendiendo el modelo. Cabe destacar que el tiempo promedio de entrenamiento para 100 épocas es de 3.500 segundos aproximadamente para el conjunto de datos pequeño, para el mediano es de aproximadamente 10 horas y el grande tardaría varios días, aunque no se llegó a probar. Se trabajó principalmente con el conjunto pequeño ya que no se disponía de tiempo suficiente para entrenar con conjuntos de datos mayores, aunque los resultados obtenidos son válidos para obtener ciertas conclusiones interesantes. Los parámetros principales del optimizador Adam y del objeto de entrenamiento con los que fue entrenado el modelo son los siguientes: *batch_size* progresivo de 1 a 32, *dropout* de 0.6, ratio de aprendizaje del 0.001, regularización L2 en 0.01 con decaimiento. A excepción del *dropout* y el *batch_size*, el resto de los parámetros no se pueden modificar directamente, hubo que importar el optimizador directamente desde la librería de THINC [1] y sustituir los valores por defecto del optimizador utilizado en spaCy por los valores propios. Se optó por aumentar el valor de la regularización L2, también conocido como *Ridge*, para que los pesos no sean tan bajos respecto al valor original.

En un principio el modelo se entrenaba con todo el conjunto de datos mezclado y se separaba el 20 % de las muestras para usarlas en validación. Por lo que durante el entrenamiento siempre se ha observado que el modelo durante las primeras épocas ya empezaba a converger en épocas muy tempranas, aparentemente puede existir un sobreajuste puesto que no es normal ese comportamiento y da la sensación de que el modelo memorizó las facturas. Entonces se decidió separar el conjunto de datos en varios subconjuntos y separar del conjunto de entrenamiento aquellas plantillas que eran distintas del resto. Con esto obtenemos varias gráficas del comportamiento del modelo con conjuntos de validación distintos y asegurar que

[1] `THINC:https://thinc.ai/`

tan independiente es el modelo del formato de una factura.

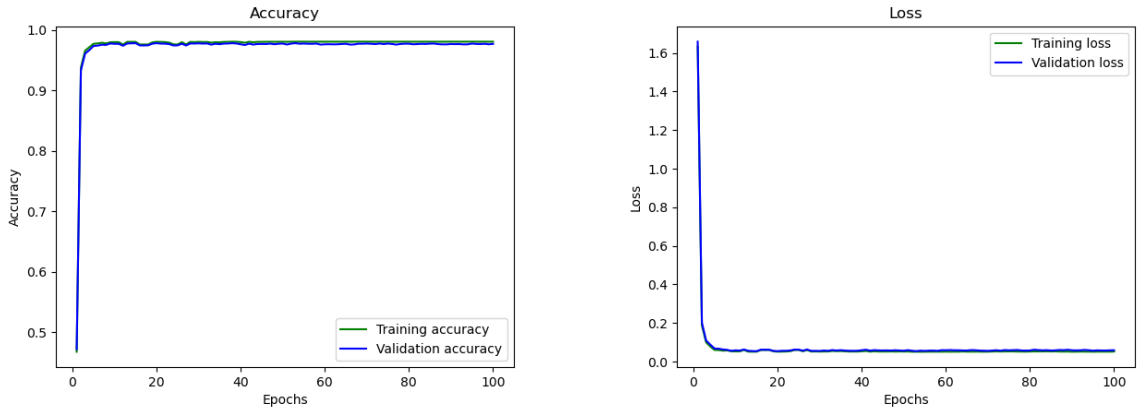


Figura 5.1: Gráficas que muestran el progreso de la precisión y la pérdida. Las líneas se encuentran solapadas.

Como conclusión de esta primera prueba como se observa en la figura 5.1, el modelo logra aprender de las facturas utilizadas para entrenar. Ya que como se comentó, la separación del conjunto de entrenamiento y validación se realizó después de mezclar las muestras, por lo que parte de los datos utilizados en la validación pueden están en el entrenamiento.

