

Análisis de las Herramientas de Procesamiento de Lenguaje Natural para estructurar textos médicos

Borja Quevedo Marcos



PROYECTO FINAL DE MÁSTER EN INGENIERÍA INDUSTRIAL

TECNUN - Universidad de Navarra

Donostia - San Sebastián

2020

Abstract

La escasez de información estructurada en el campo de la medicina a lo largo de los años imposibilita la aplicación en este área de nuevas tecnologías de Inteligencia Artificial relacionadas con el análisis de datos .

Nuevas aplicaciones de PLN han sido creadas con el objetivo de procesar textos médicos de manera automática y aumentar así la cantidad de datos estructurados. Este proyecto trata de analizar las herramientas disponibles y proponer una solución a la estructuración de textos médicos según las necesidades de la empresa Naru Intelligence.

Tras analizar y comparar las herramientas de estructuración de textos médicos existentes a día de hoy se ha observado que muchas de estas herramientas no son de calidad o no son capaces de estructurar cualquier texto. Amazon Comprehend Medical (servicio de AWS) resulta una opción interesante para estructurar textos médicos, siendo su principal limitación el idioma soportado (Inglés).

Agradecimientos

Me gustaría dar las gracias a todo el equipo de Naru Intelligence Solutions sin quienes este trabajo no hubiera sido posible. En primer lugar a Mainer y Eider por darme la oportunidad de formar parte de Naru durante estos meses, periodo en que me han hecho sentir uno más desde el primer día y durante el cual he podido *yoki yoru naru* (“convertirme en algo mejor”). A Ainhoa y Marta por estar dispuestas a ayudar en todo lo que hiciera falta cuando fuera necesario, dejando a un lado sus quehaceres para hacernos caso a Miriam y a mi. A Pedro, por solucionar la multitud de errores informáticos y a Ainara, Ainhoa y Koldo, con quienes hemos podido compartir cafés y celebrar cualquier festejo sin aparente importancia, pues una simple boda es buena excusa para compartir un café traído de la otra parte del mundo.

Este pequeño gran grupo es una muestra de la importancia de la calidad por encima de la cantidad (“*Less is more*” [Mainer Alberich]). Son un ejemplo de que el trabajo trae sus frutos, siempre y cuando exista dedicación y organización (“*planificación diaria*” [Eider Sánchez]). Mainer y Eider, dos polos opuesto atraídos por un mismo sueño: conseguir lo mejor para los suyos.

Motivación

Siempre que me he encontrado en una situación complicada he agradecido la mano que, desde mi alrededor, se me ha tendido. Ésta mano es sencilla de agradecer y más en las situaciones en que uno siente la necesidad de apoyo para salir adelante. Muchos de estos apoyos fueron descubiertos por humanos de generaciones anteriores que posiblemente hayan fallecido. Sin embargo, la huella que dejaron es tan firme que hoy en día siguen formando parte de las herramientas utilizadas para solucionar problemas. Puede que la rutina nos haga olvidar el agradecimiento que le debemos a Alexander Fleming cuando sufrimos una infección. Sin embargo, los fallecidos en el siglo XIX a causa de infecciones hubieran agradecido a más no poder un antibiótico.

Muchas veces somos ayudados por personas externas (o herramientas creadas por ellas) que no reciben el agradecimiento que merecen por nuestra parte ya que son desconocidos. He aquí la bondad del ser humano, dispuesto a ofrecer una mano al mismo precio al que recibió todo lo que tiene.

La motivación de este proyecto, al igual que la de poder trabajar como parte del equipo de Naru Intelligence, es ayudar a todos los pacientes que se encuentran en una situación complicada ofreciéndoles una herramienta capaz de mejorar su día a día. Luchar por un objetivo resulta motivante, pero luchar por un objetivo con repercusiones positivas para otras personas, resulta ilusionante.

Contenidos

1. Introducción	21
1.1. Naru	23
1.2. Inteligencia Artificial	26
1.3. Computación Cognitiva	26
1.4. Aprendizaje Automático (Machine Learning)	27
1.5. Aprendizaje Profundo (Deep Learning)	29
1.6. Lenguaje Natural	30
1.7. Procesamiento de Lenguaje Natural	31
1.7.1. Arquitectura	32
1.7.2. Complejidad	33
1.7.3. Aplicaciones	34
1.7.4. Ventajas	36
1.8. Textos Médicos	36
1.8.1. Plasmar información médica de distintas formas	37
1.8.2. Problemática para su procesamiento	38
1.9. Estructuras de Datos	38
1.10. Big Data y Análisis de datos en Medicina	40
1.10.1. Sanidad y tecnología en España	41

2. Estado del Arte	47
2.1. PLN en textos	47
2.1.1. Tareas	49
2.1.2. Métodos	51
2.1.3. Diccionarios	57
2.1.4. Corpus	59
2.1.5. Codificación	60
2.2. PLN en textos médicos	62
2.2.1. Tareas	62
2.2.2. Métodos	65
2.2.3. Diccionarios	66
2.2.4. Corpus	66
2.2.5. Codificación	66
2.3. Herramientas para aplicar PLN en estructuración de textos . .	67
2.3.1. Librerías/Paquetes	68
2.3.2. Comerciales	70
3. Hipótesis y Objetivos	81
4. Criterios analizados en las comparativas	83
5. Comparación cualitativa de las herramientas comerciales	89
5.1. Primera Selección	89
5.2. Segunda selección	91
5.3. Tercera selección	95
5.4. Conclusión de la comparación cualitativa	97
6. Comparación cuantitativa de las herramientas comerciales seleccionadas	99
6.1. Descripción de las Herramientas Seleccionadas	99
6.1.1. Text Razor	100
6.1.2. Amazon Comprehend Medical	107

6.1.3.	IBM Watson - Natural Lenguaje Understanding	117
6.2.	Materiales y Métodos	119
6.2.1.	Datos	120
6.2.2.	<i>Gold Standard</i>	121
6.2.3.	Pre-Procesamiento	122
6.2.4.	Entrenamiento y Procesamiento	122
6.2.5.	Post-Procesamiento	122
6.2.6.	Evaluación de Calidad	123
6.3.	Resultados	123
6.3.1.	Parámetros de calidad	124
6.3.2.	Análisis Complementario	126
6.4.	Discusiones	130
6.4.1.	Discusión por herramienta	130
6.4.2.	Reconocimiento de Entidades Nombradas (NER)	131
6.4.3.	Identificación de Relaciones (IR)	132
6.4.4.	Idioma	132
6.4.5.	Temática	132
6.4.6.	Interfaz y tiempo de procesamiento	132
6.4.7.	Método de funcionamiento	136
6.5.	Conclusión de la comparación cuantitativa	137
7.	Limitaciones de las herramientas seleccionadas y alternativas	139
7.1.	Text Razor	139
7.2.	Amazon Comprehend Medical	141
7.3.	IBM Watson - Natural Lenguaje Understanding	143
8.	Metodología de estructuración de textos médicos para Naru Intelligence	145
8.1.	Solución propuesta	145
8.2.	Mejoras de la herramienta comercial seleccionada	147

9. Conclusiones de la estructuración de textos médicos con herramientas comerciales	149
Bibliografía	156

Lista de Figuras

1-1. Mapa de los conceptos relacionados con la Estructuración de Textos / “(x)” :tratado en la sección “x”.	25
1-2. Diferencia entre modelos de aprendizaje automático No supervisado (izda) y Supervisado (dcha) [1].	28
1-3. Representación clásica de una Red Neuronal Artificial [2].	30
1-4. Arquitectura básica del PLN.	33
1-5. N° de publicaciones científicas en España por año [3].	41
1-6. Gasto en sistemas gestores de BD en España por año [3].	44
1-7. Volumen de datos (imágenes) en GB por 100.000 personas de población protegida [3].	44
1-8. Historial Electrónico de Salud en España [3].	45
2-1. Esquema de la Estructuración de textos médicos / “(x)” : tratado en la sección “x”.	48
2-2. Ejemplo de estructuración de textos.	50
2-3. <i>Word Embedding</i> y representación de palabras en 2D.	54
2-4. Estructurar textos médicos.	62
2-5. Ejemplos de recetas manuscritas [4].	64
4-1. Parámetros de calidad analizados.	86

5-1. Comparación cualitativa de las herramientas de PLN para estructurar textos médicos.	98
6-1. Información de las entidades proporcionada por la demo de Text Razor.	102
6-2. Información relacionada con la entidad “Anemia” proporcionada por la demo de Text Razor.	102
6-3. Entidad “dolor” detectada por Text Razor.	103
6-4. Entidades detectadas por la demo de Text Razor.	104
6-5. Ejemplo de funcionamiento de la consola de Amazon Comprehend Medical (1).	108
6-6. Ejemplo de funcionamiento de la consola de Amazon Comprehend Medical (2).	109
6-7. Respuesta de la API de Amazon Comprehend Medical en formato JSON.	110
6-8. Información de cada categoría enviada por ACM.	112
6-9. Definición de categorías de entidades de Amazon Comprehend Medical.	113
6-10. Ejemplos de negación de entidades de Amazon Comprehend Medical.	114
6-11. Relaciones en el texto de Amazon Comprehend Medical.	115
6-12. Atributos de la entidad “peso” detectados por Amazon Comprehend Medical.	116
6-13. Atributos de medicamentos de Amazon Comprehend Medical.	116
6-14. Extracción de información personal de Amazon Comprehend Medical.	117
6-15. Respuesta JSON de la herramienta de IBM Watson.	119
6-16. Proceso de evaluación de la calidad de las herramientas.	120
6-17. <i>Gold Standard</i> en la evaluación de calidad de sistemas informáticos.	121

6-18. Respuesta de la herramienta IBM.	134
6-19. Respuesta de la herramienta TR.	135
6-20. Método óptimo para estructurar textos médicos (opinión personal).	138
7-1. Forma óptima de redactar medicamentos en un texto procesado por ACM.	142
8-1. Estructuración de textos médicos con Amazon Comprehend Medical.	146
8-2. Tablas de la Base de Datos donde se almacena la información estructura por ACM.	147

Lista de Tablas

1.1. Presupuesto sanitario destinado a Tecnologías de Información y Comunicación.	42
1.2. Desglose del gasto sanitario en Plataformas Tecnológicas (PT) en España [3].	43
2.1. Calidad de la estructuración de textos en distintos estudios [5].	57
2.2. Variaciones en la expresión de medicamentos.	63
5.1. Parámetros de las herramientas en la selección 2 (Tareas:NER,Tagging y otras / EP: Entidades Personalizadas / PG:prueba gratuita / *:ofertas para estudiantes)	94
5.2. Parámetros de las herramientas en la selección 3.	97
6.1. Entidades relevantes detectas por Text Razor y Text Razor 2.0.	105
6.2. Parámetros de calidad de Text Razor y Text Razor 2.0.	106
6.3. Entidades relevantes detectadas por cada herramienta.	124
6.4. Parámetros de calidad de las herramientas.	125
6.5. Resumen de los parámetros obtenidos.	126
6.6. Términos “conflictivos” detectados por ACM y TR (1).	128

6.7. Términos “conflictivos” detectados por ACM y TR (2). 129

6.8. Ránking de las herramientas analizadas por parámetros evaluados. 137

9.1. Conclusiones de las Herramientas Comerciales evaluadas. 153

Diccionario

En este capítulo se recogen las definiciones de los términos más frecuentes y de compleja comprensión, así como el significado de las abreviaturas¹ utilizadas.

Definiciones

1. **Aprendizaje Automático** (*Machine Learning*): rama de la inteligencia artificial que tiene como objetivo conseguir máquinas que aprendan solas.
2. **Aprendizaje Profundo** (*Deep Learning*): campo del aprendizaje automático compuesto por múltiples capas de procesamiento organizadas jerárquicamente.
3. **Big Data**: conjunto de datos de distinta naturaleza, gran volumen y que se regeneran a gran velocidad.
4. **Computación Cognitiva**: sistemas de aprendizaje automático que utilizan la minería de datos, el reconocimiento de patrones y el procesamiento del lenguaje natural para imitar el funcionamiento del cerebro humano [6].

¹En caso de aparecer dos abreviaturas separadas por un guión, la primera hace referencia a la abreviatura en castellano y la segunda en inglés.

5. **Corpus:** En tareas de NER, conjunto de documentos que contienen anotaciones relacionadas con los tipos de entidades que se desean detectar.
6. **Diccionario:** En tareas de NER, conjunto de términos que, en caso de aparecer en un texto, se desean reconocer.
7. **Estructuración de texto:** proceso de extracción de información relevante de un texto formado por las tareas de NER, Etiquetado e IR.
8. **Inteligencia Artificial:** campo científico de la informática que se centra en la creación de programas y mecanismos capaces de mostrar comportamientos considerados inteligentes [7].
9. **JSON:** formato ligero de intercambio de datos [8]
10. **Lenguaje Natural:** En la filosofía del lenguaje, lengua o idioma hablado o escrito por humanos para propósitos generales de comunicación.
11. **Procesamiento de Lenguaje Natural:** campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano.
12. **Red Neuronal Artificial:** sistema artificial que, imitando la estructura de las redes neuronales biológicas, pretende alcanzar un funcionamiento similar [9]. Esta estructura computacional está organizada en capas de neuronas (elemento básico) y es muy utilizada en modelos de aprendizaje profundo.
13. **Etiquetado (*Tagging*):** Asignación de palabras a grupos definidos previamente.
14. **Token:** cadena de caracteres con significado coherente en un lenguaje de programación.

15. **Tokenization**: división de un texto en tokens.
16. **Word Embedding**: representación de palabras en vectores de número reales n-dimensionales, agrupando las que son sintáctica o semánticamente similares [10].

Abreviaturas

1. **ACM**: Amazon Comprehend Medical.
2. **API** : interfaz de programación de aplicaciones.
3. **CIE - ICD** : clasificación internacional de enfermedades.
4. **CRF** : *conditional random fields*.
5. **EFA - ADE** : evento farmacológico adverso.
6. **DL** : *deep learning*.
7. **EI - IE** : extracción de información.
8. **FP** : entidades detectadas como relevantes que no lo son.
9. **FN** : entidades relevantes que no se han detectado como tal.
10. **HCE - EMR** : historial clínico electrónico.
11. **IA - AI** : inteligencia artificial.
12. **IR - RI** : identificación de relaciones.
13. **ML** : *machine learning*.
14. **PLN - NLP** : procesamiento de lenguaje natural
15. **REN - NER**: reconocimiento de entidades nombradas en un texto.

16. **RN - NN** : red neuronal.
17. **SDK** : kit de desarrollo software
18. **SVM** : máquina de soporte vectorial.
19. **TR** : Text Razor
20. **UMLS** : sistema de unificación de términos médicos de Estados Unidos.
21. **VP** : entidades relevantes que se han detectado.

Capítulo 1

Introducción

Cada vez somos más humanos, consumimos más, generamos mayor cantidad de residuos, las líneas de producción aumentan, las tecnologías no paran de avanzar, surgen nuevas enfermedades y el volumen de datos sigue creciendo, pudiendo llegar a desbordar los sistemas informáticos.

El big data y análisis de datos son conceptos surgidos hace un tiempo pero en auge actualmente. El análisis masivo de datos requiere de información estructurada para obtener resultados valiosos ya que un archivo desordenado resulta imposible de procesar. Hoy en día abundan las fuentes de datos pero una gran cantidad de los datos recogidos se encuentran desestructurados (en el área de salud concretamente), disminuyendo el valor de la información almacenada.

En 2020, cada 73 días se duplicará la cantidad de datos médicos, de los que se estiman el 80 % están desestructurados [11]. Un solo paciente genera millones de datos (historial médico, características, datos de pruebas o monitorizados) haciendo infinita la cantidad de datos almacenables así como el número de relaciones que entre estos pueden generarse. El tiempo medio de visitas de pacientes de oncología (inferior a 30 minutos [12]) y de visitas

rutinarias (inferior a 15 minutos [13]) son muestra de la cantidad de datos generados a diario.

Si únicamente pudiera aprovecharse una pequeña parte de todos estos datos, la información obtenida sería muy valiosa. A día de hoy no se realiza una recogida estructurada de todos los datos generados en medicina. La empresa *Flatiron Health* afirma que el aprendizaje sobre salud en Estados Unidos se basa en ensayos médicos en los que apenas participa el 4% de la sociedad [14] (muestra muy pequeña de la sociedad). Las herramientas de análisis de datos son cada vez más potentes pero no así las herramientas capaces de almacenar o estructurar datos de manera ordenada haciendo que su análisis resulte óptimo. La estructuración de los textos médicos supondría un avance muy importante para el desarrollo de estas tecnologías y, sobre todo, ayudaría a aumentar el volumen de datos con los que los expertos en el análisis de datos pudieran trabajar, aportando beneficios a la medicina.

Muchas empresas y grupos de investigación están invirtiendo en este área de la salud tecnológica en la que se ha avanzado pero donde queda mucho camino por recorrer. El análisis de datos en la sanidad se prevé pueda ofrecer numerosas ventajas como la detección prematura de enfermedades, análisis de los efectos secundarios de un medicamento o tratamiento, elaboración de estadísticas o medicina personalizada entre otras. Para ello resulta indispensable disponer de una minería de datos [15].

Los sectores industriales (automoción, robótica o mecánica entre otro) reciben infinidad de datos a través de sensores que son fáciles de almacenar y analizar ya que la mayoría son valores numéricos. Gran parte de la información médica es imposible de recolectar a través de sensores ya que en muchos casos se encuentra en textos (manuscritos o tecnológicos) alfanuméricos imposibles de procesar. A pesar de que las computadoras sean capaces de reconocer y traducir texto (como los traductores de los smartphones), estos textos deben ser sencillos y seguir un orden si se desea obtener un rendimiento elevado del sistema. El vocabulario técnico preciso, las diversas gramáticas y semánti-

cas utilizadas en cada rama de la medicina y la forma de escribir de cada médico hacen que los textos médicos sean complejos y en muchos casos, no resulten nada fácil de entender para un ser humano y mucho menos para una computadora [16].

Crear una herramienta computacional capaz de, partiendo de historiales médicos de pacientes, generar una base de datos útil y de calidad para los analistas de datos sería un avance monumental. El número de datos estructurados a disposición de los analistas aumentaría de manera automática y los beneficios derivados de las conclusiones obtenidas no harían más que aportar beneficios para la medicina en general y los pacientes en particular. No obstante, los retos a los que se encuentra todo investigador que trata de crear esta soñada herramienta son numerosas.

1.1. Naru

A lo largo de la historia, la atención médica ha tenido un enfoque único para todos basado en la generalización de tratamientos y diagnósticos en base a un prototipo establecido. La iniciativa de Precisión Medica tiene como objetivo a corto plazo incrementar el uso del conocimiento genético y biológico del cáncer para obtener tratamientos más efectivos para distintas enfermedades. A largo plazo se desea generalizar este uso del conocimientos a todas las ramas de la salud [17].

Las respuestas ante un mismo procedimiento son variopintas en función de las características del paciente [11] dando pie a la aparición de eventos adversos (EAS: aparición o empeoramiento de un signo, síntoma o condición médica indeseable después de comenzar un tratamiento). En Europa y Estados Unidos 8.2 millones de pacientes con cánceres sólidos sufren hospitalizaciones debidas a EAS, generando un gasto de 246 billones de USD (el coste medio asociado a cada hospitalización de oncología asciende a \$20.000). A pesar de que el coste de algunos EAS comunes como un resfriado es de \$546, otros como

una perforación gastrointestinal pueden llegar a costar incluso más que la propia hospitalización (\$24.633). El 65 % de las hospitalizaciones se producen desde urgencias hospitalarias, incrementando la complejidad de gestión por parte del hospital al no estar estas planificadas. Además, algunos estudios han cuantificado que el 65 % de pacientes de oncología sufre al menos 1.5 hospitalizaciones no programadas a lo largo de su tratamiento [18] [19] [20] [21].

La necesidad de reducir estos Eventos Adversos Severos (así como su impacto en pacientes con cáncer durante y después del tratamiento) es clara. Naru Intelligence trabaja por prevenir estos EAS o al menos detectarlos precozmente disminuyendo el coste asociado y mejorando la calidad de vida y seguridad de los pacientes. Para ello son necesarias herramientas que mejoren tanto la gestión de los síntomas, toxicidades y eventos adversos de un paciente como la información acerca del paciente y su evolución al tratamiento de la que dispone el equipo clínico a la hora de tomar decisiones relativas al tratamiento o al seguimiento del paciente.



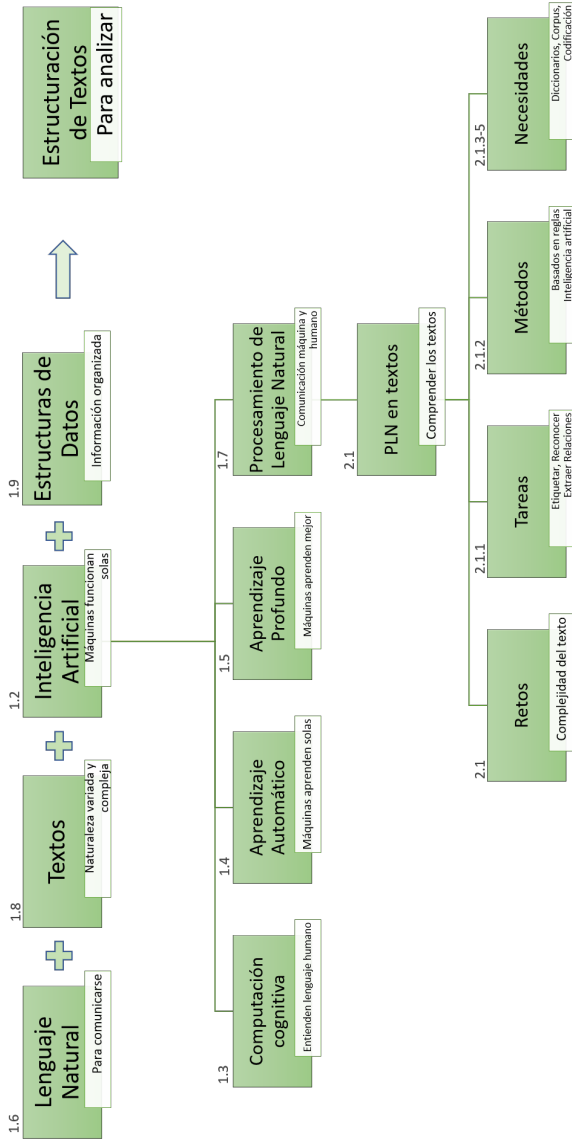


Figura 1-1: Mapa de los conceptos relacionados con la Estructuración de Textos / “(x)” :tratado en la sección “x”.

1.2. Inteligencia Artificial

No es sencillo definir Inteligencia Artificial pues no dispone de una única definición. Estrictamente se podría traducir como el programa de computación diseñado para realizar determinadas operaciones propias de la inteligencia humana [22]. De manera general se trata de conseguir que una máquina se comporte como un humano, dotándole de la capacidad de pensar aunque también hay quien dice que la Inteligencia Artificial posee distintos niveles:

1. Actuar como personas (imitar el comportamiento humano)
2. Pensar como personas (decidir, resolver problemas, aprender)
3. Actuar racionalmente (emular el comportamiento humano, conductas inteligentes)
4. Pensar racionalmente (imitar el pensamiento lógico racional)

Lo que se busca con la IA es que una máquina sea capaz de resolver un problema de la misma manera que lo haría un humano, o por lo menos, que le ayude a tomar decisiones en diversas situaciones, acercando su comportamiento al de un ser humano [23].

1.3. Computación Cognitiva

En la década de 1990 surgió, derivando de la inteligencia artificial y enfocándola hacia un aspecto más humano todavía, el concepto de “Computación Cognitiva”. ¿Quién no estaría encantado de dejar a un lado el ratón y el teclado para poder comunicarse con las computadoras a través del lenguaje? Las clases de mecanografía, al igual que los libros de texto, irán perdiendo su importancia según pase el tiempo y es probable que terminen desapareciendo. El objetivo de la computación cognitiva es la comunicación entre las máquinas y el ser humano, enseñando a estas el lenguaje de los usuarios sin

que el humano tenga que aprender el lenguaje de las computadoras, haciendo que las computadoras piensen como mentes humanas llegando a obtener conclusiones parecidas a las de una persona. La computación cognitiva se basa en sistemas de aprendizaje automáticos, basados en algoritmos complejos, que requieren el uso de minerías de datos [24] y busca crear sistemas tecnológicos de información automatizados capaces de resolver problemas sin requerir la asistencia humana [11].

Abarca, entre otros, el razonamiento, el procesamiento de lenguaje de natural, el reconocimiento tanto de habla como visual, la generación narrativa o la interacción y el diálogo entre computadoras y el ser humano. Es la culpable de que los teléfonos móviles sean capaces de traducir textos sacándoles una foto o frases activando el micrófono del dispositivo móvil [25].

Otro ejemplo de las capacidades de la inteligencia artificial es Watson, considerada la primera plataforma de computación cognitiva, que consiguió vencer a dos concursantes en un concurso de preguntas y respuestas estadounidense en 2011. Los avances desde entonces son numerosos y muchos están relacionados con el aprendizaje automático.

1.4. Aprendizaje Automático (Machine Learning)

Los seres humanos somos capaces de aprender en base a la práctica, lo cual nos ahorra mucho tiempo ya que resultaría muy costoso tener que aprender a cepillarnos los dientes cada noche antes de acostarnos o a conducir para poder llegar a la oficina cada mañana. Resulta inverosímil ¿verdad? Si la inteligencia artificial busca que las máquinas solucionen problemas como lo hace un ser humano, el aprendizaje automático trata de enseñar a las máquinas a aprender por cuenta propia en base a la experiencia. El *machine learning* se basa en la detección de patrones significativos en los datos [26].

Un ejemplo de ML sería el reconocimiento del número “2” en imágenes.

Tras mostrarle una serie de imágenes en las que aparecen distintos números (uno por imagen), el programa aprenderá que las imágenes que contengan una forma similar a “2” deben ser agrupadas conjuntamente como “2” sin haber sido programado para realizar esta función (lo mismo ocurrirá con el resto de números). [27]. Este modelo de aprendizaje se conoce como “No Supervisado”.

En los modelos “Supervisados” se genera la solución manualmente y se envía al modelo, quien aprende de ella. En el ejemplo anterior las imágenes habrían sido agrupadas previamente y el programa aprendería los patrones y características que se repiten en las imágenes de cada grupo. Tras procesar una nueva imagen, el programa compararía sus características con los patrones “aprendidos” para cada grupo y asignaría esta nueva imagen a uno de los grupos previamente definidos de manera manual.

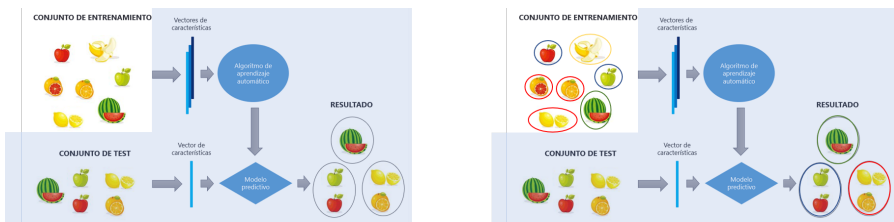


Figura 1-2: Diferencia entre modelos de aprendizaje automático No supervisado (izda) y Supervisado (dcha) [1].

Para realizar procesamiento de lenguaje natural comúnmente son utilizados distintos algoritmos de *Machine Learning* como [28]:

- *Conditional random fields (CRF)*
- *Support vector machines (SVM)*
- *Maximun entropy (ME)*
- *Structural SV (SSVM)*

1.5. Aprendizaje Profundo (Deep Learning)

El aprendizaje profundo es un campo de aprendizaje automático compuesto de múltiples capas de procesamiento organizadas jerárquicamente para aprender representaciones de datos con múltiples niveles de abstracción [29]. La primera capa se denomina “capa de entrada” y la última “capa de salida”, siendo el resto conocidas como “capas ocultas”. La salida de una capa es la entrada de la siguiente en el nivel jerárquico. La capa de entrada recibe los datos y aprende una información sencilla, enviándosela a la capa siguiente que compone una información algo más compleja y se la envía a la tercera capa (segunda capa oculta) para repetir el proceso tantas veces como capas ocultas haya.