Por lo tanto, se procedió a evaluar utilizando facturas hechas a partir de plantillas distintas y entonces se procedió a separarlas del conjunto total de facturas. Para su evaluación se realizaron cinco sesiones de entrenamiento, en cada una el conjunto de validación contenía plantillas distintas del conjunto de entrenamiento, con ello se puede intentar asegurar de que el modelo no esté convergiendo rápido porque este memorizando los datos, ya que el objetivo es que generalice el concepto de factura y no depender del formato. Los resultados obtenidos fueron los siguientes.

Cuadro 5.1: Sesiones de entrenamiento con plantillas distintas en el conjunto de validación

Model	**Training**		**Validation**		**Avg. Time**
	Accuracy	**Loss**	**Accuracy**	**Loss**	
train1	0.981	0.051	**0.678**	**0.837**	3515 seconds
train2	0.985	0.042	0.915	0.195	2010 seconds
train3	0.981	0.051	0.867	0.275	1824 seconds
train4	0.982	0.047	0.950	0.135	1874 seconds
train5	0.989	0.034	**0.739**	**0.536**	1862 seconds

Como se puede observar en la tabla 5.1 ocurre un fenómeno interesante. Todos los modelos con el conjunto de entrenamiento convergen bien, sin embargo, a la hora de validar con el conjunto de validación se puede observar como el modelo train1 y train5 dan un resultado muy inferior de lo esperado en comparación a los otros tres modelos. A continuación se mostrarán las gráficas correspondientes al entrenamiento y validación del modelo train1, train3 y train5.

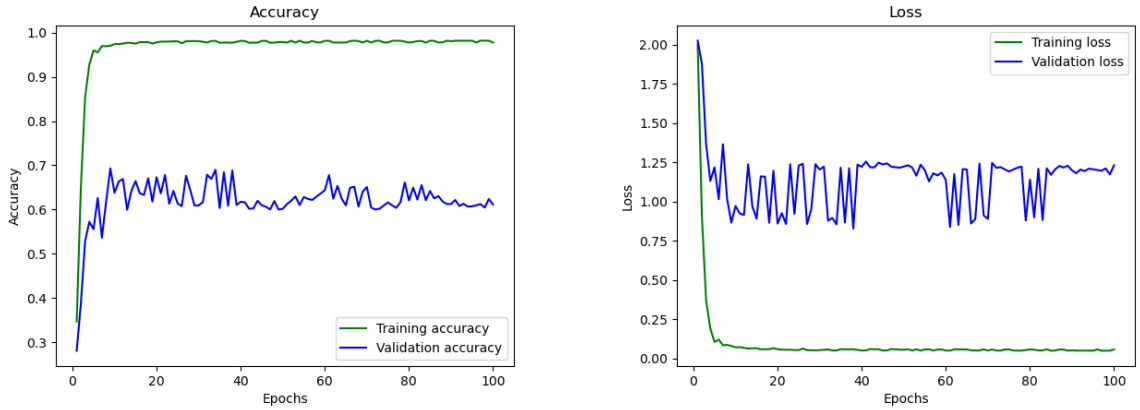


Figura 5.2: Modelo train1 - Gráfica que muestra el progreso de la precisión y la pérdida.

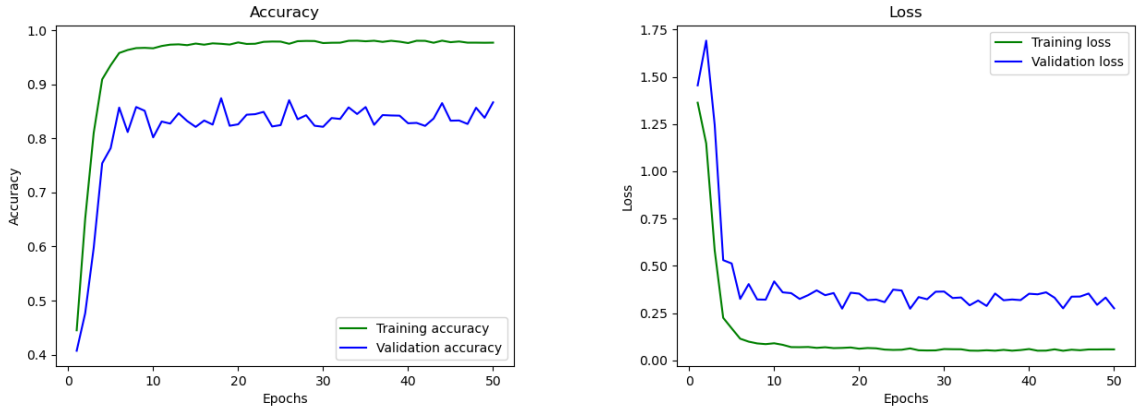


Figura 5.3: Modelo train3 - Gráfica que muestra el progreso de la precisión y la pérdida.

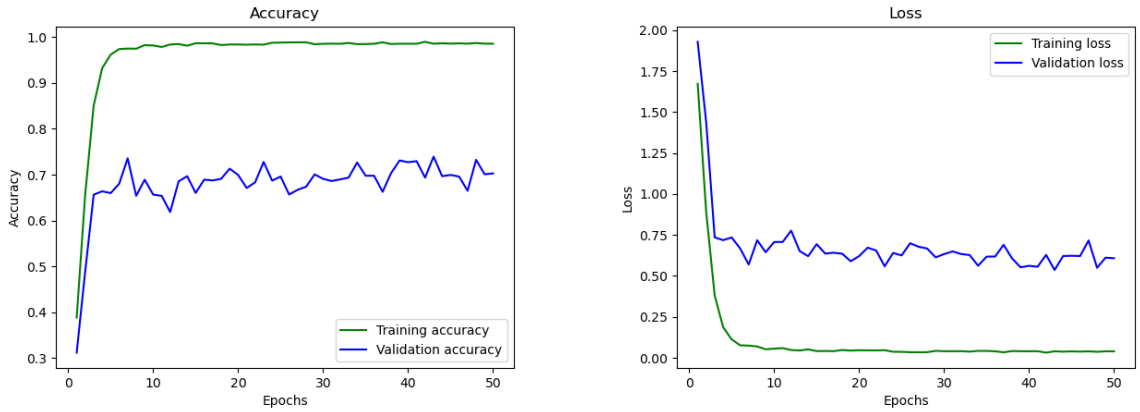


Figura 5.4: Modelo train5 - Gráfica que muestra el progreso de la precisión y la pérdida.

Tras visualizar las gráficas de todos los modelos entrenados, se puede apreciar como el conjunto de entrenamiento converge, donde la precisión tiende a 1 y la pérdida a 0, mientras que el conjunto de validación no. Esto es debido a que si se entrena al modelo con facturas y distintos datos, logra inferior la información, sin embargo al cambiar la plantilla, donde la distribución y el contexto de ciertas entidades puede ser distinto, pues no asociar por completo las entidades. Por lo tanto, se concluyó que el modelo está memorizando las facturas del conjunto de entrenamiento y no está generalizando lo suficiente para tipos de facturas que no ha visto. Entonces estamos ante un caso interesante ya no es ni sobreajuste ni subajuste. Se puede apreciar un error constante entre el conjunto de entrenamiento y validación (bias), probablemente es debido a que en las plantillas del conjunto de validación existen campos que no puede identificar el modelo y otros que si, por lo que resulta en no poder inferir correctamente los datos del conjunto de validación. Otro posible motivo al comportamiento errático de los modelos train1 y train5 es debido a que las plantillas que se encuentran en el conjunto de validación sean distintas de las que se encuentran en el conjunto de entrenamiento.

Una de las maneras existentes para mitigar los problemas de sobreajuste es tener un conjunto de muestras mayor y que exista mayor variedad entre ellas para poder generalizar las facturas. Otra forma de mejorar los resultados es modificando los parámetros de regularización como el *dropout* y L2, donde lo que se busca es normalizar el comportamiento del modelo y que sus resultados no sean tan erráticos. Uno de los objetivos fue reducir dicha varianza modificando el *dropout* y L2, y el conjunto elegido fue el *train1* 5.2 ya que de los cincos archivos separados, fue el que más varianza presentaba. Los parámetros utilizados fueron los siguientes; 30 épocas, cada 30 épocas se aumentaba el *dropout* o L2(se realizaron ambas pruebas por separado), el ratio de aprendizaje se puso por defecto a 0.001. Estos fueron los resultados del entrenamiento.