Los modelos de aprendizaje automático son capaces de aprender las características generales más importantes (en el ejemplo mencionado aprenden la forma general de los números para agruparlos). Los modelos de aprendizaje profundo van más allá pudiendo aprender características más complejas como los bordes, formas geométricas de la imagen o cambios de color [27]. Cuanto mayor sea el número de capas y neuronas del modelo, mayor será la complejidad de las características consideradas por el modelo.

Los modelos más habituales son redes neuronales estando cada capa formada por distintos números de neuronas (modelo multicapa). La red neuronal de la Figura 1-3 está compuesta por una capa de entrada formada por “n” neuronas, una capa oculta formada por “m” neuronas y la capa de salida.

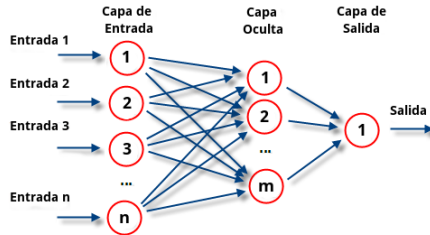


Figura 1-3: Representación clásica de una Red Neuronal Artificial [2].

1.6. Lenguaje Natural

En la filosofía del lenguaje, el lenguaje natural es la lengua o idioma hablado o escrito por humanos para propósitos generales de comunicación. Son aquellas lenguas que han sido generadas espontáneamente en un grupo de hablantes con propósito de comunicarse, a diferencia de otras lenguas (como puedan ser una lengua construida, los lenguajes de programación o los lenguajes formales) usadas en el estudio de la lógica formal, especialmente la lógica matemática [30].

El lenguaje natural es el utilizado a diario y está compuesto, además de por palabras y reglas sintácticas y semánticas, por tonos de voz y gestos que permiten a los humanos poder exteriorizar sus pensamientos y sentimientos (pudiendo comunicar lo que deseen al exterior) convirtiéndolo en un lenguaje más complejo. La ambigüedad y complejidad de las lenguas naturales puede apreciarse en el hecho de que una palabra, en función del contexto en que sea utilizada, puede tener distintos significados, en las frases hechas, ironías, metáforas o en los gestos, cuyo significado varía en función de la cultura. El lenguaje humano es tan complejo que hasta para un ser humano puede resultar, en algunos casos, complicado entender una conversación.

El lenguaje natural se divide en cinco niveles:

1. Nivel fonológico: trata la relación de las palabras con los sonidos que éstas representan.
2. Nivel Morfológico: trata de cómo las palabras se forman a través de morfemas (unidades de significado más pequeñas).
3. Nivel Sintáctico: trata de cómo las palabras se unen para formar oraciones, así como del papel de cada una de ellas en la oración.
4. Nivel Semántico: trata del significado de cada palabra, así como de la unión de palabras para formar oraciones con un significado sin tener en cuenta el contexto.
5. Nivel Pragmático: trata del uso de una oración en distintas situaciones, así como del efecto de estas en el significado de las oraciones.

1.7. Procesamiento de Lenguaje Natural

El procesamiento de lenguaje natural (PLN o NLP) es un campo de las ciencias de la computación, inteligencia artificial y lingüística que estudia las interacciones entre las computadoras y el lenguaje humano. Tiene el fin de construir sistemas y mecanismos que permitan la comunicación entre personas y máquinas por medio de lenguajes naturales. Además, busca que estos mecanismos sean lo más eficaces posible computacionalmente hablando. Se desea que el usuario pueda llegar a comunicarse con el ordenador de la misma forma que lo haría con un humano, para lo cual es fundamental que la computadora sea capaz de recibir información, entenderla y comunicarla en “lenguaje humano” [31].

Hasta la década de 1980, la mayoría de los sistemas de PLN se basaban en un complejo conjunto de reglas diseñadas a mano. A partir de finales de 1980, sin embargo, hubo una revolución en PLN con la introducción de algoritmos de aprendizaje automático para el PLN. [32].

La ciencia avanza a velocidad de vértigo y el número de novedades e inventos presentados incrementan de forma exponencial. Muchas de las tecnologías que los jóvenes de hoy en día utilizamos a diario, serían consideradas pura imaginación. Sin embargo, los niños ya las ven como algo “normal” y lo mismo ocurrirá cuando los nacidos en la década de los 90 pasemos a ser los “abuelos”. La aparición de robots y expectativas de mejora de estos de cara al futuro hacen pensar que podremos crear máquinas idénticas al ser humano. Ahora bien, ¿la comunicación humano-máquina podrá llegar a ser igual que la comunicación entre humanos? ¿Podrá una máquina pensar y expresarse igual que lo hace un humano? La traducción de palabras, inverosímil hace cuarenta años podría considerarse hoy en día algo “normal” pero, ¿cómo una máquina es capaz de interpretar una frase hecha? Que una persona viva gracias a un órgano artificial resultaba igual de inverosímil que lo que hoy nos resulta pensar que un robot pueda llegar a pensar y actuar como un humano. Sin embargo, la ciencia parece no tener barreras o, si las tiene, muy lejos.

1.7.1. Arquitectura

Atendiendo a los niveles del lenguaje natural, la arquitectura general del PLN sería la siguiente:

1. El usuario **expresa** qué es lo que quiere hacer y la computadora lo recibe.
2. La computadora realiza un **análisis morfológico** (análisis interno de las palabras) y **sintáctico** (estructura de las oraciones) mediante analizadores lexicográficos (scanner) y sintácticos (parser)
3. La computadora realiza un **análisis semántico** para conocer el significado de cada oración. Se interpretan las oraciones una vez eliminadas las ambigüedades morfosintácticas.

4. La computadora realiza un **análisis pragmático** (análisis de todas las oraciones juntas, teniendo en cuenta la situación de cada una dentro del texto recibido) tras el cual obtiene la expresión final. La computadora sabe qué va a hacer.
5. Se realiza la **ejecución** de la expresión final para obtener el resultado. Éste se proporciona al usuario.

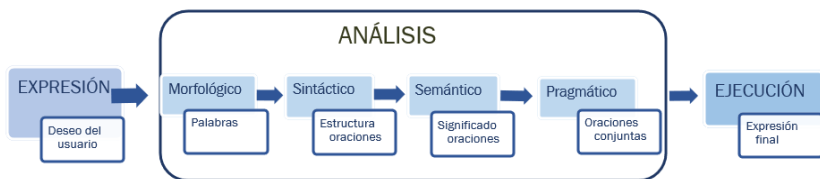


Figura 1-4: Arquitectura básica del PLN.

Para realizar PLN es necesario realizar el análisis de la información recibida. En función del tipo de procesamiento a realizar, se llevarán a cabo análisis en los distintos niveles de la arquitectura ya que no todos ellos serán útiles ni necesarios en todas las aplicaciones del PLN [33].

1.7.2. Complejidad

El nivel de complejidad del problema se observa en la comunicación intrahumana, llena de malentendidos. Además ¿cuán difícil resulta a una persona, a menudo, expresar sus sentimientos más personales? Si entender lo que un amigo quiere decirnos resulta complicado, ni qué decir tiene lo complejo que debe ser crear una máquina capaz de entender a dicho amigo.

Ya se ha mencionado la complejidad de los lenguajes naturales utilizados en el día a día por los seres humanos. El PLN resultaría imposible sin la capacidad de las máquinas de captar la información que transmite una persona

y trasladarla a otra de manera legible. En este proceso se pueden plantear dificultades en la etapa de comprensión de la información recibida o en la de expresión de información de manera comprensible.

Para comprender información comunicada por un ser humano se deben tener en cuenta aspectos como los múltiples significados de cada palabra, los acentos de cada zona, la jerga de cada lugar, expresiones típicas, lenguaje ambiguo, ironías...algunos de los cuales ya han sido solventados y otros están en proceso. El PLN está más avanzado en tratamiento de textos ya que es la forma de comunicación más fácil de interpretar y donde hay más datos (además de la sencillez de obtenerlos en formato electrónico).

A la hora de comunicar la información de manera legible a un ser humano, a la máquina se le plantean dificultades menores como el idioma o el formato de salida deseado por el usuario.

La dificultad de una tarea de PLN en textos es inversamente proporcional tanto a la estructuración y uniformidad del texto como a la complejidad semánticopragmática del mismo [34].

1.7.3. Aplicaciones

Todo el mundo hace uso del PLN prácticamente a diario a pesar de no ser consciente de ello. Algunas de las aplicaciones (ya mencionadas anteriormente) más habituales y fáciles de observar una vez profundizado en el concepto de PLN son el diccionario (traducción de palabras aisladas), traductor (traducción de frases o textos) escrito u oral o Siri (transcripción de mensajes de voz).

No obstante, existen otras muchas aplicaciones muy habituales en nuestro día a día como la extracción de información de un texto y su clasificación estructurada, revisión lingüística, respuesta a preguntas, reconocimiento del habla, procesamiento de grandes volúmenes de información (historias clínicas, clasificación de currículums u opiniones de usuarios de diversos sectores), clasificación de correos electrónicos o análisis de polaridad (positivos o negativos)

de comentarios.

Los programadores junto con los lingüistas no paran de avanzar y sueñan con conseguir más herramientas basadas en PLN capaces de generar resúmenes de textos, indizaciones automáticas,... cuyas posibilidades a día de hoy pueden parecer sueños.

A pesar de que en apartados venideros se mencionarán numerosas aplicaciones de PLN, se citan algunos ejemplos de herramientas de PLN comerciales citados en el artículo [3]:

- MedWhat permite a un paciente conocer el tipo de acciones a realizar ante una serie de síntomas.
- Doctor24 agiliza la comunicación médico-paciente mejorando la calidad de los tratamientos.
- Savana Consulta ayuda al médico en la toma de decisiones cuando se encuentra frente a un paciente, proporcionando información clave para cada caso.
- Selene Clinic agiliza la búsqueda y recuperación de historiales médicos.
- 3M ofrece un servicio de codificación de enfermedades.

PLN y Extracción de Información

El PLN está muy presente en el campo de procesamiento, recuperación y organización de información donde el objetivo final es que el usuario pueda plantearle al ordenador la pregunta tal y como se la plantearía a un humano y este le devolviera la respuesta oportuna sin tener que realizar la búsqueda a través de palabras o conceptos que el usuario considera “clave”. Dentro de la recuperación de información, la extracción de información (IE, Information Extraction) tiene como objetivo obtener información estructurada de un documento de manera automatizada. El documento, escrito en un lenguaje

natural, debe ser legible por la computadora, pero no tiene por qué estar estructurado (aunque en caso de estarlo facilitaría el trabajo de extracción ya que un documento escrito en lenguaje poco formal resulta más costoso de evaluar). Hoy en día se utilizan técnicas de PLN para procesar los documentos. Algunas empresas ofrecen servicios de PLN relacionadas con IE. La empresa “Expert System”, en concreto, ha creado un sistema capaz de extraer las palabras clave de un documento haciendo uso de la categoría gramatical de éstas.

1.7.4. Ventajas

Algunas de las ventajas que presenta el uso del PLN son: [35]:

- Ahorro de tiempo al automatizar y agilizar procesos que hace un tiempo se realizaban manualmente.
- Mejora de la comprensión semántica de los textos y la extracción estructurada de información relevante de estos.
- Ayuda en la toma decisiones de negocio relacionadas tanto con el marketing como con la estrategia, detectando posibles problemas rastreando las redes sociales

1.8. Textos Médicos

Al hablar de textos médicos podemos referirnos a folletos informativos, artículos (académicos o de divulgación científica), notas clínicas, libros, informes de diagnósticos, tratamientos, recetas médicas o prospectos entre otros. Todos ellos con fines didácticos, informativos o comunicativos por lo que sus rasgos principales son la veracidad, la precisión y la claridad. No obstante, muchos de los textos médicos que llegan a manos de los pacientes no son todo lo rigurosos que deberían ser por distintos motivos [36].

1.8.1. Plasmar información médica de distintas formas

La excelencia de un texto esta condicionada por el autor ya que no todos poseen el mismo rigor léxico. Suponiendo dos autores idénticamente excelentes que tratan de plasmar la misma información en un texto, es muy probable que exista una gran diferencia entre ambos documentos [37]. ¿A qué se deben estas diferencias?

- Es frecuente el uso de **abreviaturas** ambiguas como “TCF” (iniciales en inglés) para referirse a “factor de células T” o “fluido de cultivo de tejidos”.
- Existen **variaciones léxicas** como “radioterapia-radiación-irradia” que hacen de dos textos idénticos algo distinto (esto suponiendo que todos los médicos utilizaran y cumplieran las reglas de normalización establecidas). Para intentar unificar la terminología utilizada en los textos médicos, existen sistemas de estandarización como UMLS (sistema de lenguaje médico unificado de Estados Unidos [38]) o ICD (Calificación Internacional de Enfermedades [39]). Además un mismo término puede ser escrito de diversas formas: “N-acetilcisteína”, “N-acetil-cisteína” o “NAcetilCisteína” [40].
- El **idioma** del país de residencia o la **época** pueden ser factores que influyan en la diferencia entre dos textos médicos cuyo significado sea idéntico, ya que el vocabulario sufre modificaciones.
- No se debe dejar a un lado el hecho de que dentro de la medicina existen **numerosas áreas** (neurología, psiquiatría, geriatría o farmacología por mencionar algunas) que forman parte de esta gran ciencia, las cuales pueden utilizar términos distintos para referirse a un mismo concepto. Además de estas variaciones se han detectado diferencias entre documentos redactados en distintos hospitales [41].

- Las costumbres hacen que existan diferencias en la **forma de llamar a los medicamentos**. En España es habitual escuchar el nombre de marcas comerciales mientras que en el Reino Unido se utilizan los nombres de compuestos químicos.
- No se deben olvidar los **cruces diatópicos** (variación del significado de una palabra según la variedad geolectal dentro de una lengua).

1.8.2. Problemática para su procesamiento

A la hora de procesar textos médicos surgen problemas que dificultan la extracción de información.

Los autores de estos textos utilizan el lenguaje natural para plasmar su conocimiento y representar la información que desean, pudiendo comunicarse entre profesionales. Desgraciadamente, las ciencias computacionales no disponen de herramientas para aprovechar al máximo los grandes volúmenes de datos generados por la sanidad ya que no es posible entender la información plasmada en ellos [3].

Además de los ya mencionado problemas como el idioma, la falta de estructura o diversidad léxica, muchos médicos redactan informes a mano utilizando abreviaciones de conceptos, con letra ilegible y sin ningún orden ni rigor complicando aún más los textos.

Como puede observarse únicamente se puede asegurar una similitud entre dos textos médicos: el tema.

1.9. Estructuras de Datos

La programación de computadores, cuyo comienzo data de la década de los cincuenta, se basa en lenguajes de programación de alto nivel (difíciles de entender y escribir) para que la comunicación entre los seres humanos y las computadoras sea posible. La mejora de los programas se basa en las técni-

cas de programación que son soportadas por estos ya que logran optimizar el costo de la escritura, su mantenimiento y el uso de recursos. En pocas palabras, implementan algoritmos más eficientes ya que permiten la reutilización de componentes. No solo las técnicas de programación juegan un papel fundamental en la mejora de programas. Hoy en día las estructuras de datos son la base para el desarrollo de programas complejos y disponer de una presentación apropiada de los datos es necesario para lograr algoritmos eficientes. Trabajar con estructuras de datos supone que un mismo código pueda ser utilizado por distintas estructuras de datos [42]. Además, disponer de documentación estructurada contribuye a tener registros más completos y confiables [41].

Atendiendo a la flexibilidad de crecimiento las estructuras de datos pueden ser estáticas (no cambian una vez definidas) o dinámicas (crecen o decrecen a lo largo del tiempo). Según el recorrido de acceso a los datos están las estructuras lineales (datos unidos por una línea) o no lineales (varias secuencias de unión entre datos).

Las operaciones de búsqueda, inserción/eliminación modificación u ordenación de registros, así como la mezcla de archivos programadas para ser realizadas por la computadora son mucho más sencillas de realizar cuando los datos están organizados. En caso de carecer de estructuras adecuadas resultaría muy complicado realizarlas.

Una adecuada estructuración de los datos supone, además de un ahorro de tiempo tanto para la computadora como para el programador, la posibilidad de obtener mejores resultados cuando se analizan estos datos.

Cuando un analista de datos se dispone a obtener información y tratar un volumen masivo de datos, la probabilidad de éxito estará íntimamente relacionada con la calidad de su estructuración.

1.10. Big Data y Análisis de datos en Medicina

La medicina es una rama de la ciencia donde las aplicaciones del Big Data pueden resultar beneficiosas. En este caso se dejan a un lado los beneficios y ganancias económicas para dejar paso a la gestión de la salud. El volumen de datos que puede obtenerse en el ámbito de la medicina es enorme y tratándolos de manera correcta pueden llegar a predecirse epidemias, curar enfermos o simplemente mejorar la calidad de las personas, apoyando una medicina más personalizada [43].

Tradicionalmente se han almacenado datos relacionados con el paciente como información personal, tratamientos o pruebas de laboratorio. En el siglo XXI los avances tecnológicos se presentan en medios tecnológicos, las historias clínicas se almacenan en el ordenador, existen dispositivos capaces de medir la glucosa en sangre de pacientes con diabetes o presión arterial a tiempo real y existen monitorizaciones cuya información se encuentra en lenguaje natural. La variedad de información, al igual que su veracidad, son claves para obtener beneficios del análisis de los datos. Para ello resulta interesante poder obtener y almacenar información de fuentes de distinta naturaleza. Ahora bien, recordando las 3V y teniendo en cuenta la posibilidad de que los datos se encuentren en distinta lengua ¿cómo procesar de manera automática grandes cantidades de datos (volumen) de distinta naturaleza (variedad) a gran velocidad? Si se desea sacar el máximo provecho a los datos generados día tras día (procesarlos, almacenarlos y posteriormente analizarlos), se requieren nuevas arquitecturas y herramientas [3].

Existen numerosas empresas que hacen uso del análisis de datos en entorno médicos para obtener información relevante y poder ayudar a mejorar la calidad de los pacientes.

Un caso es el de la farmacéutica *Roche*, que adquirió en 2018 la empresa *Flatiron Health*, poseedora de una gran cantidad de clínicas oncológicas y líneas de investigación para obtener datos de pacientes. La plataforma se encarga de capturar y normalizar datos, estructurados o no, de pacientes de oncología

de distintas fuentes y se centra en comprender las posibles interacciones entre medicamentos y organismos, pudiendo obtener información sobre el impacto sobre la salud de distintos tratamientos [44]. Una gran cantidad de los datos con los que se trabaja en el continente resultan muy útiles. Es cierto que los sistemas actual no poseen las capacidades necesarias para analizar ni sacar provecho de los datos de toda la población del mundo, pero los datos utilizados son una muestra demasiado pequeña de la población. La inversión, cercana a los 2 billones de dólares, hace suponer que el futuro de la oncología pueda estar relacionado con el almacenamiento y análisis de los datos de pacientes [45].

Como podemos observar, el análisis de datos está muy presente en el mundo de la medicina y juega un papel muy importante en su futuro. Existen muchas investigaciones relacionadas con este concepto y prueba de ello es el aumento del número de publicaciones científicas sobre big data en la salud de los últimos años (Figura 1-5).



Figura 1-5: N° de publicaciones científicas en España por año [3].

1.10.1. Sanidad y tecnología en España

La tecnología forma parte de la sociedad y se encuentra en prácticamente todos los sectores, incluyendo la sanidad. El índice SEIS recoge anualmente y

de forma objetiva los datos reales sobre el gasto en Tecnologías de Información y Comunicaciones en la sanidad pública en España [39]. La Tabla 1.1 muestra el presupuesto total destinado a las TIC que, a pesar de no superar el 1.3 % del presupuesto total de la sanidad pública, ha aumentado 100 millones de € en apenas 5 años.

Tabla 1.1: Presupuesto sanitario destinado a Tecnologías de Información y Comunicación.

AÑO	Presupuesto TIC (millones de €)	% Presupuesto TIC respecto Global Sanitario
2013	624	1.20 %
2014	661	1.27 %
2015	665	1.26 %
2016	674	1.22 %
2017	696	1.22 %
2018	729	1.23 %

Sin embargo, la cantidad destinada a sistemas gestores de bases de datos supone una pequeña parte del gasto en plataformas tecnológicas (58 % del presupuesto TIC). A pesar de destinarse mayor cantidad de dinero a las TIC (Tabla 1.2), la cantidad total destinada a gestionar BD no ha aumentado con los años (Figura 1-6).

El volumen de datos almacenados en imágenes 3D sí que ha aumentado (Figura 1-7b) mientras que la cantidad de datos generados a lo largo de los años se mantiene constante (Figura 1-7a). Esto puede significar que se pretende aumentar el tamaño de la base de datos (a pesar de que ya es de un tamaño considerable). El problema de los datos generados y almacenados por el sistema sanitario de salud es que gran parte de ellos no están estructurados.

A pesar de los avances tecnológicos de que goza este sector queda mucho camino por recorrer. Dejando a un lado los avances médicos y haciendo

Tabla 1.2: Desglose del gasto sanitario en Plataformas Tecnológicas (PT) en España [3].

DESGLOSE DEL GASTO GLOBAL EN PT	Año					% variación entre los años 2018-2019
	2015	2016	2017	2018		
	en miles €	en miles €	en miles €	en miles €	% presupuesto TIC	
En CPD	53.373	47.360	49.759	51.082	7,01 %	2,66 %
En puestos de trabajo	39.831	38.956	53.496	60.432	8,29 %	12,97 %
En comunicaciones de datos	61.046	55.060	55.663	67.127	9,21 %	20,60 %
En software base de sistemas y comunicaciones	35.304	37.472	49.638	47.351	6,50 %	-4,61 %
En sistemas gestores de BD	16.585	8.560	14.228	16.883	2,32 %	18,66 %
En contratación de servicios externos	113.850	116.994	126.968	131.155	18 %	3,3 %
En personal propio	27.313	23.254	25.327	28.213	3,87 %	11,4 %
Otros gastos gestores de BD	2.040	5.099	22.590	22.416	3,08 %	-0,77 %
Gasto total en PT	349.343	332.755	397.669	424.661	58,27 %	6,79 %

mención a la gestión de la información, a 2018 los ciudadanos de apenas una comunidad autónoma no pueden acceder a su historial electrónico a través de Internet (Figura 1-8a), pero solo 9 de las 17 comunidades autónomas disponen de una aplicación para ello (Figura 1-8b) (a día de hoy existen aplicaciones para todo tipo de usos). Cuando se desea incluir o modificar datos del historial médico electrónico, únicamente 4 comunidades autónomas permiten hacerlo

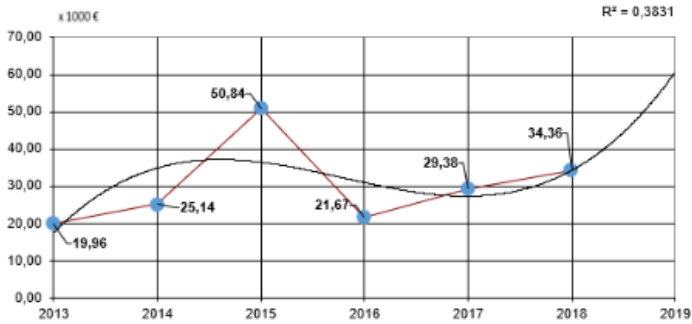
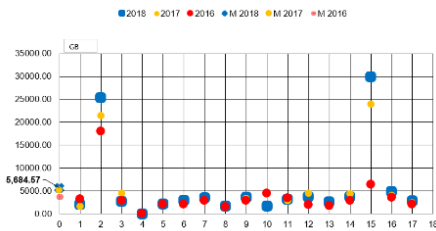
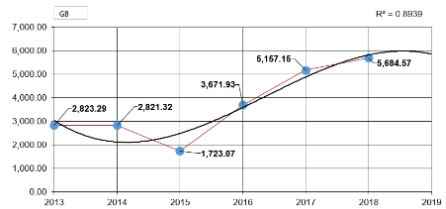


Figura 1-6: Gasto en sistemas gestores de BD en España por año [3].



(a) Generados en exploraciones



(b) Espacio medio consumido [3].

Figura 1-7: Volumen de datos (imágenes) en GB por 100.000 personas de población protegida [3].

desde Internet (Figura 1-8c).

Como cita el Informe de vigilancia tecnológica de la Universidad Politécnica de Madrid, la distribución de la cantidad destinada a las TIC deja entrever que éstas únicamente se utilizan para almacenar datos generados por el sistema sanitario de salud pero no se saca provecho de ellos [3]. Gestionar bases de datos de una forma adecuada y óptima permitiría obtener conclusiones interesantes y útiles para mejorar tanto la salud de los pacientes (mejorando

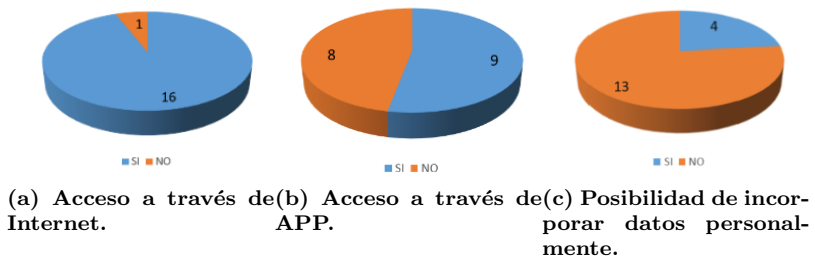


Figura 1-8: Historial Electrónico de Salud en España [3].

el diagnóstico clínico o aumentando las probabilidades de éxito de un medicamento) como la gestión de la sanidad, optimizando tiempo y dinero. Los avances tecnológicos en la salud española están a la vista pero pueden ser mejorados y mucho.

Capítulo 2

Estado del Arte

2.1. PLN en textos

Existen muchas aplicaciones de PLN para textos, algunas relacionadas con la medicina. Se han creado herramientas que permiten generar datos en el historial clínico de un paciente a medida que el médico habla [46], otras capaces de listar los problemas médicos de un HCE [47] y algunas detectar el motivo de la visita de un paciente [48]. Este capítulo trata de las aplicaciones de PLN para estructurar textos médicos, cuyo objetivo es extraer información relevante y estructurada de informes médicos, citas o historiales clínicos. El aumento de la cantidad de evaluaciones de las aplicaciones de PLN para procesar texto libre desde el 2005 [41] indica que este tipo de aplicaciones hoy en día están en desarrollo.

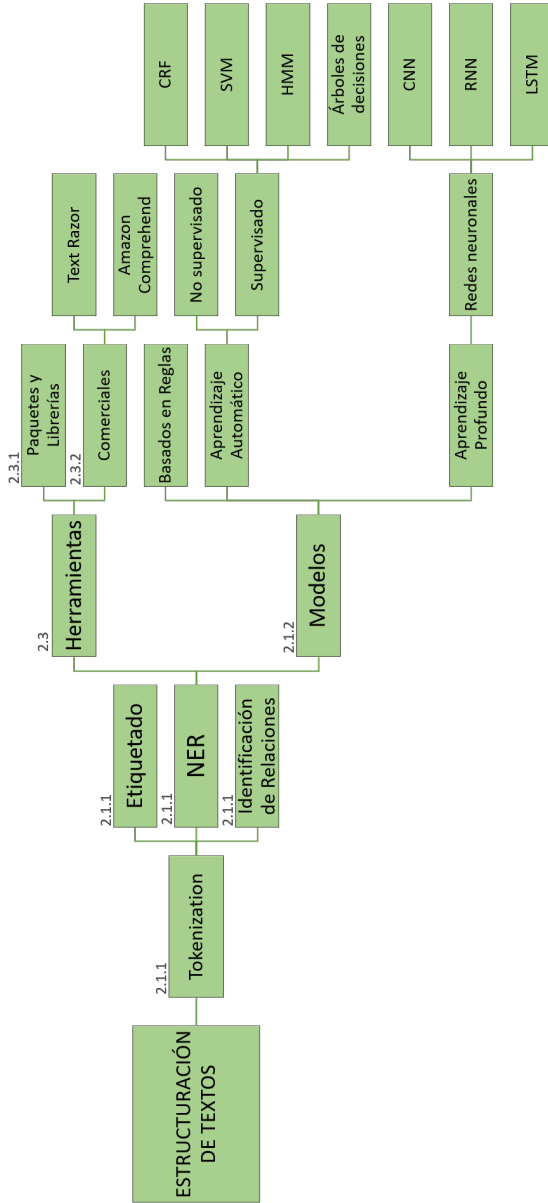


Figura 2-1: Esquema de la Estructuración de textos médicos / “(x)” : tratado en la sección “x”.