Cuadro 5.2: Entrenamiento con el conjunto train1 aumentando el dropout por cada sesión de entrenamiento.

Dropout	**L2**	**Training**		**Validation**		**Avg. Time**
		Accuracy	**Loss**	**Accuracy**	**Loss**	
0.1	0.0	0.980	0.057	**0.782**	**0.589**	1200 seconds
0.2	0.0	0.982	0.050	0.690	0.873	1123 seconds
0.3	0.0	0.976	0.062	**0.736**	**0.781**	1216 seconds
0.4	0.0	0.981	0.050	**0.726**	**0.711**	1142 seconds
0.5	0.0	0.978	0.056	**0.725**	**0.790**	1116 seconds
0.6	0.0	0.979	0.058	0.688	0.862	1029 seconds
0.7	0.0	0.947	0.139	0.590	1.060	1315 seconds
0.8	0.0	0.455	1.345	0.349	1.777	1229 seconds

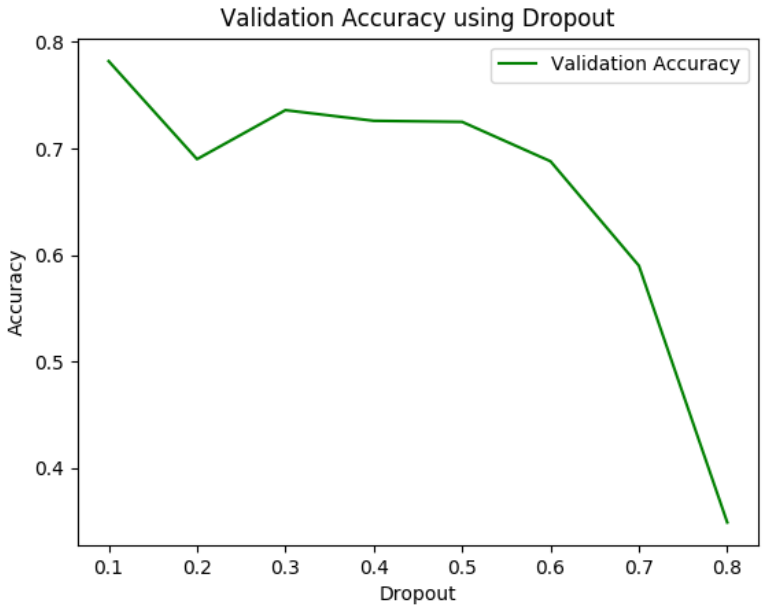


Figura 5.5: Modelo train1 - Gráfica que muestra la precisión según el valor tomado en dropout.

Como se puede observar, teniendo la regularización L2 a 0 e ir aumentando progresivamente el *dropout* desde 0.1 a 0.8, observamos resultados mejores en valores del *dropout* inferiores a 0.5 inclusive. Se logró una mejoría en la varianza en el mejor de los casos de una reducción del 11,2 %, teniendo previamente una varianza del 30 % al 19,8 %.

Se procede con la siguiente prueba, que consiste en regularizar utilizando solo L2 con un

valor fijo y sin decaimiento y manteniendo el *dropout* a 0. Los parámetros utilizados fueron los mismos que la prueba anterior.

Cuadro 5.3: Entrenamiento con el conjunto train1 aumentando la regularización L2 por cada sesión de entrenamiento.

Dropout	**L2**	**Training**		**Validation**		**Avg. Time**
		Accuracy	**Loss**	**Accuracy**	**Loss**	
0.0	0.1	0.981	0.050	0.753	0.730	1118 seconds
0.0	0.2	0.981	0.051	0.719	0.799	1234 seconds
0.0	0.3	0.981	0.049	**0.796**	**0.669**	1292 seconds
0.0	0.4	0.980	0.053	**0.799**	**0.665**	1193 seconds
0.0	0.5	0.981	0.051	0.674	1.112	1090 seconds
0.0	0.6	0.981	0.054	0.749	0.778	1277 seconds
0.0	0.7	0.981	0.049	**0.797**	**0.425**	1252 seconds
0.0	0.8	0.981	0.053	**0.779**	**0.894**	1252 seconds
0.0	0.9	0.979	0.055	0.692	1.075	1179 seconds

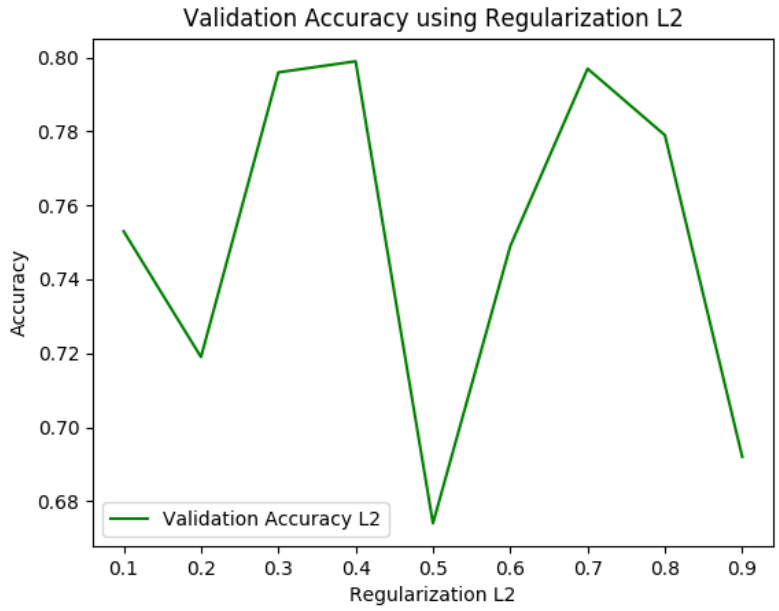


Figura 5.6: Modelo train1 - Gráfica que muestra la precisión según el valor tomado en L2.

En este caso también se puede observar, teniendo el *dropout* a 0 e ir aumentando progresivamente la regularización L2 desde 0.1 a 0.9 en cada sesión de entrenamiento y sin

decaimiento, se observa que también hay mejora en los resultados. Se logró una mejora en la varianza en el mejor de los casos de una reducción del 11,9 %, teniendo previamente una varianza del 30 % al 18,1 %.

A continuación se mostrará una serie de ejemplos de entidades reconocidas en facturas, empezando por el modelo generado en la primera prueba.


```
{
    "CP": "40465",
    "CUPS": "ES9923013353469393SACO",
    "Comercializadora": "VISALIA ENERGIA  S.L",
    "ConsumoEnergia": "365",
    "Direccion": "Callejon del Carb\u00f3n "
        "(Pr\u00e1dena del Rinc\u00f3n)",
    "Distribuidora": "",
    "FechaCargo": "04/11/1992",
    "FechaEmision": "01/11/1992",
    "FechaFin": "30/10/1992",
    "FechaInicio": "31/08/1992",
    "FinContrato": "",
    "ImporteAlquiler": "2,11",
    "ImporteEnergia": "",
    "ImporteImpuestos": "2,31",
    "ImportePotencia": "2,44",
    "ImporteTotal": "25,37",
    "NIF": "73834082M",
    "Nombre": "SALOM\u00d3N ZAPATA BUSTO",
    "NumeroContrato": "6977083232658",
    "NumeroFactura": "FI6262344204",
    "PeajeAcceso": "20DHA",
    "Poblacion": "San Crist\u00f3bal de la Vega",
    "PotenciaContratada": "6.0",
    "Provincia": "Segovia"
}
```


Listing 15: Predicción del modelo entrenado con todas las plantillas.