2.1.1. Tareas

Para poder realizar el procesamiento de textos no estructurados (análisis y extracción de la información contenida en ellos) se deben realizar algunas tareas previas. La estructuración de textos (preprocesamiento de la información para extraer datos que puedan ser analizados) consta de las siguientes tareas:

- Clasificación: clasificar el documento en base al tema.
- *Tokenization*: dividir una secuencia de caracteres en *tokens* (“pedazos”) y despreciar los caracteres sin información como signos de puntuación.
- *Named Entity Recognition (NER)*: localizar en el texto y clasificar nombres de entidades relevantes conocidas o establecidas previamente.
- Normalización: relacionar cada entidad con un identificador.
- *Sentence segmentation*: dividir el texto en frases.
- *POS Tagging* (Etiquetado): clasificar las palabras el texto (etiquetarlas) acorde a su función.
- *Relation Identification (RI)* (Identificación de relaciones - IR): reconocer las relaciones existentes entre las entidades detectadas.

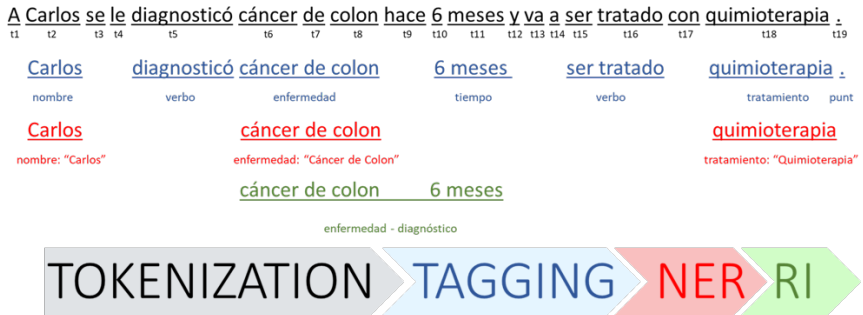


Figura 2-2: Ejemplo de estructuración de textos.

En esta etapa es aconsejable realizar una “limpieza” del texto, eliminando aquellos términos o caracteres que carecen de sentido (signos de puntuación o limpieza de caracteres en un html), y convertir a minúsculas todas las palabras a procesar [49].

Para extraer datos de un texto, de entre todas las palabras es necesario identificar las que interesan. En primer lugar se debe dividir el texto completo en palabras ya que si no se diferencian las distintas palabras del texto, resultará imposible realizar la búsqueda de una entidad. Una vez se ha detectada la palabra, ésta debe ser clasificada antes de enviar la respuesta al usuario ya que, en caso de reconocer entidades y devolverlas sin ningún orden ni criterio, los datos seguirían careciendo de valor. Muchas veces es necesario un post-procesamiento para realizar un filtrado de todas las entidades detectadas y proporcionar como resultado aquellas que resulten de interés [50].

Retos

La extracción de entidades nombradas tiene varios retos [51]:

- Variación de orden de las palabras: “perforación de la úlcera duodenal” tiene el mismo significado que “úlcera duodenal perforada”.

- Derivación: los sufijos pueden cambiar el significado de las palabras (por ejemplo: mediasatinum (nombre) - mediastinal (adjetivo))
- Inflexión: tiempo verbal, número (singular o plural), formas comparativas o superlativas.
- Sinónimos: muy habituales en medicina (por ejemplo: droga, fármaco, medicamento, preparado)
- Dos entidades pueden compartir la palabra de referencia en el texto. El término "Proteínas 91 y 84kDa" se refiere a dos entidades: "Proteína 91kDa" y "Proteína 84kDa" lo cual puede resultar ambiguo [40].
- Homógrafos:
 - o Polisemia: palabras con más de un significado (bilirrubina puede referirse a una sustancia, a un resultado o a una prueba)
 - o Abreviaciones: cada vez más utilizadas (APC puede significar Proteína C activada o Poliposis adenomatosa coli)

Una herramienta capaz de estructurar textos de manera automática debe ser capaz de lidiar con estas dificultades extrayendo la información de manera adecuada y veraz.

2.1.2. Métodos

El beneficio de invertir en IA en entornos clínicos radica en la escalabilidad de estas máquinas capaces de realizar tareas más rápido y, a veces, mejor. Además de disponer de una amplia variedad de conjuntos de datos bien estructurada para cada caso, es necesario seleccionar la arquitectura o el algoritmo más apropiado para el diseño del sistema (en muchos casos la compleja naturaleza del ámbito médico requerirá la combinación de distintos algoritmos de aprendizaje automático) [52].

Existen cuatro técnicas aplicadas a la tarea de NER basadas en distintos enfoques [53]:

Basados en reglas

Los primeros sistemas de PLN desarrollados se basan en reglas manuscritas (los datos anotados son innecesarios). Estas reglas pueden ser diseñadas en base a diccionarios específicos del ámbito de aplicación [54, 55] y patrones léxico-sintácticos [56]. Algunos de los sistemas de NER basados en reglas (la mayoría en reglas semánticas y sintácticas manuscritas) son LaSIE-II [57], NetOwl [58], Facile [59], SAR [60], FSASTUS [61] o LTG [62]. Estos sistemas presentan altas precisiones debido a la exhaustividad del léxico y baja sensibilidad debido a la especificidad de las reglas y a los diccionarios incompletos [63]. Su desarrollo supone un gran coste de tiempo [64] y en el ámbito de la medicina, donde los nombres de entidades son muy variados, tienen problemas [65].

Los algoritmos de *Machine Learning* supusieron un gran avance ya que permitieron detectar nombres con vocabularios dinámicos y fuerte variabilidad [40]. A pesar de tener que disponer del texto en formato electrónico, automatizan el proceso [66]. Para poder realizar la tarea de NER con estos algoritmos es necesario convertir el input de texto a representaciones numéricas.

Aprendizaje automático (Machine Learning) No supervisado

Existen métodos que se basan en algoritmos de aprendizaje automático no supervisados, capaces de aprender a clasificar con un conjunto de entrenamiento sin etiquetar [66] (véase Figura 1-2 en 28). Un modelo muy habitual es el basado en *Clusters* (grupos) [67]. Extrae entidades nombradas en grupos predefinidos mediante similitud de contexto (recursos y patrones léxicos o es-

tadísticas calculadas en los corpus). Para realizar este modelo únicamente son necesarias 7 reglas [68] pero no son capaces de detectar algunos patrones [65].

Aprendizaje automático (Machine Learning) Supervisado

Estos métodos requieren de datos de entrenamiento etiquetados previamente (véase Figura 1-2 en 28) para aprender y poder realizar el etiquetado de textos desconocidos. Se realiza la tarea de NER como un etiquetado secuencial. Se basan en algoritmos de aprendizaje supervisado con ingenierías complejas para detectar características. Los algoritmos de aprendizaje automático (ML) aprenden de diversos textos de entrenamiento y son capaces de reconocer patrones similares en textos desconocidos. La ingeniería de características toma un papel fundamental ya que las palabras se almacenan en vectores como conjuntos de características (valores booleanos, numéricos o nominales) que son utilizadas por el algoritmo [67, 69] (véase Figura 2-3). Esta representación es conocida como *Word Embedding*. La distancia entre los vectores (representaciones de palabras) se toma como medida de similitud [70].

Las características más utilizadas son [71]:

- Características de palabra: mayúscula, puntuación, prefijo-sufijo, forma (SOD1 = palabra + num)
- Características de contexto: conjunto de características de las “n” palabras anteriores y posteriores.
- Características de Frase: longitud, presencia de dígitos, enumeración (comienzo de frase), dos puntos al final, tiempo verbal (pasado presente futuro)
- Características de sección: títulos y subsecciones (se detecta en base a mayúsculas y tipo de letra por ejemplo)

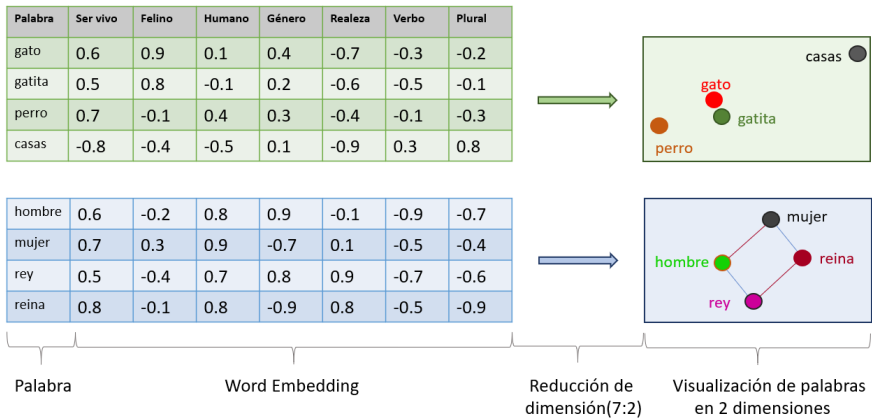


Figura 2-3: *Word Embedding* y representación de palabras en 2D.

- Características de documento: patrones de mayúscula o minúscula, longitud del documento

Estos son ejemplos de algunas características, cada método puede hacer uso de las características que considere necesarias para detectar las entidades nombradas. Basados en estas características, muchos algoritmos de ML han sido aplicados en sistemas supervisados de NER:

- Basados en la teoría de probabilidad [72]:
 - o Modelos lineales como CRF [73], que realiza un etiquetado secuencial adecuado para tareas de NER [72] (utilizado por el paquete de Java Stanford [74]).
 - o Árboles de Decisión [75]
- Basados en la teoría de gran margen [72]:
 - o Modelos relacionales como el modelo oculto de Markov (HMM) [76] o modelos de máxima entropía [77] (utilizado por la herramienta de Python NLTK [64])

o Máquinas de soporte vectorial (SVM) [78] (utilizados para detectar reacciones adversas a medicamentos [79]).

Los modelos de ML (supervisados y no supervisados - véase Figura 1-2 en página 28) a menudo requieren una gran cantidad de documentación con anotaciones para ser entrenados, lo cual supone un coste de tiempo ya que, sobre todo en el ámbito médico, las anotaciones de los textos se realizan manualmente por expertos [70].

Aprendizaje profundo (Deep Learning)

Aprendizaje profundo: identifican, sin ninguna ayuda externa, las características o representaciones necesarias para realizar la clasificación y detección de entidades. Su alta efectividad está relacionada con la capacidad de aprendizaje. Dejando a un lado la ingeniería de características necesaria para los modelos de aprendizaje supervisado [80], el rendimiento de la extracción de entidades nombradas de los métodos de aprendizaje profundo se basa, en gran medida, en la calidad de la representación del input. En estos modelos se realiza una representación distribuida de las palabras. Estas se almacenan en vectores de tantas dimensiones como características se desee almacenar y cada valor almacenado se refiere a una de las características. Este tipo de representaciones detecta características semánticas y sintácticas y se puede realizar a nivel de palabras o de caracteres (útil para obtener información de sufijos y prefijos). Las representaciones híbridas incluyen además información como similitudes léxicas o diccionarios geográficos. Una oración dividida en “tokens” es convertida en una matriz que sirve de input al modelo [81]. El modelo de aprendizaje profundo más utilizado para tareas de NER es BiLSTM-CRF. Las Redes Neuronales son las arquitecturas más utilizadas:

- Convolucionales [82–84]
- Recurrentes [85]

- Recursivas
- Transformadores profundos
- Memoria a largo-corto plazo (LSTM) [86,87]

Estos modelos obtienen mejores rendimientos en tareas de NER que los basados en aprendizaje supervisado. Sin embargo, existe margen de mejora que pasa por aplicar las ingenierías de características utilizadas por los modelos supervisados a los modelos de aprendizaje profundo [63].

En cuanto a las técnicas de aprendizaje profundo que están siendo investigadas para aplicar a tareas de NER destacan *deep multi-task learning*, *deep transfer learning*, *deep active learning*, *deep reinforcement learning*, *deep adversarial learning*, and *neural attention*:

- DL multi tarea: [88] aprende distintas tareas al mismo tiempo obteniendo mejores resultados que los algoritmos que aprenden cada tarea por separado.
- DL de transferencia: aplica el conocimiento aprendido en un dominio de entrenamiento [89]. También se llama DL adaptado al dominio.
- DL de refuerzo: el algoritmos aprende del entorno interactuando con él [90,91].
- DL adverso: [92] el modelo es entrenado con modelos adversos para hacerlo más robusto mejorando así la calidad.
- Atención neuronal: [93] permite a los modelos enfocarse en las partes del input que son relevantes y detectar así los elementos más relevantes.

Los sistemas híbridos combinan los distintos algoritmos mencionados, obteniendo las características más interesantes de cada uno mejorando así el rendimiento global [64]. [94], por ejemplo, implementa CRF tras la salida bidireccional de una red neuronal(LSTM) y [79] utiliza árboles de decisiones, máquinas de soporte de vectores y modelos de máxima entropía.

Como se observa en la Tabla 2.1, las mejoras tras la aparición de estos algoritmos es considerable¹. Cabe mencionar que los modelos de relaciones neuronales son los que mejores rendimientos han obtenido en tareas de NER [5]. Además, estudios como [95] demuestran que utilizar algoritmos de aprendizaje automático es muy útil a la hora de extraer relaciones (como medicamentos y EFAs) entre entidades médicas .

Tabla 2.1: Calidad de la estructuración de textos en distintos estudios [5].

MÉTODO		Precisión	Valor-F
Tradicional	Coincidencia léxica de patrones [96]	0.70	0.68
Algoritmos ML	Aprendizaje supervisado vía CRF [97]	0.86	0.82
	RN convolucional con atención [81]	0.82	0.83
	SVMs [79]	0.89	0.87

Las mejoras de los modelos que realizan NER y RI pasan por mejorar los modelos de *machine learning* o diseñar técnicas de *machine learning* que utilicen conocimiento externo y texto no estructurado. El progreso de PLN ha llevado a mejoras notables en NER y RI para dominio clínico. Sin embargo, queda margen de mejora en la tarea conjunta de NER-RI [98].

2.1.3. Diccionarios

En el reconocimiento de entidades nombradas se define diccionario como el conjunto de conceptos o entidades que se desean reconocer. Estos diccionarios cambian constantemente ya que se añaden nuevos conceptos y se eliminan los

¹Los conceptos de Precisión y Valor-F se encuentran explicados en la página 86. Todos los sistemas han utilizado el mismo conjunto de datos

que han quedado obsoletos. En la mayoría de los casos los diccionarios son lenguajes documentales, instrumentos que facilitan el control terminológico mediante el uso de descriptores (palabras clave que han sido normalizadas):

Los lenguajes documentales consisten en conjuntos de términos (palabras clave) que pueden sufrir modificaciones (altas, bajas y cambios de morfología). El objetivo es representar y recuperar información de manera sencilla, para lo cual contemplan reglas sobre la forma de los términos, las relaciones entre ellos y su aplicación.

El continuo de lenguajes documentales (CDL) está formado por (de simple a complejo) [99]:

- Listas de términos: no dicen nada acerca de las relaciones entre términos.
- Taxonomía: lista de términos organizada jerárquicamente en categorías y subcategorías.
- Tesoro: taxonomía con diversas capas de tratamiento para definir la relación entre los términos como relaciones de sinonimia, jerárquicas o asociativas. Se establecen un conjunto de reglas para definir las relaciones entre términos de la misma manera.
- Ontología: se trata de tesauros codificados en un formato procesable por un programa de software, basados en lógica formal y expresados como una colección de asertos utilizando relaciones como [sujeto]-[predicado]-[objeto] (ej.: [android]-[tipo de sistema operativo]-[dispositivos móviles] = Android es un sistema operativo para dispositivo móviles). Son representaciones lógicas de la realidad [100]

Los diccionarios utilizados por las distintas herramientas para realizar NER son, en la mayoría de los casos, taxonomías (mejoran la organización de los contenidos facilitando la navegación y las búsquedas) u ontologías (permiten recuperar y compartir gran cantidad de información de forma ordenada) [101].

Los modelos relacionales y XML (Lenguaje de Marcado Extensible) son los modelos más utilizados a la hora de representar ontologías [102].

2.1.4. Corpus

Para realizar PLN y *Text Mining* en textos, es necesario disponer de un corpus anotado (conjunto de documentos que contienen anotaciones relacionadas con los tipos de entidades que se desean detectar [40]). El rendimiento de los modelos supervisados depende en gran medida de la disponibilidad y calidad de los datos de entrenamiento [64]. La complejidad del texto a procesar condiciona el tipo del corpus y el coste del mismo (cuanto mayor sea la precisión y variedad de los datos a obtener, más complejo será el corpus necesario). Cuando se desea clasificar documentos el corpus estará formado por textos ejemplo ya clasificados mientras que si se quiere detectar entidades nombradas, el corpus constará de textos con todas las entidades de etiquetadas de manera adecuada.

El corpus también sirve para evaluar el rendimiento de una herramienta ya que se pueden procesar los documentos sin anotar y comparar la salida con las anotaciones generadas en el corpus [40].

En función de la forma en que se realizan las anotaciones, existen dos tipos de corpus [40]:

- *GSC* (corpus “oro”): las anotaciones son realizadas a mano por expertos siguiendo unas pautas detalladas y específicas.
- *SSC* (corpus “plata”): las anotaciones son realizadas automáticamente por sistemas computados.

Los errores de funcionamiento de un modelo pueden ser debidos a la incapacidad de detectar adecuadamente las entidades. No obstante, en caso de que el modelo generado sea de calidad pero el resultado proporcionado no sea el esperado, esto puede ser debido a errores en el corpus utilizado (un atleta

puede correr mal en una competición porque no ha aprendido del entrenador o porque se le ha enseñado mal la técnica). Estos errores se dividen en dos grupos [98]:

- Inconsistencia de las anotaciones: para que la herramienta sea capaz de identificar las entidades después del entrenamiento, la información proporcionada por el corpus tiene que ser de calidad. El corpus debe estar exento de errores de caligrafía y poseer una sintaxis adecuada con los términos definidos correctamente. Un ejemplo de este tipo de errores es olvidar un carácter en una abreviatura.
- Superposición de anotaciones: cada entidad detectada en el texto es única y su significado debe ser identificado como tal, etiquetándola en base a la función que desempeña.
 - o Anotar un término dentro de dos categorías: tratar la entidad “nausea” como EFA y síntoma simultáneamente es un ejemplo de este tipo de error.
 - o Anotar un término dos veces dentro de la misma categoría: muchas entidades del ámbito médico están formadas por más de una entidad (“cáncer de pulmón”). En este caso las entidades no deben ser tratadas individualmente sino como una única y almacenarse conjuntamente (error: Enfermedad 1: “cáncer” / Enfermedad 2: “cáncer de pulmón”)

2.1.5. Codificación

Los sinónimos, las abreviaturas o la diferencia entre la terminología coloquial y científica provocan que un mismo concepto pueda ser expresado de diversas formas. Las reglas de codificación o clasificación tienen como objetivo normalizar las entidades y generar un único identificador para cada una de ellas. Los sistemas de codificación existentes utilizan códigos alfanuméricos

como identificadores (uno por cada entidad) pudiendo estar cada código relacionado con distintos fragmentos texto (ej.: el código X que define la *sustancia líquida constituida por hidrógeno y oxígeno, sin olor, color ni sabor que se encuentra en la naturaleza en estado más o menos puro formando ríos, lagos y mares y ocupa las tres cuartas partes del planeta Tierra además de formar parte de los seres vivos* estará relacionado con “H2O”, “agua”, “water” y “ura” entre otros. La necesidad de mejorar y aumentar la documentación estructurada es evidente [103–105]. Hoy en día no existe un sistema de codificación global que contemple todos los términos (resulta imposible por la infinidad de términos existentes). Sin embargo en el ámbito médico sí que se han definido algunos que contemplan en la página 66

Estos códigos, además de normalizar las entidades, son representados vectorialmente en sistemas n-dimensionales y sirven para calcular la similitud entre conceptos [106–108]. Siendo los vectores V1 y V2 las representaciones de dos términos, la similitud entre estos se calcula como:

$$\text{similitud} = \cos(V1, V2) \quad (2.1)$$

En el ámbito médico también se utiliza la información almacenada en las ontologías para calcular la similitud entre dos conceptos [109–111]. La estructura de las ontologías (formada por nodos y vértices) es tal que la distancia entre dos conceptos es menor cuanto mayor sea su similitud. Los métodos más utilizados para calcular la similitud en base a información de ontologías son [112]:

- Basado en nodos: utilizan la etiqueta de los conceptos así como los grupos y subgrupos a los que pertenecen.
- Basado en vértices: comparan la distancia entre los términos.
- Método híbrido: unificación de los métodos anteriores.

2.2. PLN en textos médicos



PACIENTE	ENFERMEDADES	SÍNTOMAS	MEDICAMENTOS	TRATAMIENTOS	EFFECTOS SECUNDARIOS
Nombre	Cáncer de colon	Dolor abdominal	Avastin	Radioterapia	Hipertensión
Edad		Diarrea	Cyramza	Cirugía	Fatiga
Sexo		Pérdida de peso	Zaltrap		Dolor de cabeza
		Pólipo colorrectal			Úlceras en la boca

Figura 2-4: Estructurar textos médicos.

2.2.1. Tareas

La función principal del PLN para la estructuración de textos médicos es extraer información, para lo cual son necesarias las tareas de *Tokenization*, *NER*, *POS tagging* y *RI*.

A pesar de que en el etiquetado de entidades como lugares, personas o fechas los modelos de PLN existentes presentan altos rendimientos, el etiquetado de entidades médicas no es de calidad. Esto se debe a problemas

relacionados con la falta de estandarización, la alta velocidad de aparición de nuevos conceptos o los múltiples nombres utilizados para nombrar a los términos médicos entre otros [113]. A pesar de las investigaciones realizadas, no existen aplicaciones de PLN para estructurar textos que ofrezcan servicios de gran calidad. Extraer toda la información relevante de un historial clínico puede parecer sencillo pero entraña multitud de dificultades. Crear una herramienta capaz de, partiendo de un texto médico, generar una base de datos, es un reto.

Detectar el nombre de un medicamento pero no ser capaz de reconocer la dosis hace que la información quede incompleta. Además de la dificultad de disponer de datos en valor numérico y caracteres, no se debe dejar a un lado las distintas formas existentes para nombrar un mismo fármaco (o los errores gramaticales cometidos por algunos médicos). Si la computadora no es capaz de detectar estas variaciones puede surgir un exceso de variables a estudiar [114]:

“Tratamiento: tomar durante 5 días 2uds de HCT 20mg”

“Tratamiento: tomar durante 5 días 2uds de Hidrocortisona 20mg”

“Tratamiento: tomar durante 5 días 2uds de Hidcortisona 20mg”

Tabla 2.2: Variaciones en la expresión de medicamentos.

MEDICAMENTO	DOSIS
HCT	20mg
Hidrocortisona	20mg
Hidcortisona	20mg

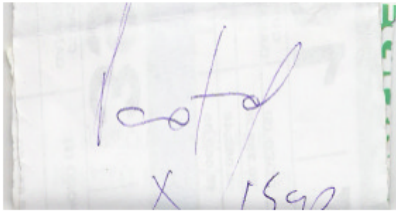
Tampoco resulta sencillo extraer información y almacenarla en función de una misma variable ”Drogas” si esta está expresada de formas distintas:

“Drogas: Si”

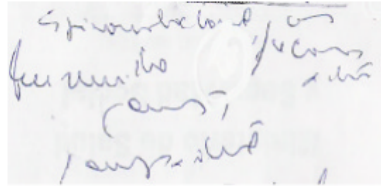
“Drogas: X”

Existen casos en los que la cantidad no se expresa mediante una variable numérica: “*Estenosis muy severa*”. ¿Cómo generar una variable cualitativa representativa para compararla y analizarla en distintos individuos?

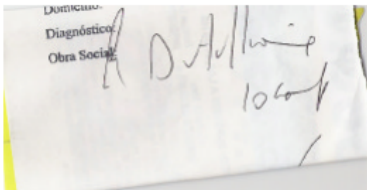
No son pocos los textos médicos manuscritos, muchos de los cuales resultan jeroglíficos para los seres humanos siendo prácticamente indescifrables para un computadora(2-5).



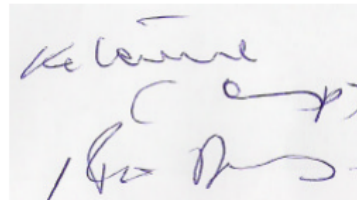
Receta 1 = Paratral ®



Receta 2 = espirolactona y furosemida



Receta 3 = butilioscina



Receta 4: ketamina

Figura 2-5: Ejemplos de recetas manuscritas [4].

Detectar una entidad independientemente del contexto no es lo mismo que detectarla teniendo en cuenta su significado dentro de la oración. En el primer caso la tarea de Tagging resulta más sencilla pues únicamente sería necesario detectar una serie de términos almacenados en una base de datos. No obstante, cuando es necesario entender el texto para poder extraer información

de valor, la tarea de tagging resulta más costosa ya que no todas las entidades (por mucho que su significado independiente sea el mismo) tendrán la misma relevancia. Un medicamento muy habitual como es la *aspirina* puede aparecer en un texto para describir la alergia del paciente o el tratamiento que se le ha recetado. El término *Diabetes* puede hacer referencia a antecedentes paternos, a la enfermedad actual del paciente o al diagnóstico realizado por el médico. La letra A puede referirse al tipo de sangre A + o simplemente ser el comienzo de un párrafo. Como puede apreciarse, entender el significado de los términos es tan importante como poder detectarlos.

La detección de secciones de un texto es una tarea que ha conseguido mejorar el rendimiento de la extracción de información (y otras aplicaciones) en muchos proyectos de PLN [115].

Algunas entidades pueden estar formadas por la unión de más de una entidad, pudiendo detectarlas de manera independiente (ej.: “carcinoma broncogénico de células pequeñas” puede ser interpretado como “carcinoma broncogénico” o “carcinoma de células pequeñas”) [116].

Estas y otras muchas son las dificultades a las que se enfrenta el PLN para poder estructurar textos médicos y generar una base de datos de valor.

2.2.2. Métodos

Los sistemas de PLN destinados a estructurar textos médicos comenzaron utilizando modelos basados en reglas. Gracias al desarrollo de las nuevas tecnologías y al avance en esta área, las herramientas emergentes utilizan complejos algoritmos de *machine learning* o *deep learning*, lo que permite estructurar textos con mayor calidad aprendiendo y mejorando a medida que se procesa la información.

2.2.3. Diccionarios

Algunas ontologías médicas (utilizadas en muchos casos por aplicaciones de PLN) están disponibles en la web vía “BioPortal” [117]. *SNOMED-CT*, *Diseases*, *FMA (Body Parts)* o *Biological Processes* son algunos ejemplos de diccionarios existentes [118].

2.2.4. Corpus

Uno de los hándicaps del PLN en medicina es la escasez de corpus anotados (relacionado, por una parte, por la privacidad de los datos). Para hacer frente al problema se pueden utilizar reglas, lo cual genera el problema de disminuir la generalización [115].

2.2.5. Codificación

En medicina existen sistemas de codificación que contienen una gran cantidad de términos (conceptos, descripciones y relaciones) con la intención de permitir la interoperabilidad entre sistemas informáticos. Los más utilizados son:

- SNOMED - CT (Nomenclatura Médica Sistematizada - Término Clínicos): estándar internacional distribuido por la Organización Internacional de desarrollo de estandarización de terminología de salud (IHTSDO) [119].
- UMLS (Sistema de Lenguaje Médico Unificado): repositorio de vocabulario médico generado por la Biblioteca nacional de Medicina de Estados Unidos [120].
- ICD - 10 (Clasificación estadística Internacional de Enfermedades): lista creada por la Organización Mundial de la Salud (OMS) [121].

2.3. Herramientas para aplicar PLN en estructuración de textos

Una herramienta computacional servirá para estructurar textos en la medida en que sea capaz de:

- Extraer la información del texto (tareas de NER y Tagging), para lo cual resulta necesario comprender el significado de los términos en el texto.
- Devolver la información de manera estructurada en base a las necesidades del usuario.

Las tareas más importantes para extraer información relevante de un texto médico son *Tokenization*, *NER*, *POS tagging* y *RI*. La estructuración de textos pasa por reconocer entidades relevantes (NER), clasificarlas (etiquetado) e identificar las relaciones entre ellas (RI) así como la información relacionada con cada entidad [98]:

- NER y Tagging: estas tareas se pueden comprender como una secuencia de etiquetado (frase=conjuntos de palabras/tokens). Las historias clínicas contienen menciones a entidades médicas como medicamentos (y sus correspondientes atributos), síntomas o AFE (efecto farmacológico adverso, aunque la mayoría de ensayos controlados aleatorios no los incluyen [98]) que deben identificarse y clasificarse de manera adecuada en distintas categorías [5].
- RI: una vez detectadas las entidades y realizado su etiquetado, se deben detectar las relaciones entre estas. Algunos ejemplos de relaciones entre entidades son:
 - o medicamento - AFE
 - o medicamento - dosis

o enfermedad-síntoma

La tarea de identificación de relaciones se divide en dos pasos:

1. ¿Existe relación? Si - No
 2. En caso de que exista, ¿de qué relación se trata?
- Tan importante como detectar entidades es extraer la información relacionada con estas (atributos). Casos habituales en los textos médicos son dosis, vía de administración o frecuencia de medicamentos, negación de un término (“sin metástasis”) o gravedad de una enfermedad (“débil”, “aguda”).

La “incrustación de palabras” (proyección de palabras en espacios vectoriales, convirtiendo la distancia en una medida de similitud entre entidades [65]) ayuda a comprender la relación entre distintas palabras en el contexto (es el secreto del éxito de muchos sistemas de PLN de aprendizaje no supervisado) [122]

Las herramientas informáticas disponibles hoy en día para estructurar textos pueden agruparse en dos grupos.

2.3.1. Librerías/Paquetes

Se entiende como librería o biblioteca el conjunto de implementaciones funcionales codificadas en un lenguaje de programación que ofrece una interfaz bien definida para la funcional que se invoca. Esta pretende ser utilizada de forma simultánea por distintos programas independientes [123].

En programación, se define paquete como una agrupación de clases e interfaces relacionadas. Estas clases son ocultas y nadie ajeno al sistema puede conocer su funcionamiento interno. [124]

Las librerías o paquetes de programación (disponibles para distintos lenguajes como C#, Python, Java o Ruby entre otros) son la base de cualquier herramienta o aplicación informática. Distintos sistemas de estructuración de

texto que han sido desarrollados se encuentran a disposición de quien los requiera, bien sea en formato descargable (códigos o ejecutables), servicios web o interfaces de aplicaciones [64]. Muchos estudios han utilizado este tipo de sistemas para generar herramientas a medidas, para lo cual es necesario disponer de un amplio conocimiento de programación y experiencia además de tiempo para programar el código.

A continuación se listan algunas de las librerías o programas que se pueden descargar para estructurar textos médicos²:

- Spark NLP
- Sci Lite Annotations
- NLTK
- Spacy
- Rasa NLU
- Nirvana
- Meta Map Transfer (MMTx)
- Apache cTAKES
- Apache Open NLP
- Allen NLP
- Mallet
- Standford Core NLP
- MedXN

²Se citan las más utilizadas, en base a artículos académicos y estudios que han sido leídos.

- KEA
- GATE
- Clear TK
- Flair
- Gensim
- Yoo Name
- Free Ling
- Polyglot
- Deep Pavlov

En distintos artículos analizados se ha observado que estas herramientas son utilizados en muchos casos para desarrollar sistemas capaces de solucionar problemas específicos. Estos programas cumplen con un objetivo concreto (como extraer entidades nombradas en textos médicos en chino [72]) pero en caso de aplicarse para otro caso (extraer entidades nombradas en textos médicos en italiano) no sirven.

2.3.2. Comerciales

La implementación de los paquetes mencionados en el apartado anterior requiere gran experiencia y conocimiento de programación [125] y como se ha comentado, su aplicación se limita a casos concretos. Hoy en día existen soluciones automatizadas ofrecidas por empresas privadas (muchas de las cuales están basadas en librerías mencionadas en la sección anterior) que hacen uso del PLN para realizar estructuración de textos. El acceso a estas herramientas se realiza vía web o mediante una API: el usuario adjunta los textos o documentos que desee procesar, la aplicación los procesa en el servidor y devuelve

los resultado al usuario [66]. Estas herramientas de aquí en adelante serán denominadas Comerciales para poder diferenciarlas de las librerías mencionadas previamente.

No son pocas las aplicaciones de PLN en la sociedad y tampoco es escaso el número de herramientas comerciales existentes para estructurar textos médicos. Aplicaciones de NER y Tagging junto con análisis de sentimiento, resumen de textos, clasificación de conceptos son más comunes de lo que cualquier ignorante del tema supondrá.

A continuación se enumeran y detallan algunas de las herramientas comerciales existentes para estructurar textos³. A pesar de que existen diferencias, las características y capacidades de todas ellas resultan similares ya que el objetivo principal es analizar textos siendo capaz de extraer la información relevante de ellos.

4 grandes multinacionales han invertido en el PLN para ofrecer servicios relacionados con la estructuración de textos:

Amazon Comprehend Medical

La mundialmente conocida Amazon ofrece un servicio para detectar información útil en textos médicos no estructurados llamada Amazon Comprehend Medical ⁴. Para hacer uso de esta herramienta es necesario disponer de una cuenta AWS (Amazon Web Services) y el costo es proporcional al uso del servicio (cantidad de texto procesado). Dispone de una primera prueba gratuita (2.500 caracteres) y en la web puede probarse el servicio. La principal pega del servicio es que, a día de hoy, únicamente procesa texto en inglés.

Dispone de dos API que pueden ser utilizadas independientemente.

³No se describen todas las herramientas detectadas durante la realización del proyecto sino las que se han considerado más potentes e interesantes tras las comparaciones realizadas

⁴A lo largo del documento puede que se haga referencia a esta herramienta mediante las siglas ACM.

- PHId identifica y extrae información médica protegida de los pacientes.
- NERe extrae relaciones y entidades con nombre médico como anatomía, condición médica, medicamento, PHI o tratamiento. En caso de estar acompañadas de atributos (valores como la dosis de un medicamento), éstos también son extraídos.

Amazon Comprehend Medical utiliza el machine learning para ir aprendiendo a medida que los usuarios hacen uso de la herramienta. Los datos de salida, acompañados de un a puntuación de confianza para conocer la calidad de los mismos, se pueden almacenar en un servicio de almacenamiento o una base de datos (aspecto interesante si posteriormente se quieren analizar).

A continuación se listan algunas empresas que han utilizado este servicio en casos reales:

- LexisNexis Legal & Professional: esta empresa que ofrece soluciones tecnológicas a empresas en general y en el ámbito legal en particular, utiliza Amazon Comprehend para detectar entidades como juez o abogado en millones de documentos.
- Vibes: la plataforma de interacción móvil procesa los mensajes de los usuarios con Amazon Comprehend para identificar el tema, extraer frases clave o detectar opiniones con el objetivo de ofrecer experiencias gratificantes al consumidor y proporcionar información relevante al equipo de márkintg.
- VidMob: La plataforma tecnológica transcribe texto de vídeos y los analiza con Comprehend para recopilar información que sea procesable y poder así generar información útil para sus clientes.
- Otras empresas como AfricanStockPhoto, Atrium, AntEater Analytics o JimmyCode

IBM Watson - Natural Language Understanding

La multinacional estadounidense IBM (International Business Machines Corporation), además de ser el padre de “Watson” (inteligencia artificial capaz de responder a preguntas formuladas en lenguaje natural) ofrece un servicio de PLN aplicado al análisis de textos no estructurados. Al igual que en el caso de Amazon Comprehend se debe disponer de una cuenta en IBM cloud para hacer uso de este servicio, cuyo coste no depende de la cantidad de texto procesada sino que dispone de tres tarifas fijas con un precio y capacidad de procesamiento establecidos. La herramienta se puede probar en la página web.

El servicio es capaz de procesar texto en 13 idiomas distintos y realizar tareas como la detección automática del lenguaje, extracción de entidades nombradas o palabras clave y análisis de sentimiento. IBM Watson NLU, además de disponer de una serie de entidades y relaciones ya establecidas, permite generar nuevas entidades personalizadas para poder realizar la extracción de información en función de la necesidad del usuario (este servicio supone un coste adicional) [127].

En el ámbito médico IBM ha desarrollado otras herramientas como Watson for Oncology (WFO) para oncología o Watson for Genetics (WFG) relacionado con los genes y las proteínas que, en caso de funcionar adecuadamente, supondrán un gran avance.

Azure Text Analytics

Microsoft no se ha quedado atrás y, al igual Amazon o IBM, ofrece un servicio de análisis de texto en Azure (la nube) con tarifas establecidas, para lo cual es necesario disponer de una cuenta Azure. Si se desea procesar poco texto, además de la prueba gratuita de 7 días y la demo disponible en la página web, existe un servicio gratuito con límite mensual.

Capaz de detectar 120 idiomas, realiza tareas como análisis de opiniones, extracción de frases clave o reconocimiento de entidades con nombre (aun-

que todas las entidades únicamente están disponibles para algunos idiomas). Dispone de pocas entidades establecidas entre las que no se encuentran aquellas relacionadas con el ámbito médico. Azure permite descargar librerías en distintos programas para analizar los textos, cuya capacidad máxima es de 5129 caracteres o 1MB. Algunas aplicaciones reales de Azure text Analytics son [128]:

- Análisis de resultados de encuestas
- Extracción de información de llamadas grabadas
- Clasificación de incidentes del soporte técnico

Google Cloud Natural Language

Para cerrar el ciclo de grandes empresas, Google también ha creado un servicio de PLN para extraer información valiosa de textos sin estructurar. El usuario puede proporcionar el contenido en varios idiomas multimedia ya que la herramienta es capaz de extraer información, además de documentos de texto, de conversaciones de audio o imágenes.

Realiza tareas de análisis de entidades (entre las que no se encuentran entidades médicas), extracción de conceptos clave del texto, análisis de sentimiento, detección de idioma o clasificación de textos en más de 700 categorías. Dispone de métodos de entrenamiento basados en Machine Learning para mejorar el servicio, los cuales pueden ser creados por el usuario.

Dispone de dos API que pueden utilizarse por separado o simultáneamente, ya que las tareas que realizan difieren un poco [129]:

- AutoML Natural Language
- API Natural Language

Google Cloud Natural Language se puede utilizar, por ejemplo, para comprender recibos y facturas extrayendo nombres, fechas y conceptos de los pagos realizados. Algunas empresas que han hecho uso de sus servicios son:

- DocuSign: utiliza AutoML para etiquetar documentos y ahorrar tiempo en el proceso que se realizaba manualmente.
- Hearst newspapers: utiliza esta herramienta de Google para conocer la acogida que sus publicaciones tienen en las audiencias y analizarlas.
- News UK: clasifica los artículos de opinión y los editoriales por temas de manera rápida y precisa con Natural Language.

Además de las herramientas ofrecidas por las conocidas multinacionales, existen empresas que se dedican a la estructuración de textos, ofreciendo herramientas que, no por ser menos conocidas, resultan ser menos interesantes.

Dandelion

SpazioDati es una startup con sede en Italia especializada en semántica y Big Data. Su actividad está muy relacionada con la informática y los datos. Entre otras, ofrece soluciones de datos empresariales relacionadas con el análisis de los datos o gráficos que facilitan las tomas de decisiones [130].

Esta empresa es la creadora de Dandelion, aplicación de PLN para estructurar textos en distintos idiomas y formas: documento de texto, url o html. Se pueden adquirir distintas tarifas en función del uso que vaya a hacerse de la herramienta (a no ser que sea sin ánimo de lucro, para investigación o educación, en cuyo caso se ofrecen planes gratuitos) y ofrecen descuentos para startups. Se puede realizar una prueba en la página web para hacerse una idea del funcionamiento.

Dandelion es capaz de extraer palabras clave o entidades (permite seleccionar las entidades que se desean extraer, entre las que se encuentran entidades médicas), clasificar textos y contenido, detectar similitud entre dos documentos, realizar análisis de sentimientos o “limpiar” el contenido (como puede ser un artículo).

El texto de entrada se envía por internet aunque también existe la posibilidad de, en casos aislados, descargar la herramienta en la computadora [131].

Repustate

Otra herramienta de PLN aplicada a textos es Repustate, que ofrece análisis de texto multilingüe en 23 idiomas. Las tarifas tienen un coste en función de la cantidad de documentos que vayan a procesarse y disponen de descuentos para los estudiantes.

A partir de un texto (acepta formato Excel), realiza extracción de entidades (personas, palabras clave, lugares, enfermedad), clasificación del texto por el tema (dispone de alrededor de 30 temas distintos) y análisis de sentimiento. Si se desea extraer información acerca de un tema desconocido para Repustate, la función “DeepSearch” (algo más cara) permite generar nuevas entidades y realizar búsquedas personalizadas. En este caso el tamaño máximo de los documentos procesables es de 5MB. Esta herramienta tiene una exactitud de 82-90 %, valor que aumenta cuando se trata de análisis de sentimientos.

La descarga de las bibliotecas está disponible para distintos lenguajes de programación (Java, C#, Ruby, Python), aunque también existe la posibilidad de procesar los textos directamente en la nube de Repustate.

Actualmente más de 3.000 organizaciones en todo el mundo (entro otras Roma, Dubai, Sidney, o Nueva York) utilizan repustate para analizar sentimientos o extraer información de los datos de la empresa, redes sociales, noticias, encuestas, blogs o foros. Algunos ejemplos reales del uso de Repustate en distintas áreas [132]:

- Salud y Farmacia: se utiliza el reconocimiento de entidades nombradas para conocer el motivo de la visita de un paciente, los medicamentos y dosis que pueden ser recetadas o el tiempo entre visitas sucesivas de un paciente.
- Automoción: visualizar la satisfacción de los clientes en base a los resultados obtenidos tras analizar los sentimientos de sus comentarios.
- Sysomos: la compañía de análisis de redes sociales con sede en Toronto, utiliza Repustate para ayudar a potenciar el análisis de texto.

Aylien

La principal actividad de Aylien (plataforma inteligente de noticias) es analizar las noticias para obtener información comercial valiosa a tiempo real. Mediante el monitoreo inteligente permite detectar riesgos a nivel mundial, busca ventajas estratégicas de inversión para sus socios y realiza análisis financieros. Sin embargo, también ofrece servicios (plataforma y aplicación) para el análisis de texto.

Mediante la plataforma TAP el usuario puede crear un modelo adaptado específico para su problema y utilizarlo cuando lo deseen. Una vez creado, este modelo es entrenado antes de poder utilizarse.

Sin embargo, la API de análisis de texto no tiene que ser creada por el cliente sino que Aylien ha realizado el trabajo. En caso de querer probarla, la empresa ofrece una prueba gratuita de 30 días de duración y a partir de entonces es necesario abonarse a una de las tarifas ofrecidas para continuar procesando textos. Ofrece descuentos para suscripciones anuales o fines académicos o de investigación

Insertando un documento de texto o url (en idiomas como Inglés, francés, alemán, español), esta aplicación permite extraer entidades ya definidas (dispone de entidades médicas), palabras clave o conceptos, realizar un resumen, clasificar el texto, realizar un análisis de sentimientos o sugerir un hashtag para compartir en redes sociales.

Existen librerías en diversos lenguajes (Java, Python, PHP, Ruby, C#) que pueden ser descargadas. [133]

Text Razor

Con sede en Londres, Text Razor⁵ es una startup fundada en 2011 que proporciona software para ayudar a los desarrolladores a incorporar rápida-

⁵A lo largo del documento puede que se haga referencia a esta herramienta mediante las siglas TR.

mente el análisis de datos a sus aplicaciones. Ofrece un servicio de análisis de texto compatible con 142 idiomas que puede ser probado en la página web.

El coste de esta herramienta está fijado por las tarifas ofrecidas, aunque existe una versión gratuita que permite realizar 500 solicitudes de máximo 20kB cada día (límite aumentable para estudiantes). Text Razor reconoce entidades nombradas (también entidades médicas), clasifica documentos, realiza etiquetado de temas, extrae relaciones y análisis de dependencia además de sinónimos, detecta desambigüedades y realiza corrección ortográfica. Las entidades definidas en la aplicación han sido extraídas de las bases de datos DBpedia y Freebase, aunque también existe la posibilidad de crear nuevas categorías y ejecutarlas en el servidor.

La principal limitación de Text Razor es el tamaño máximo de documentos procesables: 150kB (límite de la tarifa más cara, siendo 50kB y 100kB en las otras dos tarifas). A pesar de que los documentos de textos son el tipo de archivo de menor tamaño, podría resultar algo escaso [134].

3M Code Ryte Code Assist

La empresa 3M utiliza la ciencia para tener un impacto real a nivel mundial, incitando al progreso y la innovación. Entre la gran variedad de productos que ofrece está el software Code Ryte Code Assist, capaz de escanear informes médicos y obtener datos a través de herramientas de PLN.

Entre las entidades que es capaz de identificar esta aplicación web están enfermedades y tratamientos. Automatiza el proceso de codificación ya que, tras detectar cada entidad, se le asigna un código de ICD (clasificación internacional de enfermedades) o del CPT (terminología de tratamientos) [135].

Refinitiv - Open Calais

El proveedor global de datos e infraestructura de los mercados financieros Refinitiv, fundado en 2018, ofrece un servicio de análisis de texto llamado

“Etiquetado Inteligente” cuya versión gratuita es “Open Calais”.

Este paquete requiere de una cuenta en Microsoft Azure para ser utilizado ya que se funciona en la “nube”. Además de realizar etiquetado, es capaz de extraer entidades nombradas (dispone de entidades médicas definidas: tratamientos, condiciones médicas, fármacos), clasificar documentos o realizar análisis lingüístico de un texto [136].

Capítulo 3

Hipótesis y Objetivos

Dada la cantidad de herramientas de PLN disponibles para realizar estructuración de textos médicos, se considera de interés el estudio en profundidad de estas para su utilización por parte de la empresa Naru Intelligence.

El tiempo disponible para realizar el proyecto es limitado y los conocimientos informáticos se consideran insuficientes para generar una herramienta capaz de estructurar textos médicos con mayor calidad que las herramientas disponibles. Por ello en el proyecto se contemplan únicamente las herramientas comerciales (véase subsección 2.3.2) sin considerar la posibilidad de generar una herramienta de PLN personal.

Los objetivos del proyecto son:

- Analizar en detalle las características de Procesamiento de Lenguaje Natural (PLN) aplicado a textos.
- Comparar, cualitativa y cuantitativamente, las herramientas de PLN disponibles para realizar estructuración de textos médicos.
- Seleccionar la mejor herramienta de PLN para estructurar textos médi-

cos según las necesidades de Naru Intelligence.

- Proponer una metodología de estructuración de textos médicos para Naru Intelligence.
- Analizar las líneas futuras de desarrollo de las herramientas de PLN para estructurar textos médicos existentes .

Capítulo 4

Criterios analizados en las comparativas

Conocidas las herramientas para estructurar textos médicos se va a realizar un análisis comparativo de las comerciales ya que generar una herramienta capaz de estructurar textos en base a librerías descargables resultaría muy costoso y complejo.

En primer lugar se deben establecer los parámetros y características de las herramientas de PLN que se consideren más relevantes para la estructuración de textos médico. Esto facilitará la posterior comparación y decisión final acerca de qué herramientas resultan más interesantes en base a las necesidades de Naru Intelligence.

1. **Tipo de licencia y Términos de uso:** Conocer el uso que la compañía puede hacer de la información que ha sido procesada en su herramienta.
2. **Precio**
3. **Funcionamiento:** En este apartado se consideran distintos aspectos

relacionados con la capacidad de la herramienta para procesar textos de distinta naturaleza.

- Tareas: A la hora de estructurar textos médicos para realizar análisis de los datos obtenidos, las tareas de PLN más importantes son *NER*, *Tagging* y *RI*. Estas permiten extraer o agrupar conceptos para poder generar una adecuada estructura de datos que pueda procesarse (analizar o realizar búsquedas automáticas) posteriormente.
- Capacidad y tiempo de computación: la cantidad de información que la herramienta es capaz de procesar y el tiempo invertido en ello es importante. Aunque los textos no sean los archivos con mayor tamaño, al tratarse de grandes volúmenes de datos la capacidad y velocidad de la herramienta puede resultar un cuello de botella para el proceso.
- Idiomas compatibles: cuantos más idiomas entienda una herramienta mayor será la cantidad de textos que pueda procesar.
- Formato de los archivos: saber si la herramienta es capaz de extraer información tanto de textos electrónicos como manuscritos (muchas de las historias clínicas antiguas no se encuentran en formato digital) o archivos en otros formatos como audio. Además del formato de entrada el “output” (respuesta recibida por el usuario) se considera de suma importancia.
- Aplicaciones: algunas herramientas realizan selección de palabras en base a su naturaleza sintáctica pero a la hora de extraer información de textos médicos resulta interesante saber si se realiza una estructuración en base a grupos de conceptos médicos como enfermedades, síntomas o fármacos.
- Opiniones: conocer la opinión de un usuario puede ser interesante para conocer cómo funciona en la realidad la herramienta, ya que en

muchos casos las primeras impresiones no se asemejan a la realidad.

4. **Calidad**¹: Uno de los aspectos más importantes es comprobar el rendimiento de la herramienta en las tareas de extracción de entidades nombradas [137]. El impacto de proporcionar información errónea no es igual al impacto de olvidar algún dato. La evaluación a realizar será interna (en base al comportamiento dentro del ámbito para el que ha sido creada) e intrínseca (relacionada con el rendimiento de la herramienta en base al objetivo deseado) [50]. Se realizará un análisis de Sensibilidad-Especificidad ajustado a las necesidades del problema, ya que es el método más empleado cuando se realizan análisis de calidad de estas aplicaciones ([138] [5] [81] [96] [49] [122] por citar algunos).

La evaluación de calidad los sistemas de NER (Figura 4-1 se realiza mediante la comparación de la salida de estos con las anotaciones manuales. Esta comparativa será por coincidencia relajada: la clasificación se evalúa independientemente de la detección de límites (en la coincidencia exacta deben realizarse adecuadamente ambas tareas).

Para realizar la evaluación cuantitativa se definen los siguiente valores:

- Verdaderos Positivos (VP): entidades proporcionados que son relevantes.
- Falsos Positivos (FP): entidadess proporcionados que no son relevantes (sujeto a malinterpretaciones).
- Falsos Negativos (FN): entidades relevantes olvidados por la herramienta.
- Verdadenos Negativos (VN): entidades no relevantes que no han sido proporcionados (raramente son tenidos en cuenta en la evaluación de PLN [139]).

¹Este parámetro será evaluado en el Capítulo 6 tras seleccionar, en este capítulo, las herramientas más interesantes en base a los parámetros anteriores.

Una vez obtenidos estos valores, se procede a evaluar tanto la cantidad de las entidades recuperadas como la calidad de éstas en base a tres parámetros :

- Precisión (P): porcentaje de entidades proporcionadas que son relevantes.

$$\frac{VP}{VP + FP} \quad (4.1)$$

- Exhaustividad (R): porcentaje de entidades relevantes que han sido proporcionadas.

$$\frac{VP}{VP + FN} \quad (4.2)$$

- Valor-F (F-s): media armónica de P y R. Hace referencia a lo lejos que se encuentra la calidad del análisis realizado respecto de la calidad total (P=R=1).

$$\frac{2 \cdot P \cdot R}{P + R} \quad (4.3)$$

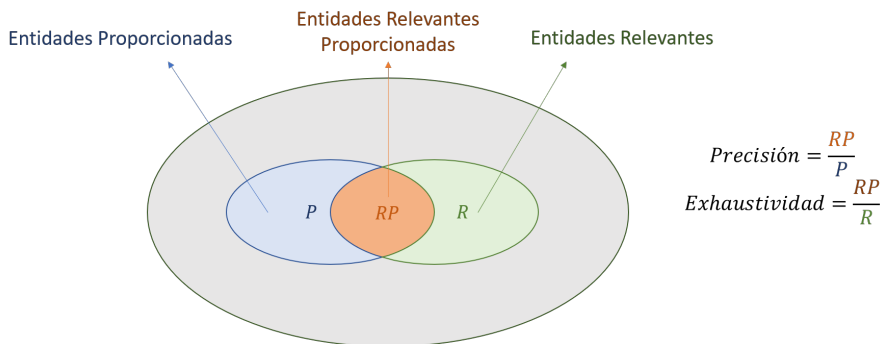


Figura 4-1: Parámetros de calidad analizados.

Este método es el más utilizado para evaluar la calidad los sistemas de PLN [139] [53]. En el capítulo 6 se añaden detalles de esta evaluación.

Todos los parámetros mencionados en este capítulo se han tenido en cuenta tanto en la comparativa cualitativa (Capítulo 5) como en la cuantitativa (Capítulo 6). No obstante, los mencionados en los puntos 1,2 y 3 de este capítulo (tipo de licencia y términos de uso, precio y funcionamiento) se han tenido en cuenta en mayor medida en la primera comparativa mientras que la calidad se ha estudiado más en profundidad en la segunda comparativa realizada.

Capítulo 5

Comparación cualitativa de las herramientas comerciales

Esta primera comparativa de las herramientas de PLN para estructurar textos médicos se divide en tres procesos de filtrado, cada cual más exigente. En cada uno de ellos se han considerado distintos criterios de los comentados en el capítulo 4.

5.1. Primera Selección

A medida que se realiza el rastreo de las distintas herramientas disponibles en el mercado, se han ido testeando las que ofrecían una *Demo* virtual y se ha analizado el uso o aplicaciones que pueden tener. Así, en esta primera selección se han elegido las herramientas más acordes al objetivo del proyecto: estructurar textos médicos. En esta lista se encuentran las que, en definitiva,

han llamado la atención y parecen ser útiles.

Algunas de las herramientas han sido descartadas sin ser analizadas en detalle por los siguientes motivos:

- No son capaces de realizar las tareas de etiquetado y reconocimientos de entidades (consideradas fundamentales para el objetivo concreto): se ha dado el caso de que alguna herramienta realiza análisis de sentimiento de alta calidad pero no realiza la función de NER.
- El idioma de los textos de entrada es muy limitado.
- La prueba realizada no ha detectado correctamente la información relevante del texto de entrada.
- Las funciones de la herramienta de PLN en textos médicos no están asociadas a estructurar textos médicos. En el caso de “Health Navigator”, por ejemplo, se realiza el PLN de texto coloquial de pacientes para obtener el motivo de la visita del paciente al médico [48], lo cual no concuerda con el objetivo.

Algunas de las herramientas descartadas en esta fase son:

- Bio Creative [140] y English Genia Corpus [141]: aplicadas en genes
- Tagger Corpus y jCORE&jPOS [142]: únicamente soportan textos en alemán.
- Ontotex: además de tener calidad limitada para reconocer entidades, solo es compatible con textos en inglés, francés, alemán y búlgaro [143].
- Spacy Named Entity Visualizer [144]: escasa calidad.
- BeCalm: está API reconoce 6 entidades únicamente, las cuales están relacionadas con genes y proteínas [145].