La mayoría de las entidades se reconocieron correctamente a excepción de 'Distribuidora', 'FinContrato' e 'ImporteEnergía' que no se reconocieron, el resultado es aceptable cuando el modelo se entrenó con todas las plantillas, sin embargo en el siguiente ejemplo podemos identificar varias anomalías. Ya que el modelo fue aislado de una plantilla que fue la que se introdujo en el conjunto de validación.

```
{
    "CP": "",
    "CUPS": "ES4359372808946026XLBZ",
    "Comercializadora": "NICOLAZA  MADURO PE\u00d1A",
    "ConsumoEnergia": "132",
    "Direccion": "",
    "Distribuidora": "ELECTRICA COSTUR,SL",
    "FechaCargo": "02/02/2020",
    "FechaEmision": "",
    "FechaFin": "12/03/2003",
    "FechaInicio": "11/01/2003",
    "FinContrato": "12/03/2003",
    "ImporteAlquiler": "",
    "ImporteEnergia": "",
    "ImporteImpuestos": "",
    "ImportePotencia": "",
    "ImporteTotal": "",
    "NIF": "767382380",
    "Nombre": "CUANT\u00cdAS CORRESPONDIENTES A PEAJES Y CARGOS",
    "NumeroContrato": "",
    "NumeroFactura": "",
    "PeajeAcceso": "",
    "Poblacion": "",
    "PotenciaContratada": "4.5",
    "Provincia": "Cantabria"
}
```


Listing 16: Predicción del modelo entrenado con algunas plantillas y validado con plantillas que no ha visto. Muestra 2

```
{
    "CP": "03600",
    "CUPS": "ES3104507140905091XNCN",
    "Comercializadora": "ENERGIA  NORDICA GAS Y ELECTRICIDAD",
    "ConsumoEnergia": "54",
    "Direccion": "Calle Cabo Tortosa (Rozas de Madrid, Las)",
    "Distribuidora": "DISTRIBUCIONES ELECTRICAS GISTAIN S.L.",
    "FechaCargo": "02/08/2008",
    "FechaEmision": "30/07/2008",
    "FechaFin": "28/07/2008",
    "FechaInicio": "29/05/2008",
    "FinContrato": "28/07/2008",
    "ImporteAlquiler": "",
    "ImporteEnergia": "",
    "ImporteImpuestos": "",
    "ImportePotencia": "",
    "ImporteTotal": "138,69",
    "NIF": "28813310S",
    "Nombre": "CUANT\u00cdAS CORRESPONDIENTES A PEAJES Y CARGOS",
    "NumeroContrato": "",
    "NumeroFactura": "",
    "PeajeAcceso": "",
    "Poblacion": "Elda",
    "PotenciaContratada": "3.5",
    "Provincia": ""
}
```


Listing 17: Predicción del modelo entrenado con algunas plantillas y validado con plantillas que no ha visto. Muestra 2

Como podemos ver en estos ejemplos de predicciones con la segunda prueba donde se cogió el modelo y se entrenó y validó con plantillas distintas. Lo que podemos deducir es que el modelo reconoce muy bien ciertas entidades como 'Provincia', 'Comercializadora', 'CUPS', 'NIF', etc. Sin embargo varias no las reconoces y esto es debido que no ha podido generalizar del todo las facturas y porque probablemente la plantilla utilizada en el conjunto de validación tiene un contexto distinto de las entidades.

Capítulo 6

Conclusiones y trabajo futuro

Uno de los aspectos más importantes del proyecto es la evaluación del entrenamiento con métricas que realmente informen del estado del modelo en el proceso de aprendizaje, así como la búsqueda de distintas arquitecturas de redes neuronales que ayuden a solventar el problema de forma eficiente y en un tiempo aceptable. A lo largo del proyecto y a medida que se va entrenando un modelo y se comprenden las métricas, uno puede observar que la calidad de los datos que posteriormente se utilizarán para entrenar, es más importante de lo que parece. Por eso es importante entender bien cuál es el objetivo por resolver y cómo se podría adecuar el conjunto de entrenamiento sin que sea muy explícito.