Las herramientas comerciales seleccionadas para ser evaluadas más a fondo en la siguiente fase son:

1. Amazon Comprehend Medical
2. IBM Watson
3. Dandelion
4. Repustate
5. Aylien
6. Text Razor
7. Open Calais
8. Savanna
9. IOMED
10. 3M Code Ryte Code Assist
11. Health Navigator
12. Google Cloud Natural Language
13. Azure Text Analytics
14. Lexalytics - Semantria

5.2. Segunda selección

En este punto se llevan a cabo rastreos de las páginas web de cada herramienta (o compañía) para conocer los parámetros que se consideran en la evaluación. A pesar de que algunos han sido considerados en la selección anterior, en esta fase se analizan los siguientes parámetros:

- Términos de uso y tipo de licencia
- Tareas
- NER del ámbito médico
- Precio
- Idiomas compatibles
- Capacidad de procesamiento

Algunas páginas web no han permitido obtener toda la información que se desea acerca de la herramienta pero ha sido posible hacerse una idea de las capacidades y limitaciones de cada una, permitiendo seleccionar las más interesantes.

En esta fase se han descartado las siguientes 6 herramientas:

- Health Navigator: la aplicación no se ajusta a las necesidades de estructuración de textos médicos ya que realiza PLN aplicado a quejas de clientes. Detecta el motivo de la visita de un paciente con gran precisión disminuyendo el tiempo invertido por el servicio de secretaría [146].
- Google Cloud Natural Language: Se ha considerado de peor calidad a la hora de estructurar textos médicos que otras herramientas disponibles ya que, a pesar de detectar los conceptos clave del texto, no los categoriza como se desea [129].
- Azure Text Analytics: A pesar de disponer de pantallazos de la biblioteca de Python descargable que dan buena impresión del funcionamiento de la herramienta, su aplicación parece estar más relacionada con el análisis de sentimiento de textos (comentarios u opiniones de usuarios) [128].
- Open Calais: se trata de un paquete disponible para usuarios de Microsoft Azure [136].

- 3M Code Ryte Code Assist: la empresa 3M ofrece un software capaz de detectar enfermedades y medicamentos de un texto médico y asignarles un código establecido. La presentación y aplicaciones son atractivas pero tanto la comunicación con la empresa como la extracción de información más exacta resultan complejas por lo que decide dejarse a un lado [135].
- Lexalytics - Semantria: la calidad de esta herramienta para estructurar textos en inglés es inferior a otras y cuando el texto de entrada se encuentra en español, no detecta ninguna entidad [147].

Las herramientas que han pasado este segundo filtro, además de no poseer limitaciones en cuanto a las políticas de privacidad y términos de uso (uso que hacen de la información de los textos procesados), realizan las tareas de reconocimiento de entidades nombradas y etiquetado. En la Tabla 5.1 se recopilan los parámetros más relevantes considerados en esta fase por herramienta.

Tabla 5.1: Parámetros de las herramientas en la selección 2 (Tareas:NER, Tagging y otras / EP: Entidades Personalizadas / PG:prueba gratuita / *:ofertas para estudiantes)

HERRAMIENTA	Funcionamiento		Limitaciones	Precio		Comentarios	
	Tareas	EP		Precio	Gratuito	Positivo	Negativo
	ACM	X		-	Solo para Inglés	0.0001\$/caracter	2.500 caract.
IBM Watson	X	X	-	0.0001\$/caracter (+extras)	300 mill. caract/mes	Multinacional con recorrido	Calidad
Dandelion	X	X	-	Gratis*	X	StartUp, descuento estudiantes	Poca experiencia
Reputate	X	X	Tamaño max 5MB	0.0099\$/documento	X	Precio, SDK, experiencia con HCE	Calidad dudosa
Aylien	X	-	Idioma limitado	0.0012\$/documento *	30 días	Calidad, SDK, descuentos, salida Excel	Idioma limitado
Text Razor	X	X	Tamaño max 100Kb	0.00004\$/KB	25MB/día *	Calidad, buena impresión	Tamaño documento

5.3. Tercera selección

La selección anterior ha servido para hacerse una idea del funcionamiento de cada herramienta, sin embargo, ésta idea ha de ser verificada ya que se ha recogido información de distintas fuentes pero no ha sido corroborada personalmente. Para ello se lleva a cabo el análisis más profundo de las herramientas seleccionadas. Además de analizar de manera exhaustiva la información de la página web, se realizan pruebas de testeo y se mide la calidad de cada una de ellas con breves fragmentos de textos médicos.

Al disponer de un número reducido de herramientas, resulta más sencillo focalizar la atención en los parámetros relevantes y realizar una comparación íntegra. En este punto se han considerado algunos parámetros que antes no habían sido evaluados en profundidad pero que son de gran importancia para evaluar el funcionamiento de las herramientas.

- Sin olvidar que el objetivo del proyecto es estructurar textos del ámbito médico, la posibilidad de etiquetar entidades y agruparlas en categorías médicas (bien sean fármacos, enfermedades, síntomas o efectos farmacológicos adversos) resulta muy interesante e incluso necesaria.
- ¿Cómo se ejecuta? Este aspecto había sido tenido en cuenta en cuanto a la privacidad de los datos, pero no se había considerado de suma importancia a la hora de hacer uso de la herramienta. Algunas herramientas ofrecen SDKs descargables para ejecutar en distintos lenguajes de programación mientras que otras ofrecen un código para ejecutar en la terminal (línea de comando).

Las 6 herramientas que se van a evaluar en esta selección se pueden dividir en dos grupos. Dos de ellas son productos de multinacionales (Amazon e IBM) mientras que las cuatro restante pertenecen a empresas más pequeñas y menos conocidas.

Amazon Comprehend ha obtenido mejores resultados que IBM Watson NLU en la prueba de testeo. Las tareas que esta herramienta dice realizar

resultan interesantes ya que es capaz de detectar, además de una serie de entidades médicas, atributos de éstas (dosis de medicamentos o valores de ciertos indicadores como el PSA entre otros). Esto es necesario para extraer información relevante de una historia clínica (pocas herramientas o ninguna ofrecen esta posibilidad). En la selección 2 se ha comentado la principal limitación de Amazon Comprehend Medical: el idioma. No obstante, la buena impresión de esta herramienta compensa la limitación de ser compatible únicamente con textos en inglés. IBM Watson ofrece servicios de ayuda en toma de decisiones como WFO o WFG (para oncología y genética). Para la toma de decisiones resulta necesario realizar las tareas de NER, tagging y RI previamente, por lo que se espera que esta herramienta sea de gran calidad. Además comenzó a desarrollarse antes que Amazon Comprehend, por lo que se deja a un lado la mala calidad del resultado obtenido en el testeo de la *demo*, considerando que se pueden obtener mejores rendimientos en caso de registrarse.

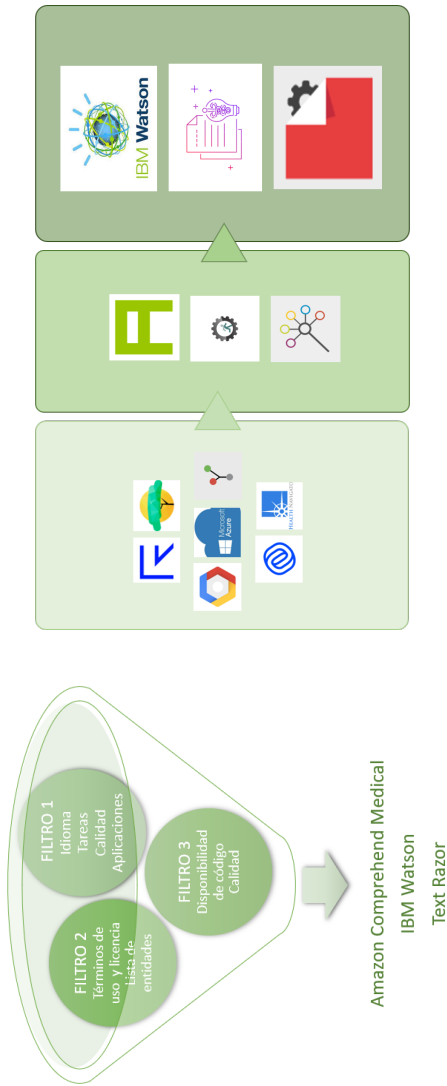
Tanto Repustate como Dandelion, una vez registrado el usuario, proporcionan un comando curl para ejecutar junto con una contraseña para llevar a cabo el etiquetado de entidades. Sin embargo, a la hora de ejecutar el código ambas dan como resultado el mismo error que parece estar relacionado con la imposibilidad de conectarse correctamente con la API. El corpus de Aylion y Dandelion es la base de datos DBPedia mientras que Text Razor utiliza conjuntamente DBPedia y FreeBase gracias a lo cual es capaz de ofrecer un mejor etiquetado de las entidades (es capaz de reconocer más términos ya que Freebase es más extensa que Wikipedia). Además, la respuesta del soporte técnico ha sido muy rápida y de calidad. Las 500 solicitudes diarias que permite procesar la licencia gratuita resultaron ser el principal inconveniente de esta herramienta en la selección 2. En esta fase, a la hora de realizar las pruebas, apenas se han superado las 150 solicitudes diarias por lo que 500 solicitudes resultan más que suficientes. La Tabla 5.2 recoge los resultados obtenidos en esta última selección:

Tabla 5.2: Parámetros de las herramientas en la selección 3.

HERRAMIENTA	DESCARTADA	MOTIVO
Amazon Comprehend		Buena calidad, multitud de posibilidades y detección de atributos
IBM Watson		Multinacional con recorrido y otros servicios médicos
Dandelion	X	Error de comunicación con la API (código curl)
Repustate	X	Error de comunicación con la API (código curl)
Aylien	X	Aplicada a análisis de noticias y menos entidades que TR
Text Razor		Calidad, precio, posibilidades y buen soporte técnico

5.4. Conclusión de la comparación cualitativa

En base a la comparación cualitativa llevada a cabo (véase Figura 5-1) y a falta de analizar el rendimiento, de todas las herramientas comerciales disponibles para estructurar textos médicos Amazon Comprehend Medical y Text Razor han resultado ser las más interesantes. IBM Watson NLU se considerada de interés debido a los servicios WFO y WFG desarrollados por esta multinacional, pues se cree que la tarea de NER debe formar parte de estos poniendo en duda la mala impresión de esta primera comparativa.



(a) Criterios evaluados en cada fase de filtrado. (b) Herramientas comerciales seleccionadas en cada filtrado.

Figura 5-1: Comparación cualitativa de las herramientas de PLN para estructurar textos médicos.

Capítulo 6

Comparación cuantitativa de las herramientas comerciales seleccionadas

6.1. Descripción de las Herramientas Seleccionadas

Tras seleccionar las herramientas más adecuadas para realizar estructuración de textos médicos acorde a este proyecto (ACM, TR e IBM Watson), estas han sido probadas para comprobar su funcionamiento y medir así el rendimiento. A continuación se detallan aspectos de cada una de las herramientas seleccionadas en el capítulo 5, así como el proceso llevado a cabo en el periodo de prueba.

6.1.1. Text Razor

Para realizar la prueba de esta herramienta ha sido necesaria la adquisición de la licencia gratuita con la que se disponía de 500 solicitudes (1 solicitud equivale a 50kB de texto) diarias. Una vez registrado, cada usuario dispone de una clave para ejecutar las solicitudes.

Se ha decidido descargar la SDK de Text Razor para Python (también está disponible en Java y PHP). La herramienta permite realizar distintas tareas pero este estudio únicamente se va a centrar en *NER* y *Tagging*. Se pueden modificar tanto el formato del texto de entrada (texto, url, html) como el idioma (en caso de no fijarse se detecta automáticamente). Además de estas, la herramienta permite fijar otra larga lista de parámetros.

Tras realizar el reconocimiento de entidades (definidas en las bases de datos de DBpedia y FreebaseDB), se almacenan distintos parámetros con información acerca de cada una de ellas. Los utilizados para extraer información de manera ordenada han sido:

- ID: nombre de la entidad
- DBpedia Types: nombre del tipo de entidad de la base de datos DBpedia a la que pertenece.
- Freebase Types: nombre del tipo de entidad de la base de datos Freebase a la que pertenece.¹
- Confidence Score: porcentaje de confianza de que la entidad es correcta.
- Matched Text: fragmento del texto que coincide con la entidad mencionada.

Esta herramienta permite la posibilidad de añadir un diccionario personal con todas las entidades deseadas. Para ello es necesario fijar el nombre del diccionario (ej.: Enfermedades), el ID de cada entidad (ej.: Cáncer) y el texto

¹Lista de tipos de entidades disponible en “www.textrazor.com/types”

que, en caso de ser detectado en el texto de entrada, se relacionad con esta entidad (ej.: “cancer”). A pesar de que la mayoría de entidades únicamente se relacionen con un texto, existe la posibilidad de relacionar más de un texto a un mismo ID . Esto resulta útil para detectar entidades con abreviaturas y almacenarlas en una única entidad (ej.:Diccionario=Constantes Vitales; ID=Tensión arterial; Texto1:“tensión arterial”; Texto2=“TA”). Para la licencia gratuita esta función está restringida a un único diccionario, por lo que no ha sido posible su utilización.

En un primer procesamiento realizado, las entidades médicas detectadas son de calidad ya que la precisión es alta . Sin embargo, hay bastante términos médicos que no han sido identificados por la herramienta (exhaustividad menor). En base a estos resultados, se procede a realizar el entrenamiento (modificación de la solución enviada por Text Razor) fijando la atención en mejorar el filtrado de términos para intentar detectar mayor cantidad de entidades relevantes mejorando así la exhaustividad.

Mejoras Realizadas - “Text Razor 2.0”

Con los resultados obtenidos en los primeros procesamientos, se ha procedido a estudiar en profundidad el funcionamiento de la herramienta para poder sacarle el máximo provecho y extraer la información de la mejor manera posible. A continuación se explican los pasos seguidos durante el entrenamiento de Text Razor²

En las entidades reconocidas inicialmente existen duplicados. En caso de que un término se repita a lo largo del texto, éste es almacenado tantas veces como aparece en el texto (de la misma manera que lo haría una nueva entidad). Para disponer de una lista de entidades no repetidas se realiza un filtrado de todas las detectadas.

Conociendo los distintos tipos de entidades disponibles en DBPedia y Free-

²Todas las mejoras mencionadas en este apartado se han realizado mediante distintos recursos de programación en Python.

base (véase Figura 6-1), para agrupar los medicamentos, enfermedades, tratamientos y síntomas, entre todas las entidades detectadas se ha realizado un filtro en base a los tipos ya definidas en las bases de datos mencionadas. Para aumentar la cantidad de entidades detectadas para cada grupo, se ha realizado el filtro en base a dos posibles tipos: DBpedia Types y Freebase Types. Mientras que a una entidad pertenece únicamente a un tipo de DBpedia, ésta entidad puede pertenecer a más de un tipo de entidad en Freebase (Figura 6-1). Es por ello por lo que hay entidades que se duplican y almacenan en dos grupos distintos (ej.: Anemia se almacena en “Enfermedades” y “Síntomas” ya que pertenece a los tipos “/medicine/disease” y “/medicine/symptom” (Figura 6-2)). Para evitarlo, además de disminuir el nº de FreebaseTypes relacionado con cada grupo, se han generado condiciones para que algunos términos sean asignados a su grupo correspondiente.

La paciente sufre **cáncer de mama**.

Words Phrases Relations **Entities** Meaning Dependency Parse

Entity	Confidence Score	Relevance Score	DBpedia Type	Freebase Type
Cáncer de mama (/m/0j8hd) (Q128581)	19.28	0.252	Disease	/fictional_universe/medical_condition_in_fiction /film/film_subject /organization/organization_sector /medicine/risk_factor /people/cause_of_death /medicine/disease /book/book_subject

Figura 6-1: Información de las entidades proporcionada por la demo de Text Razor.

Entity	Confidence Score	Relevance Score	DBpedia Type	Freebase Type
Anemia (/m/0icdk) (Q3445)	4.292	0.1447	Disease	/fictional_universe/medical_condition_in_fiction /book/book_subject /medicine/symptom /medicine/disease /medicine/risk_factor /people/cause_of_death /medicine/icd_9_classification

Figura 6-2: Información relacionada con la entidad “Anemia” proporcionada por la demo de Text Razor.

¿Y por qué no utilizar una única base de datos?

- Utilizar únicamente la base de datos Freebase no resolvería el problema

ya que se la superposición de tipos seguiría existiendo.

- Estructurar la información en base a DBPedia no es óptimo ya que la variedad de tipos de entidades es más reducida que la de Freebase (ej.: en la Figura 6-2 anemia pertenece a un único grupo (*Disease*) de DBPedia y a 7 de Freebase).

A medida que se han procesado los *training texts* se han realizado asignaciones manuales de entidades a grupos mediante condiciones en el código. En caso de que la herramienta no detecte una entidad, ésta es incluida manualmente en el grupo correspondiente.

Para obtener información relacionada con el paciente se ha llevado a cabo un procedimiento similar. Éste ha sido repetido para cada uno de los conceptos que se desea obtener: sexo, edad, peso, ocupación, estado civil y grupo sanguíneo.

El nombre de cada entidad detectada que se almacena es el *entity.id*. En el caso de que se detecte y almacene información acerca de “un dolor”, únicamente se almacena el término Dolor (Figura 6-3) ya que es el id asignado a esta entidad. Una mejora realizada es almacenar también, en caso de existir, el tipo de dolor que ha sido detectado en el texto.

ENFERMEDAD ACTUAL: **nauseas**, **dolor de cabeza**, **dolor** en hemitorax izquierdo, **odinofagia**, **fatiga**, **disnea** y **tos** húmeda productiva.

Entity	Confidence Score	Relevance Score	DBpedia Type	Freebase Type
Enfermedad (/m/027x3) (Q12139)	20.42	0.3496		/people/cause_of_death /visual_art/art_subject /organization/organization_sector /book/book_subject /internet/website_category /media_common/quotations_subject
Náusea (/m/0961c) (Q186689)	11.21	0.1114	Disease	/medicine/risk_factor /medicine/disease /medicine/symptom
Dolor (/m/062c2) (Q81938)	6.218	0.3266	Disease	/book/book_subject /medicine/icd_9_cm_classification /medicine/symptom /medicine/disease

Figura 6-3: Entidad “dolor” detectada por Text Razor.

En el caso de disponer de un texto con la dosis necesaria de cada medica-

mento, Text Razor detecta de manera independiente la entidad numérica como un número y el medicamento como un fármaco (Figura 6-4). Para aumentar la calidad de la información extraída, se ha procedido a almacenar tanto el nombre del medicamento como la dosis de este. Este mismo procedimiento se ha llevado a cabo con los distintos indicadores.

Deberá tomar **ibuprofeno 600 mg**

Words Phrases Relations **Entities** Meaning Dependency Parse

Entity	Confidence Score	Relevance Score	Ontopia Type	Freebase Type
ibuprofeno (/m/01443g) (Q198969)	4.707	0.2034	ChemicalSubstance ChemicalCompound	/medicine/drug_ingredient /medicine/drug /medicine/medical_treatment /chemistry/chemical_compound /medicine/condition_prevention_factors
600	0.5	0	Number	

Figura 6-4: Entidades detectadas por la demo de Text Razor.

Con la información almacenada en las distintas cadenas de Python, se ha generado un archivo Excel que hace de base de datos cuyas columnas son:

- Paciente: Información acerca del paciente como sexo, edad, peso, talla, ocupación, estado civil o grupo sanguíneo
- Medicamentos: nombre del medicamento y la dosis suministrada.
- Enfermedades
- Síntomas
- Tratamientos
- Indicadores: en caso de detectar algún indicador, este se almacena junto con el valor medido.

Con estos cambios se realiza de nuevo el procesamiento de todos los textos para comprobar si existe mejoría en los resultados al procesar “Text Razor” y “Text Razor 2.0”. Como se aprecia en la Tabla 6.1, la cantidad de entidades detectadas en los *training texts* ha aumentado hasta el 75 % mientras que

en los *test texts* se reconocen prácticamente las mismas entidades. Los resultados proporcionados al procesar los *training texts* han aumentado mientras que apenas existe diferencia alguna entre el procesamiento de los *test texts* con Text Razor y Text Razor 2.0 (Tabla 6.2). En general, la precisión ha aumentado un poco (+0.02) pero es la exhaustividad el parámetro que mayor mejoría ha obtenido (+0.66), lo que significa que la herramienta, tras haber sido entrenada, es capaz de reconocer mayor cantidad de términos médicos (sobre todo en los textos conocidos). El Valor-F aumenta hasta 0.79.

Tabla 6.1: Entidades relevantes detectas por Text Razor y Text Razor 2.0.

TEXTO	ENTIDADES RELEVANTES	HERRAMIENTA	
		Text Razor	Text Razor 2.0
TEST	Totales	10	11
	%	34.48	37.93
TRAIN	Totales	53	68
	%	58.24	74.73
TOTALES	Totales	63	79
	%	52.50	65.83

Tabla 6.2: Parámetros de calidad de Text Razor y Text Razor 2.0.

TEXTO	PARÁMETRO	HERRAMIENTA	
		Text Razor	Text Razor 2.0
TEST	VP	10	11
	FP	0	0
	FN	18	17
TRAIN	VP	53	69
	FP	2	1
	FN	38	23
TEST	P	1	1
	E	0.36	0.39
	V-F	0.53	0.56
TRAIN	P	0.96	0.99
	E	0.58	0.75
	V-F	0.73	0.85
TOTAL	VP	63	79
	FP	2	1
	FN	56	40
	P	0.97	0.99
	E	0.53	0.66
	V-F	0.68	0.79

6.1.2. Amazon Comprehend Medical

Esta herramienta es la que mejor impresión ha dejado en la comparación cualitativa. Los servicios ofrecidos por Amazon Web Services (AWS) son muy utilizados por multitud de empresas lo cual fortalece la impresión positiva generada. Amazon Comprehend Medical (ACM) es un servicio de PLN que facilita el uso de aprendizaje automático para extraer información médica relevante de textos médicos ofrecido por AWS. Utiliza modelos avanzados de ML y DL para extraer información que posteriormente puede ser tratada o procesada para distintas aplicaciones como toma de decisiones, análisis de datos o gestión de ensayos clínicos. Este servicio puede integrarse con otros servicios de AWS como S3 (repositorio de ficheros) o AWS Lambda.

El coste del servicio es proporcional a la cantidad de información procesada. Para realizar la prueba se han utilizado menos de los 2.5 millones de caracteres que se permiten procesar de manera gratuita por lo que no ha supuesto ningún coste.

El servicio ofrece dos API para integrar en distintas aplicaciones. Sin embargo, para procesar los textos y obtener resultados de manera sencilla y visual, a diferencia de con las otras dos herramientas probadas en las que se ha descargado un SDK, se ha utilizado la consola de Amazon Comprehend (véase la diferencia entre las Figuras 6-5, 6-6 y 6-7). Cada categoría está relacionada con un color y las entidades se codifican en base a estos.

Input text

Supported languages [↗](#)

The patient is a 40yo mother, highschool teacher
 HPI: sleeping trouble on present dosage of clonidine.
 Severe rash on face and leg, slightlu irctchy.
 Meds:
 -Vyvanse 50mgs po at breakfast (daily)
 -Clonidine 0.2mgs 1 and 1/2 tabs po qhs.
 HEENT: boggy inferior turbinates, no oropharyngeal lesion.

294 of 20000 characters used.

Clear text
Analyze

Insights info

Analyzed text

The patient is a **40yo** mother, **highschool teacher**
● Age (40yo) ● Profession (highschool teacher)

HPI: **sleeping trouble** on present dosage of **clonidine**.
● Dx name (sleeping trouble) ● Generic name (clonidine)

Severe **rash on** **face and** **leg**, **slightlu irctchy**.
● Dx name (rash) ● System organ site (face) ● System organ site (leg) ● System organ site (slightlu)

Meds: - **Vyvanse** **50mgs** **po** **at breakfast (** **daily**)
● Brand name (Vyvanse) ● Dosage (50mgs) ● Route or mode (po) ● Frequency (at breakfast) ● Frequency (daily)

- **Clonidine** **0.2mgs** **1 and 1/2 tabs** **po** **qhs**.
● Generic name (Clonidine) ● Strength (0.2mgs) ● Dosage (1 and 1/2 tabs) ● Route or mode (po) ● Frequency (qhs)

HEENT: **boggy inferior turbinates**, **no** **oropharyngeal lesion**.
● System organ site (HEENT) ● Dx name (boggy inferior turbinates) ● Direction (inferior) ● System organ site (turbinates) ● Dx name (oropharyngeal lesion)

▼ Results (14)

All
< 1 >
⚙

Entity	Type	Category	Traits
40yo 0.9994 score	● Age	Protected health information	-
highschool teacher 0.9858 score	● Profession	Protected health information	-
sleeping trouble 0.8142 score	● Dx name	Medical condition	Symptom 0.7631 score
clonidine 0.9978 score	● Generic name	Medication	-

Figura 6-5: Ejemplo de funcionamiento de la consola de Amazon Comprehend Medical (1).

6. Comparación cuantitativa de las herramientas comerciales 109

rash 0.9928 score	● Dx name	Medical condition	Symptom 0.7867 score
face 0.992 score	● System organ site	Anatomy	-
leg 0.9963 score	● System organ site	Anatomy	-
slightllu 0.3448 score	● System organ site	Anatomy	-
<input type="checkbox"/> Vyvanse 0.9995 score	● Brand name	Medication	-
— 50mgs 0.9791 score	● Dosage	Medication	-
— po 0.9997 score	● Route or mode	Medication	-
— at breakfast 0.9707 score	● Frequency	Medication	-
— daily 0.9852 score	● Frequency	Medication	-
<input type="checkbox"/> Clonidine 0.9982 score	● Generic name	Medication	-
— po 0.9997 score	● Route or mode	Medication	-
— 0.2mgs 0.8703 score	● Strength	Medication	-
— 1 and 1/2 tabs 0.6173 score	● Dosage	Medication	-
— po 0.9995 score	● Route or mode	Medication	-
— qhs 0.9983 score	● Frequency	Medication	-
HEENT 0.9895 score	● System organ site	Anatomy	-
boggy inferior turbinates 0.8377 score	● Dx name	Medical condition	Sign 0.9032 score
turbinates 0.6176 score	● System organ site	Anatomy	-
oropharyngeal lesion 0.8976 score	● Dx name	Medical condition	Sign, Negation 0.9497, 0.963 score
inferior 0.898 score	● Direction	Anatomy	-

► Application integration

Figura 6-6: Ejemplo de funcionamiento de la consola de Amazon Comprehend Medical (2).

```

{
  "Entities": [
    {
      "Id": 0,
      "BeginOffset": 10,
      "EndOffset": 14,
      "Score": 0.9876197576522827,
      "Text": "left",
      "Category": "ANATOMY",
      "Type": "DIRECTION",
      "Traits": []
    },
    {
      "Id": 1,
      "BeginOffset": 15,
      "EndOffset": 19,
      "Score": 0.9820258021354675,
      "Text": "lung",
      "Category": "ANATOMY",
      "Type": "SYSTEM_ORGAN_SITE",
      "Traits": []
    }
  ],
  "UnmappedAttributes": []
}

```

Figura 6-7: Respuesta de la API de Amazon Comprehend Medical en formato JSON.

Tras procesar los textos y recibir la respuesta en la interfaz web, la lista de entidades detectadas (archivo en formato “.json”) ha sido almacenado para futuros usos.

Esta herramienta permite procesar documentos individuales o por grupos de hasta 25 documentos, siendo el tamaño máximo de cada documento 20kb. El servicio no está disponible en todas las regiones y únicamente permite procesar texto en inglés ³.

Al procesar un texto se detectan entidades médicas como afecciones médicas, tratamientos, pruebas y resultados de las mismas, medicamentos (incluyendo atributos como dosis, frecuencia o vía de administración) o tratamientos.

De cada entidad se proporciona la siguiente información (Figura 6-8):

³Motivo por el cual todas las capturas de pantalla de ACM contienen el texto en inglés.

- Entidad: referencia textual al nombre del objeto relevante.
- Categoría: categoría generalizada a la que pertenece la entidad.
- Tipo: tipo de entidad detectada (dentro de la categoría).
- Atributo: información relacionada con la entidad.
- Rasgo: información adicional que detecta ACM (por ejemplo Negación de entidad: “sin metástasis”).
- Confianza: nivel de confianza en la entidad detectada. También se añade un nivel de confianza de la relación cuando se incluyen atributos de una entidad.
- Ubicación: límites de la entidad en el texto (consola gráficamente, en la API por valores numéricos)

```

{
  'Entidades' : [
    {
      'Id' : 123 ,
      'BeginOffset' : 123 ,
      'EndOffset' : 123 ,
      'Score' : ... ,
      'Text' : 'string' ,
      'Category' : 'MEDICATION' | 'MEDICAL_CONDITION' | 'PROTECTED_HEALTH_INFC
      'Type' : 'NAME' | 'DOSIS' | 'ROUTE_OR_MODE' | 'FORM' | 'FRECUENCIA'
      'Rasgos' : [
        {
          'Nombre' : 'SIGN' | 'SÍNTOMA' | 'DIAGNÓSTICO' | 'NEGACIÓN' ,
          'Puntuación' : ...
        }
      ],
    },
  ],
  'Atributos' : [
    {
      'Tipo' : 'NOMBRE' | 'DOSIS' | 'ROUTE_OR_MODE' | 'FORM' | 'FREC
      'Calificación' : ... ,
      'RelaciónPuntuación'
      ,
      'BeginOffset' : 123 ,
      'EndOffset' : 123 ,
      'Text' : 'string' ,
      'Traits' : [
        {
          'Name' : 'SIGN' | 'SÍNTOMA' | 'DIAGNÓSTICO' | 'NEGATION'
          'Score' : ...
        }
      ]
    }
  ]
}

```

Figura 6-8: Información de cada categoría enviada por ACM.

ACM diferencia cinco categorías y dentro de cada una se definen (en algunos casos) tipos de entidades, atributos y rasgos (Figura 6-9).

INFORMACIÓN DE CADA ENTIDAD DETECTADA POR CATEGORÍAS			
CATEGORÍA	TIPO	ATRIBUTOS	RASGOS
Anatomía	-Órgano	-Dirección (izda., dcha., lateral, dorsal, superior, ...)	
Condición Médica (síntomas o diagnóstico de afecciones)	-Nombre Dx	Agudeza (crónica, aguda, repentina, persistente, o gradual)	-Diagnóstico (causa/resultado de un síntoma) -Negación (resultado negativo o -Signo (informado por médico) -Síntoma (informado por
Medicamento	-Marca (nombre con -Genérico	-Dosis (cantidad ordenada) -Duración (tiempo que se debe -Forma -Frecuencia (de administración) -Tasa (de administración) -Vía (de administración) -Fuerza (de la medicación)	-Negación (indicación de que no se toma un medicamento)
Información Médica Personal	-Dirección -Edad -Correo -ID (seguridad social) -Nombre -Teléfono o Fax -Profesión		
Procedimiento Tratamiento Prueba	-Nombre Procedimiento: acción única Test: para medir un valor Tratamiento: intervención a lo largo de un tiempo)	-Valor (resultado de la prueba) -Unidad (acompaña al valor)	

Figura 6-9: Definición de categorías de entidades de Amazon Comprehend Medical.

Tras procesar un texto de neurología proporcionado por un experto en el área, Amazon Comprehend Medical ha detectado el 80 % de las entidades relevantes (algunas de ellas muy técnicas). En base al procesamiento realizado se han corroborado muchas de las funcionalidades que, en la página web, ACM dice realizar.

Además de detectar entidades médicas, la herramienta detecta negaciones de estas (Figura 6-10).



Figura 6-10: Ejemplos de negación de entidades de Amazon Comprehend Medical.

La identificación de relaciones de esta herramienta es de gran calidad. Detecta entidades junto con atributos cualitativos de pruebas realizadas como “valores de biopsia anormales”, “fluido cerebrospinal normal”, “niveles de neurofilamentos elevados” o “nódulo discreto” tanto en el texto como en la lista de entidades (Figura 6-11).

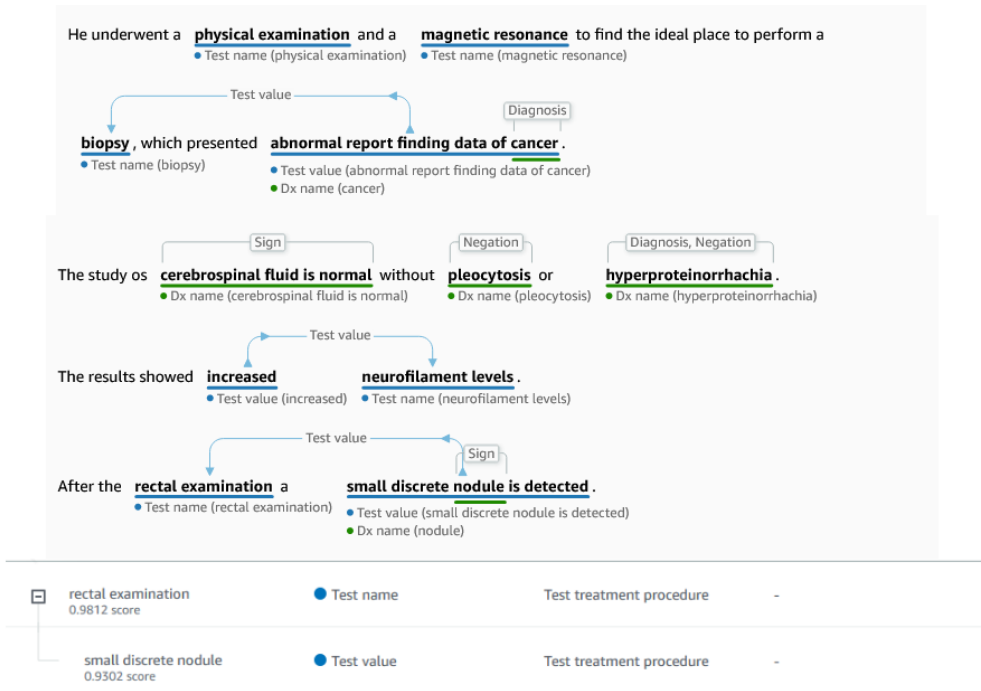


Figura 6-11: Relaciones en el texto de Amazon Comprehend Medical.

También es capaz de detectar atributos cuantitativos como el peso (Figura 6-12) o atributos de medicamentos (Figura 6-13).

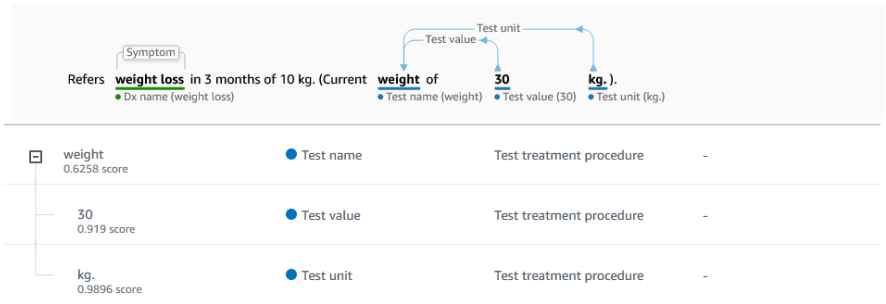


Figura 6-12: Atributos de la entidad “peso” detectados por Amazon Comprehend Medical.

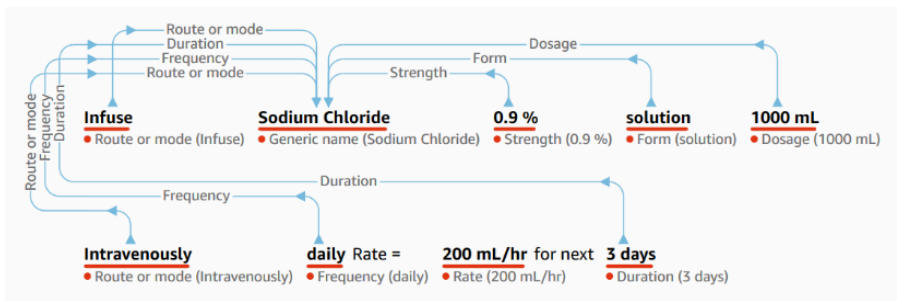


Figura 6-13: Atributos de medicamentos de Amazon Comprehend Medical.

Además de entidades relacionadas con el ámbito médico, Amazon Comprehend Medical permite detectar la información personal relacionada con los pacientes (Figura 6-14), lo cual resulta útil para relacionar un texto con el paciente.



Figura 6-14: Extracción de información personal de Amazon Comprehend Medical.

Las posibilidades que ofrece Amazon Comprehend Medical para extraer información de textos médicos desestructurados son numerosas y, como se verá en el apartado venidero, de gran calidad.

6.1.3. IBM Watson - Natural Language Understanding

La prueba de la herramienta IBM Watson NLU se ha realizado por tratarse de una multinacional que lleva varios años desarrollando soluciones de inteligencia artificial para la medicina. Una de las herramientas más conocidas (WFO) está destinada al soporte en la toma de decisiones en el ámbito oncológico. Para poder tomar decisiones respecto a un paciente y el tratamiento que le es asignado es necesario conocer la situación del paciente. Para ello es indispensable ser capaz de comprender la información expresada en los textos por lo que se ha considerado que la tarea de reconocimiento de entidades debe ser algo que Watson realice con gran rendimiento. Por este motivo se ha dejado a un lado la mala impresión dada en las primeras selecciones y se ha realizado la prueba de la versión gratuita.

Tras registrarse en IBM Cloud al usuario se le proporciona un código para realizar el procesamiento del texto. El plan gratuito permite el procesamiento de 300 millones de caracteres mensuales para la tarea de reconocimiento de entidades nombradas (en caso de querer realizar otras tareas esta cantidad se reduce a la mitad).

La herramienta permite realizar diversas tareas pero la prueba se ha centrado, al igual que en el resto de herramientas, en el reconocimiento de entidades. Ya que IBM Watson NLU detecta, además de entidades, palabras clave

en el texto, estas han sido consideradas en la solución a pesar de no poder realizar ningún tipo de agrupación ya que únicamente se detecta el fragmento de texto y no proporciona más información que su ubicación y el intervalo de confianza.

El procesamiento se realiza llamando a la interfaz de la aplicación a través de un archivo “.py” y el resultado se muestra en formato JSON (Figura 6-15).

Esta aplicación, a diferencia de TR e igual que ACM, no se ha podido “modificar” por lo que únicamente se ha llevado a cabo la generación de un archivo “.xls” para almacenar la lista de entidades y palabras clave obtenidas del texto.

Tras realizar el procesamiento de un texto con 23 entidades proporcionadas por el experto en neurología, IBM Watson apenas ha reconocido 1 entidad médica (Tabla 6.4 en la página 125). Se ha llevado a cabo el procesamiento del mismo texto en inglés pero el resultado apenas ha variado (3 entidades) por lo que el rendimiento no depende del lenguaje. A pesar de que se han detectado 24 palabras clave (entra las que se encuentran muchos de los conceptos que se desean estructurar), la utilidad de esta respuesta es muy limitada por la ausencia de información proporcionada.


```

C:\Users\Borja Quevedo\Desktop\Programas\IBMWatson>python IBMdetectarEntidadesKeywords2.py
TEXTO:
La paciente sufre cáncer de mama y será sometida a quimioterapia.
{
  "usage": {
    "text_units": 1,
    "text_characters": 65,
    "features": 2
  },
  "language": "es",
  "keywords": [
    {
      "text": "c\u00e1ncer de mama",
      "relevance": 0.942469,
      "count": 1
    },
    {
      "text": "paciente",
      "relevance": 0.357554,
      "count": 1
    },
    {
      "text": "quimioterapia",
      "relevance": 0.098839,
      "count": 1
    }
  ],
  "entities": [
    {
      "type": "HealthCondition",
      "text": "c\u00e1ncer",
      "relevance": 0.963296,
      "mentions": [
        {
          "text": "c\u00e1ncer",
          "location": [
            18,
            24
          ]
        }
      ]
    },
    {
      "count": 1
    }
  ]
}
C:\Users\Borja Quevedo\Desktop\Programas\IBMWatson>

```

Figura 6-15: Respuesta JSON de la herramienta de IBM Watson.

6.2. Materiales y Métodos

Tras realizar la comparativa de todas las herramientas que hacen uso de PLN para la extracción de información de un texto, en este apartado se procede a analizar la calidad de las herramientas Text Razor, Amazon Comprehend Medical e IBM Watson para la estructuración de textos médicos. En concreto se evalúan la capacidad de realizar las tareas de extracción de entidades nombradas (NER), etiquetado (Tagging) e identificación de relaciones (RI).

La Figura 6-16 resume el proceso llevado a cabo para testear las herra-

mientas de estructuración de textos médicos y evaluar la calidad de estas.

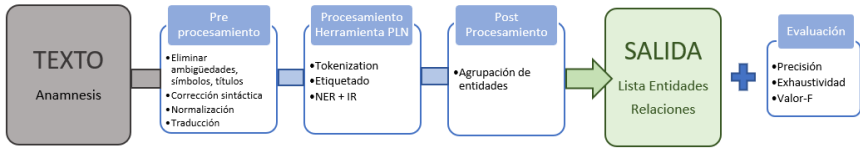


Figura 6-16: Proceso de evaluación de la calidad de las herramientas.

6.2.1. Datos

Los textos a procesar por las herramientas son anamnesis de neurología y oncología, información aportada por el paciente y por otros testimonios para confeccionar su historial médico. Estos textos se han dividido en dos grupos.

- *Training Text*: conjunto de textos utilizados para realizar el entrenamiento de las herramientas. Conociendo el resultado que se desea obtener, estos textos se utilizan para realizar modificaciones en el código (en caso de ser posible) con el objetivo de ir mejorando el resultado proporcionado por las herramientas.
- *Test Text*: conjunto de textos utilizados para comprobar la calidad de la herramienta en la realidad. Estos textos son “desconocidos” para la herramienta, ya que nunca antes han sido procesados por la misma.

Los *training texts* son fragmentos de historias clínicas de pacientes de oncología (se han obtenido cuatro ejemplos anónimos [148]). Un experto en neurología ha proporcionado seis textos generados por él que simulan anamnesis de pacientes de neurología, los cuales han sido utilizados como *test texts*.

6.2.2. *Gold Standard*

Las evaluaciones del rendimiento de sistemas informáticos para medicina se basan en la respuesta generada manualmente por expertos y conocida como *Reference Standard* o *Gold Standard* [149].

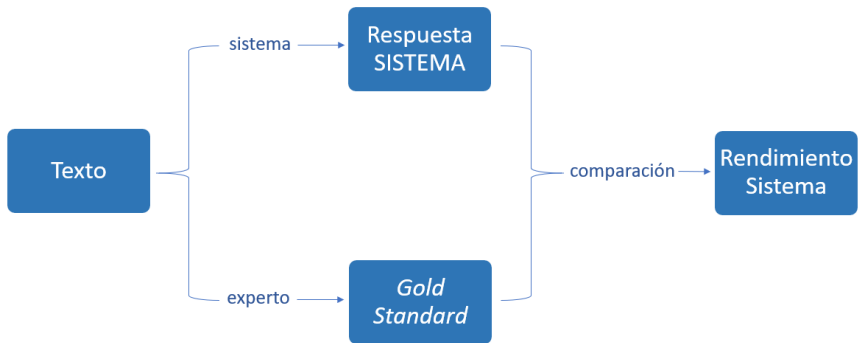


Figura 6-17: *Gold Standard* en la evaluación de calidad de sistemas informáticos.

Resulta primordial disponer de un modelo adecuado en base a la respuesta que se desea obtener por parte de la herramienta de PLN para conocer la calidad de esta [150], [116], [47], [151].

En el caso de los *training texts* el *Gold Standard* ha sido generado por el estudiante y revisado posteriormente por personas con conocimiento médico mientras que en el caso de los *test texts* ha sido el neurólogo quien ha proporcionado una lista de entidades que desea obtener de las anamnesis neurológicas. Éstas se han agrupado de manera manual para poder definir los tipos de entidades médicas (síntomas, enfermedades, indicadores (genes y proteínas), medicamentos, tratamientos) que se consideran importantes. En base a estos grupos, se ha generado de manera manual el modelo *Gold Standard* (solución que desea obtenerse de cada uno de los textos procesados) que servirá para

realizar la evaluación de calidad de las herramientas.

6.2.3. Pre-Procesamiento

Los *training texts* han sido preprocesados. Esta tarea ha consistido en la revisión de la ortografía, normalización de los términos médicos (se ha comprobado que todos estuvieran escritos correctamente) y la eliminación de símbolos ambiguos. Se han mantenido números y letras mayúsculas. Para comprobar el rendimiento en un ámbito lo más real posible los *test texts* no han sido modificados sino que se han procesado tal como se recibieron del neurólogo (cabe destacar que son textos libres sin caracteres ambiguos). Para comparar el rendimiento de las herramientas en el procesamiento de textos en distintos idiomas, los textos han sido traducidos haciendo uso del traductor *DeepL* disponible online [152].

6.2.4. Entrenamiento y Procesamiento

La herramienta Text Razor ha sido entrenada con los *training texts* ya que el código proporcionado por la empresa puede ser modificado (el proceso llevado a cabo se explica en el apartado 6.1.1). Esto no ocurre con Amazon Comprehend Medical e IBM Watson NLU, con las que únicamente se han procesado los distintos textos sin modificar el código. Se desea conocer el funcionamiento de los productos ofrecidos por las compañías y modificar las herramientas afectaría a su rendimiento, por lo que para estas herramientas tanto los *training* como los *test texts* resultan desconocidos a la hora de ser procesados. Los textos son procesados por las 3 herramientas comerciales así como por la versión “Text Razor 2.0” entrenada.

6.2.5. Post-Procesamiento

Una vez se detectan las entidades, se filtran las entidades que se desean obtener (relacionadas con la medicina) y se agrupan adecuadamente. Estas

entidades se almacenan en archivos excel para facilitar la posterior comparación de las respuestas. Las entidades detectadas que no están relacionadas con la medicina y, por lo tanto, no se consideran relevantes, han sido dejadas de lado pero su detección no se ha considerado un “error” de la herramienta.

6.2.6. Evaluación de Calidad

Para evaluar la calidad de las herramientas se comparan los resultados obtenidos por las herramientas con el *Gold Standard* de cada texto y se lleva a cabo el análisis de Especificad-Exhaustividad explicado en la sección 4 (página 83). Se han llevado a cabo dos evaluaciones. En el caso de la evaluación “No Exhaustiva” se ha contabilizado como VP aquellas entidades que contengan al menos un término de la entidad solución del *Gold Standard*. En el caso de la evaluación “Exhaustiva” (utilizada para obtener las conclusiones finales) únicamente se contabiliza como VP aquellas entidades obtenidas que coinciden completamente con las entidades solución del *Gold Standard* (en el siguiente apartado se analizan únicamente los resultados obtenidos tras la evaluación “Exhaustiva” ya que se consideran los más relevantes y aplicables a la realidad).

El número de VP,FP y FN contabilizados ha sido en base a una clasificación binaria en la que una entidad está bien detectada o no. En caso de que una entidad aparezca dos veces en la respuesta, esta ha sido contada como una única y no se penalizada dicha duplicidad (aun y todo se ha programado la solución con el objetivo de evitar duplicados).

6.3. Resultados

Para calcular la cantidad de entidades que cada herramienta reconoce se procesan los textos médicos y se comparan las entidades detectadas con las definidas como relevantes en el *Gold Standard* sin tener en cuenta la agrupación. En secciones venideras se evalúa también la identificación de relaciones entre

dichas entidades y la detección de información expuesta en el texto relacionada con las entidades reconocidas como atributos de medicamentos, gravedad de enfermedades o entidades formadas por múltiples términos entre otras.

Las tablas 6.3 y 6.4 recogen los resultados obtenidos tras el procesamiento de los textos en las distintas herramientas.

6.3.1. Parámetros de calidad

Tabla 6.3: Entidades relevantes detectadas por cada herramienta.

TEXTO	ENTIDADES RELEVANTES DETECTADAS	IDIOMA					
		INGLÉS			ESPAÑOL		
		IBM	TR	AWS	IBM	TR	AWS
TEST	Totales	3	16	23	1	10	0
	%	10.34	55.17	79.31	3.45	34.48	0
TRAIN	Totales	28	54	82	7	53	29
	%	30.77	59.34	90.11	7.69	58.24	31.87
TOTAL	Totales	31	70	105	8	63	29
	%	25.83	58.33	87.50	6.67	52.50	24.17

Tabla 6.4: Parámetros de calidad de las herramientas.

TEXTO	PARÁMETRO	IDIOMA					
		INGLÉS			ESPAÑOL		
		IBM	TR	AWS	IBM	TR	AWS
TEST	VP	3	16	23	1	10	0
	FP	2	0	4	0	0	11
	FN	25	12	5	27	18	28
TRAIN	VP	28	54	82	7	53	29
	FP	2	2	1	2	2	5
	FN	63	37	9	84	38	62
TEST	P	0.60	1	0.85	1	1	0
	E	0.11	0.57	0.82	0.04	0.36	0
	V-F	0.18	0.73	0.84	0.07	0.53	0
TRAIN	P	0.93	0.96	0.99	0.78	0.96	0.85
	E	0.31	0.59	0.90	0.08	0.58	0.32
	V-F	0.46	0.73	0.94	0.14	0.73	0.46
TOTAL	VP	31	70	105	8	63	29
	FP	4	2	5	2	2	16
	FN	88	49	14	111	56	90
	P	0.89	0.97	0.95	0.80	0.97	0.64
	E	0.26	0.59	0.88	0.07	0.53	0.24
	V-F	0.40	0.73	0.92	0.12	0.68	0.35

Tabla 6.5: Resumen de los parámetros obtenidos.

IDIOMA	HERRAMIENTA	PARÁMETRO DE CALIDAD		
		Precisión	Exhaustividad	Valor F
INGLÉS	IBM Watson	0.89	0.26	0.40
	Text Razor	0.97	0.59	0.73
	Amazon CM	0.95	0.88	0.92
ESPAÑOL	IBM Watson	0.80	0.07	0.12
	Text Razor	0.97	0.53	0.68
	Amazon CM	0.64	0.24	0.35
MEDIA	IBM Watson	0.84	0.16	0.26
	Text Razor	0.97	0.56	0.70
	Amazon CM	0.78	0.56	0.63

6.3.2. Análisis Complementario

Una vez analizados los rendimientos de las tres herramientas se ha decidido dejar a un lado IBM Watson. A pesar de que Amazon Comprehend Medical ha resultado ser la que mejor estructuración de textos médicos realiza, se ha decidido comprobar su comportamiento y el de Text Razor en diversas situaciones que, se cree, pueden darse de manera frecuente (texto procesado en inglés):

- Abreviaturas: muy comunes en los textos médicos [153].
- Correcciones gramaticales: como errores ortográficos o inversión del orden de letras en una palabra.
- Medicamentos

Las Tablas 6.6 y 6.7 recogen una lista de términos conflictivos⁴ y si han sido o no detectados por las herramientas ACM y TR.

⁴Se han seleccionado aquellos que están más relacionados con la actividad de Naru Intelligence (oncología).

Tabla 6.6: Términos “conflictivos” detectados por ACM y TR (1).

DIFICULTAD	TEXTO PROCESADO	ENTIDAD	¿DETECTADO?	
			ACM	TR
ABREVIATURAS	BP	Tensión Arterial	x	-
	HTN	Hipertensión	-	-
	QT	Quimioterapia	-	-
	QAD	Quimioterapia intensiva o a alta dosis	-	-
	CIS	Carcinoma in situ	x	-
	PSA	Prueba de antígeno prostático específico	x	-
	CAM	Medicina alternativa y complementaria	-	-
	LFT	Prueba de función hepática	x	-
		Prueba de función pulmonar	x	-
	AF	Fibrilación auricular o aleteo	x	-
	sats	Saturaciones (ej.: oxígeno)	x	-
	AP	Anteroposterior	-	-
		Abdominoperineal	-	-
	IHD	Enfermedad hisquémica de corazón	x	-
	RC	Remisión completa	-	-
	RP	Remisión parcial	x	-
EE	Enfermedad estable	-	-	
NKDA	Alergia desconocida a medicamento	-	-	
ERRORES GRAMATICALES	artralgý	Artralgia	-	-
	bipsy	Biopsia	x	-
	chemotherapy	Quimioterapia	x	-
	colpscopy	Colposcopía	x	-
	hyperglucemy	Hiperglicemia	x	-
	masectomy	Mastectomía	x	-

Tabla 6.7: Términos “conflictivos” detectados por ACM y TR (2).

DIFICULTAD	TEXTO PROCESADO	ENTIDAD	¿DETECTADO?	
			ACM	TR
DISLEXIAS	canecr	Cáncer	x	-
	lymphadenectmoy	Linfanectomía	x	-
	sugrery	Cirugía	x	-
	metsatasis	Metástasis	x	-
	nodlue	Nódulo	x	-
	osteosarocma	Osteosarcoma	x	-
MEDICAMENTOS	avastin	Avastin	x	x
	aldesleukina	Aldesleukina	x	-
	carboplatin	Carboplatin	x	x
	chlorpromazine	Clorpromazina	x	x
	dacarbazine	Dacarbazina	x	x
	dexamethasone	Dexametasona	x	x
	granisetron	Granisetron	x	x
	ifosfamide	Ifosfamida	x	x
	irinotecan	Irinotecan	x	x
	lorazepam	Lorazepam	x	x
	methotrexate	Metotrexato	x	x
	opdivo	Opdivo	x	x
	ordasentron	Ordasentron	x	-
	sylatron	Sylatron	x	x
toremifine	Toremifeno	x	-	
ENTIDADES CONFLICTIVAS DETECTADAS			35 / 45	12 / 45

6.4. Discusiones

6.4.1. Discusión por herramienta

Text Razor

La precisión de esta herramienta es bastante elevada ya que se devuelven las entidades pertenecientes al grupo “Medicina” y apenas aparecen FP. Sin embargo, los FP detectados son significativos ya que, por ejemplo, en los textos procesados detecta la entidad “metástasis” pero no su negación. No es capaz de detectar todas las entidades obteniendo un valor-f de 0.73. La información proporcionada en la respuesta es limitada (el rendimiento de Text Razor cuando se realiza una evaluación menos exhaustiva aumenta hasta 0.86). Esta herramienta es, con diferencia, la más adecuada para procesar textos médicos en español hoy en día ya que su rendimiento no depende en gran medida del idioma.

Text Razor 2.0

Los resultados obtenidos al procesar los *training texts* en español tras el entrenamiento de Text Razor han mejorado ya que la herramienta ha sido entrenada con estos. Sin embargo, al procesar los *test texts* se ha obtenido una respuesta muy similar a la proporcionada por la herramienta Text Razor (sin entrenar) y lo mismo ocurre con los *training texts* en inglés. La cantidad de entidades relevantes en los *test texts* es menor que en los *training texts* por lo que el valor-f medio (tras procesar todos los textos) de TR 2.0 es 0.1 mayor que el de TR.

Amazon Comprehend Medical

Cuando se procesan textos médicos en inglés el rendimiento de esta herramienta es muy elevado. Ha obtenido un V-F=0.92 por lo que resulta ser,

con diferencia, la herramienta más adecuada para estructurar textos médicos. Los pocos FP detectados son poco significativos ya que se tratan de términos no relacionados con la medicina. No obstante, únicamente está preparada para procesar textos en inglés ya que, al analizar el mismo texto en español el rendimiento sufre un descenso notable (sobre todo cuando se trata de textos más técnicas como los *test texts* de neurología). Además de detectar un gran número de entidades es capaz de detectar los atributos de estas y obtener relaciones entre las entidades, proporcionando información detalla del texto de entrada. La interfaz de esta herramienta resulta útil ya que permite observar las relaciones entre los términos (detectadas con gran precisión) tanto en el propio texto como en la lista de entidades que proporciona. Además, la API puede integrada en otros servicios ofrecidos por AWS de manera sencilla.

IBM Watson

Apenas detecta el 25 % de las entidades totales. Es capaz de detectar el 30 % de entidades relacionadas con oncología y apenas el 10 % cuando se trata de textos con vocabulario más complejo de neurología. Esto puede ser debido a que IBM está mas desarrollada en este área [154]. A pesar de que la precisión es alta (0.89), la exhaustividad es muy baja (0.26). Se podría decir que las pocas entidades que es capaz de detectar están relacionadas con la medicina, pero el valor-f obtenido es muy bajo (0.4), por lo que esta herramienta no es adecuada para detectar entidades médicas en textos. Cuando se procesan textos en español apenas es capaz de reconocer alguna entidad.