Incluso los procesos de limpieza son importantes ya que se puede estar eliminando símbolos o palabras que pueden aportar información relevante al contexto. Durante el desarrollo del proyecto se tenía un conjunto de datos que no era adecuado para el modelo y la mayoría de sus predicciones eran erróneas. Sin embargo, cuando se cambió la estructura del conjunto de datos, que es la mostrada a lo largo de la memoria, las predicciones mejoraron por completo.

Hacer uso de las redes neuronales fue un completo acierto, ya que el mercado muestra una clara tendencia a implementarlas con distintos enfoques como CUTIE [27] y cada vez están en más desuso métodos más convencionales, aunque siguen siendo útiles en tareas específicas.

Respecto al proyecto, los resultados mostrados a raíz del entrenamiento del modelo denotaron que las predicciones se realizan con una tasa de acierto bastante buena cuando la distribución de la factura es similar a alguna con las que se entrenó el modelo. Aunque para obtener dichos resultados a raíz de una buena generalización por parte del modelo, habría hecho falta más variedad de facturas las cuales no se tenían y por ello se optó por realizar simulaciones.

Para futuras implementaciones se tiene pensado probar a anotar la mayor cantidad de

características posibles ya que esto puede favorecer la generalización del texto, junto al aumento de la variedad de facturas. Otro posible enfoque que se le podría dar al problema sería hacer uso de redes convolutivas basadas en grafos, ya que varias empresas del mercado hacen uso de ellas y obtienen resultados aceptables.

Bibliografía

[1] TB Brown, B Mann, N Ryder, M Subbiah, J Kaplan, P Dhariwal, A Neelakantan, P Shyam, G Sastry, A Askell, et al. Language models are few-shot learners. arxiv 2020. *arXiv preprint arXiv:2005.14165*.

[2] Frederik Bussler. Will gpt-3 kill coding?, Jul 2020.

[3] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[6] Matthew Honnibal and Ines Montani. spacy 101: Everything you need to know · spacy usage documentation.

[7] Matthew Honnibal and Ines Montani. spacy · industrial-strength natural language processing in python, Feb 2015.

[8] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[9] Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. *Finding Similar Items*, page 68–122. Cambridge University Press, 2 edition, 2014.

[10] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.

[12] Christopher Olah. Understanding lstm networks, Aug 2015.

[13] Rasmus Berg Palm, Florian Laws, and Ole Winther. Attend, copy, parse end-to-end information extraction from documents. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 329–336. IEEE, 2019.

[14] Rasmus Berg Palm, Ole Winther, and Florian Laws. Cloudscan - A configuration-free invoice analysis system using recurrent neural networks. *CoRR*, abs/1708.07403, 2017.

[15] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[16] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[17] M. Rusiñol, T. Benkhelfallah, and V. P. dAndecy. Field extraction from administrative documents by incremental structural templates. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1100–1104, 2013.

[18] Anuj Sable. Building custom deep learning based ocr models, Aug 2020.

[19] Guergana Savova, Wendy Chapman, Noemie Elhadad, and Martha Palmer. Temporal histories of your medical event, 2011.

[20] Daniel Schuster, Klemens Muthmann, Daniel Esser, Alexander Schill, Michael Berger, Christoph Weidling, Kamil Aliyev, and Andreas Hofmeier. Intellix - end-user trained information extraction for document archiving. 08 2013.

[21] Sam Shead. Why everyone is talking about the a.i. text generator released by an elon musk-backed lab, Jul 2020.

[22] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016.

[23] R. Smith. An overview of the tesseract ocr engine. In *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, volume 2, pages 629–633, 2007.

[24] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. *arXiv preprint arXiv:1702.02098*, 2017.

[25] A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, L Kaiser, and I Polosukhin. Attention is all you need. arxiv 2017. *arXiv preprint arXiv:1706.03762*, 2017.

[26] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.

[27] Xiaohui Zhao, Zhuo Wu, and Xiaoguang Wang. CUTIE: learning to understand documents with convolutional universal text information extractor. *CoRR*, abs/1903.12363, 2019.

[28] Xizhou Zhu, Dazhi Cheng, Zheng Zhang, Stephen Lin, and Jifeng Dai. An empirical study of spatial attention mechanisms in deep networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6688–6697, 2019.