6.4.2. Reconocimiento de Entidades Nombradas (NER)

Amazon Comprehend Medical es la herramienta que mayor cantidad de entidades es capaz de reconocer, seguida de Text Razor y a años luz de IBM Watson (Table 6.3).

6.4.3. Identificación de Relaciones (IR)

Amazon Comprehend Medical es la herramienta que mejor detecta las relaciones entre distintas entidades reconocidas (véase Figuras 6-11 en la página 115). La diferencia de rendimiento con el resto de las herramientas en esta tarea es muy grande (influenciado también por la diferencia de calidad en tareas de NER).

6.4.4. Idioma

Todas las herramientas presentan mejores rendimientos al procesar textos en inglés. Sin embargo, los resultados de Text Razor son los que menos varían al analizar textos en inglés y español. Esto puede deberse a que se trate de la única herramienta que utiliza métodos de coincidencia léxica para detectar entidades. En ese caso sería capaz de detectar términos definidos en las ontologías DBpedia y Freebase disponibles en distintos idiomas.

6.4.5. Temática

Todas las herramientas detectan mayor cantidad de entidades relacionadas con la oncología (*train*) que con la neurología (*test*) lo cual se debe a la complejidad de los términos. Las herramientas reconocen entidades definidas en el *corpus*. Detectar enfermedades comunes como “cáncer” resulta más sencillo que reconocer otras menos habituales y específicas de cada rama de la medicina como “tricotilomanía” (sentir el deseo de arrancarse cualquier rastro de pelo del cuerpo).

6.4.6. Interfaz y tiempo de procesamiento

Además de la calidad se evalúan parámetros que, una vez trabajado con las herramientas, ha sido posible medir y también se consideran importantes.

Tanto la **interfaz** como la forma de mostrar los resultados de Amazon Comprehend Medical resulta la más útil para que el usuario pueda obtener la información del texto (Figuras 6-5 y 6-6 en las páginas 108 - 109). Además del formato visual de la interfaz, permite obtener la lista de entidades detectadas en formato JSON que puede ser utilizado para analizar o procesar esta respuesta. IBM y Text Razor ofrecen interfaces muy visuales en sus versiones de prueba pero al ejecutar las herramientas con Python la respuesta se proporciona, únicamente, en formato JSON (Figuras 6-18 y 6-19).

Examine a news article or other content

Text URL

El paciente sufre cáncer de mama.

Spanish

For results unique to your business needs consider building a [custom model](#)

* This system is for demonstration purposes only and is not intended to process Personal Data. No Personal Data is to be entered into this system as it may not have the necessary controls in place to meet the requirements of the General Data Protection Regulation (EU) 2016/679.

By using this application, you agree to the [Terms of Use](#)

Analyze

Sentiment Emotion Keywords Entities Categories Concept Syntax

Semantic Roles

Extract people, companies, organizations, cities, geographic features, and other information from the content.

[JSON](#)

Name Type Score

cáncer HealthCondition 0.88

(a) Demo disponible Online.

```
C:\Users\Borja_Quevedo\Desktop\Programas\IBMWatsonPython\IBHdetect
La paciente sufre cáncer de mama y será sometida a quimioterapia.
{
  "usage": {
    "text_units": 1,
    "text_characters": 85,
    "features": 2,
    "languages": "es",
    "keywords": [
      {
        "text": "cúebnincen de mama",
        "relevance": 0.942469,
        "count": 1
      },
      {
        "text": "paciente",
        "relevance": 0.357554,
        "count": 1
      },
      {
        "text": "quimioterapia",
        "relevance": 0.498839,
        "count": 1
      }
    ],
    "entities": [
      {
        "type": "HealthCondition",
        "text": "cúebnincen",
        "relevance": 0.963286,
        "mentions": [
          {
            "text": "cúebnincen",
            "location": [
              18,
              24
            ]
          }
        ]
      }
    ]
  }
}
```

(b) Código adquirido.

Figura 6-18: Respuesta de la herramienta IBM.

Incontinencia urinaria

Words Phrases Relations **Entities** Meaning Dependency Parse

Entity	Confidence Score	Relevance Score	DBpedia Type	Freebase Type
Incontinencia urinaria (/m/038h3) (Q281440)	4.485	0.0528	Disease	/medicine/risk_factor /medicine/symptom /medicine/disease /business/consumer_product

Picor (prurito)

Words Phrases Relations **Entities** Meaning Dependency Parse

Entity	Confidence Score	Relevance Score	DBpedia Type	Freebase Type
Picor (prurito) (/m/046lmg) (Q199602)	2.058	0.3168	Disease	/medicine/symptom /medicine/nd_3_cm_classification /medicine/disease
Picor (prurito) (/m/046lmg) (Q199602)	1.018	0.08161	Disease	/medicine/symptom /medicine/nd_3_cm_classification /medicine/disease

Menstruación

Words Phrases Relations **Entities** Meaning Dependency Parse

Entity	Confidence Score	Relevance Score	DBpedia Type	Freebase Type
Menstruación (/m/0259nc3) (Q12171)	4.054	0		/film/film_subject /biology/genus_ontology_group /medicine/risk_factor /book/book_subject /biology/genus_group /medicine/disease_cause

(a) Demo disponible Online.

```
#IMPORTACIÓN DE LIBRERÍAS
import textazor
import numpy as np
import pandas as pd

#API "TEXTRAZOR"
#Se llama a la API y extrae las entidades
textazor.api_key = 'f229ca146940857ee4877b7602b6cd69f3000b29e9cc06018cfe638d'
client = textazor.TextRazor(extractors=["entities","words"])
response=client.analyze('La paciente sufre cáncer de mama y será sometida a quimioterapia')
response.entities()
for entity in response.entities():
    print(response.entities())

[TextRazor Entity b'cxc3\xa1ncer' at positions [3], TextRazor Entity b'cxc3\xa1ncer de mama' at positions [3, 4, 5], TextRazo
n Entity b'quimioterapia' at positions [10]]
[TextRazor Entity b'cxc3\xa1ncer' at positions [3], TextRazor Entity b'cxc3\xa1ncer de mama' at positions [3, 4, 5], TextRazo
n Entity b'quimioterapia' at positions [10]]
[TextRazor Entity b'cxc3\xa1ncer' at positions [3], TextRazor Entity b'cxc3\xa1ncer de mama' at positions [3, 4, 5], TextRazo
n Entity b'quimioterapia' at positions [10]]
```

(b) Código adquirido.

Figura 6-19: Respuesta de la herramienta TR.

Los textos procesados apenas superan los 2.000 caracteres por lo que el **tiempo de procesamiento** no se ha podido evaluar. En el caso de procesar una cantidad elevada de textos tampoco se considera un parámetro relevante ya que, al tratarse de caracteres su procesamiento es rápido (en comparación con otros formatos de archivos). Por muy elevado que fuera el tiempo de procesamiento de estas herramientas se trataría de una cantidad muy pequeña en comparación con el tiempo dedicado a analizar los datos obtenidos.

6.4.7. Método de funcionamiento

Los resultados del procesamiento explicado en la subsección 6.3.2 muestran que:

- Amazon Comprehend Medical: A pesar de no ser capaz de detectar algunas abreviaturas, reconoce todas las entidades mal escritas así como los medicamentos.
- Text Razor: no detecta ninguna abreviatura ni entidad escrita de manera incorrecta pero si que reconoce 12 de los 15 medicamentos utilizados en oncología (posiblemente definidos en las ontologías Freebase o DBpedia).

El efecto del método utilizado para detectar entidades en el rendimiento de las herramientas es evidente. Cuando Text Razor procesa texto “complejo” (con errores gramaticales o abreviaturas) la capacidad de detectar entidades nombradas disminuye. Esto ocurre en menor medida en el caso de Amazon Comprehend Medical, la cual utiliza modelos de ML y DL más versátiles. Es muy común que los textos médicos no sean precisos y contengan erratas (véase sección 1.8). Además la cantidad de términos médicos no deja de crecer ya que, fruto de las investigaciones y nuevas soluciones en esta área, cada día aparecen nuevos medicamentos, enfermedades o tratamiento.

6.5. Conclusión de la comparación cuantitativa

Tras la comparación cuantitativa realizada, la herramienta de PLN para realizar estructuración de textos médicos que mayor rendimiento ha obtenido es Amazon Comprehend Medical. El servicio de Amazon Web Services detecta prácticamente todas las entidades médicas contempladas en el *Gold Standard* y proporciona gran parte de la información relacionada con estas.

Text Razor se considera la herramienta adecuada para estructurar textos médicos en Español (u otro idioma compatible con TR) ya que ACM únicamente reconoce textos en Inglés. En caso de poder hacer uso de un traductor la mejor solución es realizar un preprocesamiento del texto (traducción) para procesarlo con Amazon Comprehend Medical.

IBM Watson NLU no se ha desarrollado en tareas de NER por lo que no es aconsejable su uso para realizar estructuración de textos médicos.

Tras realizar el “entrenamiento” de las herramientas se obtienen mayores rendimiento al procesar textos conocidos. Esta mejora desaparece cuando los textos procesados son desconocidos por lo que las mejoras realizadas se consideran inútiles para aplicaciones futuras.

Tabla 6.8: Ránking de las herramientas analizadas por parámetros evaluados.

HERRAMIENTA	Calidad			INTERFAZ	PRECIO
	NER	IR	En Español		
Amazon Comprehend	1	1	2	1	3
Text Razor	2	3	1	2	1
IBM Watson	3	3	3	2	2

Solución comercial para estructurar textos médicos

Se espera que Amazon Comprehend Medical sea mejorada pudiendo procesar textos en cualquier idioma. Hasta entonces, la solución para estructurar textos médicos pasa por utilizar ACM junto con un traductor (herramienta de PLN) cuando el texto de entrada esté en un idioma distinto al anglosajón.

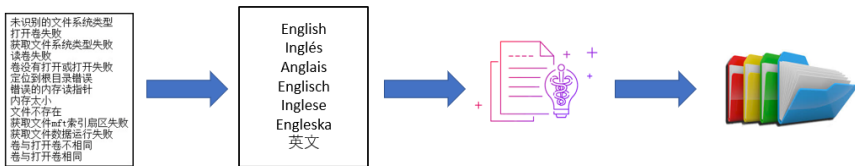


Figura 6-20: Método óptimo para estructurar textos médicos (opinión personal).

Capítulo 7

Limitaciones de las herramientas seleccionadas y alternativas

Además de comentar las desventajas de estructurar textos con cada una de las herramientas comerciales, en este capítulo se especifican una serie de soluciones a las limitaciones detectadas que se considera podrían mejorar los resultados obtenidos. Las líneas futuras de desarrollo de las herramientas de PLN para estructurar textos están relacionadas con estas limitaciones y sus diversas soluciones.

7.1. Text Razor

- Text Razor realiza tareas de NER y etiquetado pero no identifica relaciones entre las entidades ni devuelve información detallada de estas. La herramienta está preparada únicamente para reconocer entidades

mediante coincidencia léxica siendo incapaz de detectar negaciones de estas. Los falsos positivos generados al reconocer términos sin atender al contexto pueden ser significativos (véase conclusiones Text Razor en página 130) .

- A la hora de etiquetar las entidades reconocidas, estas se agrupan en grupos definidos en “DBpedia” y “FreeBase”. Tras procesar los textos algunas entidades han sido duplicadas en la respuesta final ya que el filtrado se ha realizado en base a ambas ontologías (véase página 102). Para solucionar este duplicado bastaría con realizar la agrupación en base a una única base de datos (se recomienda elegir FreeBase para aumentar la cantidad de entidades detectadas. No obstante, permanece cerrada desde el año 2016 por lo que podría quedar obsoleta, siendo incapaz de reconocer entidades surgidas tras su cierre). Esta solución acarrearía la disminución de los parámetros de calidad al disminuir la cantidad de entidades reconocidas. Generar un número infinito de condiciones para detectar todas las entidades deseadas podría solucionar este problema pero resulta imposible de programar.
- El problema anterior está relacionado con la imposibilidad de esta herramienta de comprender el significado del texto. Text Razor utiliza coincidencia léxica para detectar las entidades y la única manera de ser capaz de reconocer el significado que cada palabra tiene en el texto sería modificar el funcionamiento interno de la herramienta. Este cambio supondría un cambio completo de Text Razor, tratándose, pues, de un problema irresoluble.
- A pesar de que es capaz de detectar entidades en distintos idiomas, este aspecto resulta un problema cuando se desea “individualizar” la herramienta para un caso concreto. La versión mejorada de Text Razor (TR 2.0) ha sido programada para procesar texto en Español por lo que en caso de querer procesar un texto en un idioma diferente, los

resultados serían idénticos a los obtenidos por Text Razor.

7.2. Amazon Comprehend Medical

- El principal problema de esta herramienta es el idioma ya que únicamente está preparada para procesar textos en inglés (véase comparativa de calidad por idiomas en la Tabla 6.5 de la página 126). Si se desea estructurar un texto en otro idioma basta con preprocesar el texto mediante un traductor y post-procesar la respuesta obtenida para obtener la estructuración en el idioma del texto original. Esta solución puede alterar la respuesta ya que la traducción, en caso de no ser de calidad, modificaría el significado del texto original así como la respuesta recibida por el usuario.
- A la hora de detectar atributos de medicamentos listados en una misma línea (separados por espacios en blanco) se han detectado malinterpretaciones en la respuesta: dos medicamentos distintos han sido relacionados con una misma dosis. Se recomienda separar los medicamentos mencionados consecutivamente en del texto de entrada mediante saltos de línea y no mediante espacios en blanco en una misma frase (Figura 7-1).
- Otro aspecto a considerar es el coste de procesamiento. Este es proporcional a la cantidad de texto procesado, 0.0001\$ cada caracter (una cara completa aproximadamente: 600 palabras, 3.000 caracteres, 0.3\$). Se trata de la herramienta más costosa.

Las limitaciones de Amazon Comprehend Medical en comparación con TR e IBM-NLU resultan mínimas. La interfaz de la web es muy útil para estructurar textos ya que subraya las entidades por colores y muestra las relaciones en el texto, además de proporcionar un listado con la información relevante detectada por cada entidad (véanse Figuras 6-5 en la página 108 y 6-6 en 109). No obstante, esta interfaz no sirve para



Figura 7-1: Forma óptima de redactar medicamentos en un texto procesado por ACM.

realizar la estructuración de una gran cantidad de textos (necesidad de Naru Ingelligence) ya que el *input* es introducido manualmente por el usuario. Para poder extraer información de un conjunto de textos es necesario disponer del código para conectarse con el servicio a través de un programa como Python o Java ¹ A pesar de que el objetivo de este proyecto es estructurar un texto médico cualquiera con rendimientos elevados, relacionado con el uso de esta herramienta por parte de Naru Intelligence se considera de interés la posibilidad de procesar grandes

¹Este problema lo tienen todas las herramientas pero se comenta en este apartado ya que, al tratarse de la herramienta más adecuada para estructurar textos, es la limitación más destacada.

cantidades de texto almacenando la información detectada en ellos de manera automática. En el capítulo 8 se explica la solución propuesta con la que es posible procesar archivos de texto ubicados en una carpeta y almacenar la información relevante detectada en una base de datos MySQLite3 de manera automática.

7.3. IBM Watson - Natural Lenguaje Understanding

- Tal y como ha dado impresión en la comparativa cualitativa, el rendimiento de IBM Watson NLU para reconocer entidades nombradas en un texto deja mucho que desear (más teniendo en cuenta la calidad de otras herramientas analizadas). A pesar de tratarse de una multinacional con numerosos servicios en el ámbito de la salud (como WFO o WFG), no ha sido desarrollado en este área. Se omite comentar las limitaciones de una herramienta que se considera no ha sido trabajada por sus creadores tanto como otras.

Capítulo 8

Metodología de estructuración de textos médicos para Naru Intelligence

8.1. Solución propuesta

En este capítulo se propone una solución para estructurar textos médicos según las necesidades de la empresa Naru Intelligence.

La herramienta de PLN seleccionada para ello es Amazon Comprehend Medical. Tras procesar el texto de entrada, la consola de AWS ofrece una respuesta muy visual que facilita la comprensión de la información relevante del texto. No obstante, para los analistas de datos (Naru Intelligence entre ellos) esta interfaz carece de utilidad. ACM proporciona a su vez un archivo JSON con toda la información relevante filtrada del texto, siendo este archivo

el que resulta de interés para la propuesta realizada.

El proceso descrito a continuación y resumido en la Figura 8-1 permite estructurar una cantidad indefinida de textos (".txt") y almacenar la información detectada por ACM (tanto entidades como información relacionada con estas) en una base de datos (MySQLite3).

La metodología de estructuración de textos médicos para Naru Intelligence consta de los siguientes pasos:

1. En primer lugar se genera una base de datos MySQLite3 (descrita en la Figura 8-2) diseñada en base a las categorías de entidades definidas por ACM (véase Figura 6-9 en la página 113).
2. Se procesan los textos almacenados en una carpeta con Amazon Comprehend Medical (la conexión con ACM se realiza a través de unas pocas líneas de código), obteniendo como respuesta los archivos ".json" en los que se recoge la información relevante detectada.
3. Estos archivos generados por ACM son procesados mediante un código personal (programado en Python) que almacena la información en la Base de Datos.

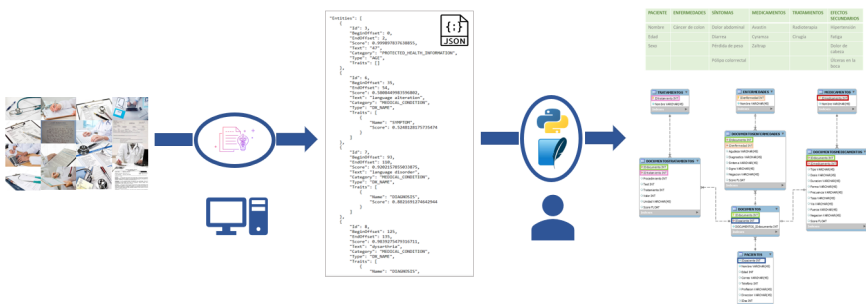


Figura 8-1: Estructuración de textos médicos con Amazon Comprehend Medical.

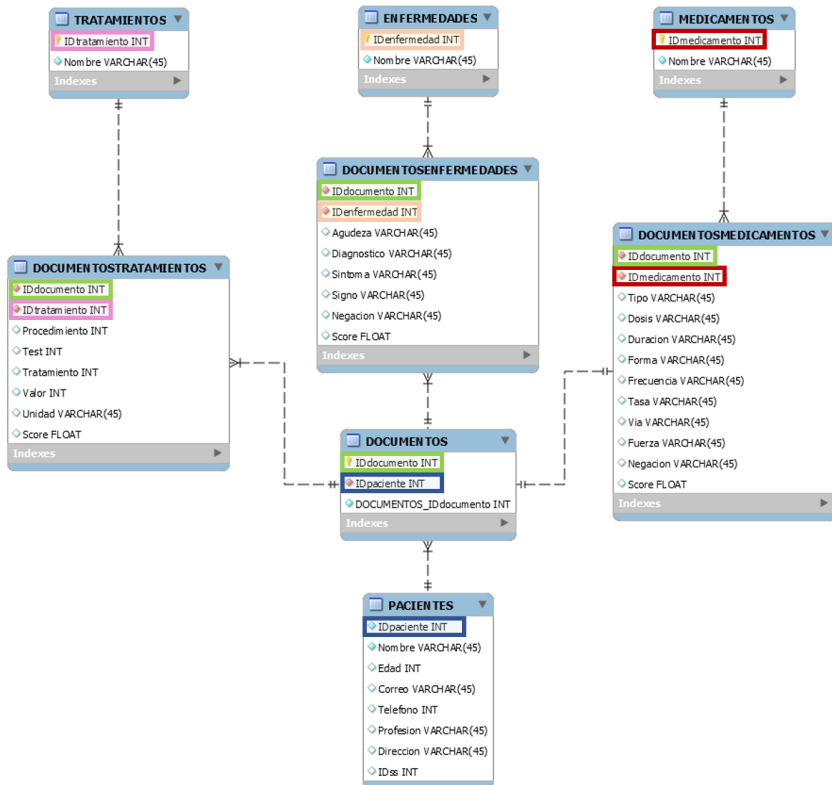


Figura 8-2: Tablas de la Base de Datos donde se almacena la información estructura por ACM.

8.2. Mejoras de la herramienta comercial seleccionada

A pesar de la gran calidad de la interfaz, a la hora de trabajar con la respuesta en formato JSON se han detectado aspectos a mejorar.

- Identificación de entidades: cuando una entidad es detectada, no se le

asigna un código identificativo sino que únicamente se almacena el fragmento de texto detectado así como la información relacionada con esta. Esto hace que comparar documentos y analizar, por ejemplo, la frecuencia de aparición de entidades en estos, resulte complejo. Las entidades “estenosis” y “estnosis” pueden ser almacenadas como independientes. Para resolver este problema se ha utilizado la función “*SequenceMatcher*” del módulo *diffib* [155] que devuelve un valor entre 0 y 1 proporcional a la similitud entre dos cadenas de caracteres. Cada entidad detectada se compara con las entidades almacenadas en su tabla correspondiente (“Enfermedades”, “Medicamentos” o “Tratamientos”) y únicamente se almacena como “nueva” entidad si el valor devuelto por *SequenceMatcher* es menor de 0.9.

- Sentencias: tras generar la Base de Datos y rellenarla con la información detectada, se han generado instancias para poder obtener datos de manera sencilla. Entre las sentencias destacan las que permiten:
 - o Conocer la lista de medicamentos, tratamientos y enfermedades detectadas por documento.
 - o Representación gráfica de la cantidad de entidades detectadas por documento, divididas por categorías de entidades.
 - o Las entidades más frecuentes en los documentos procesados, separadas por grupos.
 - o Identificar los textos en los que aparece una entidad específica, así como la información relacionada con esta en cada uno de ellos.

Capítulo 9

Conclusiones de la estructuración de textos médicos con herramientas comerciales

Acerca de las herramientas comerciales disponibles para estructurar textos (obtener información útil, de manera ordenada y estructurada) se ha concluido lo siguiente (véase Tabla 9.1 donde se resume este capítulo):

- **No existe servicio informático para estructurar grandes cantidades de textos.**

A pesar que se lleven realizando intentos de aplicar el PLN para la estructuración de textos médicos (véanse Figura 2-1 en la página 48 y Figura 2-4 en la página 62) desde hace tiempo [156] [157], no existe una capaz de realizar esta tarea para cualquier texto médico.

- **TAREAS - Estructurar no es etiquetar.**

Muchas de las herramientas que a priori dicen estructurar textos no lo hacen. Es verdad que realizan reconocimiento y etiquetado de entidades pero existe una diferencia considerable entre detectar términos y estructurar la información de un texto.

Bastantes herramientas son capaces de realizar Reconocimiento de Entidades Nombradas (con mejor o peor calidad). No obstante, Amazon Comprehend Medical es la única herramienta que detecta la información relacionada con las entidades reconocidas, así como las relaciones entre estas. Identificar relaciones entre entidades es fundamental para estructurar textos. Esta tarea resulta incluso más importante que la de NER ya que la falta de información puede acarrear una mal interpretación de las entidades detectadas.

Cuanto más técnico (específico de un tema concreto) sea el vocabulario del texto, peor será la estructuración realizada. Se ha observado que la capacidad de las herramientas para estructurar textos es proporcional a la frecuencia de uso en el día a día de las personas.

- **COMPRESIÓN SINTÁCTICA - Solo algunas de las herramientas basadas en ML y DL comprenden el significado del texto.**

Muchas herramientas se topan con la dificultad de comprender el significado de las entidades detectadas, siendo incapaces de diferenciar una enfermedad mencionada como antecedente familiar de una sufrida por el paciente. Una cosa es detectar y clasificar sintácticamente las entidades del texto y otra estructurar la información plasmada en el mismo de manera adecuada (para lo que resulta necesario comprender el significado del texto). Solo algunas de las herramientas basadas en aprendizaje automático (ACM) son capaces de comprender el texto procesado. ¹

¹Relacionado con la falta de herramientas capaces de reconocer la información de un

- **IDIOMA - estructuración de calidad de textos en inglés.**

A pesar de que la lista de idiomas soportados por las herramientas es larga, el idioma del texto condiciona la calidad de la estructuración. Todas las herramientas obtienen mejores resultados cuando se procesa texto en inglés y la más interesante (ACM) únicamente es compatible con este idioma. Cuando se desea estructurar un texto escrito en un idioma diferente al anglosajón la mejor opción es hacer uso de un traductor. La traducción resulta necesaria tanto para procesar el texto como para realizar el post-procesamiento de la respuesta obtenida por la herramienta, ya que el usuario desea recibir la información en el mismo idioma que el texto enviado. En este caso, a pesar de disponer una herramienta de estructuración de calidad, la respuesta se puede ver alterada ya que las traducciones no están exentas de posibles malinterpretaciones.

- **FORMATO - Las herramientas disponibles permiten procesar textos en formato electrónico.**

La existencia de herramientas de PLN para convertir texto manuscrito a formato informático permite la posibilidad de estructurar textos escritos a mano. En estos casos (al igual que en la traducción) la calidad del pre-procesamiento influye en el resultado de la estructuración.

La mayoría de herramientas devuelven un archivo en formato “.json”, útil para analistas de datos pero poco intuitiva para el usuario. Solo algunas de las herramientas disponen de una interfaz de salida visual que aporta beneficios a un usuario sin conocimientos informáticos.

- **COSTE.**

La mayoría de paquetes descargables son gratuitos pero algunas de las

texto, cabe poner en duda el funcionamiento de los sistemas de apoyo a la decisión clínica (CDSS) existentes. Estos necesariamente deberían realizar una función de NER previa a su tarea principal ya que la información médica debe ser comprendida para poder actuar en base a ella. No obstante, se ha observado que algunas empresas que ofertan CDSSs no ofrecen servicios de NER de calidad, lo cual pone en duda la precisión de estas soluciones de soporte.

herramientas que mejor estructuran textos tienen un coste. El procesamiento de 10.000 caras (6.000.000 palabras, 30.000.000 caracteres, 300.000 unidades.^a 0.01 % la unidad) con Amazon Comprehend Medical supone un coste total de 3.000 dólares. Coste aproximado ACM para procesar una cara: 0.3\$.

- **PRIVACIDAD - asegurada en todas las herramientas.**

En los términos de uso de las herramientas disponibles se asegura no utilizar la información procesada para obtener beneficio personal, por lo que la privacidad de la información procesada (aspecto fundamental y más en el ámbito médico) no corre peligro.

- **INDIVIDUALIZACIÓN - Se puede generar una herramienta “a medida” en función de la información que se desea extraer de un conjunto de textos concreto.**

A pesar de no existir una herramienta capaz de estructurar cualquier texto (independientemente del tema, idioma o formato), es posible entrenar un código para que sea capaz de estructurar la información de un número reducido de textos con un rendimiento elevado. Esto ha sido realizado en multitud de casos con distintas aplicaciones. Sin embargo, este proceso no es trabajo de la computadora sino que es el programador el encargado de crear el código en base al texto de entrada. La calidad de esta estructuración disminuye considerablemente cuando el texto de entrada es “desconocido” (como ha ocurrido con Text Razor 2.0 al procesar los *test texts*).

La Tabla 9.1 recoge las capacidades de cada herramienta comercial evaluada en relación con las conclusiones comentadas en este capítulo.

Tabla 9.1: Conclusiones de las Herramientas Comerciales evaluadas.

ASPECTO CONSIDERADO		TR	ACM	IBM
Tareas		NER	NER + RI	NER
Comprensión sintáctica		No	Si	No
Idioma		Varios	Inglés	-
Formato	Input	Electrónico	Electrónico	Electrónico
	Output	JSON	JSON + Interfaz	JSON
Coste		0	0.3	0
Privacidad		Si	Si	Si
Individualización		Si	No	No

Conclusión General

El objetivo de este proyecto era analizar las herramientas de estructuración de textos médicos. Se ha comprobado que existen un gran número de librerías, paquetes y programas informáticos que hacen uso del PLN para obtener información de un texto de distintas maneras. No obstante, las herramientas más avanzadas tienen su aplicación en la extracción de sentimientos, etiquetado de entidades (nombres, ciudades, fechas) o traducción. Estas aplicaciones resultan de gran utilidad pero no sirven para el objetivo del proyecto. Algunas de ellas pueden ser aplicadas a la estructuración de textos pero se ha observado que la dificultad de esta tarea es mayor de la que se pensaba en un primer momento.

Generar una herramienta capaz de estructurar un texto específico de manera adecuada ha sido posible. Sin embargo, generar una herramienta capaz de estructurar cualquier texto de manera adecuada resulta muy complejo ya que, además de detectar las entidades, debe ser capaz de entender el significado de estas en su contexto. Además, la estructura de la respuesta variará en función de la utilidad que se desee dar a los datos por lo que generar una

herramienta para estructurar cualquier texto en base a cualquier necesidad es, a día de hoy, imposible.

Sin embargo, hay herramientas como Amazon Comprehend Medical que detectan entidades nombradas de un texto con una gran precisión y son capaces de extraer la información relevante de los mismos. Podría decirse que ACM es una herramienta que sirve para estructurar textos médicos pero, ¿qué le falta? El texto es introducido manualmente por el usuario y no es posible almacenar la información de manera automática, pudiendo obtener una base de datos que contenga la información “histórica” procesada (esto resulta imprescindible para realizar la estructuración de un gran conjunto de textos de forma “rápida”). Amazon debería plantearse ofrecer un servicio de post procesamiento capaz de almacenar la información detectada en base a los requerimientos del usuario (esto resultaría de gran utilidad para Naru Intelligence) y más disponiendo de servicios de almacenamiento de información como S3.

A pesar de no ser igual a la estructuración que una persona podría realizar, llevando a cabo un proceso adecuado y haciendo uso de la herramienta de PLN adecuada, tal y como se ha comprobado, es posible extraer la información relevante de un texto médico de manera automática.

Beneficios personales

Además de cumplir con los objetivos del proyecto, gracias a los compañeros con los que he tenido la suerte de trabajar y a los inesperados problemas contra los que he tenido que lidiar, desde el comienzo en Septiembre he tenido la posibilidad de mejorar como persona y adquirir conocimientos que considero muy útiles para mi futuro.

Formar parte de Naru Intelligence me ha permitido conocer de primera mano la dureza del emprendimiento así como la fortaleza de la ilusión. Trabajar en equipo no es solo realizar trabajos conjuntamente sino contar con la herramienta más útil cuando realizas una tarea individual: el compañero.

La planificación y estructuración realizada al comienzo no tiene nada que ver con la que finalmente se lleva a cabo. Sin embargo, y a pesar de los cambios que van surgiendo sobre la marcha, planificar es imprescindible para avanzar hacia el objetivo final sin estancarse. “Invertir una semana en aprendizaje puede reducir un mes el tiempo de trabajo total”. Empezar antes no conlleva concluir antes ya que, si no se disponen de los conocimientos necesarios, será más costoso seguir el trazado planificado.

La lectura de artículos me ha permitido aprender cómo aprender de ellos, tanto de la estructura (compartida por todos los que tratan temas relacionados) como del contenido de estos.

Bibliografía

- [1] <http://www.diegocalvo.es/aprendizaje-no-supervisado/>. Online; Accedido: 07/11/2019.
- [2] https://es.wikipedia.org/wiki/Perceptr%C3%B3n_multicapa. Online; Accedido: 7/11/2019.
- [3] UPM-Centro de Apoyo a la Innovación Tecnológica. Big data en salud. 2015.
- [4] Marcela Fontana. Seguridad y evitabilidad de errores de la receta médica: una revisión del tema.
- [5] Michel Riveill Romaric Pighetti Edson Florez, Frederic Precioso. Named entity recognition using neural networks for clinical notes. 2018.
- [6] <https://searchdatacenter.techtarget.com/es/definicion/Computacion-cognitiva-o-computo-cognitivo>. Online; Accedido: 8/11/2019.
- [7] <https://www.salesforce.com/mx/blog/2017/6/Que-es-la-inteligencia-artificial.html>. Online; Accedido: 8/11/2019.
- [8] <http://www.json.org/>. Online; Accedido: 21/11/2019.

- [9] Iñaki Inza y Abdelmalik Moujahid Pedro Larrañaga. Tema 8. redes neuronales. *Departamento de Ciencias de la Computación e Inteligencia Artificial*, UPV - EHU.
- [10] Iker García Ferrero. Estudio de word embeddings y métodos de generación de meta embeddings. 2018.
- [11] Kesey O'neil y Dawson Friedland MOhamed Nooman Ahmed, Andeep S. Toor. Cognitive computing and the future of health care. 2017.
- [12] Susan Eggly Louis A. Penner Rifky Tkatch Jennifer Vichich y Terrance L. Albrecht Lauren M. Hamel, Robert Chapman. Measuring the use of examination room time in oncology clinics: A novel approach to assessing clinic efficiency and patient flow. 2014.
- [13] https://www.consalud.es/profesionales/medicina/medicos-exigen-aumentar-10-minutos-tiempo-medio-paciente-consulta_62874_102.html. Online; Accedido: 25/09/2019.
- [14] Flatiron Health. Flatiron health 's bold proposition to fight cancer with big data. <https://fortune.com/2014/06/12/flatiron-healths-bold-proposition-to-fight-cancer-with-big-data/>, June 2014. Online; Accedido: 18/09/2019.
- [15] Universidad de Alcalá. <https://www.master-data-scientist.com/importancia-analisis-datos-en-medicina/>. Online; Accedido: 20/09/2019.
- [16] <https://www.oreilly.com/ideas/lessons-learned-building-natural-lang> Online; Accedido: 24/09/2019.
- [17] <https://ghr.nlm.nih.gov/primer/precisionmedicine/initiative>. Online; Accedido: 03/10/2019.

- [18] Tancredi DJ Romano PS Bold RJ y Joseph JG Whitney RL, Bell JF. Hospitalization rates and predictors of rehospitalization among individuals with advanced cancer in the year after diagnosis. 35(31):3610–3617, 2017.
- [19] Hayat MJ y Atkins RL. Fessele KL. Predictors of unplanned hospitalizations in patients with nonmetastatic lung cancer during chemotherapy. *Oncol Nurs Forum*, 44(5), 2017.
- [20] Kim A Cloutier M Gauthier-Loiselle M Gagnon-Sanschagrín P y Guerin A Wong W, Yim YM. Assessment of costs associated with adverse events in patients with cancer. 13(4), 2018.
- [21] Angela Meekins Marie Topor-Elizabeth B. Lamont y Martin L. Brown Joan L. Warren, K. Robin Yabroff. Evaluation of trends in the cost of initial cancer treatment. *Journal of the National Cancer Institute*, 100(12):888–897, 2018.
- [22] Alan García. <https://www.the-emag.com/theitmag/blog/computacion-cognitiva-es-una-realidad>. Online; Accedido: 19/90/2019.
- [23] <https://gestion.pe/tecnologia/inteligencia-artificial-historia-orig> Online; Accedido: 19/09/2019.
- [24] <https://www.universidadviu.es/la-computacion-cognitiva/>. Online; Accedido:19/09/2019.
- [25] Micro Focus. Augmented inteligente: Helping humans make smarter decisions.
- [26] Shai Ben-David Shai Shalev-Shwartz. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

- [27] <https://www.indracompany.com/es/blogneo/deep-learning-sirve>. Online; Accedido: 7/11/2019.
- [28] Lu X Gao K Jiang M Xu H. Lei J, Tang B. A comprehensive study of named entity recognition in chinese clinical text. *J Am Med Inform Assoc*, 2014.
- [29] Y. Bengio y G. Hinton Y. LeCun. Deep learning. 2015.
- [30] https://es.wikipedia.org/wiki/Lenguaje_natural. Online; Accedido=17/09/2019.
- [31] <http://procesamientolenguajerecuperacion.50webs.org/>. Online; Accedido=17/09/2019.
- [32] https://es.wikipedia.org/wiki/Procesamiento_de_lenguajes_naturales. Online; Accedido=17/09/2019.
- [33] Jaime Priona Quispe Augusto Cortez Vásquez, Hugo Vega Huerta. Procesamiento de lenguaje natural. 2009.
- [34] <https://sistemas.com/lenguaje-natural.php>. Online; Accedido=17/09/2019.
- [35] <http://www.iic.uam.es/inteligencia/aplicaciones-procesamiento-lenguaje-natural/>. Online; Accedido=17/09/2019.
- [36] Fernando A. Navarro. La precisión del lenguaje en la redacción médica. *Penacea*.
- [37] Malcom Marsh. Algunas consideraciones sobre la traducción médica. <https://cvc.cervantes.es/lengua/aproximaciones/marsh.htm>. Online; Accedido: 18/09/2019.

- [38] Jorge Comas Francesc Solsona Rui Alves Anabel Usié, Joaquim Cruz. Chener: a tool for the identification of chemical entities and their classes in biomedical literature. *Journal of Cheminformatics*, 2015.
- [39] Servicios Sociales e Igualdad de España Ministerio de Sanidad. Cie-10-es, clasificación internacional de enfermedades. 2018.
- [40] JL Oliveira D Campos, S Matos. *Theory and Applications for Advanced Text Mining*. InTEch.
- [41] Riikka Vuokko, Päivi Mäkelä-Bengs, Hannele Hyppönen, Minna Lindqvist, and Persephone Doupi. Impacts of structuring the electronic health record: Results of a systematic literature review from the perspective of secondary use of patient data. *International Journal of Medical Informatics*, 97:293 – 303, 2017.
- [42] Juan José Puello Fuentes. Importancia de las estructuras de datos. Fundación Universitaria Tecnológico de Información.
- [43] ¿Cómo beneficia el Big Data a la medicina? <https://www.amirsalud.com/beneficio-big-data-medicina>, 2019. Online; Acceso: 17/09/2019.
- [44] Reenita Das. The flatiron health aquisition. <https://www.forbes.com/sites/reenitadas/2018/02/26/flatiron-health-acquisition-a-shot-in-the-arm-for-roches-oncology-#7dc4975f3f60>, February 2018. Online; Acceso: 17/09/2019.
- [45] Katherine Noyes. Flatiron health's bold proposition to fight cancer with big data. <https://fortune.com/2014/06/12/flatiron-healths-bold-proposition-to-fight-cancer-with-big-data>, June 2014. Online; Accedido: 17/09/2019.
- [46] https://www.youtube.com/watch?time_continue=68&v=GJGVJ202diM. Online; Accedido: 26/09/2019.

- [47] Murthy Devarakonda y Ching-Huei Tsou. Automated problem list generation from electronic medical records in ibm wtason. 2015.
- [48] <https://www.healthnavigator.com/index.html?p=5539.html>. Online; Accedido: 30/09/2019.
- [49] Paul Goldberg Alejo Nevado-Holgado Luka Gligic, Andrey Kormilitzin. Named entity recognition in electronic health records using transfer learning bootstrapped neural networks. 2019.
- [50] Maria Liakata Angus Roberts-Anoop D. Shah Katherine Morley David Osborn Joseph Hayes Robert Stewart Johnny Downs Wendy Chapman y Rina Dutta Sumithra Velupillai, Hanna Suominen. Using clinical natural language processing for health outcomes research: Overview and actionable suggestions for future advances. 2018.
- [51] Wendy W Chapman Prakas M Nadkarni, Lucila Ohno-Machado. Natural language processing: an introduction. July 2011.
- [52] Sebastian Gehrmann Patrick D Tyler Edward T Moseley Eric T Carlson David W Grant Yeran Li Jonathan Welt Leo Anthony Celi Joy T. Wu, Franck Dernoncourt. Behind the scenes: A medical natural language processing project. 2017.
- [53] Jianglei Han y Chenliang Li Jing Li, Aixin Sun. A survey on deep learning for named entity recognition. 2018.
- [54] D. Downey A.-M. Popescu T. Shaked S. Soderland D. S. Weld y A. Yates O. Etzioni, M. Cafarella. Unsupervised named entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1), 2005.
- [55] S. Sekine and C. Nobata. Definition, dictionaries and tagger for extended named entity hierarchy. 2004.

- [56] S. Zhang and N. Elhadad. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. *J. Biomed Inform*, 46, 2013.
- [57] S. Azzam C. Huyck B. Mitchell H. Cunningham y Y. Wilks K. Humphreys, R. Gaizauskas. University of sheffield: Description of the lasie-ii system as used for muc-7. 1998.
- [58] G. Krupka y K. IsoQuest. Description of the nerowl extractor system as used for muc-7. 2005.
- [59] F. Rinaldi y D. Mowatt W. J. Black. Facile: Description of the ne system used for muc-7. 1998.
- [60] T. Hampton y M. Ramos-Santacruz C. Aone, L. Halverson. Sra: Description of the ie2 system used for muc-7. 1998.
- [61] J. Bear D. Israel M. Kameyama D. Martin K. Myers y M. Tyson D. E. Appelt, J. R. Hobbs. Sri international fastus system: Muc-6 test results and analysis. 1995.
- [62] M.Moens y C. Grover A.Mikheev. Named entity recognition without gazetteers. 1999.
- [63] Vikas Yadav y Steven Bethard. A survey on recent advances in named entity recognition from deep learning models. 2018.
- [64] Ridong Jiang, Rafael E. Banchs, and Haizhou Li. Evaluating and combining name entity recognition systems. In *Proceedings of the Sixth Named Entity Workshop*, pages 21–27, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [65] Montserrat Marimon Martin Krallinger Jordi Armengol-Estapé, Felipe Soares. Pharmaconer tagger: a deep learning-based tool for automatically finding chemicals and drugs in spanish medical texts. June 2019.

- [66] Andre Lamurias and Francisco M. Couto. Text mining for bioinformatics using biomedical literature. In Shoba Ranganathan, Michael Gribskov, Kenta Nakai, and Christian Schönbach, editors, *Encyclopedia of Bioinformatics and Computational Biology*, pages 602 – 611. Academic Press, Oxford, 2019.
- [67] D. Nadeau y S. Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 2007.
- [68] M. Collins y Y. Singer. Unsupervised models for named entity classification. 1999.
- [69] S. Sekine y E. Ranshoof. Named entities: recognition, classification and use. 2009.
- [70] Qiaozhu Mei Joshua C. Denny y Hua Xu Yukun Chen, Thomas A. Lasko. A study of active learning methods for named entity recognition in clinical text. *Journal of Biomedical Informatics*, 2015.
- [71] Svetlana Kiritchenko Joel Martin y Xiaodan Zhu Berry de Bruijn, Colin Cherry. Machine-learned solutions for three stages of clinical information extraction: the state of the art at i2b2 2010. 2011.
- [72] Xuequin Lu Kaihua Gao Min Jiang y Hua Xu Jianbo Lei, Buzhou Tang. A comprehensive study of named entity recognition in chinese clinical text. 2013.
- [73] A. McCallum y F.C. Pereira J. Lafferty. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.
- [74] Trond Grenager Jenny Rose Finkel and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. 2005.
- [75] J. R. Quinlan. Induction of decision trees. 1(1):81–106, 1985.

- [76] S. R. Eddy. Hidden markov models. 6(3):361–365, 1996.
- [77] Jagat Narain Kapur. *Maximum-Entropy Models in Science and Engineering*.
- [78] E. Osuna J. Platt y B. Scholkopf M. A. Hearst, S. T. Dumais. Support vector machines. *IEEE Intelligent Systems and Their Applications*, 1998.
- [79] Harsha GurulingappaAbdul Mateen-RajpuLuca Toldo. Extraction of potential adverse drug events from medical case reports. *Journal of biomedical semantics*, 2012.
- [80] Zhiyuan Ma Tong Ruan Huanhuan Zhang y Ping He Jui Xue, Yangming Zhou. Fine-tuning bert for joint entity and relation extraction in chinese medical text.
- [81] Alistair Willis Trung Huynh, Yulan He and Stefan Ruger. Adverse drug reaction classification with deep neural networks.
- [82] T. H. Nguyen y R. Grisham. Relation extraction: Perspective from convolutional neural networks. *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, 2015.
- [83] B. Xiang y B. Zhou C. dos Santos. Classifying relations by ranking with convolutional neural networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 1:626–634, 2015.
- [84] G. Zhou y J. Zhao D. Zeng, K. Liu. Relation classification via convolutional deep neural network. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, page 2335–2344, 2014.

- [85] D. Zhang y D. Wang. Relation classification via recurrent neural network. 2015.
- [86] G. Li Y. Chen H. Peng y Z. Jin Y. Xu, L. Mou. Classifying relations via long short term memory networks along shortest dependency paths. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.*, 2015.
- [87] X. hu y M. Yang S. Zhang, D. Zheng. Bidirectional long short-term memory networks for relation classification. *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, pages 73–78, 2015.
- [88] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.
- [89] S. J. Pan y Q. Yang. A survey on transfer learning. *IEE transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.
- [90] M. L. Littman y A. W. Moore L. P. Kaelbling. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.
- [91] R. S. Sutton y A.G. Barto. Introduction to reinforcement learning. *MIT press Cambridge*, 135, 1998.
- [92] D. Lowd y C. Meek. Adversarial learning. pages 641–647, 2005.
- [93] D. Britz. Attention and memory in deep learning and nlp. *Online: <http://www.wildml.com/2016/01/attention-and-memory-in-deeplearning-and-nlp>*, 2016.
- [94] H. Yu A. Jagannatha. Structured prediction models for rnn based sequence labeling in clinical text. 2016.
- [95] Ruogu Fang Ragnhildur I Bjarnadottir William R Hogan Ypnghui Wu Xi Yang, Jiang Bian. Identifying relations of medications with adverse

- drug events using recurrent convolutional neural networks and gradient boosting. *Journal of the American Medical Informatics Association*, 2019.
- [96] Azadeh Nikfarjam y Graciela H. González. Pattern mining for extraction of mentions of adverse drug reactions from user comments. 2011.
- [97] Karen O'Connor Rachel Ginn Graciela González Azadeh Nikfarjam, Abeed Sarker. Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features. *Journal of the American Medical Informatics Association*, 22, May 2015.
- [98] James G. Arbogast Kevin A. Halbritter Deepthi Yellapragada Sushmitha Regulapati Pavani Bandaru V. Jogannathan, Charles J. Mullett. Assessment of commercial nlp engines for medication information extraction from dictated clinical notes.
- [99] <https://www.lluiscodina.com/taxonomias-ontologias>. Online; Accedido: 4/11/2019.
- [100] James Firby William A Baumgartner Helen L Johnson Philip V Ogren y K Bretonnel Cohen Lawrence Hunter, Zhiyong Lu. Opendmap: An open source, ontology-driven concept analysis engine, with applications to capturing knowledge regarding protein transport, protein interactions and cell-type-specific gene expression. *BMC Bioinformatics*, 2008.
- [101] Víctor Hugo Medina García Flor Nancy Díaz Piraquive, Luis Joyanes Aguilar. Taxonomía, onología y folkosomía, ¿qué son y qué beneficio u oportunidades presentan para los usuarios de la web? *Universidad & Empresa*, 8(16):242–261, 2009.
- [102] Lipyew Lim y Min Wang Anastasios Kementsietsidis. Supporting ontology-based keyword search over medical databases. 2008.

- [103] R. Cornet M. Dorrepaal D. Dangelmans y M.W.M. Jaspers F. Bakhshi-Raiez, N.F. de Keizer. A usability evaluation of a snomed ct based compositional interface terminology for intensive care. *International Journal of Medical Informatics*, 81:351–362, 2012.
- [104] Jane Taggart Pradeep Ray y Hairong Yu Alireza Rahimi, Siaw-Teng Liaw. Validating an ontology-based algorithm to identify patients with type 2 diabetes mellitus in electronic health records. *International Journal of Medical Informatics*, 83(10):768–778, 2014.
- [105] Anthony Nguyen Anton Bergheim Bevan Koopman, Guido Zuccon. Automatic icd-10 classification of cancers from free-text death certificates. *International Journal of Medical Informatics*, 84:956–965, 2015.
- [106] Jeff Mitchell y Mirella Lapata. Vector-based models of semantic composition. *Association for Computational Linguistics*, 2008.
- [107] Greg Corrado y Jeffrey Dean Tomas Mikolov, Kai Chen. Efficient estimation of word representations in vector space. *Proceedings of ICLR Workshop*, 2013.
- [108] Richard Socher y Christopher Manning Jeffrey Pennington. Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.
- [109] Sylvie Ranwez Stefan Janaqi y Jacky Montmain Sébastien Harispe, David Sánchez. A framework for unifying ontology-based semantic similarity measures: A study in the biomedical domain. *Journal of Biomedical Informatics*, 2014.
- [110] André O. Falcao Phillip Lord y Francisco M. Couto. Catia Pesquita, Daniel Faria. Semantic similarity in biomedical ontologies. *PLOS Computational Biology*, 5(7):1–12, 07 2009.

-
- [111] Feichen Shen Majid Rastegar-Mojarad y Hongfang Liu Liwei Wang, Yanshan Wang. Discovering associations between problem list and practice setting. *BMC Medical Informatics and Decision Making*, 2019.
- [112] Anthony Nguyen y Karin Verspoor Hamed Hassanzadeh. Quantifying semantic similarity of clinical evidence in the biomedical literature to facilitate related evidence synthesis. *Journal of Biomedical Informatics*, 2019.
- [113] Graciela Gonzalez Robert Leaman. Banner: An executable survey of advances in biomedical named entity recognition. 2008.
- [114] Comunidad de Madrid. Estandarización de abreviaturas, símbolos y expresiones utilizados en la prescripción y la administración de medicamentos.
- [115] Markus Kreuzthaler y Stefan Schulz Alexandra Pomares-Quimbaya. Current approaches to identify sections within clinical narratives from electronic health records: a systematic review. 2019.
- [116] Randolph A. Miller y Anderson Spickard Joshua C. Denny, Jeffrey D. Smithers. “understanding” medical school curriculum content using knowledgemap. *Journal of the American Medical Informatics Association*, 10(4), 2003.
- [117] Nigam H. Shah Paul R. Alexander Csongor Nyulas Tania Tudorache y Mark A. Musen Patricia L. Whetzel, Natalya F. Noy. Biportal: enhanced functionality via new web services from the national center for biomedical ontology to access and use ontologies in software applications. *Nucleic Acids Research*, 39, 2014.
- [118] Clement Jonquet Daniel Rubin Annie P Chiang y Mark A Musen Nigam H Shah, Nipun Bhatia. Comparison of concept recognizers for building

- the openbiomedical annotator. *AMIA Summit on Translational Bioinformatics*, 2009.
- [119] <http://www.snomed.org>. Online; Accedido: 5/11/2019.
- [120] <https://www.nlm.nih.gov/research/umls/index.html>. Online; Accedido: 5/11/2019.
- [121] <https://www.who.int/classifications/icd/icdonlineversions/en/>. Online; Accedido: 5/11/2019.
- [122] Keun Ho Ryu Erdenebileg Batbaatar. Ontology-based healthcare named entity recognition from twitter messages using a recurrent neural network approach. 2019.
- [123] [https://es.wikipedia.org/wiki/Biblioteca_\(inform%C3%A1tica\)](https://es.wikipedia.org/wiki/Biblioteca_(inform%C3%A1tica)). Online; Accedido:11/11/2019.
- [124] Chaim Oropeza García. Paquetes de programación en java. <https://es.slideshare.net/tibu85/paquetes-de-programacion>. Online; Accedido: 11/11/2019.
- [125] April Liang Dmytro Lituiev Peter Chang Jae Ho Sohn Yunn-Yi Chen Benjamin L.Franc Bonnie Joe Dexter Hadley Hari M. Trivedi, Maryam Panahiazar. Large scale semi-automated labeling of routine free-text clinical records for deep learning. 2018.
- [126] <https://aws.amazon.com/es/comprehend/medical/>. Online; Accedido: 01/10/2019.
- [127] <https://www.ibm.com/watson/services/natural-language-understanding/>. Online; Accedido: 02/10/2019.
- [128] <https://azure.microsoft.com/es-es/services/cognitive-services/text-analytics/>. Online; Accedido:01/10/2019.

- [129] <https://cloud.google.com/natural-language/?hl=es>. Online; Accedido: 01/10/2019.
- [130] <https://spaziodati.eu/en/>. Online; Accedido: 02/10/2019.
- [131] <https://dandelion.eu/>. Online; Accedido: 02/10/2019.
- [132] <https://www.repustate.com/>. Online; Accedido: 02/10/2019.
- [133] <https://aylien.com/>. Online; Accedido: 02/10/2019.
- [134] <https://www.textrazor.com/>. Online; Accedido: 02/10/2019.
- [135] https://www.3m.com/3M/en_US/health-information-systems-us/support/outpatient-ambulatory/?utm_term=hcbg-his-faccode-en_us-lead-codeassist_pdp-own-3mcom-na-learn-na-ne19. Online; Accedido: 01/10/2019.
- [136] <https://www.refinitiv.com/en/products/intelligent-tagging-text-analytics>. Online; Accedido: 30/09/2019.
- [137] Yangyang Li Fang Liu Shengqun Fang Wencheng Sun, Zhiping Cai and Guoyan Wang. Data processing and text mining technologies on electronic medical records: A review. *Journal of Healthcare Engineering*, 2018.
- [138] Abhyuday N Jagannatha and Hong Yu. Bidirectional RNN for medical event detection in electronic health records. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 473–482, San Diego, California, June 2016. Association for Computational Linguistics.
- [139] George Hripcsak y Adam S. Rothschild. Agreement, the f-measure, and reliability in information retrieval. 2005.

- [140] http://biocreative.sourceforge.net/bionlp_tools_links.html. Online; Accedido: 67/09/2019.
- [141] <https://orbit.nlm.nih.gov/browse-repository/dataset/human-annotated/83-genia-corpus>. Online; Accedido: 26/09/2019.
- [142] Erik Faessler Johannes Hellrich, Franz Matthies and Udo Hahn. Sharing models and tools for processing german clinical texts. 2015.
- [143] <https://www.ontotext.com/>. Online; Accedido: 8/10/2019.
- [144] <https://explosion.ai/demos/displacy-ent>. Online; Accedido: 26/09/2019.
- [145] Lars Juhl Jensen. Tagger: Becalm api for rapid named entity recognition. *bioRxiv*, 2017.
- [146] <https://www.healthnavigator.com/>. Online; Accedido: 01/10/2019.
- [147] <https://www.lexalytics.com/>. Online; Accedido: 01/10/2019.
- [148] <https://es.slideshare.net>. Online; Accedido: 16/10/2019.
- [149] George Hripcsak y Adam Wilcox. Reference standards, judges and comparison subjects: Roles for experts in evaluating system performance. *Journal of the American Medical Informatics Association*, 9(1), 2002.
- [150] Paolo Ferragina y Massimiliano Ciaramita Marco Cornolti. A framework for benchmarking entity-annotation systems. 2013.
- [151] Shrimalini Jayaramaraja Dario Giuse Theodore Speroff Steven H. Brown Peter L. Elkin y Michael E. Matheny Glenn T. Globbel, Ruth Reeves. Development and evaluation of raptat: A machine learning system for concept mapping of phrases from medical narratives. *Journal of Bio-medical Informatics*, 2014.

-
- [152] <https://www.deepl.com/es/translator>. Online; Accedido: 13/11/2019.
- [153] Blake Colman Christopher J Yates Fanny Liew y David Smallwood Sarah Holper, Rahul Barmanray. The ambiguous medical abbreviation (ama) study - challenges and opportunities. *Internal Medicine Journal*, 2019.
- [154] <https://www.ibm.com/es-es/marketplace/clinical-decision-support-oncology>. Online; Accedido: 21/10/2019.
- [155] <https://docs.python.org/2/library/difflib.html>. Online; Accedido: 28/11/2019.
- [156] Peter J. Haug Stéphane Meystre. Natural language processing to extract medical problems from electronic clinical documents: Performance evaluation. 2005.
- [157] Peter J. Haug Stéphane Meystre. Randomized controlled trial of an automated problem list with improved sensitivity. 77, 2008.